PROJECT REPORT

PROJECT TITLE : **Emerging Methods for Early Detection of Forest Fires.**

TEAM ID : PNT2022TMID14485.

TEAM MEMBERS : VINITH KUMAR S

RAHUL R

SELVAM R

SHIYAM PRASAD S

1.INTRODUCTION

1.1 Project overview

Forest fire detection can be used in operation of dangerous fire disasters in places where in it is risky for fire brigade worker to take the operation under control. Images of the surroundings around and use those images to detect firer using image processing algorithms.

1.2 Purpose

Purpose of this project is to detect forest fire at the earliest using algorithms so that it is easy to notify fire fighters about the intensity and other such information's of fire. It will be useful in places where people don't visit inside deep forest either for tourism or camping and if fire is detected we can extinguish at the earliest by using a automated water hose or foam before it spreads through entire forest.

2. LITERATURE SURVEY

2.1 Existing problem

According to the National Fire Protection Agency, there were 48 on-duty firefighter fatalities in 2019. Heart attacks remain the leading cause of death for firefighters, as the result of overexertion and stress. Thousands more are injured at fire/incident scenes. In addition to the more immediate risks of injury and death, firefighters face a 250 percent greater risk of cancer compared to the general population, due to consistent exposure to carcinogens. According to recent research published by the IAFC, an average 63 percent of all firefighters will contract cancer at some point in their lifetime. Meanwhile, more firefighters are suffering from behavioral health issues (such as anxiety, depression, PTSD and suicidal ideation), and many of them don’t seek help — or lack adequate healthcare access. Their safety training is also often diluted because they have more training topics to cover, and therefore fewer hours spent on each topic. Fire chiefs often lacks the resources and educational programs to help their team members in need.

2.2 LITERATURE SURVEY

[1] **R. Manikandan, K. Subash, T. Joshua sujith,**Development of an industrial firefighting rover, 2021, In this paper, explains the making of firefighting rover that helps while salvaging activity. With the assistance of robot control through a remote framework incorporated with the RF module. Vision and its control are done in the versatile application and the control is unmistakably on the premise of remote transmission.

[2] **Raghavendran, P.s.Raghavendran& Muthusamy, Suresh & Kumar, R & Kumar, R & K, Mahendran & Swathi, S & Kamesh, L & Sanjay, R**, 2018. An Intelligent Remote Controlled Fire Fighting Machine for Autonomous Protection of Human beings. A fire fighting machine is designed, which can extinguish the fire with the help of fire fighter away from the hazardous area. Also it can reach critical areas like damaged and demolished buildings.

[3] **Shubham Choudhury, Sahil Sawant, LaukikBidwalkar, MayureshMarathe;Siuli Das**, 2020. Design and Implementation of Autonomous Rover for Wildfire Extinguishing ,Fire

recognition and extinguishing it are the dangerous occupations that perpetually put the life of a fireman in harm's way. By placing a versatile wanderer to play out this assignment in a fire-inclined region, it can help to stay away from untoward episodes or lives lost.

[4] **Hong, Ji Hyeon; Min, Byung-Cheol, Taylor, Julia M**,2012, NL-based communication with firefighting robots,Fire figthers put themselves in harms's way while saving others and may even lose their lives in certain situation, such as toxic fumes, extreme heat, or inhaling smoke.

[5] **J. Suresh**, Fire-fighting robot, 2017, explains about the designing and implementing an autonomous robot capable of detecting and extinguishing flames, disasters can be avoided with minimal risk to human life. In this research, we illustrate an autonomous robot capable of detecting flames indoors and manoeuvring towards the flame to extinguish it with the help of carbon dioxide.

2.3 Problem Statement Definition

Emerging Methods for Early Detection of Forest Fires project is to detect forest fire at the earliest using algorithms so that it is easy to notify fire fighters about the intensity and other such information's of fire.

3. IDEATIOIN AND PROPOSED SOLUTION

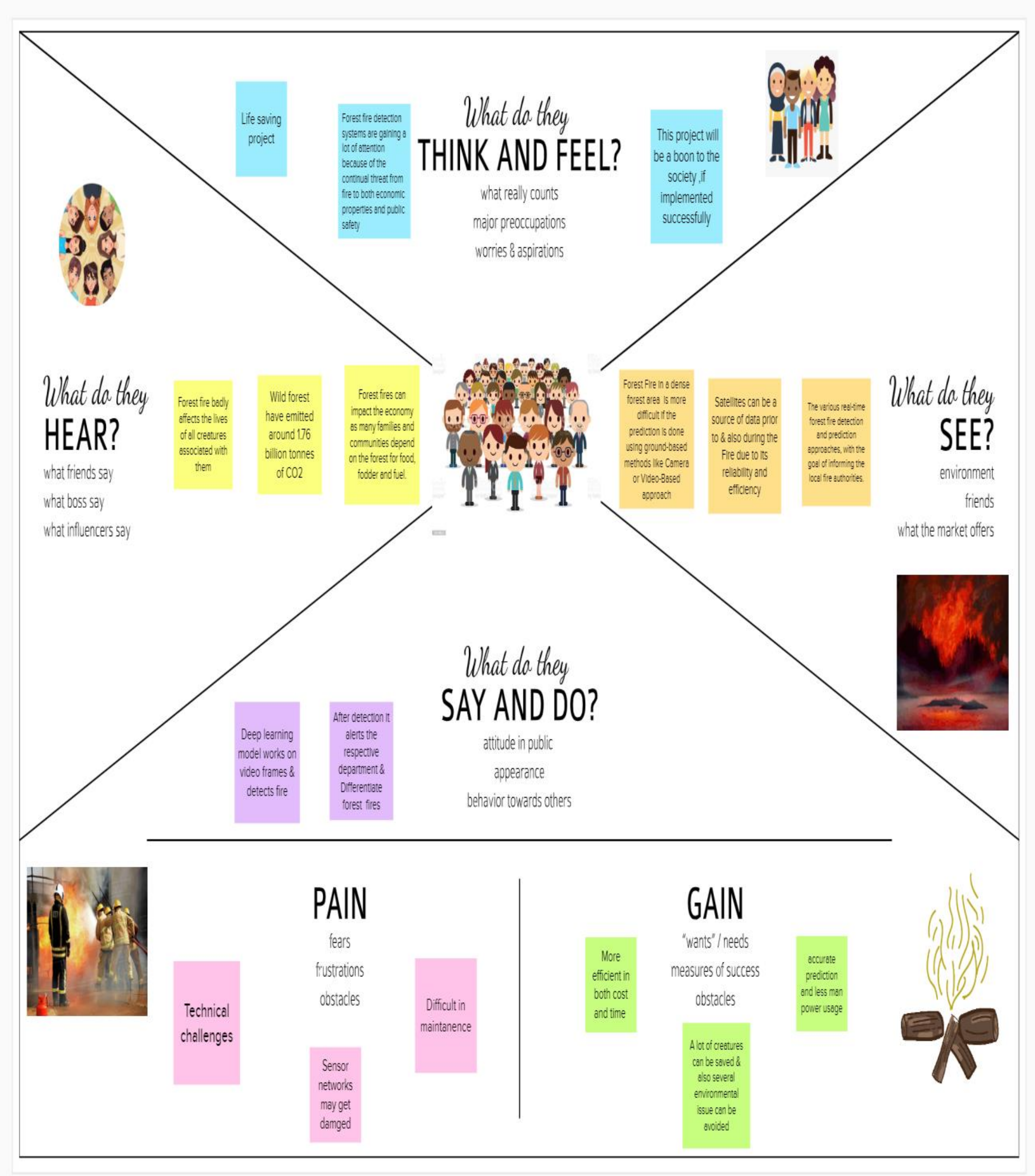
3.1 Empathy Map Canvas

1. An empathy map is a collaborative tool teams can use to gain a deeperinsight intotheir customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Grayand has gained much popularity within the agile community.

2. An empathy map is an effective visualization template that helps analyze the behavior and emotions of customers and users. Empathy maps not only detect thebehaviors but highlight possible mediums for brands to communicate with their customers in abetter way

3. Empathy maps can also be used to collect data directly from the users. Used alongside user interviews, survey answers, etc., you can also have a user fill in anempathy map themselves. This often reveals aspects of the user that may have remained unsaid or not thought of.

4. Each of the four quadrants comprises a category that helps us delve into the mind of the user. The four empathy map quadrants look at what the user says, thinks,feels, and does.



What do they THINK AND FEEL?

what really counts

major preoccupations

worries & aspirations

Life saving project

Forest fire detection systems are gaining a lot of attention because of the continual threat from fire to both economic properties and public safety

This project will be a boon to the society ,if implemented successfully

What do they HEAR?

what friends say

what boss say

what influencers say

Forest fire badly affects the lives of all creatures associated with them

Wild forest have emitted around 1.76 billion tonnes of CO2

Forest fires can impact the economy as many families and communities depend on the forest for food, fodder and fuel.

What do they SEE?

environment

friends

what the market offers

Forest Fire in a dense forest area is more difficult if the prediction is done using ground-based methods like Camera or Video-Based approach

Satellites can be a source of data prior to & 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.

What do they SAY AND DO?

attitude in public

appearance

behavior towards others

Deep learning model works on video frames & detects fire

After detection it alerts the respective department & Differentiate forest fires

PAIN

fears

frustrations

obstacles

Technical challenges

Sensor networks may get damged

Difficult in maintanence

GAIN

"wants" / needs

measures of success

obstacles

More efficient in both cost and time

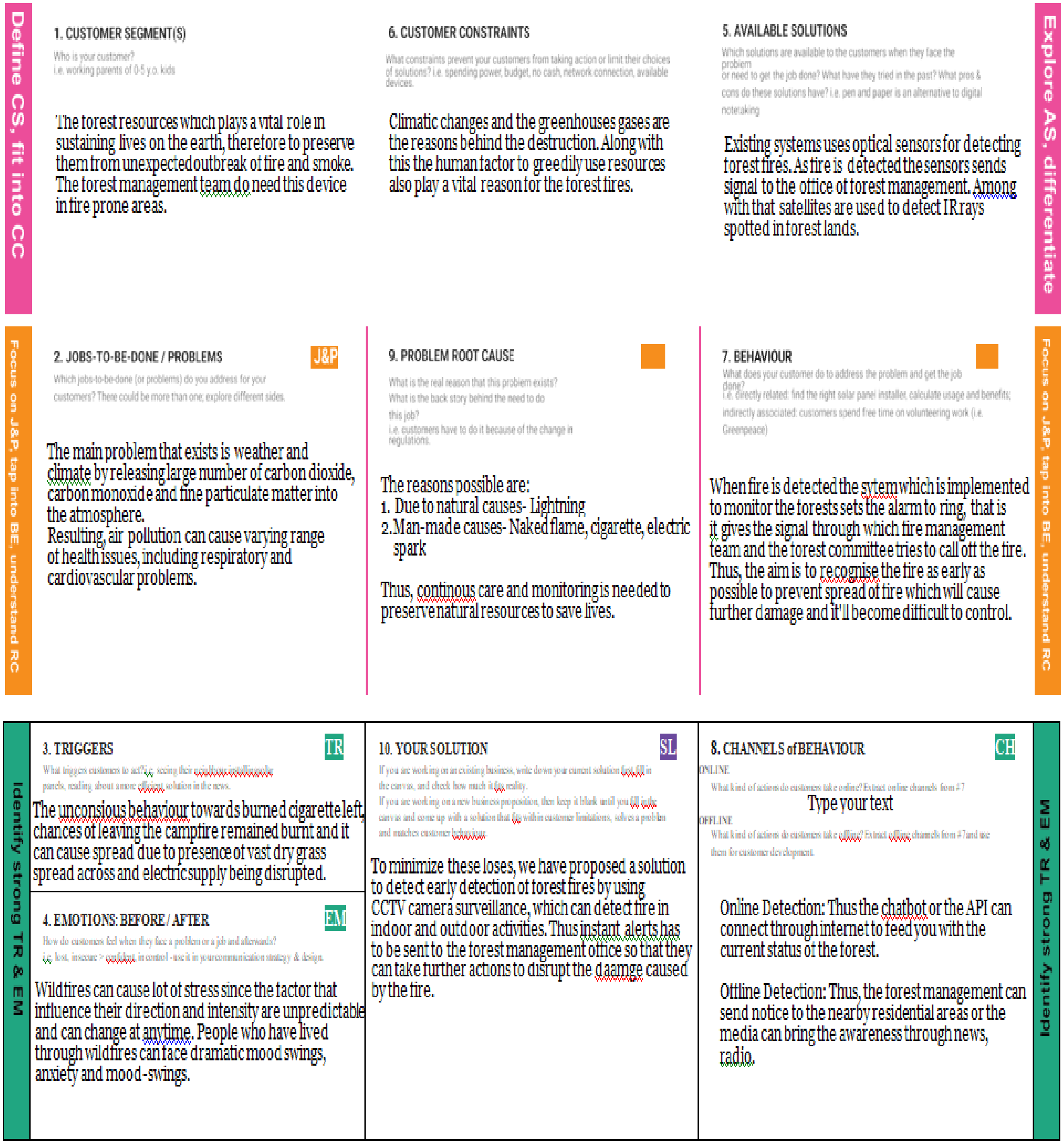
A lot of creatures can be saved & also several environmental issue can be avoided

accurate prediction and less man power usage

3.2 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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. Unplanned and abrupt forest fires are a major cause of forest degradation, while a controlled fire to manage and check the spread of unwanted forest fires serves as the action to improve the forest. So therefore, we've to detect prevention measure which should be taken to identify the fire prone areas and the tools which needed to be developed to minimize the loss and as well as implement forest fire committee to work for the reduction of damage caused.
2.	Idea / Solution description	Our solution aims at collecting the vast range of dataset to test and train the model regularly by using **CNN** where the system can detect immediately if any ignition of fire is found, where the video can be surveyed by satellite. Then **Cloudant DB** is brought to use where the large amount of data is stored and fetched which acts as a server. **Open CV** acts as a tool for processing videos which are captured. To send alerts to forest committee **Twilio API** is used where alerts are passed on detection. **Watson Assistant** also a chatbot tool which can help you monitor if any guide is needed.
3.	Novelty / Uniqueness	Existing system uses electronic sensors to detect forest fire and smoke. The change in temperature indicates the presence of forest fire and smoke in a region which can be detected by the sensors using radiation heat. As forests are in remote area it's difficult for installation and maintenance of sensors. Our proposed system depends on using AI to make it cheaper and easier for the forest management. Accuracy and timely prediction using AI, CNN and API made it possible.
4.	Social Impact / Customer Satisfaction	Forest fires are dangerous for the existence of life as they carry wildlife and natural resources which gives life to various living bodies. Thus, fires are occurred expectedly or unexpectedly which has to be prevented as earlier as we can. Therefore, forest management should be active enough to be aware and keeping an eye to check the forest fields regularly. As, what we save, saves us should be the

3.3 Problem Solution fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

Who is your customer?
i.e. working parents of 0-5 y.o. kids

The forest resources which plays a vital role in sustaining lives on the earth, therefore to preserve them from unexpectedoutbreak of fire and smoke. The forest management team do need this device in fire prone areas.

6. CUSTOMER CONSTRAINTS

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Climatic changes and the greenhouses gases are the reasons behind the destruction. Along with this the human factor to greedily use resources also play a vital reason for the forest fires.

5. AVAILABLE SOLUTIONS

Which solutions are available to the customers when they face the problem
or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Existing systems uses optical sensors for detecting forest fires. As fire is detected the sensors sends signal to the office of forest management. Among with that satellites are used to detect IR rays spotted in forest lands.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS

J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

The main problem that exists is weather and climate by releasing large number of carbon dioxide, carbon monoxide and fine particulate matter into the atmosphere.
Resulting, air pollution can cause varying range of health issues, including respiratory and cardiovascular problems.

9. PROBLEM ROOT CAUSE

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

The reasons possible are:

1. Due to natural causes- Lightning
2. Man-made causes- Naked flame, cigarette, electric spark

Thus, continous care and monitoring is needed to preservenatural resources to save lives.

7. BEHAVIOUR

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

When fire is detected the sytem which is implemented to monitor the forests sets the alarm to ring, that is it gives the signal through which fire management team and the forest committee tries to call off the fire. Thus, the aim is to recognise the fire as early as possible to prevent spread of fire which will cause further damage and it'll become difficult to control.

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

Identify strong TR & EM

3. TRIGGERS

TR

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

The unconsious behaviour towards burned cigarette left, chances of leaving the camptire remained burnt and it can cause spread due to presence of vast dry grass spread across and electric supply being disrupted.

4. EMOTIONS: BEFORE / AFTER

EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Wildfires can cause lot of stress since the factor that influence their direction and intensity are unpredictable and can change at anytime. People who have lived through wildfires can face dramatic mood swings, anxiety and mood-swings.

10. YOUR SOLUTION

SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

To minimize these loses, we have proposed a solution to detect early detection of forest fires by using CCTV camera surveillance, which can detect fire in indoor and outdoor activities. Thus instant alerts has to be sent to the forest management office so that they can take further actions to disrupt the daamge caused by the fire.

8. CHANNELS of BEHAVIOUR

CH

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

Type your text

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

Online Detection: Thus the chatbot or the API can connect through internet to feed you with the current status of the forest.

Offline Detection: Thus, the forest management can send notice to the nearby residential areas or the media can bring the awareness through news, radio.

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

1. The system shall take training sets of fire images and recognize whether there is a fire or the beginning of a fire (smoke) or if there is no fire
2. The system shall send a notification to the admin when it recognizes a fire in the image given
3. The system shall take real inputs of camera images and determine whether the image contains a fire or not
4. The system shall be able to take images with a variety of sizes and convert it to one fixed image to be used throughout the application.
5. The system shall run as a service on either a Windows or Linux operating system. 6. In the event that the computer on which the system is running shuts down, the system service should start automatically when the computer restart.

4.2 Non Functional Requirement

1. The system shall provide following facility that will allow web pages that the user is permitted to access. The system must support the following facility:

   a. Send alert message

   b. Customer data management

2. The system shall allow the user's status to be stored for the next time he returns to the web site. This will save the user x minutes per visit by not having to reenter already supplied data.

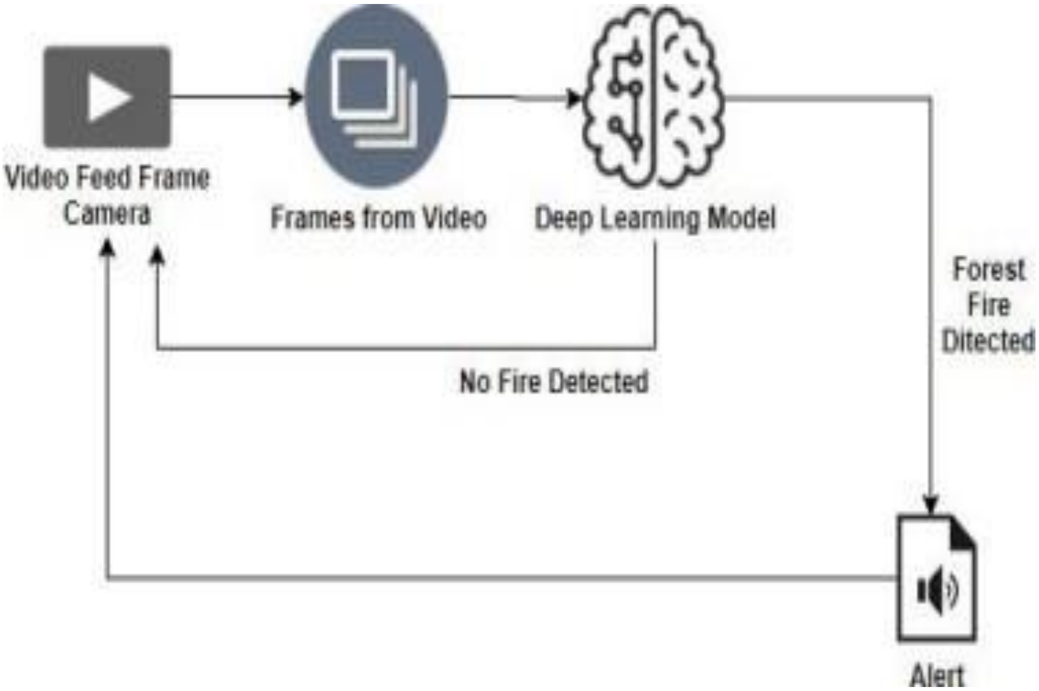
3. The system shall provide information about event log of forest.

5. PROJECT DESIGN

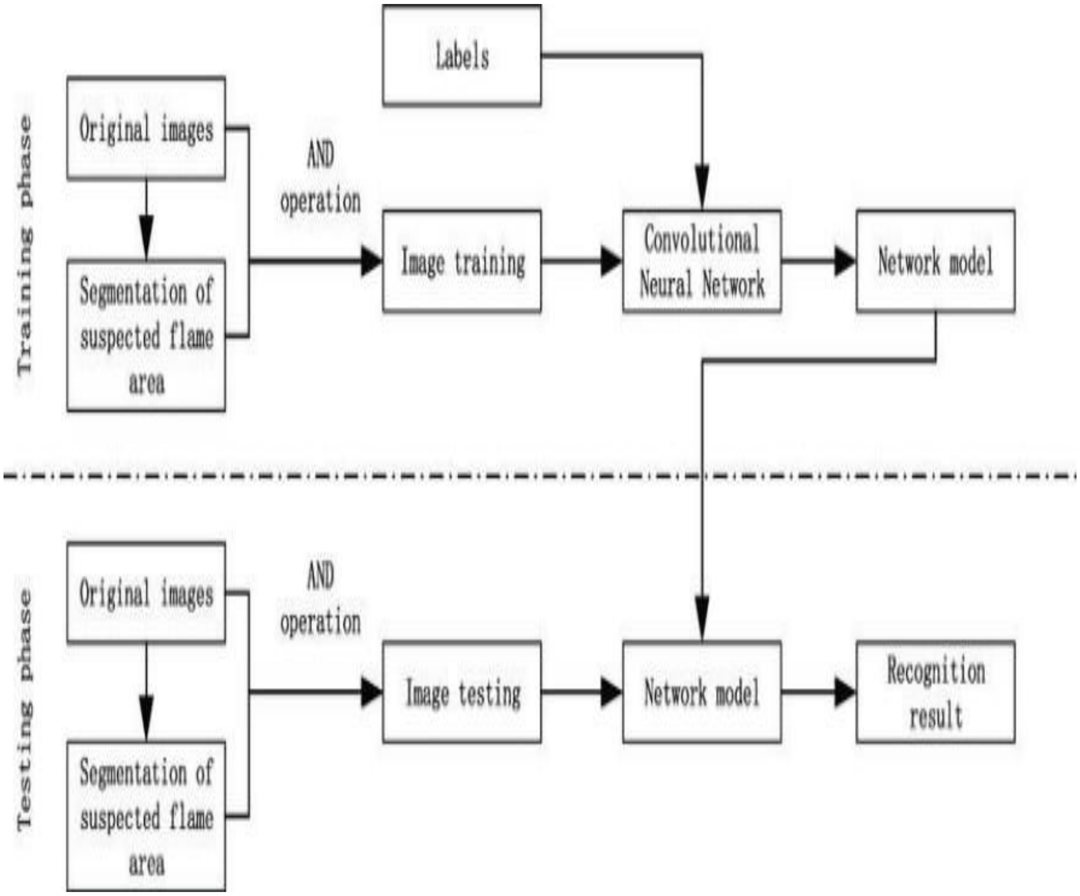
5.1 Data Flow Diagrams

It is difficult to predict and detect Forest Fire in a sparsely populated forest area. 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

- A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system.
- A neat 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.

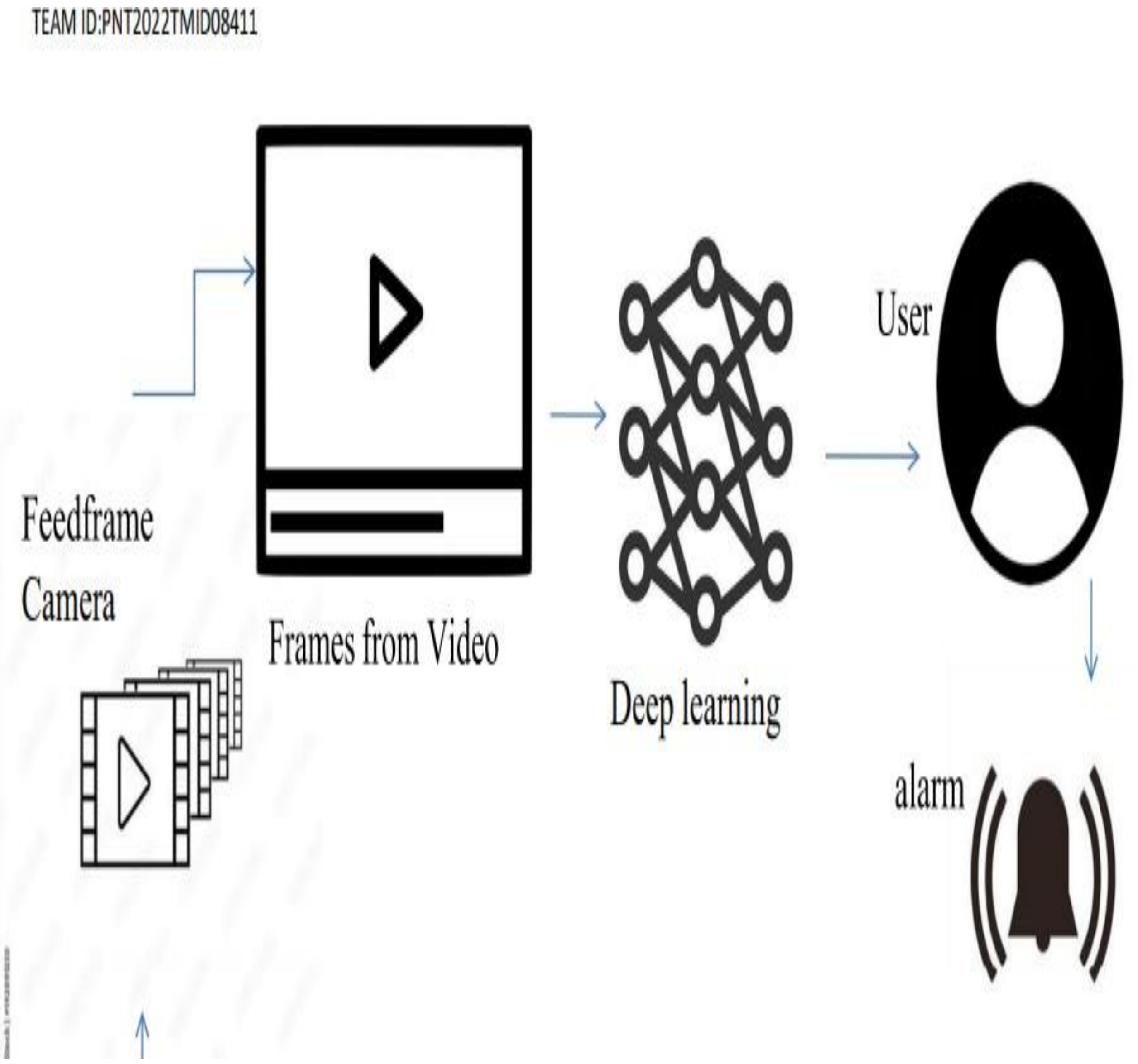


Architecture Diagram



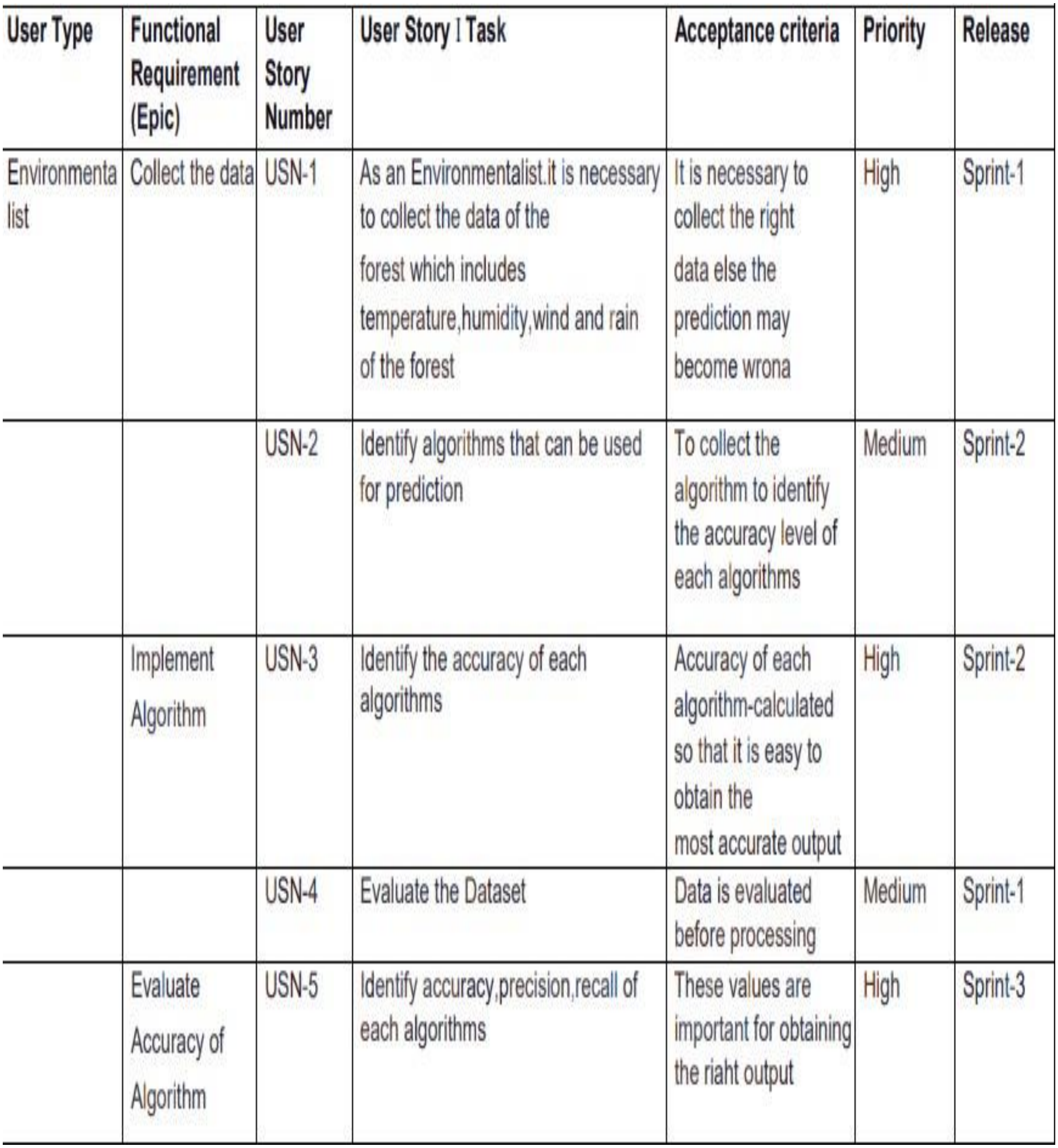
Data Flow Diagrams

5.2 Solution & Technical Architecture



Technical Architecture

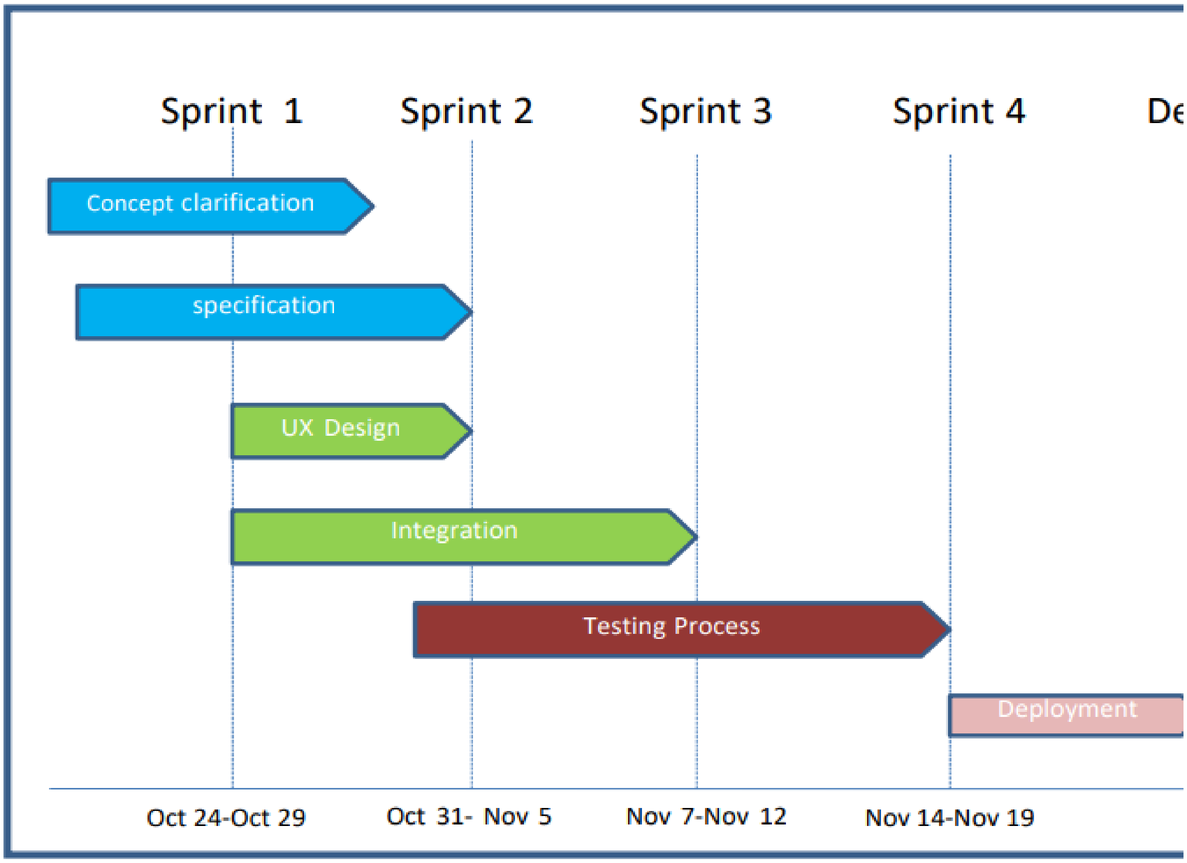
5.3 User Stories



User Type	Functional Requirement (Epic)	User Story Number	User Story I Task	Acceptance criteria	Priority	Release
Environmentalist	Collect the data	USN-1	As an Environmentalist.it is necessary to collect the data of the forest which includes temperature,humidity,wind and rain of the forest	It is necessary to collect the right data else the prediction may become wrona	High	Sprint-1
		USN-2	Identify algorithms that can be used for prediction	To collect the algorithm to identify the accuracy level of each algorithms	Medium	Sprint-2
	Implement Algorithm	USN-3	Identify the accuracy of each algorithms	Accuracy of each algorithm-calculated so that it is easy to obtain the most accurate output	High	Sprint-2
		USN-4	Evaluate the Dataset	Data is evaluated before processing	Medium	Sprint-1
	Evaluate Accuracy of Algorithm	USN-5	Identify accuracy,precision,recall of each algorithms	These values are important for obtaining the riaht output	High	Sprint-3

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation



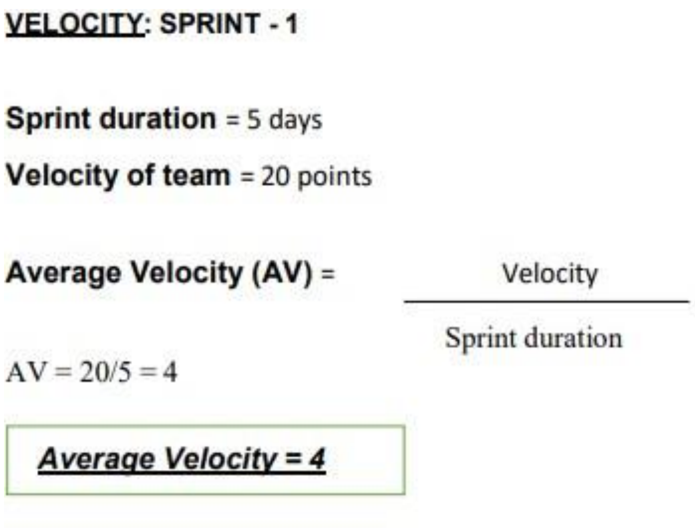
6.2 Sprint Delivery Schedule

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

6.3 Reports from JIRA

JIRA has categorized reports in four levels, which are −

- Agile
- Issue Analysis
- Forecast & Management
- Others



VELOCITY: SPRINT - 1

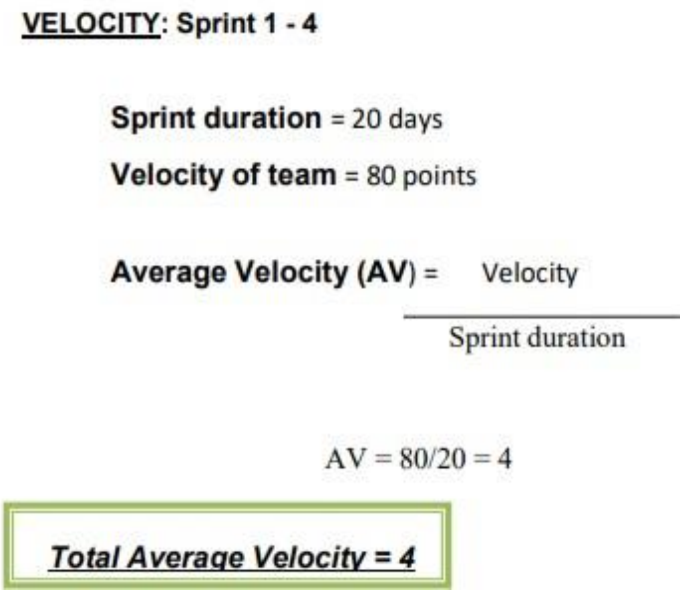
Sprint duration = 5 days

Velocity of team = 20 points

Average Velocity (AV) = $\frac{\text{Velocity}}{\text{Sprint duration}}$

AV = 20/5 = 4

Average Velocity = 4



VELOCITY: Sprint 1 - 4

Sprint duration = 20 days

Velocity of team = 80 points

Average Velocity (AV) = $\frac{\text{Velocity}}{\text{Sprint duration}}$

AV = 80/20 = 4

Total Average Velocity = 4

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

Feature 1

```
import warnings
warnings.filterwarnings("igno
re") import numpy as np
import sys
import tensorflow as tf
from distutils.version import
StrictVersion from collections
import defaultdict
from object_detection.utils import ops as utils_ops


##import serial
##ser = serial.Serial(port = "COM7", baudrate = '9600')


# This is needed since the notebook is stored in the
object_detection folder. sys.path.append("..")



if StrictVersion(tf._version__) < StrictVersion('1.9.0'):
 raise ImportError('Please upgrade your TensorFlow installation to v1.9.* or
 later!')



from utils import label_map_util
```

```
    from utils import visualization_utils as vis_util


    MODEL_NAME = 'inference_graph'
    PATH_TO_FROZEN_GRAPH = MODEL_NAME +
    '/frozen_inference_graph.pb' PATH_TO_LABELS =
    'training/labelmap.pbtxt'


    detection_graph = tf.Graph()
    with
     detection_graph.as_default():
     od_graph_def =
     tf.GraphDef()
     with
      tf.gfile.GFile(PATH_TO_FROZEN_GR
      APH, 'rb') as fid: serialized_graph =
      fid.read()
      od_graph_def.ParseFromString(serialized
      _graph)
      tf.import_graph_def(od_graph_def,
      name='')


category_index =
label_map_util.create_category_index_from_labelmap(PATH_TO_LABELS,
use_display_name=True)


    def
      run_inference_for_single_image(im
      age, graph): if 'detection_masks' in
      tensor_dict:
```

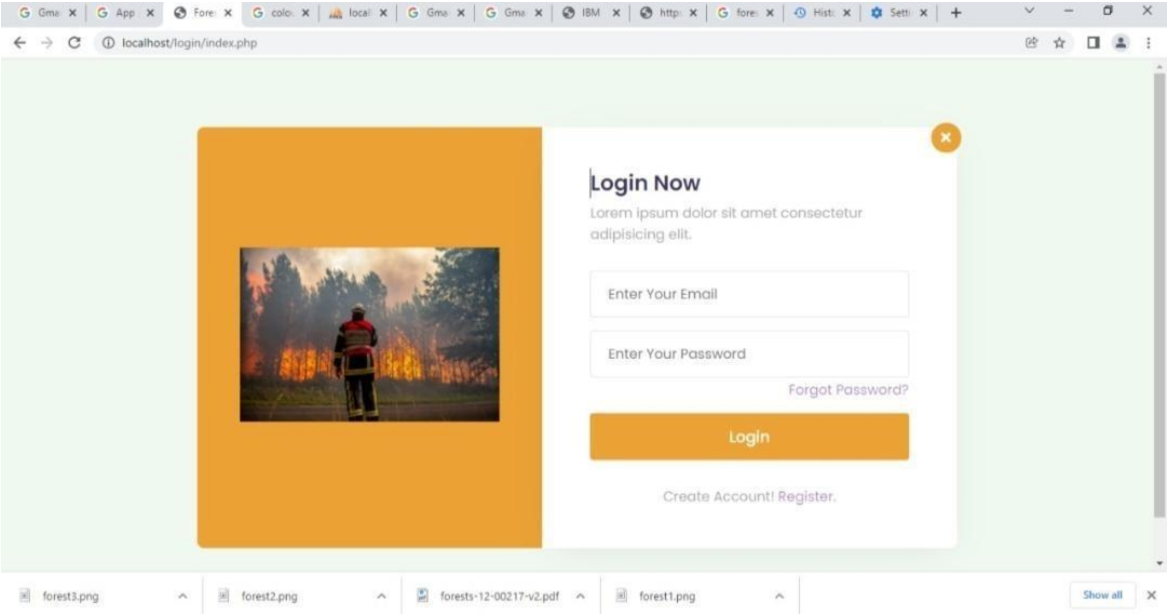
```
    # The following processing is only for single
    image detection_boxes =
    tf.squeeze(tensor_dict['detection_boxes'], [0])
    detection_masks =
    tf.squeeze(tensor_dict['detection_masks'], [0])

    # Reframe is required to translate mask from box coordinates to
image coordinates and fit the image size.

    real_num_detection =
    tf.cast(tensor_dict['num_detections'][0], tf.int32)
    detection_boxes = tf.slice(detection_boxes, [0, 0],
    [real_num_detection, -1])

    detection_masks = tf.slice(detection_masks, [0, 0, 0], [real_num_detection, -1, -
    1]) detection_masks_reframed =
    utils_ops.reframe_box_masks_to_image_masks(

      detection_masks, detection_boxes, image.shape[0],
    image.shape[1]) detection_masks_reframed = tf.cast(

      tf.greater(detection_masks_reframed, 0.5), tf.uint8)

    # Follow the convention by adding back the
    batch dimension
    tensor_dict['detection_masks'] =
    tf.expand_dims(

      detection_masks_reframed, 0)

  image_tensor = tf.get_default_graph().get_tensor_by_name('image_tensor:0')


  # Run inference

  output_dict = sess.run(tensor_dict,

              feed_dict={image_tensor: np.expand_dims(image, 0)})
  # all outputs are float32 numpy arrays, so convert
  types as appropriate output_dict['num_detections'] =
  int(output_dict['num_detections'][0])
  output_dict['detection_classes'] = output_dict[
    'detection_classes'][0].astype(np.uint8)


  output_dict['detection_boxes'] =
  output_dict['detection_boxes'][0]
  output_dict['detection_scores'] =
  output_dict['detection_scores'][0] if
  'detection_masks' in output_dict:
    output_dict['detection_masks'] = output_dict['detection_masks'][0]
  if output_dict['detection_classes'][0] == 1 and
   output_dict['detection_scores'][0] > 0.70: print('FIRE')
  winsound.Beep(frequency, duration)


 if output_dict['detection_classes'][0] == 2 and
   output_dict['detection_scores'][0] > 0.70: print('FIRE')


winsound.Beep(frequency, duration)


 if ((output_dict['detection_classes'][0] == 1 or
output_dict['detection_scores'][0] < 0.70) and
(output_dict['detection_classes'][0] == 2 or
output_dict['detection_scores'][0] < 0.70)):
   print('No

  Fire') return

  output_dict
```

Feature 2

```
import cv2

cap = cv2.VideoCapture(0)


try:
 with
  detection_graph.as_default
  ():with tf.Session() as
  sess:
    # Get handles to input and output
    tensors ops =
    tf.get_default_graph().get_operati
    ons()
    all_tensor_names = {output.name for op in ops for
    output in op.outputs}tensor_dict = {}
    for key in [
     'num_detections', 'detection_boxes',
     'detection_scores','detection_classes',
     'detection_masks'
    ]:
     tensor_name = key + ':0'
     if tensor_name in all_tensor_names:
       tensor_dict[key] =
      tf.get_default_graph().get_tensor_by_name(
      tensor_name)


    while True:
     ret, image_np = cap.read()
     # Expand dimensions since the model expects images to have shape: [1,
     None,
None,
3]

     image_np_expanded =
     np.expand_dims(image_np,  axis=0)# Actual
     detection.
        utput_dict =
              run_inference_for_single_imagiimage_np,
              detection_graph)# Visualization of the results of
              a detection.
              vis_util.visualize_boxes_and_labels_on_image_
              array(
       image_np,
       output_dict['detection_boxes'],
       output_dict['detection_classes'],
       output_dict['detection_scores'],
        category_index,
        instance_masks=output_dict.get('detectio
        n_masks'),
        use_normalized_coordinates=True,
        line_thickness=8)
      cv2.imshow('Frame',
      cv2.resize(image_np,(800,600))) if
      cv2.waitKey(1) == ord('q'):
        cap.release()
        cv2.destroyAllWin
        dows()break
except Exception as e:print(e)
cap.release()
```

8. TESTING

8.1 Test Cases

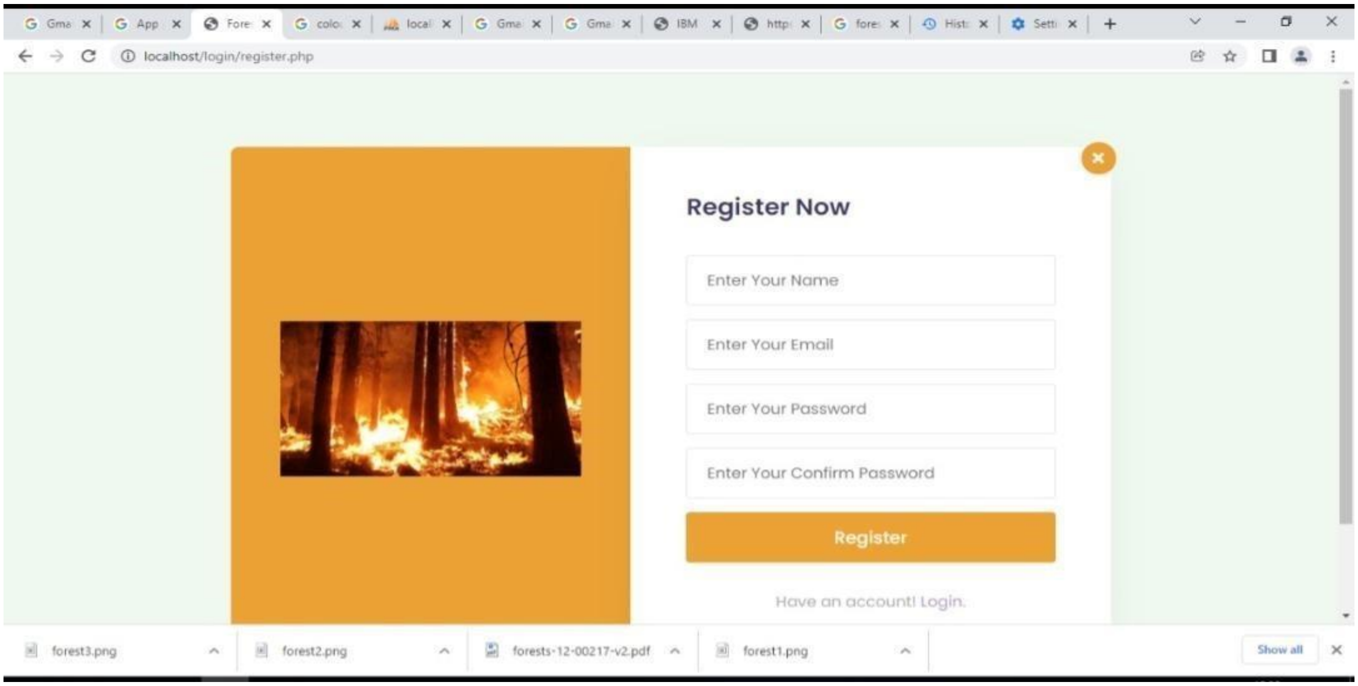


localhost/login/index.php

Login Now

Lorem ipsum dolor sit amet consectetur adipisicing elit.

Login Page



Register Now

Enter Your Name

Enter Your Email

Enter Your Password

Enter Your Confirm Password

Register

Have an account! Login.

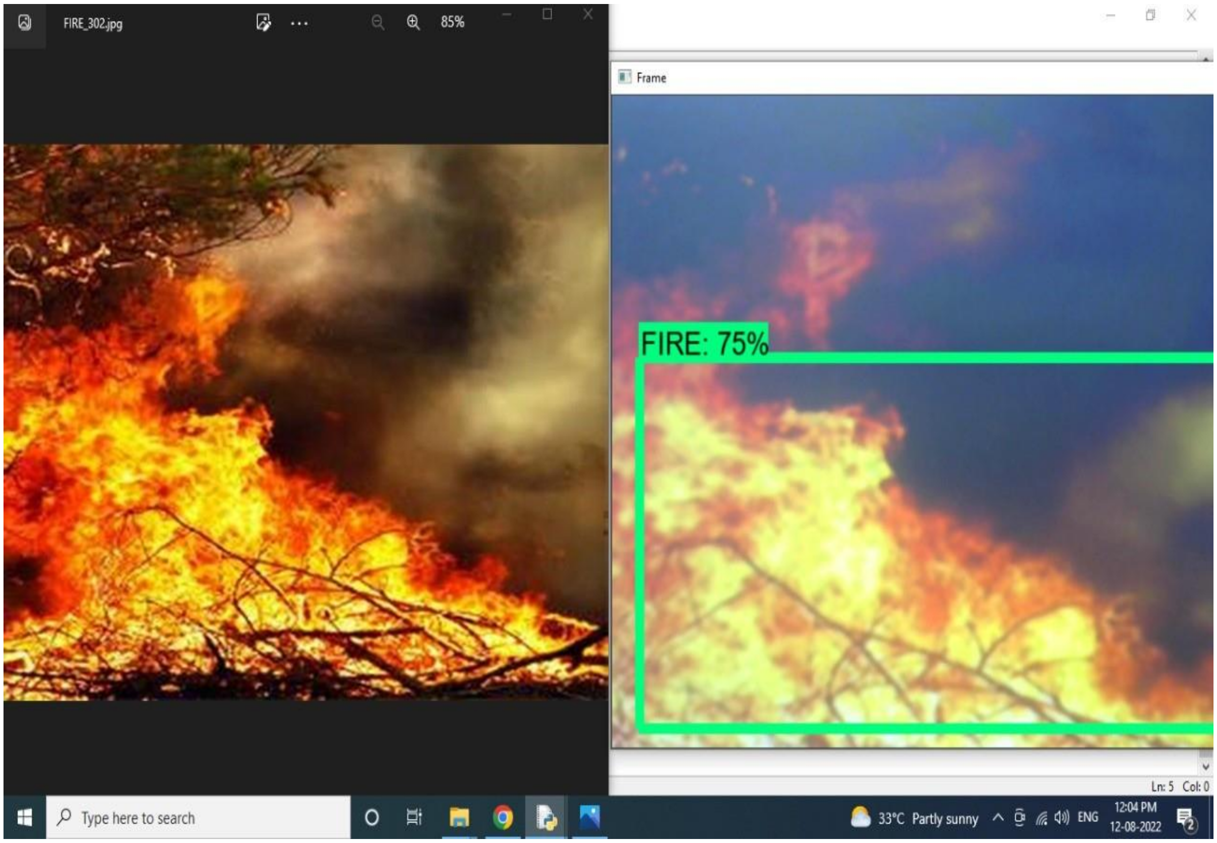
Register Page

8.2 User Acceptance Testing

1. This sort of testing is carried out by users, clients, or other authorised bodies toidentify the requirements and operational procedures of an application or piece of software.
2. The most crucial stage of testing is acceptance testing since it determineswhether or not the customer will accept the application or programmer.
3. It could entail the application's U.I., performance, usability, and usefulness. It isalso referred to as end-user testing, operational acceptance testing, and user acceptance testing (UAT).

9. RESULT

Performance Metrics



Value obtained from three sensor, if any Infrared ray detected, it gives output as IR detected, Sensor activated! Similarly, if there is any temperature change it will show Abnormal temperature and its intensity. For any smoke detection it output as Smoke detected and sensorvalue.Above image is result obtained from the trained ML model showing count for damaged and intact homes.

10. ADVANTAGES & DISADVANTAGES

Advantages:

1. It refreshes the habitat zones: Fire clears out plants and trees to make more natural resources available to the habitat. Fewer trees mean more water becomes available for the remaining plants and animals that call the area their home. New grass and shrubs are food sources for a number of animals as well. A ground cover that comes back after a fire becomes anew micro-habitat. Everything is refreshed with a fire.

2. Low-intensity fires don't usually harm trees: The bark of a tree is like an armored shell against fire, pests, and other things that could damage them. Most forest fires burn at low- temperature levels when conditions are optimal and this causes minimal damage to the trees ofthe forest when it occurs.

The end result is a clearing of the ground floor of the forest while thetrees are able to continue standing majestically.

3. Decreases the Wastes on Forests: Forests have a lot of waste that ends up building up over time and these wastes can help create wildfires. If a large wildfire breaks out it might take weeks to control it and the damage it can cause is just too extensive to understand for us. Waste such as dead leaves on the ground can be pretty useful for wildfires to feed on and smallforest fires just deal with these wastes properly without going out of control.

Disadvantages:

1. A forest fire sets up the potential for soil erosion to occur: Forest fires clear the underbrush away and encourage new growth, but there is a period of time between the fire andthe new growth where the forest is vulnerable.

2. Forest fires always bring death in some form: Maybe it's just the weak plants of the forest that are killed during a fire, but there is always some sort of death that happens when afire occurs. Sometimes it is the firefighters who are tasked with stopping the fire. It could beanimals or pets.

3. Uncontrolled fires can cause localized air pollution: Despite the amount of global development that has occurred, there are many forests that are difficult or nearly impossible toreach. Fires in these areas are left to burn in an uncontrolled fashion and this creates air pollution which can affect the local environment and make it difficult to breathe.

11. CONCLUSION

This project will help in early detection of forest fire and the prevention. It also involves the risk factor of analyzing the drone images of affected areas using machine learning algorithm whichovercomes the existing project. This system detects the fire conditions in a short time before any fire accidents spreads over the forest area. The scope of using video frames in the detection of fire using machine learning is challenging as well as innovative. If this system with less error rate can be implemented at a large scale like in big factories, houses, forests, it is possible to prevent damage and loss due to random fire accidents by making use of the Surveillance System.

12. FUTURE SCOPE

Future Scope In future, we are planning to install smart water tank system in dense forest where reachability of resources and firefighters is difficult. In addition to that we will be updatingthe system with more features and reliability. We will also include a high pitch sound system that will keep away the animals from the site of fire.The proposed system can be developed to more advanced system by integrating wireless sensors with CCTV for added protection and precision. The algorithm shows great promise in adapting to various environment.

13. APPEND

Source Code:

```
import warnings
warnings.filterwarnings("ignore")
import numpy as np

import sys

import tensorflow as tf

from  distutils.version  import  StrictVersionfrom
collections import defaultdict

from object_detection.utils import ops as utils_ops

##import serial

##ser = serial.Serial(port = "COM7", baudrate = '9600')

# This is needed since the notebook is stored in the object_detection folder.
sys.path.append("..")


if StrictVersion(tf._version__) < StrictVersion('1.9.0'):

  raise ImportError('Please upgrade your TensorFlow installation to v1.9.* orlater!')

from utils import label_map_util

from  utils  import  visualization_utils  as  vis_util


MODEL_NAME = 'inference_graph'

PATH_TO_FROZEN_GRAPH = MODEL_NAME + '/frozen_inference_graph.pb'
PATH_TO_LABELS = 'training/labelmap.pbtxt'
```

```
detection_graph = tf.Graph()

with detection_graph.as_default():

  od_graph_def = tf.GraphDef()

  with tf.gfile.GFile(PATH_TO_FROZEN_GRAPH, 'rb') as fid:
    serialized_graph = fid.read()
    od_graph_def.ParseFromString(serialized_graph)
    tf.import_graph_def(od_graph_def, name='')


category_index = label_map_util.create_category_index_from_labelmap(PATH_TO_LABELS,
use_display_name=True)


def run_inference_for_single_image(image, graph):if
    'detection_masks' in tensor_dict:

        # The following processing is only for single image detection_boxes =
        tf.squeeze(tensor_dict['detection_boxes'], [0]) detection_masks =
        tf.squeeze(tensor_dict['detection_masks'], [0])

        # Reframe is required to translate mask from box coordinates to image
coordinates and fit the image size.

        real_num_detection = tf.cast(tensor_dict['num_detections'][0], tf.int32)
        detection_boxes = tf.slice(detection_boxes, [0, 0], [real_num_detection,

-1])

        detection_masks = tf.slice(detection_masks, [0, 0, 0],

[real_num_detection, -1, -1]) detection_masks_reframed
        =

utils_ops.reframe_box_masks_to_image_masks(

            detection_masks, detection_boxes, image.shape[0], image.shape[1])
        detection_masks_reframed = tf.cast(
            tf.greater(detection_masks_reframed, 0.5), tf.uint8)
        # Follow the convention by adding back the batch dimension
        tensor_dict['detection_masks'] = tf.expand_dims(

            detection_masks_reframed, 0)
    image_tensor =

tf.get_default_graph().get_tensor_by_name('image_tensor:0')


    # Run inference

    output_dict  =  sess.run(tensor_dict,

                            feed_dict={image_tensor: np.expand_dims(image, 0)})


    # all outputs are float32 numpy arrays, so convert types as appropriate
    output_dict['num_detections'] = int(output_dict['num_detections'][0])
    output_dict['detection_classes'] = output_dict[

        'detection_classes'][0].astype(np.uint8)


    output_dict['detection_boxes'] = output_dict['detection_boxes'][0]
    output_dict['detection_scores']  =   output_dict['detection_scores'][0]if
    'detection_masks' in output_dict:

        output_dict['detection_masks']  =   output_dict['detection_masks'][0]if
    output_dict['detection_classes'][0] == 1 and

output_dict['detection_scores'][0] > 0.70:
      print('FIRE')


    if output_dict['detection_classes'][0] == 2 and
output_dict['detection_scores'][0] > 0.70:
      print('FIRE')
    if ((output_dict['detection_classes'][0] == 1 or
output_dict['detection_scores'][0] < 0.70) and
(output_dict['detection_classes'][0] == 2 or
output_dict['detection_scores'][0] < 0.70)):
      print('No Fire')

    return  output_dict




import cv2
cap = cv2.VideoCapture(0)


try:
    with detection_graph.as_default():with
        tf.Session() as sess:
                # Get handles to input and output tensors ops =
                tf.get_default_graph().get_operations()
                all_tensor_names = {output.name for op in ops for output in
op.outputs}
                tensor_dict = {}
                for key in [
                  'num_detections', 'detection_boxes', 'detection_scores',
                  'detection_classes', 'detection_masks'
                ]:
                    tensor_name = key + ':0'
                    if tensor_name in all_tensor_names:

                        tensor_dict[key] = tf.get_default_graph().get_tensor_by_name(tensor_name)


                while True:

                        ret, image_np = cap.read()

                    # Expand dimensions since the model expects images to haveshape:
[1, None, None, 3]

                    image_np_expanded = np.expand_dims(image_np,   axis=0)#
                    Actual detection.

                    output_dict = run_inference_for_single_image(image_np,
detection_graph)

                    # Visualization of the results of a detection.
                    vis_util.visualize_boxes_and_labels_on_image_array(

                        image_np,
                        output_dict['detection_boxes'],
                        output_dict['detection_classes'],
                        output_dict['detection_scores'],
                        category_index,

                        instance_masks=output_dict.get('detection_masks'),
                        use_normalized_coordinates=True, line_thickness=8)

                    cv2.imshow('Frame',   cv2.resize(image_np,(800,600))) if
                    cv2.waitKey(1) == ord('q'):

                        cap.release()
                        cv2.destroyAllWindows()
                        break

except Exception as e:
    print(e) cap.release()
```

MODEL MAIN:

```
from__future___import absolute_import
from__future___import division

from__future___import  print_function

from absl import flags


import tensorflow as tf



from object_detection import model_hparams
from object_detection import model_lib




flags.DEFINE_string(

    'model_dir', None, 'Path to output model directory '
    'where event and checkpoint files will be written.')

flags.DEFINE_string('pipeline_config_path',  None,  'Path  to  pipeline  config  ''file.')

flags.DEFINE_integer('num_train_steps',  None,  'Number  of  train  steps.')
flags.DEFINE_boolean('eval_training_data', False,

                    'If training data should be evaluated for this job. Note ''that
                    one call only use this in eval-only mode, and '
                    '`checkpoint_dir` must be supplied.')

flags.DEFINE_integer('sample_1_of_n_eval_examples', 1, 'Will sample one of'

                    'every n eval input examples, where n is provided.')

flags.DEFINE_integer('sample_1_of_n_eval_on_train_examples', 5, 'Willsample '
                    'one of every n train input examples for evaluation, '
                    'where n is provided. This is only used if '
                    '`eval_training_data` is True.')

  flags.DEFINE_string(

    'hparams_overrides', None, 'Hyperparameter overrides, ''represented as
    a string containing comma-separated ' 'hparam_name=value pairs.')

  flags.DEFINE_string(

    'checkpoint_dir', None, 'Path to directory holding a checkpoint. If '
    '`checkpoint_dir` is provided, this binary operates in eval-only mode, '


'writing resulting metrics to `model_dir`.')flags.DEFINE_boolean(

    'run_once', False, 'If running in eval-only mode, whether to run just ''one
    round of eval vs running continuously (default).'

  )

  FLAGS = flags.FLAGS




  def main(unused_argv): flags.mark_flag_as_required('model_dir')
   flags.mark_flag_as_required('pipeline_config_path')

   config = tf.estimator.RunConfig(model_dir=FLAGS.model_dir)


   train_and_eval_dict = model_lib.create_estimator_and_inputs(
     run_config=config,
     hparams=model_hparams.create_hparams(FLAGS.hparams_overrides),
     pipeline_config_path=FLAGS.pipeline_config_path,
     train_steps=FLAGS.num_train_steps,
    sample_1_of_n_eval_examples=FLAGS.sample_1_of_n_eval_examples,
    sample_1_of_n_eval_on_train_examples=(

        FLAGS.sample_1_of_n_eval_on_train_examples))
estimator = train_and_eval_dict['estimator'] train_input_fn
= train_and_eval_dict['train_input_fn'] eval_input_fns =
train_and_eval_dict['eval_input_fns']

eval_on_train_input_fn = train_and_eval_dict['eval_on_train_input_fn']
predict_input_fn = train_and_eval_dict['predict_input_fn']

train_steps = train_and_eval_dict['train_steps']




if FLAGS.checkpoint_dir:

  if FLAGS.eval_training_data:
    name = 'training_data'

    input_fn = eval_on_train_input_fn

  else:

    name = 'validation_data'

    # The first eval input will be evaluated.
    input_fn = eval_input_fns[0]

  if FLAGS.run_once:
    estimator.evaluate(input_fn,

                       num_eval_steps=None, checkpoint_path=tf.train.latest_checkpoint(

                           FLAGS.checkpoint_dir))

  else:

    model_lib.continuous_eval(estimator, FLAGS.checkpoint_dir, input_fn,
                              train_steps, name)
  else:
    train_spec, eval_specs = model_lib.create_train_and_eval_specs(train_input_fn,
        eval_input_fns,
        eval_on_train_input_fn,
        predict_input_fn,
        train_steps,
        eval_on_train_data=False)


    # Currently only a single Eval Spec is allowed. tf.estimator.train_and_evaluate(estimator,
    train_spec,  eval_specs[0])



if__name___== '_main_':
  tf.app.run()
```

GITHUB LINK :

https://github.com/IBM-EPBL/IBM-Project-10353-1659173520