PROJECT REPORT

Emerging Methods for Early Detection of Forest Fires

TEAM ID : PNT2022TMID37312

TEAM MEMBERS :

SONIA V [TL] – 311619104071

PADMAVATHY J [TM1]- 311619104052

RAKSHANA S [TM2] – 311519104055

SAILAKSHMI M [TM3]- 311619104059

INTRODUCTION

OVERVIEW :

- The environmental challenges the world faces nowadays have never been greater or more complex. Global areas covered by forests and urban woodlands are threatened by natural disasters that have increased dramatically during the last decades, in terms of both frequency and magnitude. Large-scale forest fires are one of the most harmful natural hazards affecting climate change and life around the world. Thus, to minimize their impacts on people and nature, the adoption of well-planned and closely coordinated effective prevention, early warning, and response approaches are necessary.

- This paper presents an overview of the optical remote sensing technologies used in early fire warning systems and provides an extensive survey on both flame and smoke detection algorithms employed by each technology. Three types of systems are identified, namely terrestrial, airborne, and spaceborne-based systems, while various models aiming to detect fire occurrences with high accuracy in challenging environments are studied. Finally, the strengths and weaknesses of fire detection systems based on optical remote sensing are discussed aiming to contribute to future research projects for the development of early warning fire systems.

PURPOSE :

Forest fires are a universal problem that both confronts and confounds many countries. Such fires not only destroy large number of natural resources, but also destroys wildlife and their natural habitat, wreaks general havoc on ecosystems and creates environmental pollution. For all that, firefighting is one of today's most important matters for natural and environmental resource protection and preservation.

LITERATURE SURVEY

EXISTING PROBLEM :

- Forest-fire detection is a real-time problem. In fact, early fire detection should be carried out in few seconds or minutes at large. Moreover, the location of fire with enough resolution is also very important. The combination of minimal delay and resolution makes not yet some detection techniques such as satellite-based techniques (Rauste, 1996). However, these satellite technologies seem to be very useful to activate early detection systems, to tune their parameters according to the current conditions, and to validate alarms. The variability of the detection conditions in natural environments plays a significant role. In fact, the detection problem is more complex than in other industrial fields, and then, the direct application of some detection technologies fails. That is particularly true in detection systems based on the analysis of visual images. Visual sensors only can be used in some conditions (appropriated lighting conditions) and the reliability of the detection process is low when considering all possible forest environments. However, some existing systems based on colour analysis (De Vries and Den Breejen E., 1993) seem to be very promising.

- Infrared detection is the basis of some existing detection systems such as the BOSQUE system from FABA-BAZAN (Gandia et al., 1994) and the BSDS system from FISIATELETRON (Laurent and Neri, 1996). The performance of these systems varies. All the systems are able to detect early fires of small dimensions from several Kilometres. The Bosque system is capable of detecting a one square meter at ten kilometres or a ten square meter fire at twenty kilometres (Gandia et al., 1994). The cost and maintenance of the high-resolution infrared cameras with cooling systems is currently a significant drawback. However, new technology developments such as the uncooled infrared cameras based on microbolometers could significantly cut the cost in the near future (Unewisse et al., 1995). Another significant drawback of existing system is the false alarm rate (De Vries, 1997). Other sensors technologies have been applied such as the application of radiometers (see Fig. 1) that provides the temperature of a given point. However, these technologies can be considered mainly as a way to confirm a given alarm (Lorentz et al., 1997). Thus, it can be seen that all the existing sensors for forest-fire detection have some drawbacks. On the other hand, forest-fire detection involves substantial amount of heuristic knowledge. Experts detect forest fires using not only images but also terrain information, meteorological information, and knowledge about human activities in the field. This paper proposes the consideration of these sources of information to improve the reliability of automatic detection systems.

REFERENCES :

CITATIONS

[1] Celik (2007) proposed a generic model for fire and smoke detection without the use of sensors. Fuzzy based approach is used in this system. Color models such as YCbCr, HSV are used for fire and smoke detection. The fire is detected using YCbCr color model samples because it distinguishes luminance and chrominance. Y, Cb, Cr color channels are separated

from RGB input image. A pixel is more likely a fire pixel if intensity of Y channel is greater than channel Cb and Cr. In the above image set, the pixel is fire pixel as the intensity of Y channel is greater than Cb and Cr channel. HSV color model is used for Smoke detection as is does not show chrominance characteristics as fire. As smoke is the early indicator of fire it should be detected at lower temperature, here its color varies from white-bluish to white, the saturation is low which satisfies the HSV color model property. As like smoke, sky also has grayish color property and it may be identified as smoke. This problem is rectified by Motion Property, where sky will be removed.

[2] Paulo Vinicius Koerich Borges proposed a fire detection method based on probabilistic method and classification. Computer vision based approach is used in this approach. Though this approach is used surveillance it is also used to automatic video classification for retrieval of fire catastrophes in databases of newscast content. There are large variations in fire and background characteristics depending on the video instance. The proposed method observes the frame-toframe changes of low-level features describing potential fire regions. These features include color, area size, surface coarseness, boundary roughness, and skewness within estimated fire regions. Bayes classifier is used for fire recognition. In addition, apriori knowledge of fire events captured in videos is used to significantly improve the results.The fire region is usually located in the center of each frame. This fact is used to model the probability of occurrence of fire.

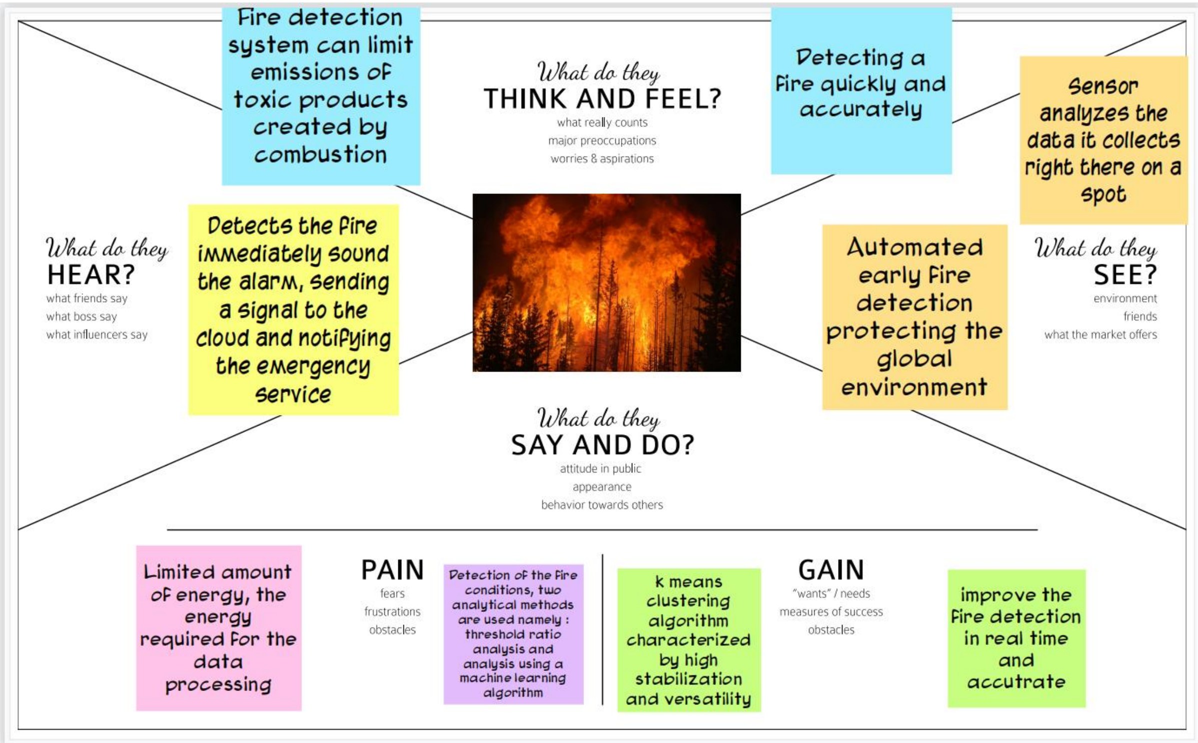
PROBLEM STATEMENT DEFINTION :

Forest fires lead to destruction of forest wealth and not only that it also destroys the flora and fauna which causes harm to biodiversity. Forest are great resources and to preserve them is a major challenge. As, they are irreparable damage to the ecosystem, so forest fire detection and prevention is utmost important and best way to tackle this problem. But the forest fire early detection and prevention is another major challenge faced all over the world. Several methods for controlling and monitoring of fires have been proposed. In earlier days, manned observation towers were used but this technique was inefficient and failed. After that satellite and camera imaging technologies were tried but this also proved inefficient and ineffective. For example, cameras were installed at different sites in forest but these provide only line of sight pictures. For a very large areas alert system is required as it is really tedious task to monitor all the images.

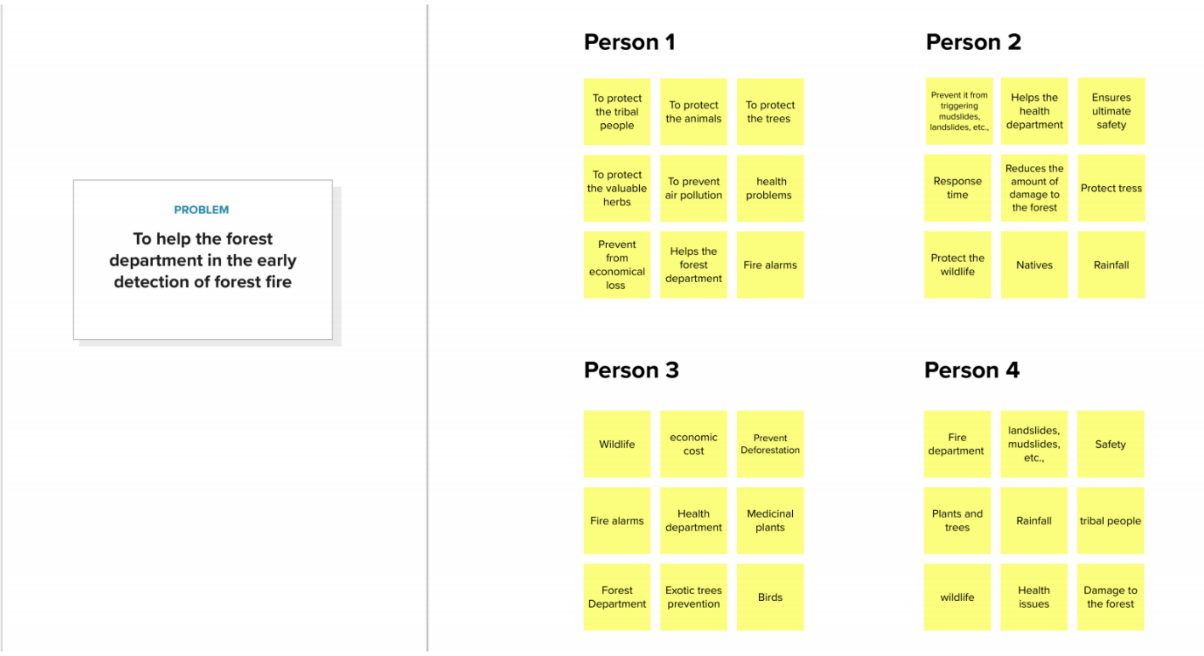
Figure 1 Amazon Rainforest Fire

IDEATION AND PROPOSED SOLUTION

EMPATHY MAP :



IDEATION AND BRAINSTORMING :

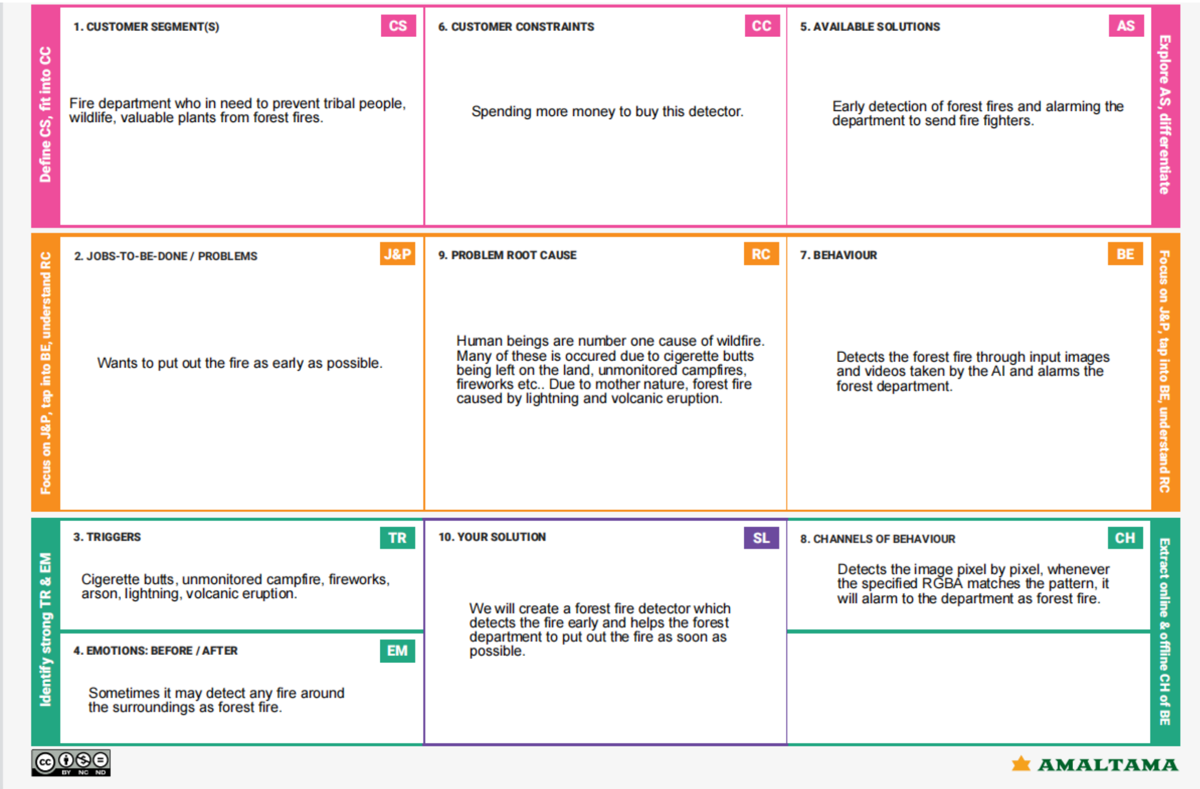


PROPOSED SOLUTION :

S.NO.	PARAMETER	DESCRIPTION
1.	PROBLEM STATEMENT (PROBLEM TO BE SOLVED)	Detection of forest fire and smoke in wildland areas is done through remote sensing-based methods such as satellites, high-resolution static cameras fixed on the ground, and unmanned aerial vehicles (UAVs).
2.	IDEA/SOLUTION DESCRIPTION	Detection of forest fires using CO2 sensors Existing detection methods such as satellite and optical systems can cover large areas; satellite systems identify infrared signatures, while optical systems look for smoke plumes.

3.	NOVELITY/UNIQUENESS	A fire detection system uses a smoke detector to detect a fire before it actually starts. An effective fire detection system eliminates damage by ensuring that a fire can be prevented before it even starts. A fire detector may also have a direct connection to an alarm monitoring centre
4.	SOCIAL IMPACT/CUSTOMER SATISFACTION	Blocked roads and railway lines, electricity, mobile and land telephone lines cut, destruction of homes and industries, and the way of life of many communities are annual news stories and the balance of the catastrophe caused by fire results in a wealth of articles, editorials and communications.
5.	BUSINESS MODEL(REVENUE MODEL)	Installing security and fire systems contractors too often leave the profits from selling service agreements for other competitors to procure from their clientele. Learn how marketing a service sales strategy can generate consistent recurring revenue from existing customers, as well create new ones
6.	SCALABILITY OF THE SOLUTION	The completely modular system makes it easily expandable and business efficient for customized fire detection, with significant cost reductions.

PROPOSED SOLUTION FIT :



REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENT

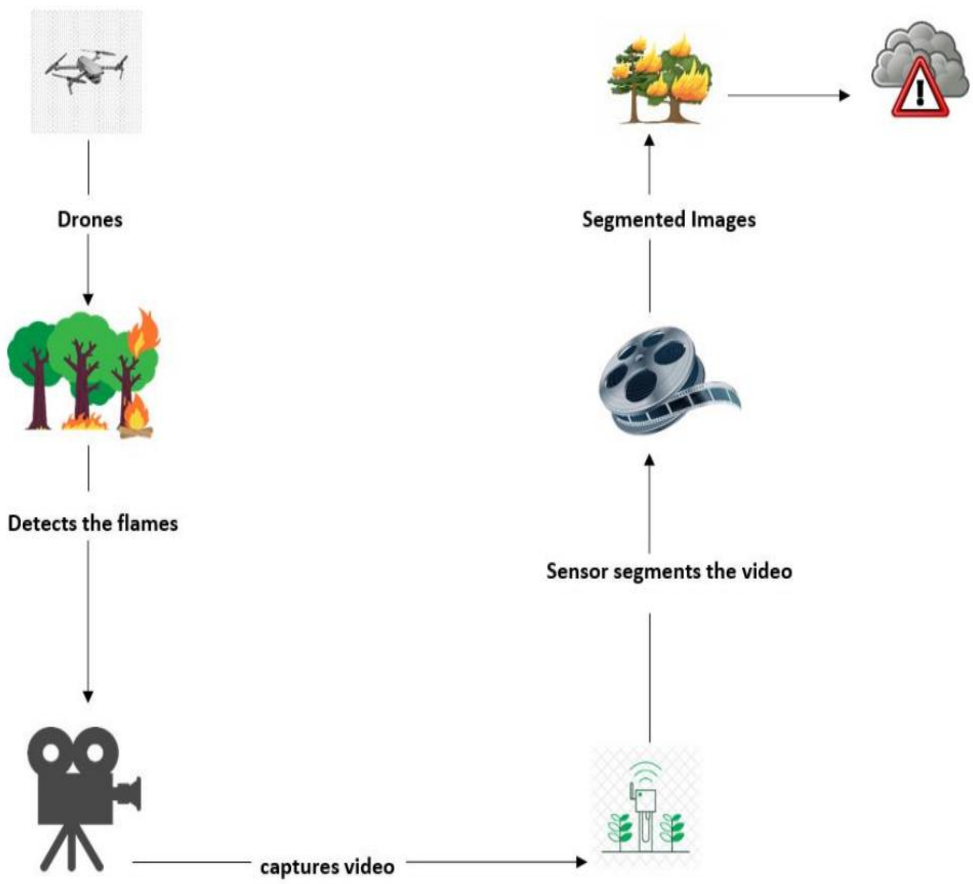
The system shall send a notification to the admin when it recognizes a fire in the image given the system shall take real inputs of satellite images and determine whether the image contains a fire or not The system shall be able to take images with a variety of sizes and convert it to one fixed image to be used throughout the application The system shall run as a service on either a Windows or Linux operating system. In the event that the computer on which the system is running shuts down, the system service should start automatically when the computer restarts Detection of forest fire and smoke in wildland areas is done through remote sensing-based methods such as satellites, high-resolution static cameras fixed on the ground, and unmanned aerial vehicles (UAVs).

NON FUNCTIONAL REQUIREMENT

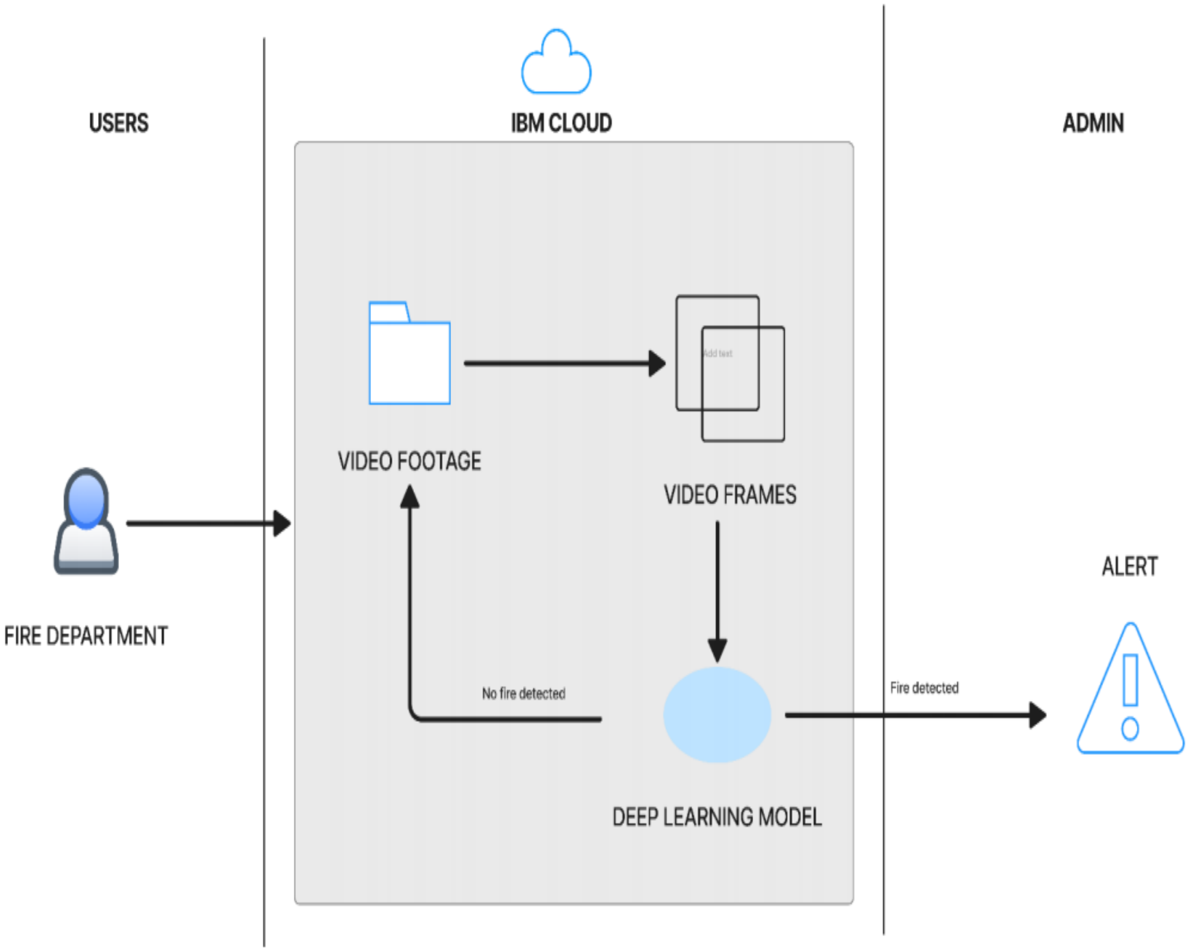
Wireless Sensor Networks (WSNs) can be used for many applications, such as industrial automatic control, remote environmental monitoring, and target tracking. The similar system is promising applications in forest fires can make a real-time monitoring and detection. Wireless sensor network consists of numerous small nodes in most situations, which small nodes are deployed in remote and inaccessible hostile environments or over large geographical areas. The large number of small nodes sense environmental changes and report them to cluster head node over network architect, which the deployment and maintenance should be easy and scalable. A new approach for forest fire monitoring and detection was present in this paper, which it is by using data aggregation in wireless sensor network. The proposed approach can provide faster and efficiently reaction to forest fires while consuming economically WSN's energy, which has been validated and evaluated in extensive simulation experiments.

PROJECT DESIGN

DATA FLOW DIAGRAM :



SOLUTION AND TECHNICAL ARCHITECTURE :



PROJECT PLANNING AND SCHEDULING

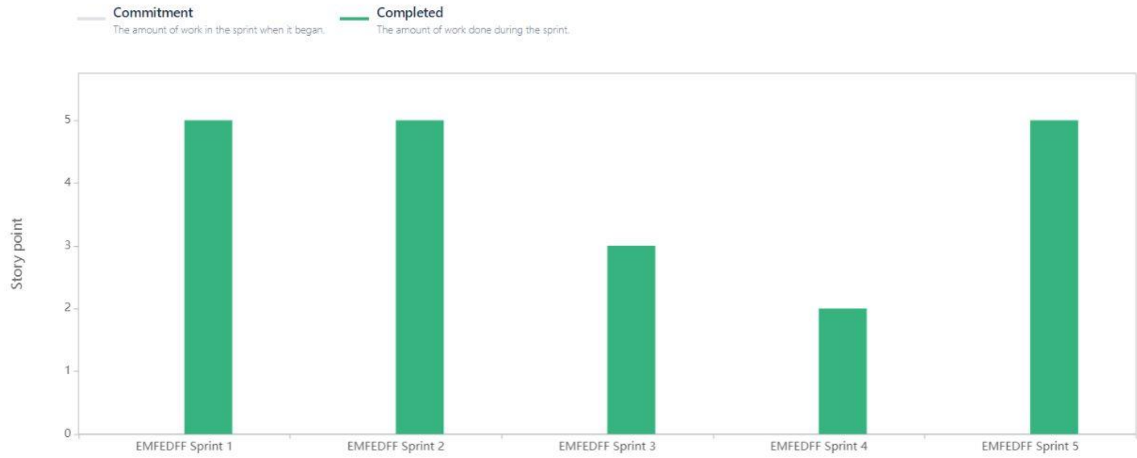
SPRINT PLANNING AND ESTIMATION :

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collecting the appropriate data sets. It is the actual data set used to train the model for performing various actions.	5	Medium	Sonia, Rakshana, Sailakshmi, Padmavathy
Sprint-2	Image Processing	USN-2	In Image processing, the dataset images are to be preprocessed before giving it to the model. The first step is usually importing the libraries that will be needed in the program	5	High	Sailakshmi, Sonia
Sprint-3	Model Building	USN-3	Here, the model is built by initializing the model, adding convolution layer, adding pooling layer, flatten layer and full connection layer which include hidden layer.	3	High	Padmavathy, Sailakshmi
Sprint-3	Video Analysis	USN-4	OpenCV is an open-source library that provides us with the tools to perform almost any kind of image and video processing	2	High	Sonia, Rakshana
Sprint-4	Train CNN Model	USN-5	The Convolutional Neural Network model will be trained and processed.	5	High	Sailakshmi, Sonia, Rakshana, Padmavathy

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	31 Oct 2022	20	31 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	9 Nov 2022	20	9 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	16 Nov 2022	20	16 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

REPORTS FROM JIRA :



Sprint	Commitment	Completed
EMFEDFF Sprint 1	0	5
EMFEDFF Sprint 2	0	5
EMFEDFF Sprint 3	0	3
EMFEDFF Sprint 4	0	2
EMFEDFF Sprint 5	0	5

Figure 2 Velocity Report Graph

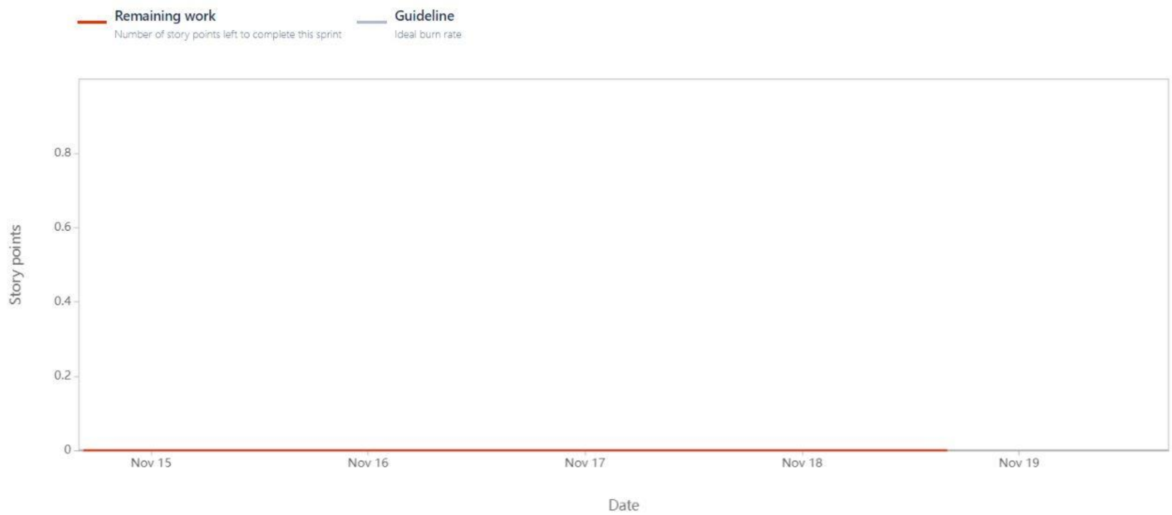


Figure 3 Burn Down Graph

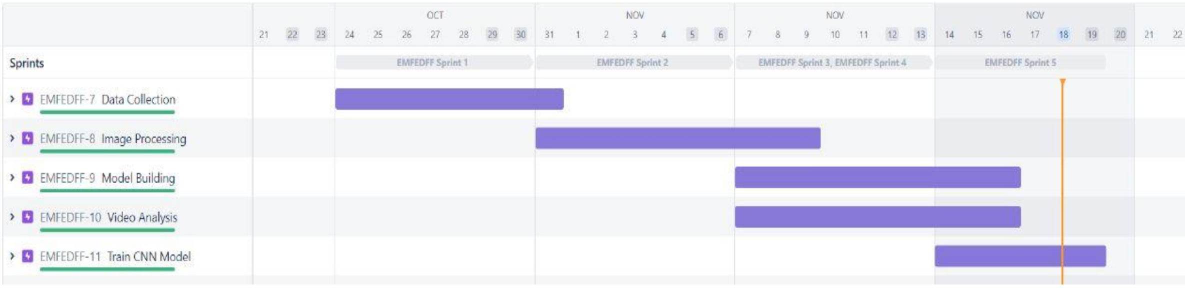


Figure 4 Road Map

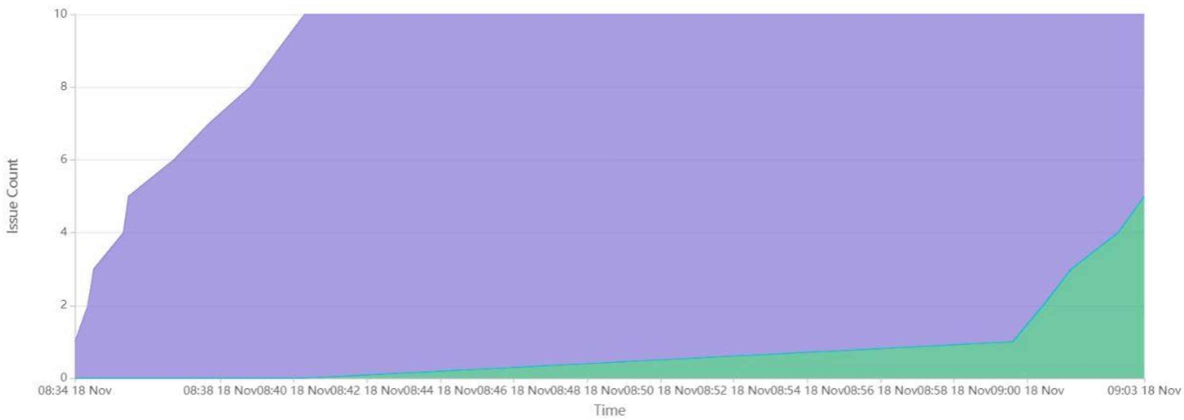


Figure 5 Cumulative Flow Graph

CODING AND SOLUTION

FEATURE 1 :

- **DATA COLLECTION AND IMAGE PROCESSING :**

In this phase, the datasets are collected for the project and the images from the datasets are processed accordingly

- CODE :

```
#!/usr/bin/env python
# coding: utf-8
# In[1]:
import zipfile
# In[2]:
import tensorflow as tf
# In[3]:
with zipfile.ZipFile('archive.zip', 'r') as zip_ref:
    zip_ref.extractall()
# In[4]:
#IMAGE PROCESSING
# In[5]:
import keras
# In[6]:
from keras.preprocessing.image import ImageDataGenerator
# In[7]:
train_datagen=ImageDataGenerator(rescale=1./255,
                                 shear_range=0.2,
                                 rotation_range=180,
                                 zoom_range=0.2,
                                 horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
# In[8]:
x_train =
train_datagen.flow_from_directory(r'./Dataset/Dataset/train_set',target
_size=(128,128),
                                     batch_size=32,
                                     class_mode='binary')
# In[9]:
x_test =
test_datagen.flow_from_directory(r'./Dataset/Dataset/test_set',target_s
ize=(128,128),
                                     batch_size=32,
                                     class_mode='binary')
# In[10]:
x_test.class_indices
```

FEATURE 2 :

❖ **MODEL BUILDING, VIDEO ANALYSIS AND TRAINING THE CNN MODEL**

In this phase, the model for detecting the forest fire is built and the video analysis for detecting fire is made. The Convolutional Neural Network model is trained.

❖ **CODE :**

```
# In[11]:
#MODEL BUILDING
# In[12]:
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')
# In[13]:
model = Sequential()
# In[14]:
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='re
lu'))
# In[15]:
model.add(MaxPooling2D(pool_size=(2,2)))
# In[16]:
model.add(Flatten())
# In[17]:
model.add(Dense(150,activation='relu'))
# In[18]:
model.add(Dense(1,activation='sigmoid'))
# In[19]:
model.compile(loss='binary_crossentropy', optimizer="adam",
metrics=["accuracy"])
# In[20]:
model.fit(x_train, epochs=5, validation_data=x_test, batch_size=32)
# In[21]:
model.save("forest1.h5")
# In[22]:
import numpy as np
# In[23]:
predictions = model.predict(x_test)
predictions = np.round(predictions)
# In[24]:
predictions
# In[25]:
print(len(predictions))
# In[26]:
from keras.models import load_model
#import image class from keras
import tensorflow as tf
from tensorflow.keras.preprocessing import image
#import numpy
```

```
import numpy as np
#import cv2
import cv2
# In[27]:
model = load_model(r'forest1.h5')
# In[28]:
def predictImage(filename):
  img1 = image.load_img(filename,target_size=(128,128))
  Y = image.img_to_array(img1)
  X = np.expand_dims(Y,axis=0)
  val = model.predict(X)
  print(val)
  if val == 1:
    print(" fire")
  elif val == 0:
      print("no fire")
# In[30]:
predictImage(r'C:\Users\PADMAVATHY\Desktop\IBM-
project\Dataset\Dataset\test_set\with fire\louisiana_forest_fire.jpg')
# In[31]:
```

#OPENCV For Video Processing

```
# In[32]:
pip install twilio
# In[33]:
pip install playsound
# In[34]:
#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[35]:
#load the saved model
model = load_model(r'forest1.h5')
#define video
video = cv2.VideoCapture('VideoCapture.mp4')
#define the features
name = ['forest','with forest']
# In[36]:
# Creating An Account in Twilio Service
# In[37]:
account_sid='ACb64c545bdd1f4134f579761c63d979d0'
auth_token='b1d8a216257673a32d5a60b6e6ce2a9e'
client=Client(account_sid,auth_token)
message=client.messages .create(
      body='Forest Fire is detected, stay alert',
      from_='+18583305228',
      to='+919361637171'
)
print(message.sid)
# In[37]:
```

#Sending Alert Message

```
# In[38]:
from keras.preprocessing import image
# In[38]:
from matplotlib import pyplot as plt
#import load model from keras.model
from keras.models import load_model
#import image from keras
from tensorflow.keras.preprocessing import image
img1 = image.load_img(r'C:\Users\PADMAVATHY\Desktop\IBM-
project\Dataset\Dataset\test_set\with
fire\deerfire_high_res_edit.jpg',target_size=(128,128))
Y = image.img_to_array(img1)
x = np.expand_dims(Y,axis=0)
val = model.predict(x)
plt.imshow(img1)
plt.show()
if val==1:
  print('Forest fire')
# In[40]:
if val==1:
  print('Forest fire')
  from twilio.rest import Client
  print('Forest fire')
  account_sid='ACb64c545bdd1f4134f579761c63d979d0'
  auth_token='b1d8a216257673a32d5a60b6e6ce2a9e'
  client=Client(account_sid,auth_token)
  message=client.messages   .create(
      body='Forest fire is detected, stay alert!',from_='+18583305228',
      to='+919361637171')
  print(message.sid)
  print("Fire detected")
  print("SMS Sent!")
elif val==0:
  print('No Fire')


#!/usr/bin/env python
# coding: utf-8
# In[4]:
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3
def __iter__(self): return 0
# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage.
It includes your credentials.
# You might want to remove those credentials before you share the
notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='TZF0HDYpFwXDzuDIpXfWmCnIpwByLziPqcp9CfptkWaL',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-
storage.appdomain.cloud')
bucket = 'cnn-donotdelete-pr-chivj41em9ccer'
object_key = 'archive.zip'
```

```
streaming_body_1 = cos_client.get_object(Bucket=bucket,
Key=object_key)['Body']
# Your data file was loaded into a botocore.response.StreamingBody
object.
# Please read the documentation of ibm_boto3 and pandas to learn more
about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
# In[5]:
from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)
# In[6]:
pwd
# In[10]:
import os
filenames=os.listdir('/home/wsuser/work/Dataset/Dataset/train_set')
# In[11]:
get_ipython().system('pip install libgl1-mesa-dev')
import tensorflow as tf
import numpy as np
from tensorflow import keras
import os
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
# In[12]:
#Define the parameters/arguments for ImageDataGenerator class
# In[13]:
train=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=
180,zoom_range=0.2,horizontal_flip=True)
train = ImageDataGenerator(rescale=1/255)
test = ImageDataGenerator(rescale=1/255)
# In[14]:
# Applying ImageDataGenerator functionality to trainset
# In[15]:
x_train =
train.flow_from_directory("/home/wsuser/work/Dataset/Dataset/train_set"
,
                                           target_size=(64,64),
                                           batch_size = 32,
                                           class_mode = 'binary' )
# In[16]:
#Applying ImageDataGenerator functionality to testset
# In[17]:
x_test =
test.flow_from_directory("/home/wsuser/work/Dataset/Dataset/test_set",
                                           target_size=(64,64),
                                           batch_size = 32,
                                           class_mode = 'binary' )
# In[18]:
x_test.class_indices
# In[19]:
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
```

```
from keras.layers import MaxPooling2D
from keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')
# In[20]:
#Initializing the model
# In[21]:
model =Sequential()
# In[22]:
# Add CNN Layer
# In[23]:
model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu
'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
# In[24]:
# Add Hidden Layer
# In[25]:
model.add(Dense(150,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
# In[26]:
# Configure the learning process
# In[27]:
model.compile(loss = 'binary_crossentropy',
              optimizer = "adam",
              metrics = ["accuracy"])
# In[28]:
model.save("/home/wsuser/work/archive(1)/forest1.h5")
# In[29]:
predictions = model.predict(x_test)
predictions = np.round(predictions)
# In[30]:
predictions
# In[31]:
print(len(predictions))
# In[33]:
from keras.models import load_model
import tensorflow as tf
from tensorflow.keras.preprocessing import image
import numpy as np
# In[34]:
model = load_model("/home/wsuser/work/archive(1)/forest1.h5")
# In[35]:
def predictImage(filename):
  img1 = image.load_img(filename,target_size=(64,64))
  Y = image.img_to_array(img1)
  X = np.expand_dims(Y,axis=0)
  val = model.predict(X)
  print(val)
  if val == 1:
    print(" fire")
  elif val == 0:
      print("no fire")
# In[37]:
predictImage("/home/wsuser/work/Dataset/Dataset/test_set/with
fire/19464620_401.jpg")
# In[38]:
pip install twilio
```

```
# In[39]:
pip install playsound
# In[40]:
import numpy as np
from keras.preprocessing import image
from keras.models import load_model
from twilio.rest import Client
from playsound import playsound
# In[41]:
model = load_model(r'/home/wsuser/work/archive(1)/forest1.h5')
name = ['forest','with forest']
# In[42]:
#Creating An Account In Twilio Service
# In[43]:
pip install pygobject
# In[44]:
def message(val):
  if val==1:
    from twilio.rest import Client
    print('Forest fire')
    account_sid='AC6be2d13a80de59f51a5fe3ba2bf9d6f1'
    auth_token='00ac87e22f4bbc807a00a5ca30eedd1e'
    client=Client(account_sid,auth_token)
    message=client.messages      .create(
        body='forest fire is detected, stay alert',
        #use twilio free number
        from_='+14793912961',
        #to number
        to='+917358579433')
    print(message.sid)
    print("Fire detected")
    print("SMS Sent!")
  elif val==0:
    print('No Fire')
# In[46]:
from matplotlib import pyplot as plt
from keras.models import load_model
from tensorflow.keras.preprocessing import image
img1 = image.load_img('/home/wsuser/work/Dataset/Dataset/test_set/with
fire/deerfire_high_res_edit.jpg',target_size=(64,64))
Y = image.img_to_array(img1)
x = np.expand_dims(Y,axis=0)
val = model.predict(x)
plt.imshow(img1)
plt.show()
message(val)
# In[48]:
img2 =
image.load_img('/home/wsuser/work/Dataset/Dataset/test_set/forest/beaut
iful_morning_mountain_forest_scenery_free_stock_photos_picjumbo_DSC0217
6.jpg',target_size=(64,64))
Y = image.img_to_array(img2)
x = np.expand_dims(Y,axis=0)
val = model.predict(x)
plt.imshow(img2)
plt.show()
message(val)
# In[49]:
```

```
from ibm_watson_machine_learning import APIClient
wml_credentials={"url":"https://us-
south.ml.cloud.ibm.com","apikey":"TFXoHzN3M76f8UM68mdo_MshGtF2Dk1H56fJ6
7oDagbV"}
client=APIClient(wml_credentials)
# In[50]:
def guid_from_space_name(client,space_name):
    space=client.spaces.get_details()
    return(next(item for item in space['resources']if
item['entity']["name"]==space_name)['metadata']['id'])
# In[51]:
space_uid=guid_from_space_name(client,'imageclassification')
print("Space UID= "+space_uid)
# In[52]:
client.set.default_space(space_uid)
# In[53]:
client.software_specifications.list()
# In[54]:
software_spec_uid=client.software_specifications.get_uid_by_name("tenso
rflow_1.15-py3.6")
software_spec_uid
# In[55]:
keras
```

TESTING

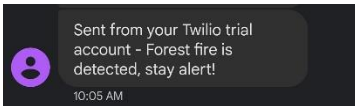
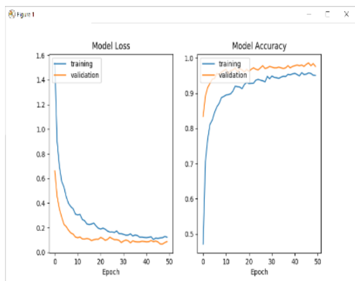
PERFORMANCE TESTING :

Project Development Phase

Model Performance Test

Model Performance Testing:

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

	Parameter	**Values**	**Screenshot**
1.	Model Summary	This project is created using AI and ML, which detects the forest fire and sends the alert message to the respective individual using Twilio API. Keras and Open CV are also effectively used in the project.	 *Figure 6Alert Message*
2.	Accuracy	Training Accuracy – 95% Validation Accuracy -62%	
3.	Confidence Score (Only Yolo Projects)	Confidence Score – 97%	

USER ACCEPTANCE TEST CASE :

Defect Analysis:

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

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	5	1	1	1	8
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	7	2	4	10	23
Not Reproduced	0	0	0	0	0
Skipped	0	0	1	1	2
Won'tFix	0	3	2	1	6
Totals	15	9	11	14	49

Test Case Analysis:

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5

Client Application	30	0	0	30
Security	2	0	0	2
Out source Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

ADVANTAGES AND DISADVANTAGES :

Wireless sensor network technology normally deploys large number of small, low cost sensors fairly densely that can observe and influence the physical world around them by gathering physical information, transforming it into electrical signals, sending it to a remote location to do some analysis, and deploying the results in different applications. By this way there is no need to build towers or set up complicated communication links such as microwave and satellite. It can be deployed anywhere even in inaccessible places.

This technology can provide real-time monitoring, where it can provide information at the ignition instance or at very small delays, depending on the node used in wake-up/sleep schedule.

This technology works on short communication links fashion. As a result, more accurate information with less delay can be provided for the fire fighters.

Using this technology for forest fire application requires a large number of randomly deployed nodes to provide a reliable network. Based on what has been discussed above, it can define the key issues of this network for this application in the following points.

(i) Localisation: all the previous work used a GPS or fixed the nodes in a known place.

(ii) Coverage: the nodes deployed randomly a full coverage almost impossible.

(iii) Network life span: For sensor nodes working on batteries, it is impossible to go back to each node in the forest and recharge it again.

(iv) Fire detection method: this is the heart of the application; it should be precise and reliable. In the previous work they used other kinds of sensors like IP cameras or other technologies like satellite images or superoptical tower cameras. They integrated the sensor nodes information with data bases, weather stations, fuel and weather indexing models, and many other models and processing techniques to reduce the number of false alarms. In order to detect the fire, each node was provided with many sensors to get the environment parameters such as temperature, wind speed, relative humidity, and fire flickering to define a fire incident in the first place.

CONCLUSION AND FUTURE SCOPE

Evolution emerges in the processing, computation, and algorithms. This strives many researchers to pay attention in many domains where they work in the processing of surveillance video streams so that abnormal or unusual actions could be detected. The usage of UAVs is recommended in the detection of forest fire due to the high mobility and ensures the coverage areas at various altitudes and locations at a low cost. Hence, an efficient and scalable UAV is used for detection. This work aims in developing the 3D model for the captured scene. YOLOv4 tiny network is deployed to detect the fire. The accuracy of the detection rate achieved through this model is 91%. The proposed model outperforms the other existing techniques in terms of detecting in the early stage. However, this model is sensitive to the forest with dense fogs and clouds. This is because smoke appears as the same as fog, and the model may misclassify the fog as smoke. As our future works, focus to meet practical detection and meet the necessity of early detection including the generation of the mixed reality model of the forest fire area that gives more information, and prevention analysis will be made easy. The 3D modelling techniques presented in this paper can also be extended to various natural disaster prediction models.

APPENDIX

Source Code 1 :

https://github.com/IBM-EPBL/IBM-Project-32408-1660209538/tree/main/Final_Submission/Final_Code

GitHub :

https://github.com/IBM-EPBL/IBM-Project-32408-1660209538

Demo Video Link :

https://github.com/IBM-EPBL/IBM-Project-32408-1660209538/tree/main/Final_Submission/Demo_Video