DETECTION OF FIRE COMBUSTION IN FORESTS USING IBM WATSON STUDIO

1.INTRODUCTION:

1.1)Overview :

Generally Artificial Intelligence is simply the ability of computer to exhibit “intelligence”. This intelligence can either mimic human intelligence or observe real world problems and intelligently find solutions. Our project Detection of fire combustion in forests using ibm Watson studio is all about building the model which comes under artificial intelligence.

Forest fires represent a real threat to human lives , ecological systems, and infrastructure and forest fires are a major environmental issue, creating economic and ecological damage while endangering human lives. There are typically about 100,000 wildfires in the United States every year. Hundreds of millions of hectares are destroyed by wildfires each year in the world. Forest fires destroy a total area of 3.5 to 4.5 million km2.Increase in forest fires resulted in an increased motivation for developing fire warning systems for early detection of wildfires. Over 9 million acres of land have been destroyed due to treacherous wildfires. Sensor technology has been widely used in fire detection, usually depending on pressure, humidity, and temperature and also chemical parameters like carbondioxide, nitrogen dioxide etc. So it requires some useful detection techniques ,such that the forest fires can be detected quickly without any late and without more energy usage.

- **Data collection**

○ Collect the dataset or create the dataset

- **Image Preprocessing**

○ Import the libraries

  - Import ImageDataGenerator Library
  - Configure ImageDataGenerator Class
  - Apply ImageDataGenerator functionalities to the trainset and testset
- **Model Building**
  - Initializing the model
  - Adding convolution layers
  - Adding pooling layers
  - Flatten layer
  - Full connection layers which include hidden layers
  - Train the model
  - Save the model
  - Make predictions
- **Video Analysis**
  - Open cv for video processing
  - Creating account in twilio service
  - Sending alert message
- **Training CNN model on IBM:**

- Train model on IBM
- Store model on IBM
- Download stored model to localsystem

1.2)PURPPOSE :

As it is difficult to predict and detect Forest fire in a sparsely populated area and it is more difficult it the prediction is done using ground-based methods like camera or video-based approach. Sattelites can be 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 authorities. As forests are protectors of earth's ecological balance. Unfortunately, the forest fire is usually only observed when it has already spread over a large area,making its control and stoppage arduous and even impossible at times.

So the main purpose of this project is to predict the fire combustion in forest and send the alert message to the higher authorities ,this also includes video streaming and alerting . This model can overcome the loss of greenery in the world bit faster.

2.LITERATURE SURVEY

Various researchers and students have published related work in national and international research papers, thesis to understand objects, types of algorithm they have used and various techniques for imagepreprocessing ,deployment of models ,building of models etc.

[1] Yinglian et al proposed forest fire disaster prevention algorithm based on image processing.this algorithm depends on fire and smoke color properties to identify the fire.

2.1)Existing Problem :

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

2. As we all are aware that there are many sensor technologies that has been widely used in fire detection which gives beep sounds when they detect fire, but sometimes they also give false alerts ,even if it sensed smoke in the forest which may also be fog or some other materials like cigar smoke etc.

2.2)Proposed Solution:

1. The proposed solution for the problem is first we need to create a dataset containing the pictures of forests without fire and forests with fire.

2. Our proposed solution is useful to overcome the exixting system in an intelligent manner.

3. Initially we import all necessary dependencies after their installation. after that we divide our datasets into trainset and testset,

then after initializing model, CNN layers, hidden layers,

Output layers, and then training the model, testing the model, at last we save our model

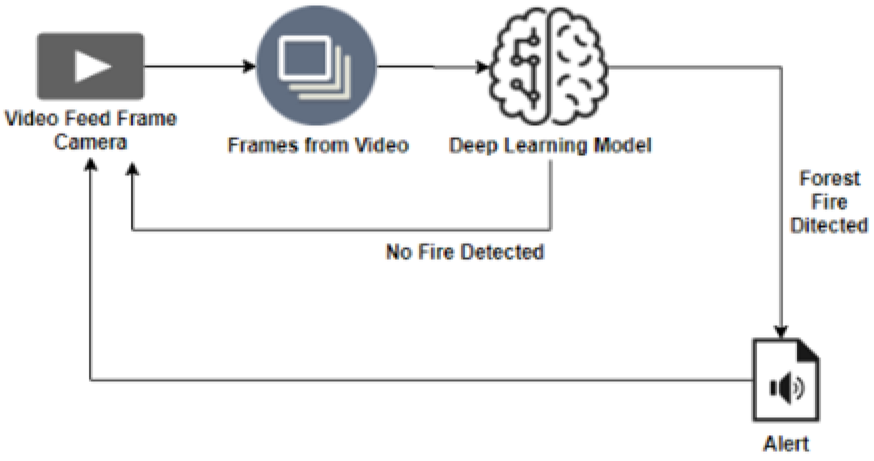
4. In this the user interacts with the web camera to read the video. Once the input image from the video frame is sent to the model, then if the fire is detected then it will be displayed on the console and the alerting sound will be generated, and an alert message will be sent to the authorities.

5. So in this manner ,our project will be useful in detecting fire combustions in the forests.

3.Theoretical Analysis

Users will give the inputs to UI and the inputs are given to the machine learning model. The model gets trained data that has already been given. The model then evaluates the inputs given by the user and predicts the forest fire combustion.

3.1) Block Diagram :



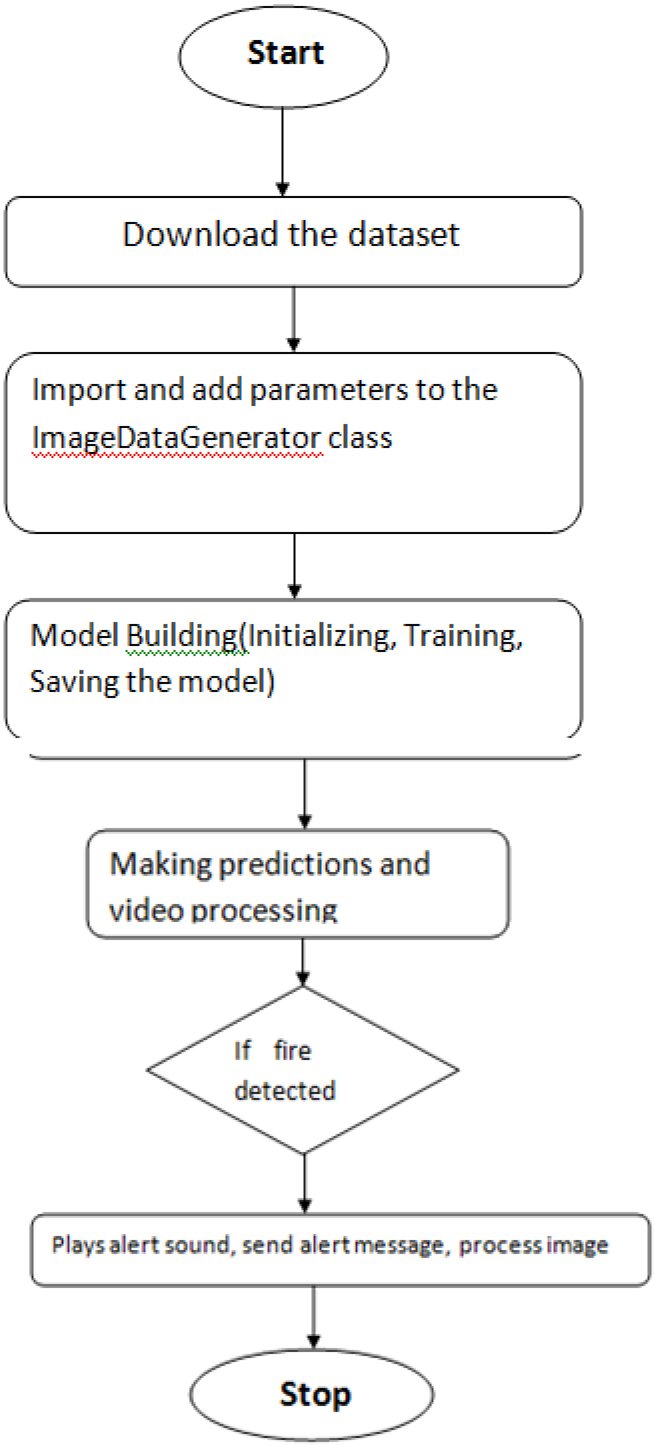
3.2) Software Designing:

First we need to write machine learning algorithm in the notebook and then we train the machine learning algorithm using the data. And then we deploy our model in IBM Watson studio and download our model from that Watson studio, then we perform video streaming and analysis. By doing this we get alert message from twilio and also alert sound .

4.Experimental Investigations:

The most common hazard in forests is forests fire.Forest fires are as old as the forest themselves. They pose a threat not only to the forest wealth but also to the entire regime to fauna and flora seriously disturbing the bio-diversity and the ecology and environment of a region. As We depend on forests for our survival, from the air we breathe to the wood we use. Besides providing habitats for animals and livelihoods for humans,forests also offer watershed protection, prevent soil erosion and mitigate climate change. So forest fires can cause a depth loss for humans, environment, animals. Forest fires are caused by Natural causes as well as Man Made causes. The prevention of human-caused fires through education and environmental modification. The major cause of this failure is the piecemeal approach to the problem. Both the national focus and the technical resources required for sustaining a systematic forest fire management programme are lacking in the country. Taking into consideration the serious nature of the problem, it is necessary to make some major improvements in the detection of forest fires.So Moreover we can use machine learning algorithms like CNN ,Opencv, Deep Learning for classifying the data.

5.Flowchart :


Start
Download the dataset
Import and add parameters to the ImageDataGenerator class
Model Building(Initializing, Training, Saving the model)
Making predictions and video processing
If fire detected
Plays alert sound, send alert message, process image
Stop

6.Result:

This forest fire detection process constitutes of three steps. These steps are mentioned below.

1. First we use the prediction by training and testing the data which is analysed during the process of accuracy calculation.

2. Input can be taken from the user or from any video which may include the images of forest with fire or without fire.

3. The output is generated related to the analysis based on the data collected related to the prediction of fire combustion in forest on various conditions.

7.Advantages and Disadvantages:

Advantages:

- It results in Early Fire Detection
- It Decreases Risk of Damage due to Fire
- Early detection can enable us to avoid serious desrtruction, so it is of extreme importance.
- Do not need as many interior lines.
- Reduce the man power and Time consumption
- Low power consumption.

Disadvantages:

- Timing and spacing of spots is some what difficult.
- Sometimes prediction can be false due to some discrepencies.
- It can give alert sound , Even if there is little fire in the image which will be disturbance for sometimes.

8.Applications :

This project will be applicable in different ways such as

- It is used to predict the fires in the forest whenever it detects fire .
- It is also used to send alert message to verified mobile number in Twilio message service.
- It is used to play an alert sound when it detects fire in the forest.
- And also displays a detected image with fire predicted message on it.

9.Conclusion:

This method proves the accuracy related to the prediction of fire related to the forests can be calculated using the datasets and model which helps in the increase of prediction of fires in the forests and manages the issues related to the forest fires prediction and maintenance. By using the dataset, by training the model we created the api of the model which will predict he forest fires.

10.Future Scope:

The prediction of Fires form the forests has a great advantage related to the data collected from the performance of the model based on various input images. This helps to increase the performance of the model to detect the fires in the forest based upon the fire images.This makes prediction of fires in the forests easier with a great accuracy.

11.Bibliography:

1. https://link.springer.com/article/10.1007/s13753-019-00233-1

2. https://www.researchgate.net/publication/300416078_Detection_of_forest_fires_using_machine_learning_technique_A_perspective

12.Appendix

12.1) Source code:

Cell-1:

```
#import keras library

import tensorflow

import keras

#import ImageDataGenerator class from keras

from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Cell-2:

```
#Define the parameters /arguments for ImageDataGenerator class
from tensorflow.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)
```

cell-3:

```
train_datagen
```

cell-4:

```
test_datagen
```

cell-5:

```
#: Applying ImageDataGenerator functionality to trainset.
#give the path of training images folder
x_train =train_datagen.flow_from_directory(r'D:\internship\project\Dataset\train_set',
                                           target_size = (64,64),
                                           batch_size = 32,
                                           class_mode = 'binary')
```

cell-6:

```
#: Applying ImageDataGenerator functionality to testset.
#give the path of testing images folder
x_test =
test_datagen.flow_from_directory(r'D:\internship\project\Dataset\test_set',
                                 target_size = (64,64),
                                 batch_size = 32,
                                 class_mode = 'binary')
```

cell-7:

```
'''import model building libraries'''


#To define linear intialisation import Sequential
from tensorflow.keras.models import Sequential
#To add layers import Dense
from tensorflow.keras.layers import Dense
#To create Convolution kernel import Convolution2D
from tensorflow.keras.layers import Convolution2D
#import Maxpooling layer
from tensorflow.keras.layers import MaxPooling2D
#import Flatten layer
from tensorflow.keras.layers import Flatten
```

cell-8:

#intializing the model

```
model =Sequential()
```

cell-9:

#add convolutional layer

```
model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
```

cell-10:

#add maxpooling layer

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

cell-11:

#add flatten layer

```
model.add(Flatten())
```

cell-12:

#add hidden layer

#model.add(Dense(150,activation='relu'))

```
model.add(Dense(kernel_initializer="uniform",activation="relu",units=150))
```

cell-13:

#add output layer

```
model.add(Dense(units=1,kernel_initializer="uniform",activation='sigmoid'))
```

cell-14:

#to check the indices

```
print(x_train.class_indices)
```

cell-15:

```
#Training the model

model.fit_generator(x_train,steps_per_epoch=14,

                    epochs=10,validation_data=x_test,

                    validation_steps=4)
```

cell-16:

```
#save the model

model.save("forest1.h5")
```

cell-17:

```
#import numpy

import numpy as np

#import load_model from keras.model

from tensorflow.keras.models import load_model

#import image class from keras

from tensorflow.keras.preprocessing import image

#load the saved model

model = load_model("forest1.h5")

#give any random image path

img = image.load_img(r'D:\internship\project\Dataset\train_set\with fire\with fire (2).jpg',

                     target_size = (64,64))

x = image.img_to_array(img)
```

```
#expand the image shape
x = np.expand_dims(x,axis = 0)
pred = model.predict_classes(x)
pred
```

cell-18:

```
print(pred[0])
```

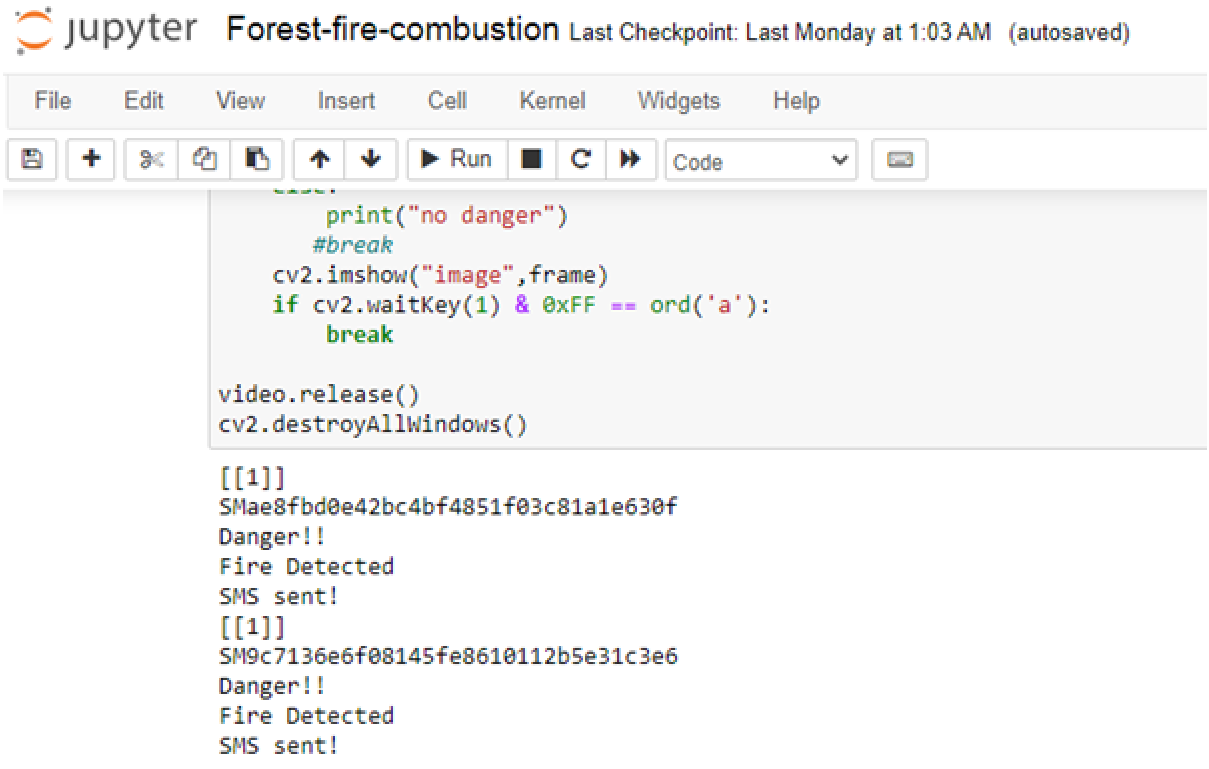
cell-19:

```
import cv2
#import facevec
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models  import load_model
from twilio.rest import Client
from playsound import playsound

model = load_model(r'forest1.h5')
video = cv2.VideoCapture(0)
name = ['forest','with fire']

while(1):
    success, frame = video.read()
    cv2.imwrite("image.jpg",frame)
```

```
img = image.load_img("image.jpg",target_size = (64,64))

x  = image.img_to_array(img)

x = np.expand_dims(x,axis = 0)

a=np.argmax(model.predict(x),axis=1)

#pred = model.predict_classes(x)

p = a[0]

print(pred)

cv2.putText(frame, "predicted  class = "+str(name[p]), (100,100),

            cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,0), 1)



pred = model.predict_classes(x)

if pred[0]==1:

    account_sid = 'AC36cb2559eb78f334eba463d94c2e52d3'

    auth_token = '74f7293f74f1fc4eccccc0f2d84f4e980'

    client = Client(account_sid, auth_token)


    message = client.messages \

    .create(

     body='Forest Fire is detected, stay alert',

     from_=' +16107561017', #twilio free number

     to='+919347515147')
```

```
        print(message.sid)
        print("Danger!!")
        print('Fire Detected')
        print ('SMS sent!')
        playsound(r'D:\internship\alert.wav')
        #break
    else:
        print("no danger")
       #break
    cv2.imshow("image",frame)
    if cv2.waitKey(1) & 0xFF == ord('a'):
        break

video.release()
cv2.destroyAllWindows()
```

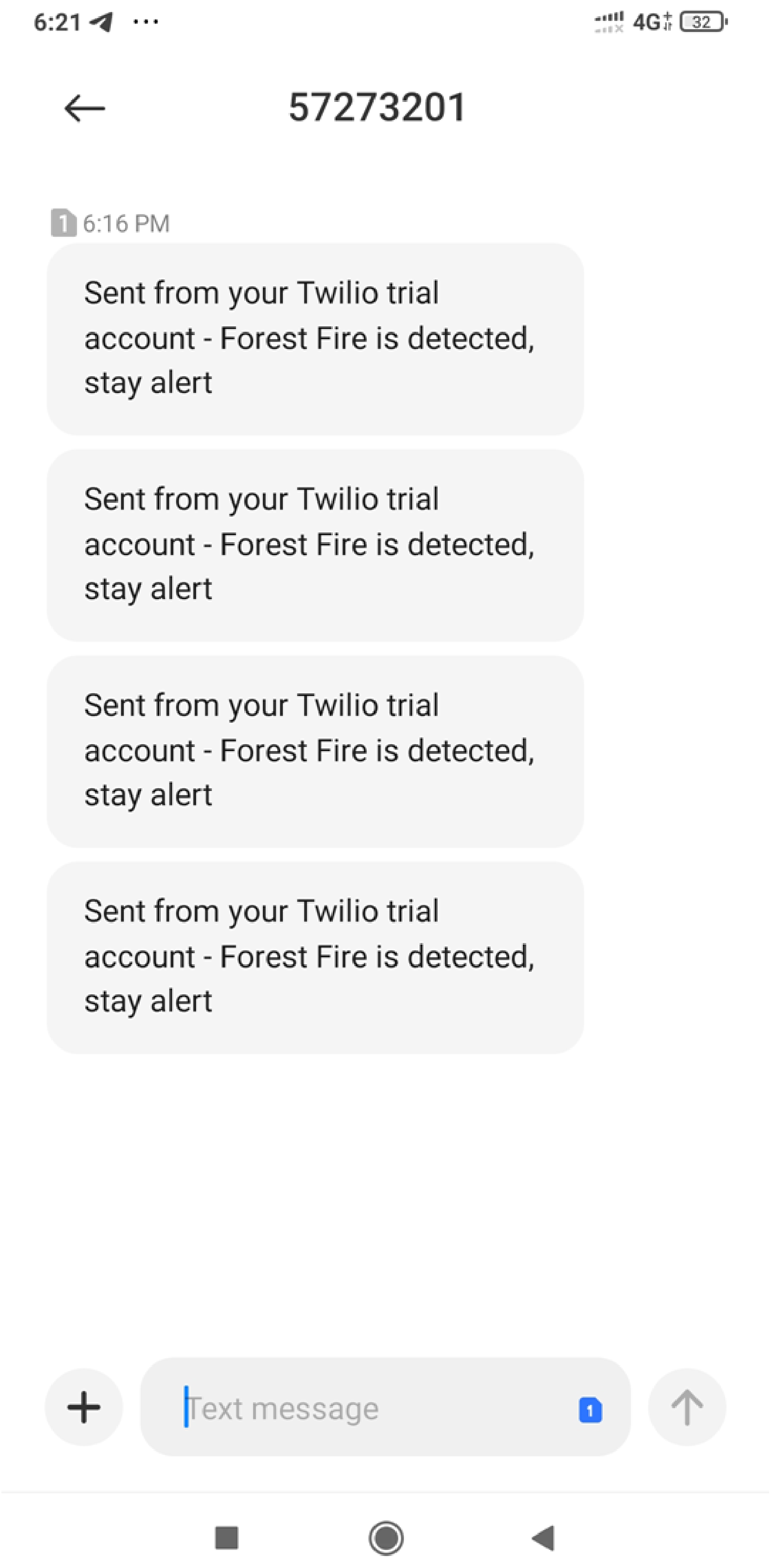
12.2)UI Output Screenshots:


jupyter
Forest-fire-combustion Last Checkpoint: Last Monday at 1:03 AM (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

```
    else:
        print("no danger")
       #break
    cv2.imshow("image",frame)
    if cv2.waitKey(1) & 0xFF == ord('a'):
        break

video.release()
cv2.destroyAllWindows()
```

```
[[1]]
SMae8fbd0e42bc4bf4851f03c81a1e630f
Danger!!
Fire Detected
SMS sent!
[[1]]
SM9c7136e6f08145fe8610112b5e31c3e6
Danger!!
Fire Detected
SMS sent!
```

1. **Alert messages from twilio service**



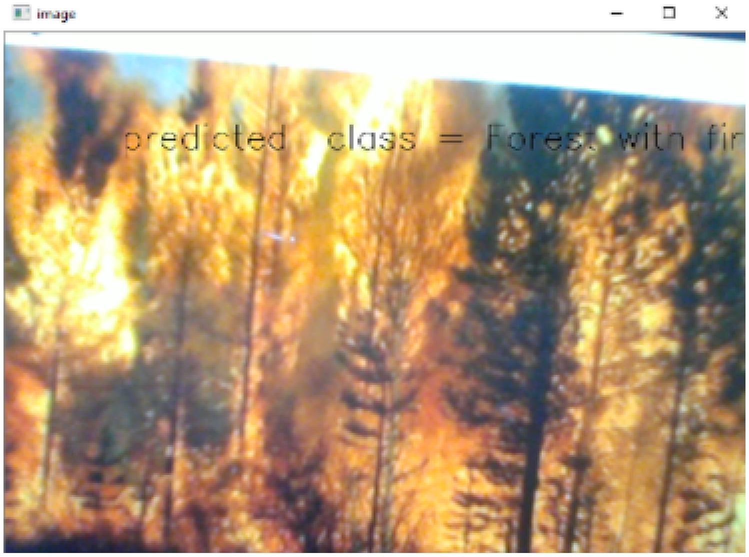
Sent from your Twilio trial account - Forest Fire is detected, stay alert

Sent from your Twilio trial account - Forest Fire is detected, stay alert

Sent from your Twilio trial account - Forest Fire is detected, stay alert

Sent from your Twilio trial account - Forest Fire is detected, stay alert

1. **And also its opens a prompt box containing the detected image with predicted message on it**



predicted class = Forest with fir

1. **And also its opens a prompt box containing the detected image with predicted message on it**