1.INTRODUCTION

1.1 Overview

Forests, which are diverse centres 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 such a 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 mother nature 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. Interestingly, a higher percentage of forest fires occur due to human activities.

The goal of the project is to develop a forest fire detection system that can identify forest fires in their early phases.

2.Literature survey

2.1 Existing problem

Every year, there are an estimated 340,000 premature deaths from respiratory and cardiovascular issues attributed to wildfire smoke.

The increasing frequency and severity of wildfires pose a growing threat 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.

Some of the existing solutions for solving this problem are:

Technology

The present technology includes particle and smoke detection systems, which are commonly used in facilities and families. These systems detect moisture in a space and determine whether the current atmosphere is safe or if an alarm should be triggered. The same way that a fire alarm works by spraying water throughout the room to put out the fire.

Fire fighter

To tackle fire problems, highly trained humans are used. Firefighters employ techniques and trucks to suppress forest fires throughout the conditions.

The priority of a firefighter is to protect people and reduce the number of people killed or injured by fire. Firefighting and property damage are the second and third priorities, respectively.

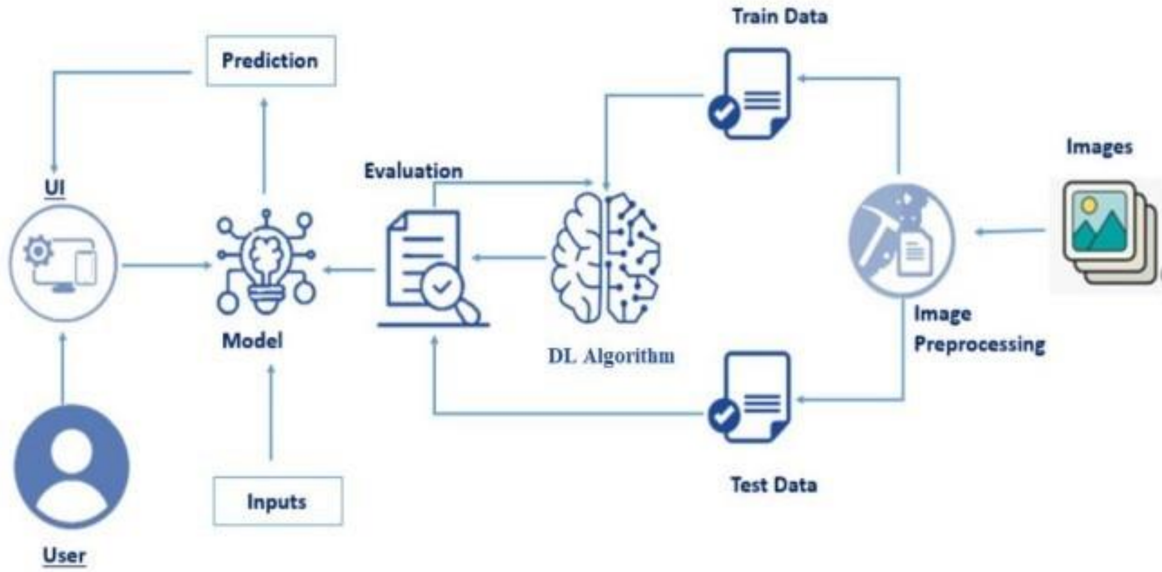
2.2 Proposed solution

The following paper describes the system to detect fire before becoming a big flame of destruction:

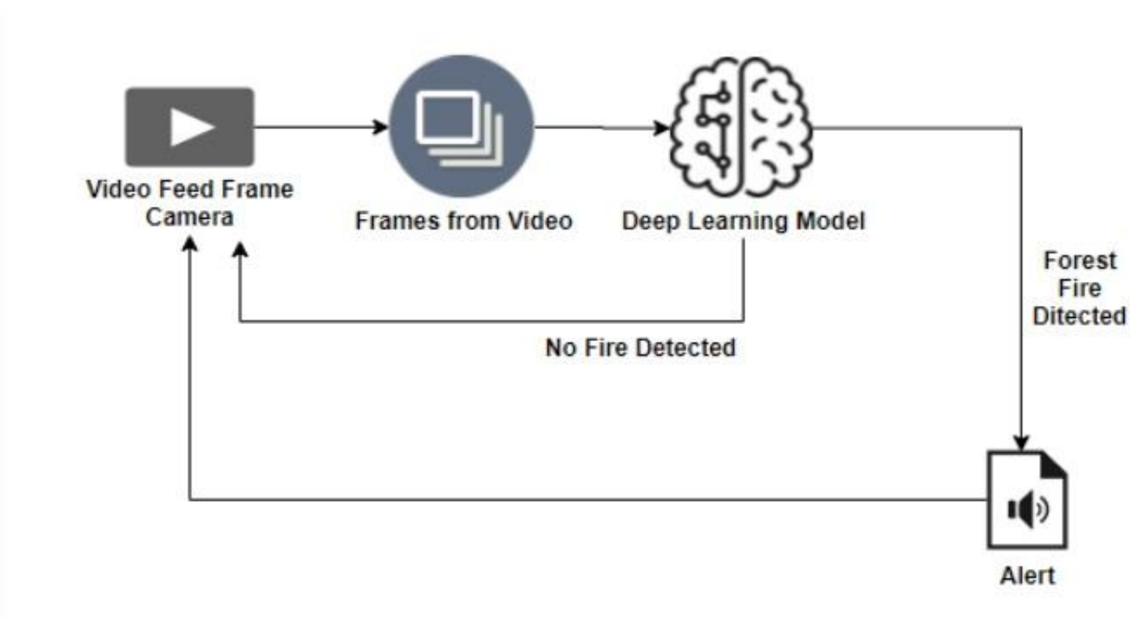
1. To build a system to detect the fire in woods through image processing.
2. To overcome the physically and molecular dynamic to detect fire for faster response.
3. A conventional neural network is being used to develop a model used to train through various images. This system will help to detect fire with before response system to prevent huge destruction.

3.Theoritical Analysis

3.1 Block Diagram



Architecture:



3.2 Hardware/Software Designing

Hardware Requirements:

Operating System	Windows, Mac, Linux
CPU (for training)	Multi Core Processors (i3 or above/equivalent)
GPU (for training)	NVIDIA AI Capable / Google's TPU
Webcam	Integrated or External with Full HD Support

Software Requirements:

Python	v3.9.0 or Above
Python Packages	flask, tensorflow, opencv-python, keras, numpy, pandas, VirtualNet, pillow
Web Browser	Google Chrome or any modern web browser.
IBM Cloud (for training)	Watson Studio- Model Training & Deployment as Machine Learning Instance

4.Experimental Investigations

Training and Testing using Dataset Provided:

Creating the model

```
[12] import tensorflow
     from tensorflow.keras.models import Sequential
     from tensorflow.keras import layers
     from tensorflow.keras.layers import Dense,Flatten
     from tensorflow.keras.layers import Conv2D,MaxPooling2D
     from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
[13] #initializing the model
     model=Sequential()
```

```
[14] #adding convolution layer
     model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
```

```
[15] #adding maxpooling layer
     model.add(MaxPooling2D(pool_size=(2,2)))
```

```
[16] #adding flatten layer
     model.add(Flatten())
```

```
[17] #adding hidden layer
     model.add(Dense(32))
```

```
#adding output layer
model.add(Dense(2,activation='softmax'))
```

5. Flowchart

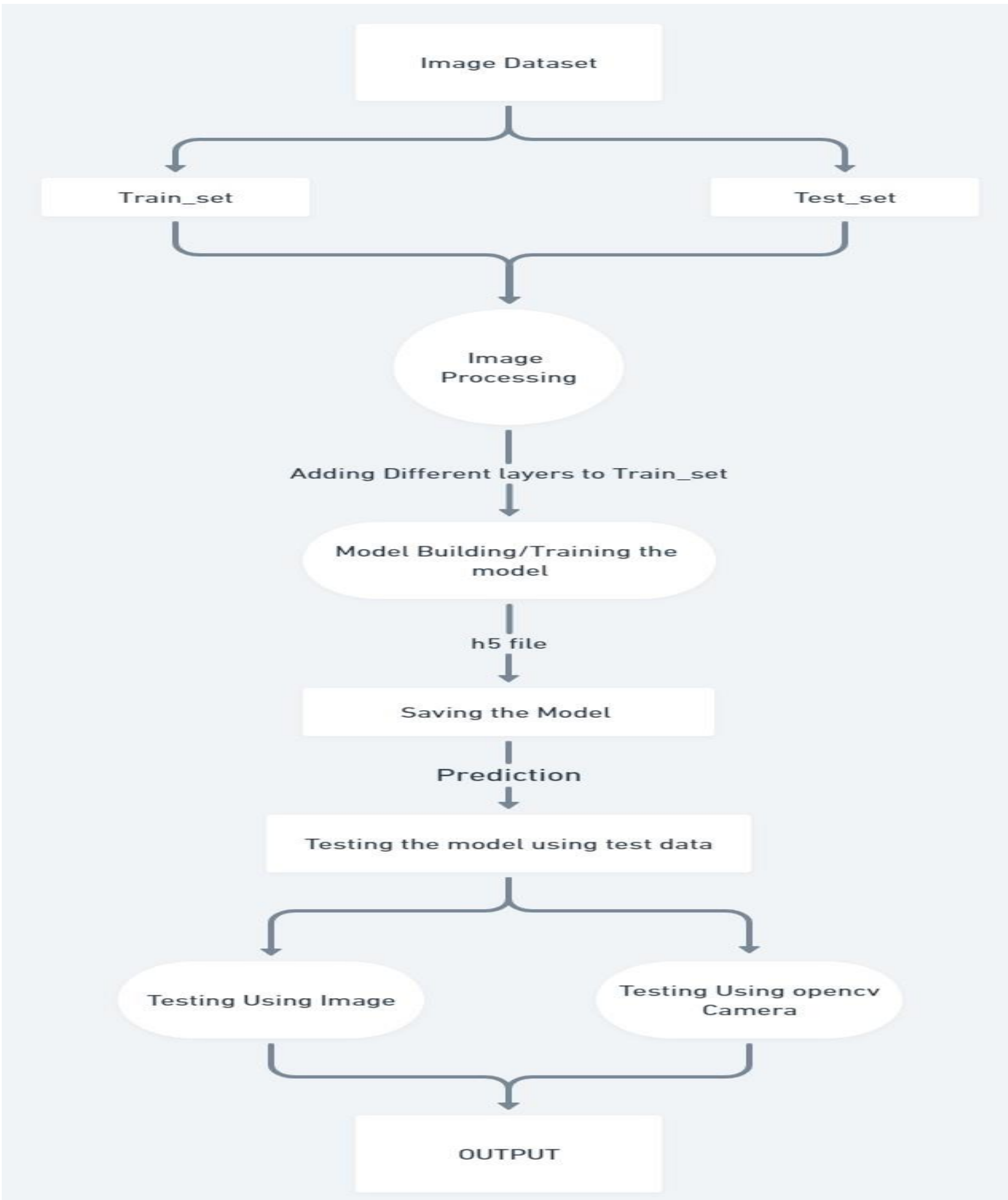


Image Dataset
Train_set
Test_set
Image
Processing
Adding Different layers to Train_set
Model Building/Training the
model
h5 file
Saving the Model
Prediction
Testing the model using test data
Testing Using Image
Testing Using opencv
Camera
OUTPUT

6.Result

The proposed procedure was implemented and tested with set of images. The sets of images of forest with fire and normal for training dataset and set of images of forest fire and normal for testing dataset. Once the model recognises the appropriate result on the screen.

<u>Some examples images of the output are provided below:</u>

Saving our model

```
[24] model.save('forest1.h5')
```

Predicting our results

```
[25] from tensorflow.keras.models import load_model
     from tensorflow.keras.preprocessing import image
     model = load_model("forest1.h5")
```

```
[26] img = image.load_img(r"/content/drive/MyDrive/forest fire/Dataset/train_set/with fire/with fire (14).jpg",target_size= (64,64))#loading of the image
     x = image.img_to_array(img)
     x = np.expand_dims(x,axis = 0)#changing the shape
     #pred = model.predict_classes(x)#predicting the classes
     preds=model.predict(x)
     preds=np.argmax(preds,axis=1)
     preds

     1/1 [==============================] - 0s 139ms/step
     array([1])
```

```
[28] img
```

```python
[29] index=['Forest','With Fire']
     result = str(index[preds[0]])
     result
```

```
'With Fire'
```

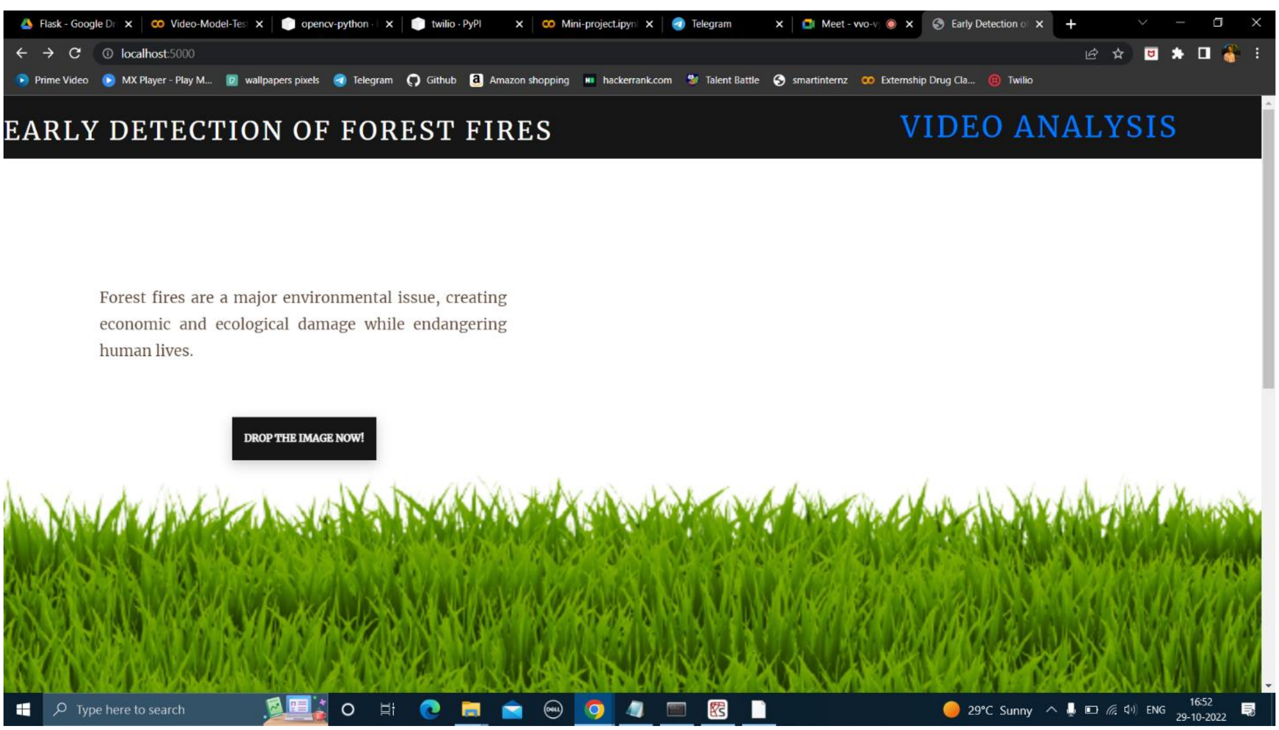
```python
[34] img = image.load_img(r"/content/drive/MyDrive/forest fire/Dataset/test_set/forest/0.64133000_1519374442_forest_deep.jpg",target_size= (64,64))#loading of the image
     x = image.img_to_array(img)
     x = np.expand_dims(x,axis = 0)
     preds=model.predict(x)
     preds=np.argmax(preds,axis=1)
     preds
```

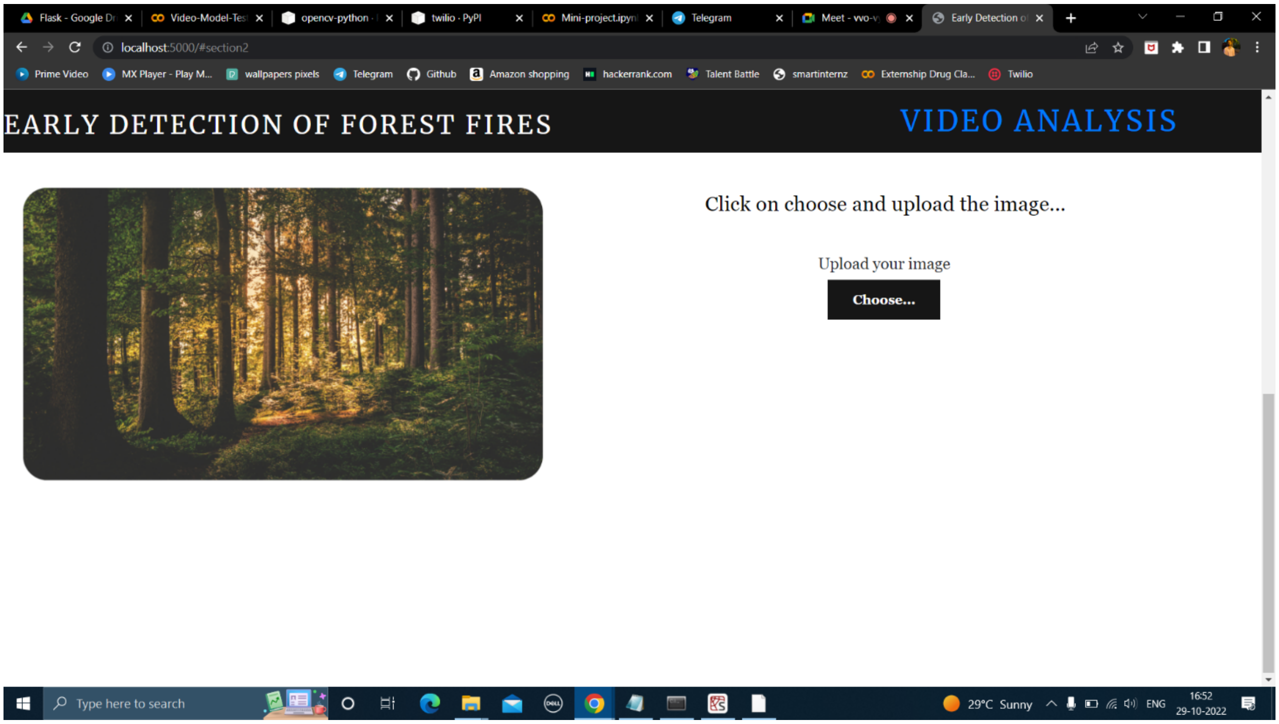
```
1/1 [==============================] - 0s 29ms/step
array([0])
```

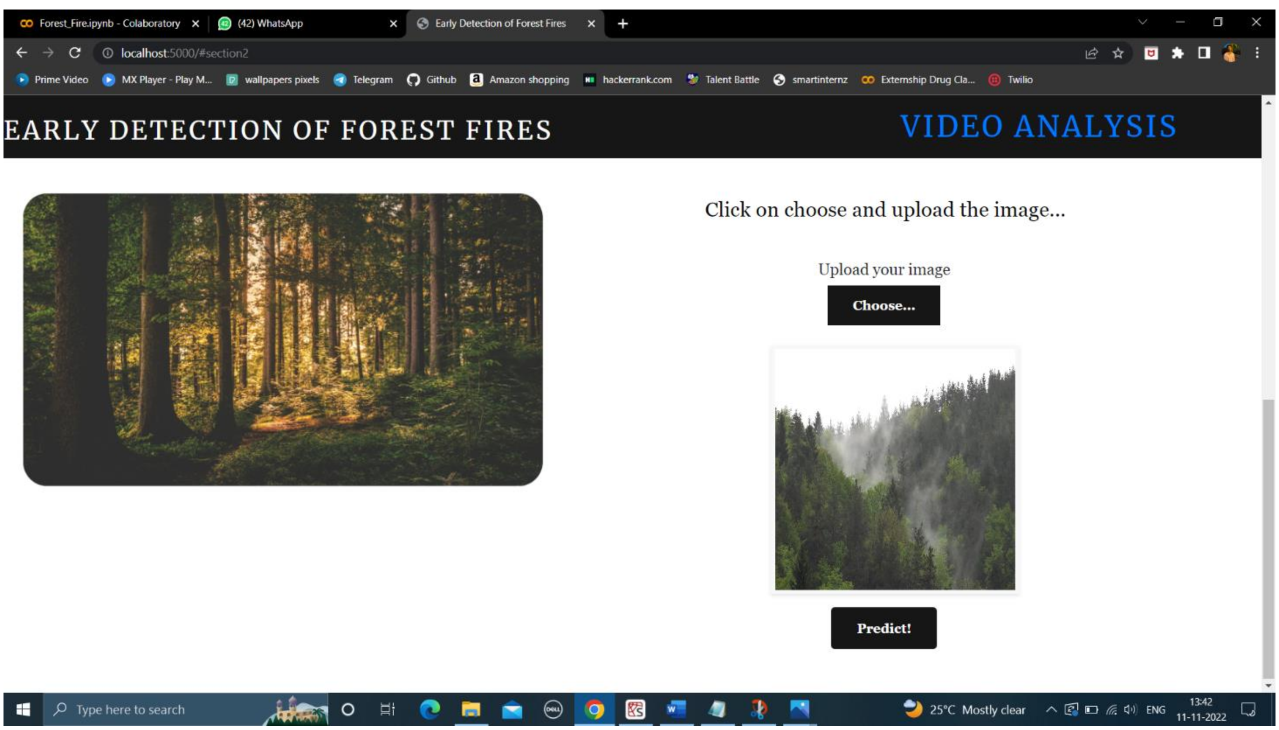
```python
img
```

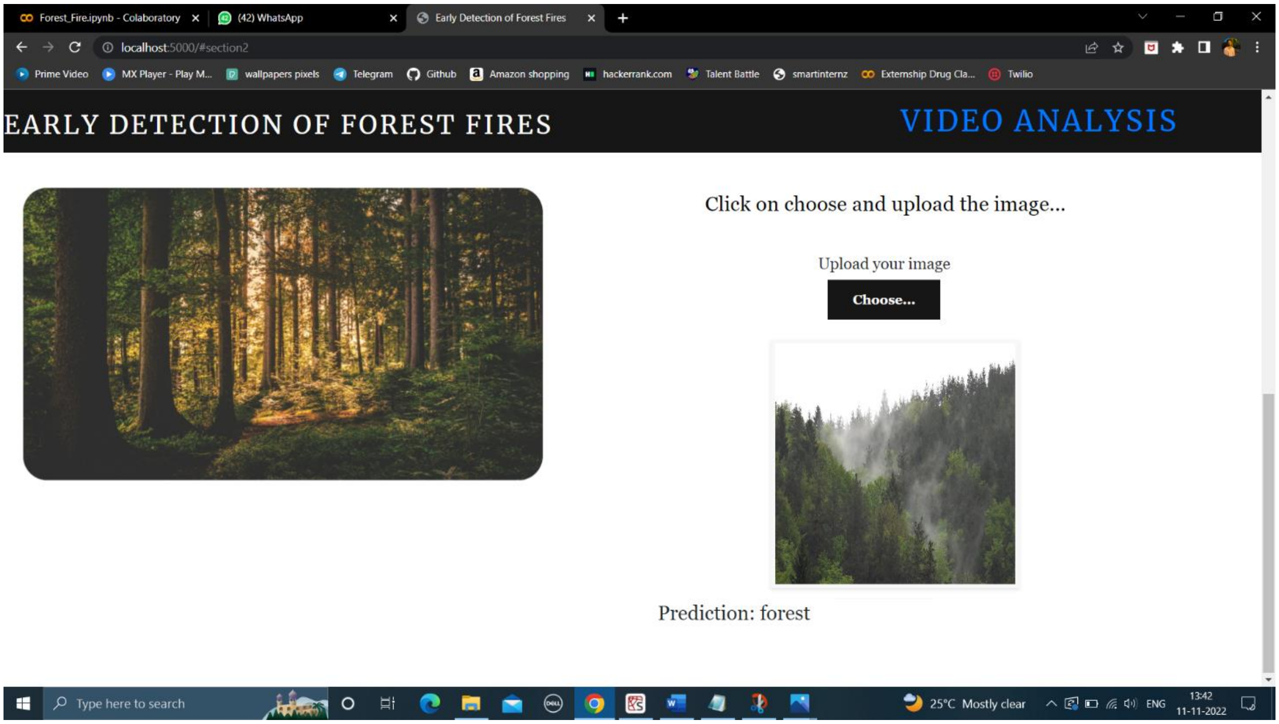
```python
[36] index=['Forest','With Fire']
     result=str(index[preds[0]])
     result
```

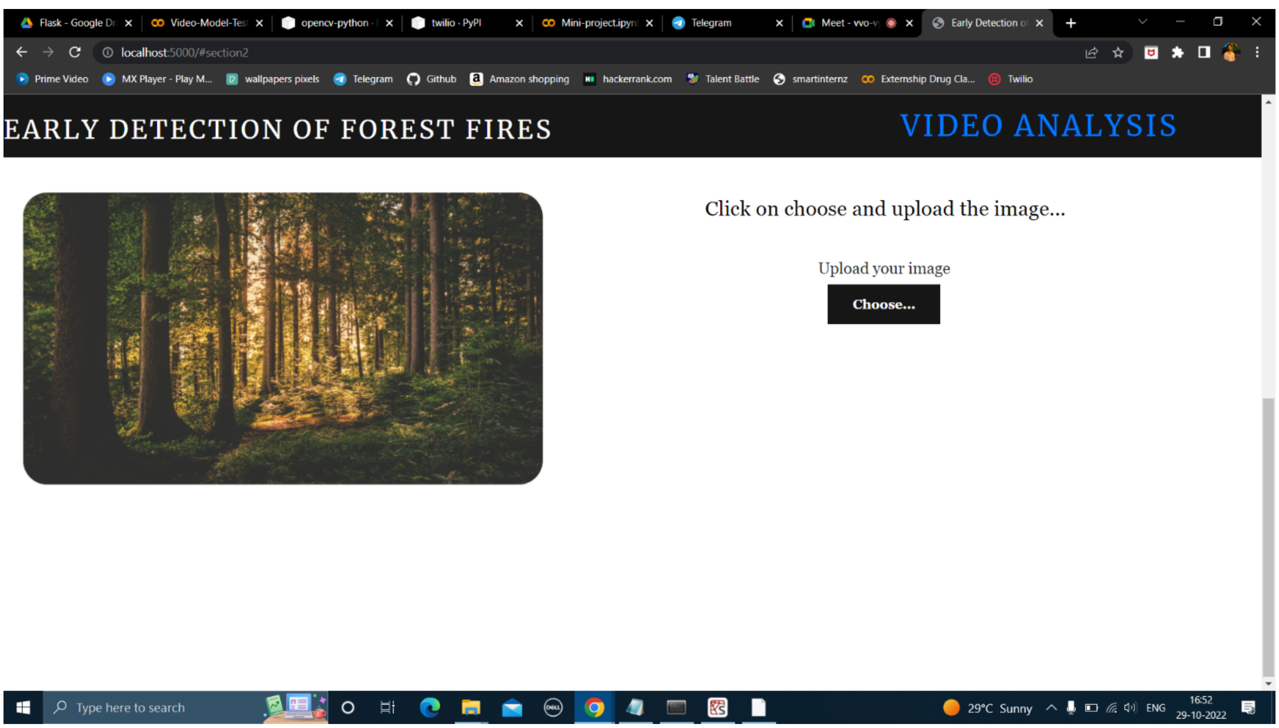
```
'Forest'
```

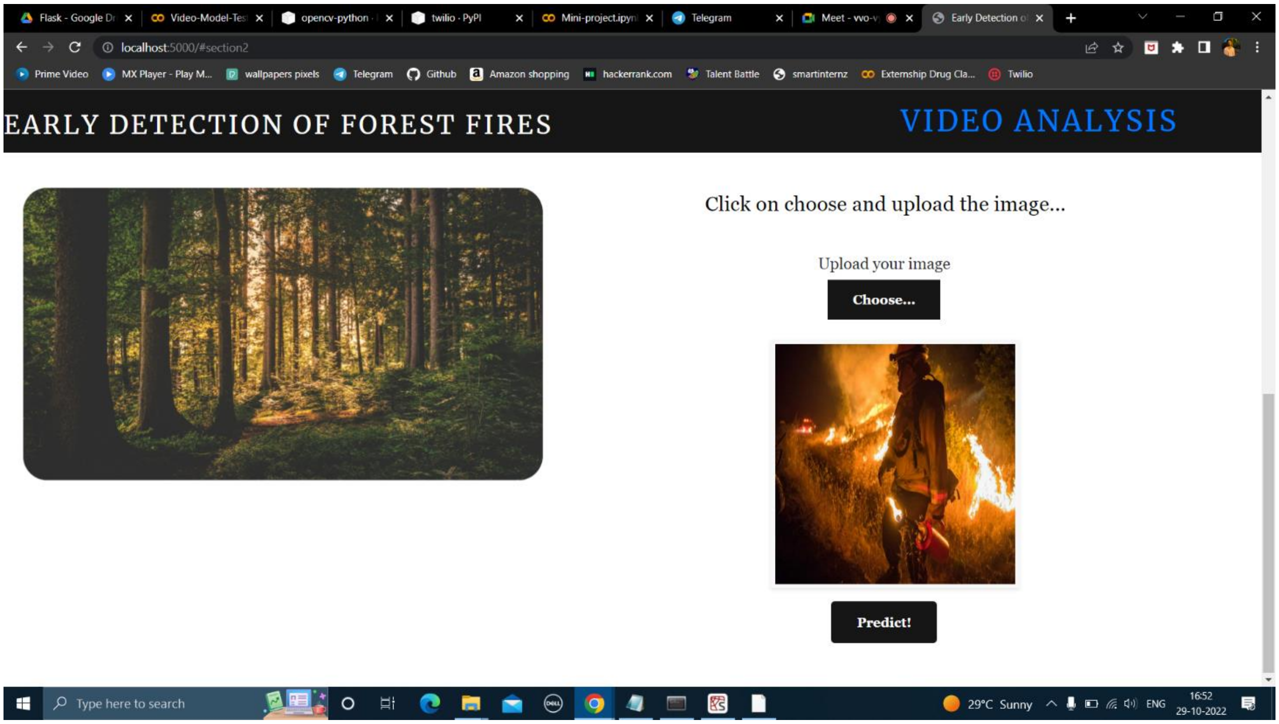



localhost:5000/#section2
EARLY DETECTION OF FOREST FIRES
VIDEO ANALYSIS
Click on choose and upload the image...
Upload your image
Choose...
Type here to search
29°C Sunny
16:52
29-10-2022

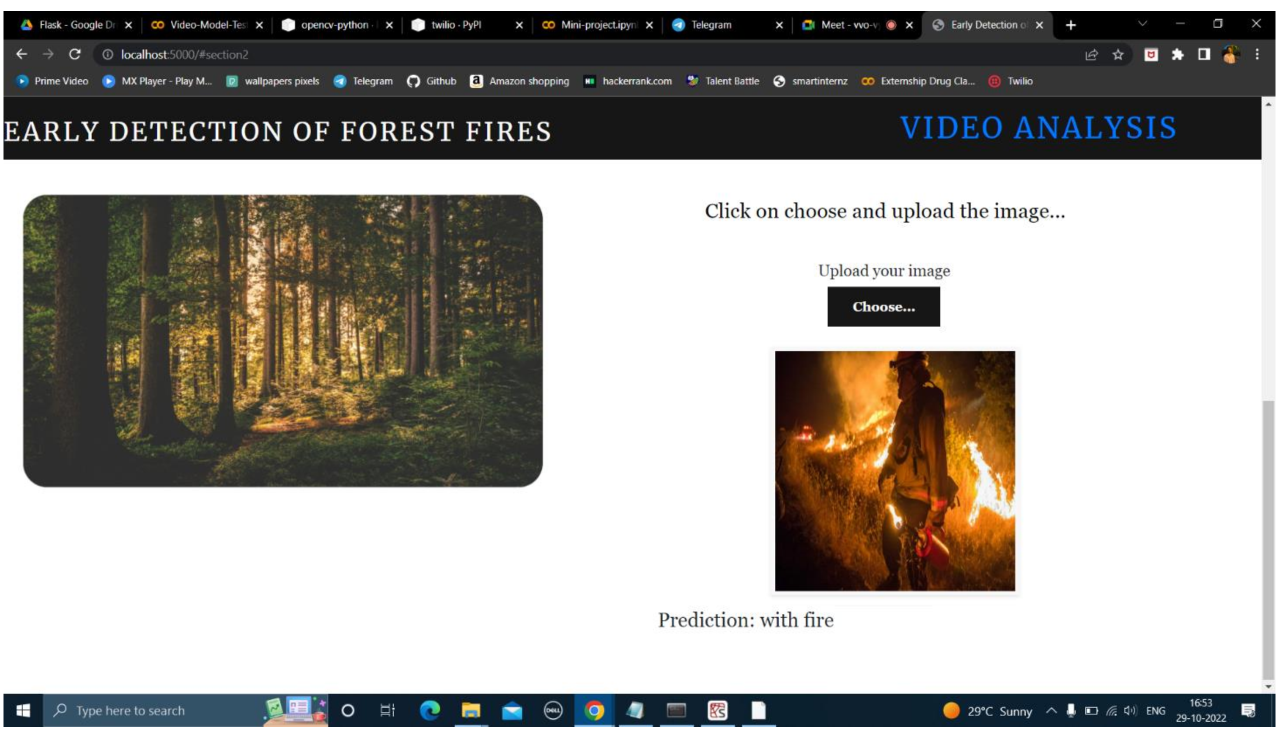

localhost:5000/#section2
EARLY DETECTION OF FOREST FIRES
VIDEO ANALYSIS
Click on choose and upload the image...
Upload your image
Choose...
Predict!
Type here to search
25°C Mostly clear
13:42
11-11-2022


Forest_Fire.ipynb - Colaboratory
(42) WhatsApp
Early Detection of Forest Fires
localhost:5000/#section2
Prime Video
MX Player - Play M...
wallpapers pixels
Telegram
Github
Amazon shopping
hackerrank.com
Talent Battle
smartinternz
Externship Drug Cla...
Twilio
EARLY DETECTION OF FOREST FIRES
VIDEO ANALYSIS
Click on choose and upload the image...
Upload your image
Choose...
Prediction: forest
Type here to search
25°C Mostly clear
ENG
13:42
11-11-2022

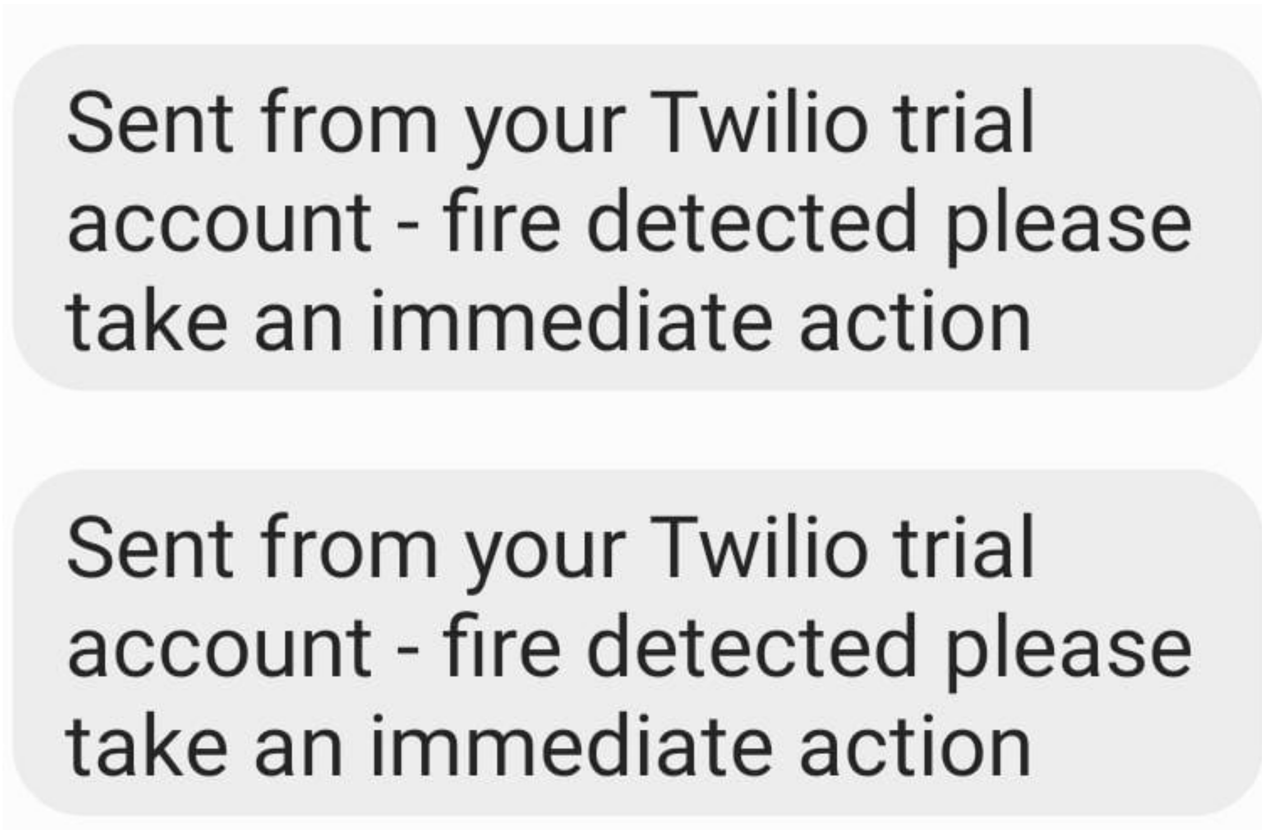

Flask - Google Dr
Video-Model-Tes
opencv-python
twilio · PyPI
Mini-project.ipyn
Telegram
Meet - wvo-v
Early Detection o
localhost:5000/#section2
Prime Video
MX Player - Play M...
wallpapers pixels
Telegram
Github
Amazon shopping
hackerrank.com
Talent Battle
smartinternz
Externship Drug Cla...
Twilio
EARLY DETECTION OF FOREST FIRES
VIDEO ANALYSIS
Click on choose and upload the image...
Upload your image
Choose...
Type here to search
29°C Sunny
ENG
16:52
29-10-2022



- In Video Analysis we can capture the image of forest through OpenCV window:



- But when we try to show the fire video the OpenCV window does not open and simply send SMS to phone number that registered



7. Advantages and Disadvantages

- **Advantages:**

1. The proposed model can be used in combination with a night camera 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 accurate outcome when detecting flame destruction on ability.
3. model can be implemented in mobile 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.

8. Applications:

1. 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 number of dangers that should be treated and those that should not. extra assistance In contacting fire fighters for assistance system

9. Conclusion

Forest fires are a major cause of rain forest and savanna degradation. This model will aid in minimising 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 recognises them and determines whether or not to send a threat alert. The model extracts data from video feeds and defines image processing into RGB data for signal response modelling.

10. Future Scope

The 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.

11.BIBLOGRAPY

1. Environment Setup:https://www.youtube.com/watch?v=5mDYijMfSzs

2. Forest fire Dataset:
https://drive.google.com/drive/folders/1vq8TRFWE7WH7_https://drive.google.com/drive/folders/1vq8TRFWE7WH7_-dsqKAmvjJAsaxx-kPQ?usp=sharingdsqKAmvjJAsaxx-kPQ?usp=sharing

3. Keras Image Processing Doc: https://keras.io/api/preprocessing/image/

4. Keras Image Dataset from Directory Doc:
https://keras.io/api/preprocessing/image/#imagedatasetfromdirect ory-function

5. CNN using TensorFlow: https://www.youtube.com/watch?v=umGJ30-15_A

6. OpenCV Basics of Processing
Image:https://www.youtube.com/watch?v=mjKd1Tzl70I

7. Flask Basics:https://www.youtube.com/watch?v=lj4I_CvBnt0

8. IBM Academic Partner Account Creation: https://www.youtube.com/watch?v=x6i43M7BAqE

9. CNN Deployment and Download through IBM Cloud:
https://www.youtube.com/watch?v=BzouqMGJ
41k

12. Appendix

App.py

```
from __future__ import division, print_function
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from flask import Flask, request, render_template,url_for
from werkzeug.utils import secure_filename
import cv2
import smtplib
from twilio.rest import Client

global graph
#graph=tf.get_default_graph()
# Define a flask app
app = Flask(__name__)
model = load_model('forest1 (2).h5')


print('Model loaded. Check http://127.0.0.1:5000/')




@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('digital.html')


@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
```

```
        # Get the file from post request
        f = request.files['image']

        # Save the file to ./uploads
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        img = image.load_img(file_path, target_size=(64,64))

        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)

        #with graph.as_default():
        preds = np.argmax(model.predict(x))
        index = ["forest","with fire"]
        print(preds)
        text = index[preds]
        return text

@app.route('/video', methods=['GET', 'POST'])
def opencv():
    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)
        pred=np.argmax(model.predict(x),axis=1)
        #pred = model.predict_classes(x)
        pred=model.predict(x)
        #p = pred[0]
        p=int(pred[0][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(x)
        pred=np.argmax(model.predict(x),axis=1)
        print(pred)
        #cv2.putText(frame, "predicted  class = "+str(name[pred]), (100,100), 
cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,0), 1)
        if pred[0]==1:
            
            cv2.putText(frame, "predicted  class = Fire Detected" ,(100,100), 
cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,0), 1)
            account_sid = 'AC8b5221b34406db8b4cc42c18afb15901'
            auth_token = '17f628fc77996909ec040b321b771173'
            client = Client(account_sid, auth_token)

            message = client.messages.create(
                messaging_service_sid='MGaf95938e5e0ffcf0cae9f7732329b67d',
             body='fire detected please take an immediate action',
             to='+919347657960'  )
            print(message.sid)

            print('Fire Detected')
            print ('SMS sent!')
            #return 'Fire Detected'
            return render_template('video.html',pred="Fire Detected Alert Notification Sent")
            break
        else:
            cv2.putText(frame, "predicted  class = No Danger",(100,100), 
cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,0), 1)
            print("no danger")
           #break
        cv2.imshow("image",frame)

        if cv2.waitKey(1) & 0xFF == ord('a'):
            break

    video.release()
    cv2.destroyAllWindows()
    return render_template('digital.html')

if __name__ == '__main__':
    app.run(threaded = False)
```