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.

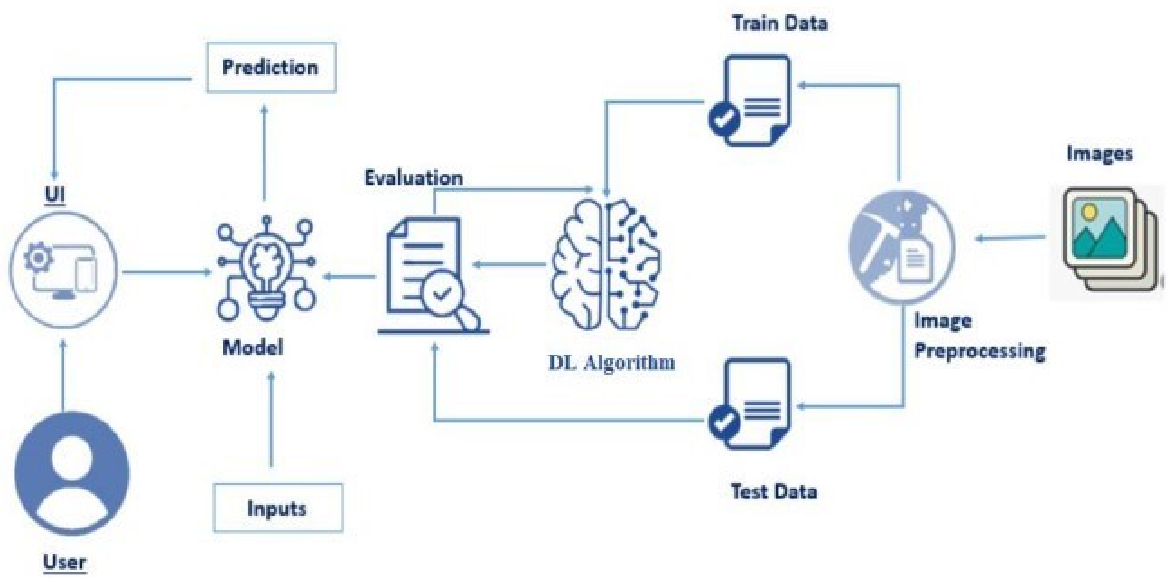
2.2 Proposed solution

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

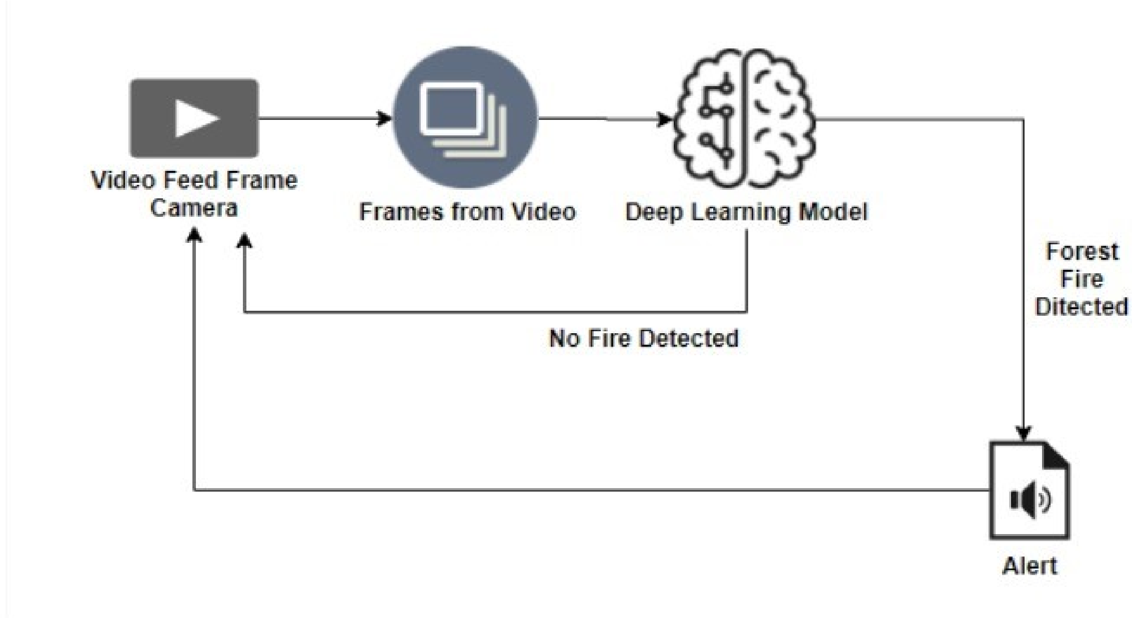
1. To build a system to detect the fire in woods through image processing.
2. To overcome the physically and molecular dynamic to detect fire for faster response.
3. A conventional neural network is being used to develop a model used to train through various images. This system will help to detect fire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

Model Building

In [4]:
```
'''import model building libararies'''

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')
```

In [5]:
```
#initializing the model
model=Sequential()
```

In [6]:
```
#add convolution layer
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```

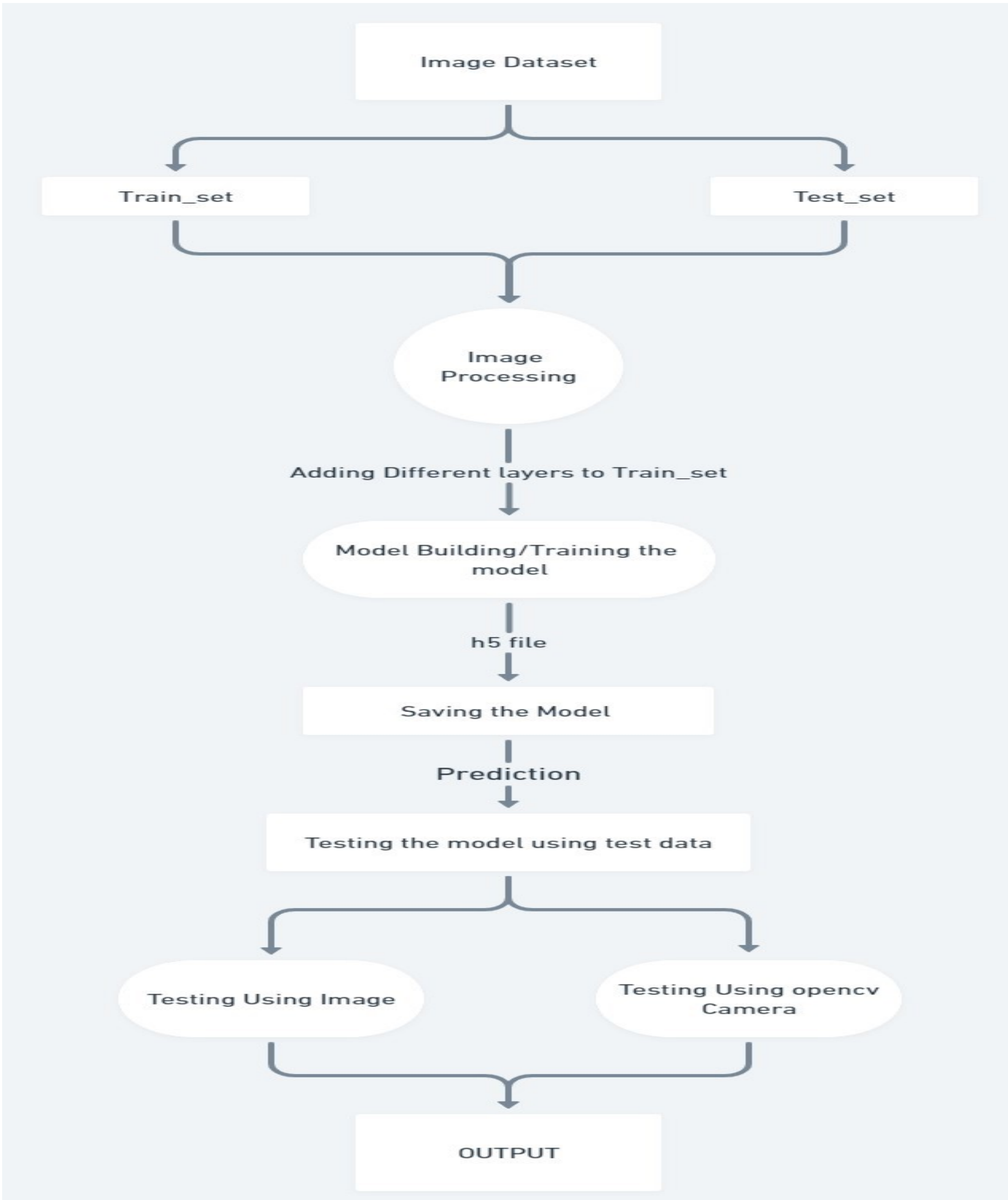
In [7]:
```
#add maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [8]:
```
#add flatten layer
model.add(Flatten())
```

In [9]:
```
#add hidden layer
model.add(Dense(units=150,activation='relu'))
```

In [10]:
```
#add output layer
model.add(Dense(units=1,activation='sigmoid'))
```

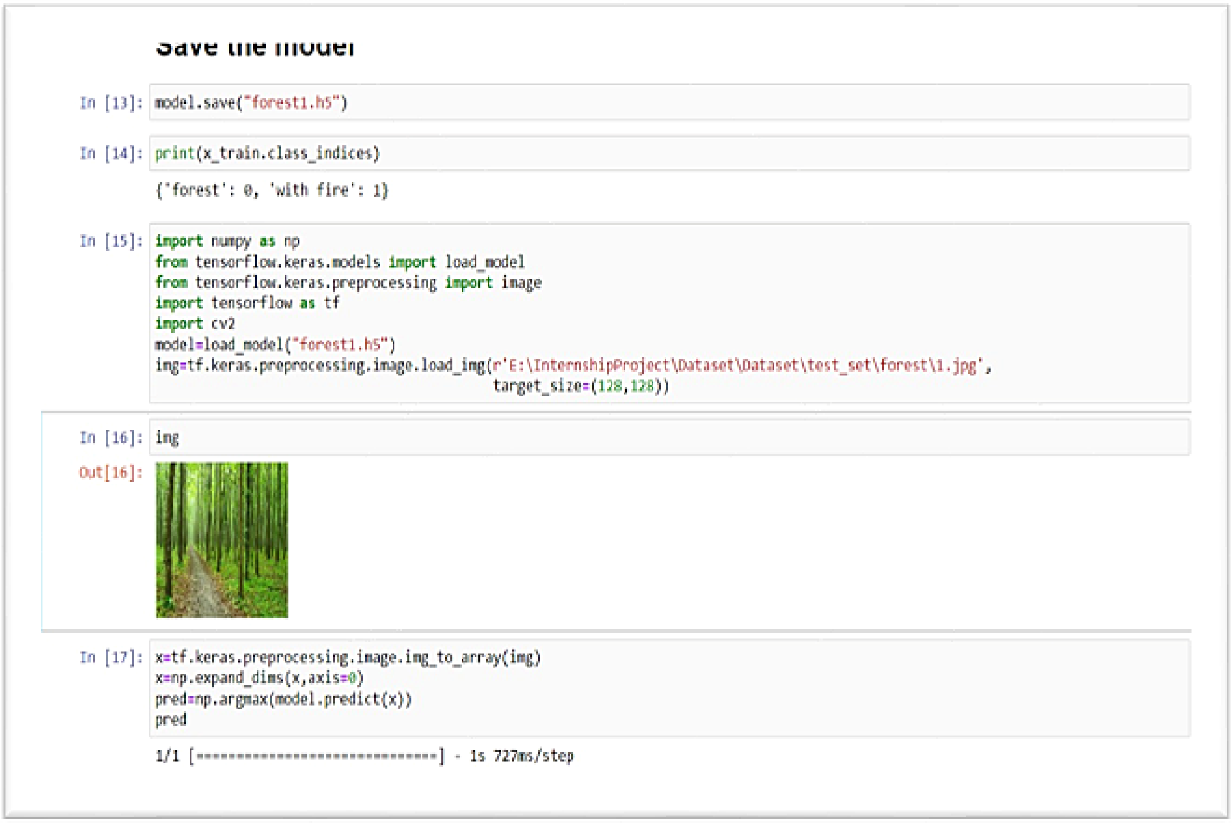
5. Flowchart



6.Result

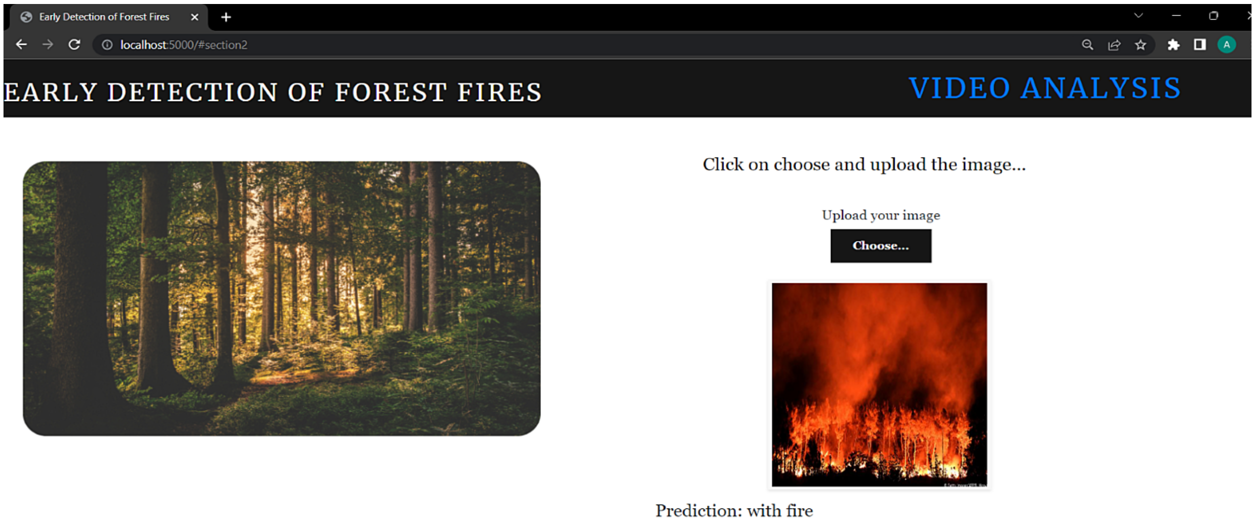
The proposed procedure was implemented and tested with set of images. The sets of images of forest with fire and normal for training dataset and set of images of forest fire and normal for testing dataset. Once the model recognises the appropriate result on the screen.

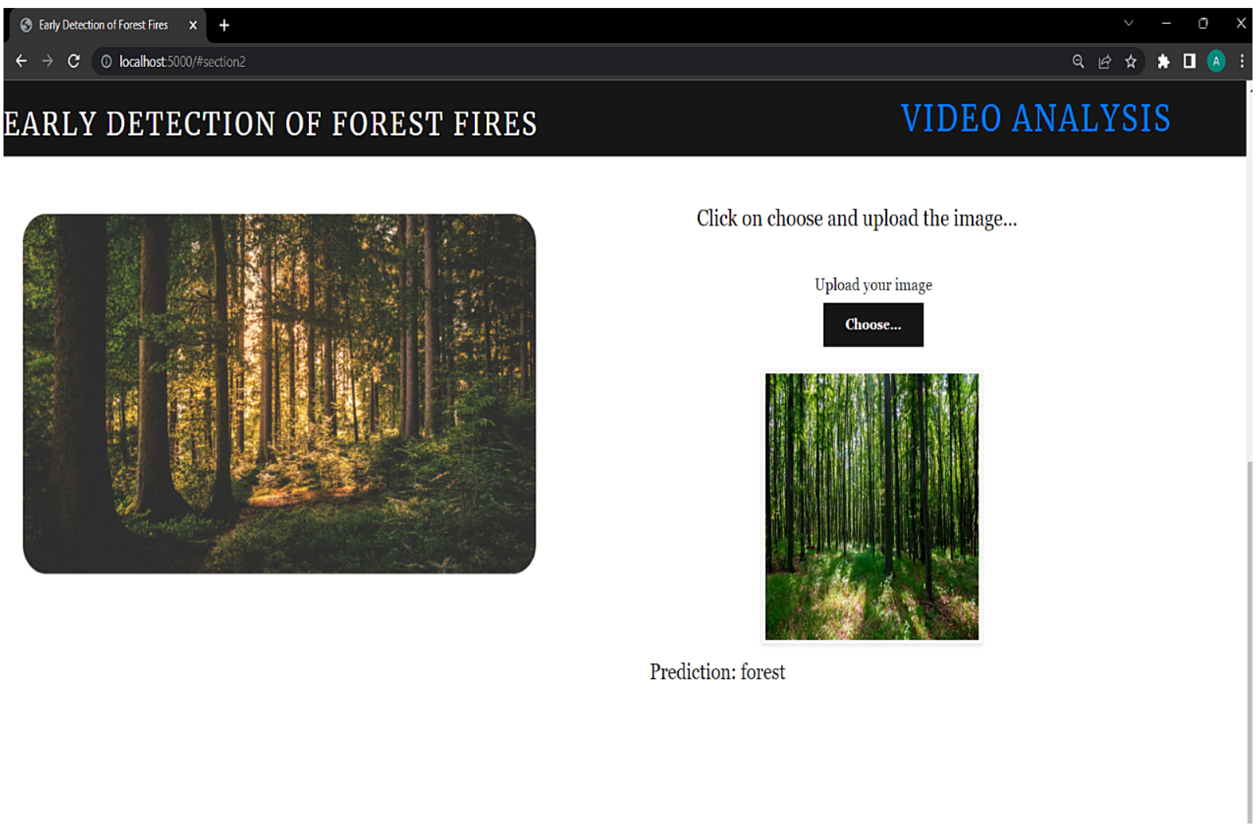
<u>Some examples images of the output are provided below:</u>



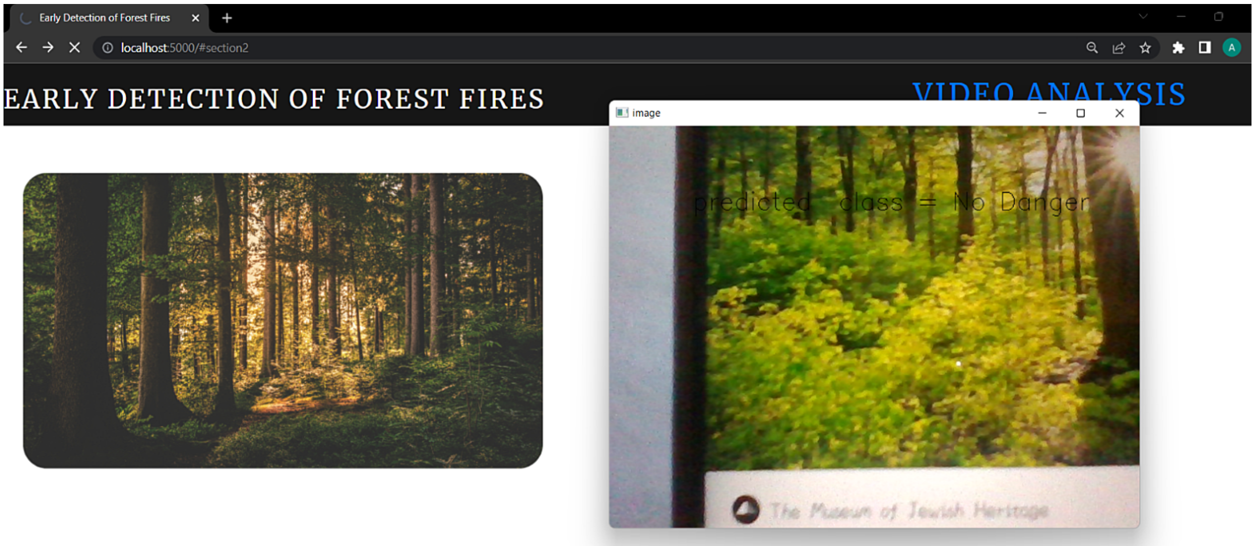
Save the model

```
In [13]: model.save("forest1.h5")

In [14]: print(x_train.class_indices)

{'forest': 0, 'with fire': 1}

In [15]: import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import tensorflow as tf
import cv2
model=load_model("forest1.h5")
img=tf.keras.preprocessing.image.load_img(r'E:\InternshipProject\Dataset\Dataset\test_set\forest\1.jpg',
                                          target_size=(128,128))

In [16]: img

Out[16]:

In [17]: x=tf.keras.preprocessing.image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred=np.argmax(model.predict(x))
pred

1/1 [==============================] - 1s 727ms/step
```

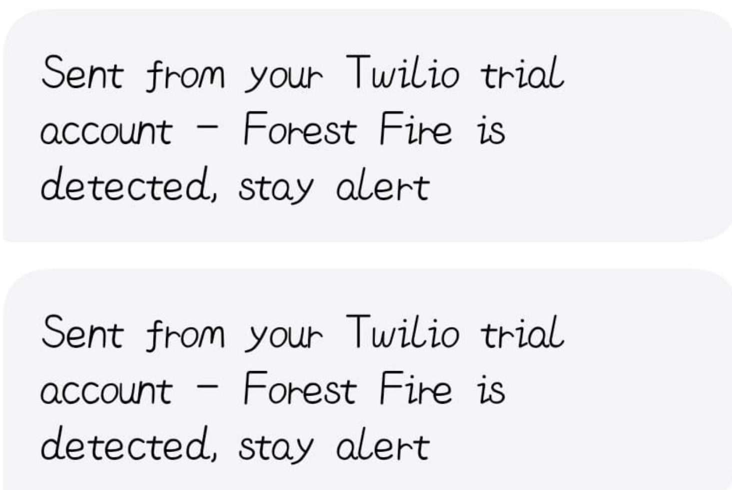




- In Video Analysis we can capture the image of forest through OpenCV window:



- 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=x6i43M7_BAqE
9. CNN Deployment and Download through IBM Cloud:
https://www.youtube.com/watch?v=BzouqMGJ
41k

12. Appendix

IBM Model Training & Download Code:

/ Emerging Methods For Early Dete... / ForestFire

```
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.7->tensorflow) (1.26.7)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard~=2.7->tensorflow) (3.3)
Requirement already satisfied: oauthlib>=3.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.7->te
nsorflow) (3.2.0)
```

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

```python
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)
```

```python
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.
client_298d59510e9e45f380332a73c0f8068d = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='qtl0qZJfhJO9aEXVPdbQwOHJAsdWJuM5jMOkojFeIPDC',
    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')

streaming_body_6 = client_298d59510e9e45f380332a73c0f8068d.get_object(Bucket='emergingmethodsforearlydetectiono-donotdelete-pr-fyqh5iurozb62h', Key='ForestFireDataset.zip')['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/
```

-

Watson Studio | Search in your workspaces | Buy | Anusha Bonthu's Account | Dallas | AB

:ts / Emerging Methods For Early Dete... / ForestFire

```
# pandas documentation: http://pandas.pydata.org/
```

[6]:
```
from io import BytesIO
import zipfile
unzip = zipfile.ZipFile(BytesIO(streaming_body_6.read()),'r')
file_paths = unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

[7]:
```
pwd
```

[7]:
```
'/home/wsuser/work'
```

10]:
```
ls -1
```

```
Dataset/
```

11]:
```
import os
filenames = os.listdir('/home/wsuser/work/Dataset/train_set')
```

13]:
```
x_train=train_datagen.flow_from_directory(r'/home/wsuser/work/Dataset/train_set',
                                          target_size=(128,128),
                                          batch_size=32,
                                          class_mode='binary')
x_test=test_datagen.flow_from_directory(r'/home/wsuser/work/Dataset/test_set',
                                        target_size=(128,128),
                                        batch_size=32,
                                        class_mode='binary')
```

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

14]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')
```

Watson Studio | Search in your workspaces | Buy | Anusha Bonthu's Account | Dallas | AB

/ Emerging Methods For Early Dete... / ForestFire


```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')
```


```
model=Sequential()
```

```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```

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

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

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

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

```
model.compile(loss = 'binary_crossentropy',
              optimizer="adam",
              metrics=["accuracy"])
```

```
model.fit_generator(x_train,steps_per_epoch=14,
                    epochs=15,validation_data=x_test,
                    validation_steps=4)
```

```
Epoch 1/15
14/14 [==============================] - 21s 1s/step - loss: 2.4567 - accuracy: 0.6789 - val_loss: 0.1582 - val_accuracy: 0.9339
Epoch 2/15
14/14 [==============================] - 20s 1s/step - loss: 0.3046 - accuracy: 0.8647 - val_loss: 0.1272 - val_accuracy: 0.9587
Epoch 3/15
14/14 [==============================] - 19s 1s/step - loss: 0.2071 - accuracy: 0.9266 - val_loss: 0.0872 - val_accuracy: 0.9752
Epoch 4/15
```

Watson Studio   Search in your workspaces   Buy   Anusha Bonthu's Account   Dallas   AB

/ Emerging Methods For Early Dete... / ForestFire

```
[ ]: from tensorflow.keras.models import Sequential
     from tensorflow.keras.layers import Dense
     from tensorflow.keras.layers import Convolution2D
     from tensorflow.keras.layers import MaxPooling2D
     from tensorflow.keras.layers import Flatten
     import warnings
     warnings.filterwarnings('ignore')
```

```
[ ]: model=Sequential()
```

```
[ ]: model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```

```
[ ]: model.add(MaxPooling2D(pool_size=(2,2)))
```

```
[ ]: model.add(Flatten())
```

```
[ ]: model.add(Dense(units=150,activation='relu'))
```

```
[ ]: model.add(Dense(units=1,activation='sigmoid'))
```

```
[ ]: model.compile(loss = 'binary_crossentropy',
                   optimizer="adam",
                   metrics=["accuracy"])
```

```
[ ]: model.fit_generator(x_train,steps_per_epoch=14,
                         epochs=15,validation_data=x_test,
                         validation_steps=4)
```

```
Epoch 1/15
14/14 [==============================] - 21s 1s/step - loss: 2.4567 - accuracy: 0.6789 - val_loss: 0.1582 - val_accuracy: 0.9339
Epoch 2/15
14/14 [==============================] - 20s 1s/step - loss: 0.3046 - accuracy: 0.8647 - val_loss: 0.1272 - val_accuracy: 0.9587
Epoch 3/15
14/14 [==============================] - 19s 1s/step - loss: 0.2071 - accuracy: 0.9266 - val_loss: 0.0872 - val_accuracy: 0.9752
Epoch 4/15
```

Emerging Methods For Early Dete... / ForestFire

```
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client) (2.0.4)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (1.20.3)
Installing collected packages: watson-machine-learning-client
Successfully installed watson-machine-learning-client-1.0.391
```

```
from ibm_watson_machine_learning import APIClient
wml_credentials={
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey":"2cwsGDroOFChi1TJiNyWCibU0LC60FQRqksfaFTsSEiK"
}
client=APIClient(wml_credentials)
```

```
client=APIClient(wml_credentials)
```

```
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'])
```

```
space_uid=guid_from_space_name(client,'Detection Of Forest Fires')
print("Space UID= "+space_uid)
```

```
Space UID= 6d88acdc-e675-4f1f-a43f-69e0cdf6e8f6
```

```
client.set.default_space(space_uid)
```

```
'SUCCESS'
```

```
client.software_specifications.list()
```

```
----------------------------  ------------------------------------  ----
NAME                          ASSET_ID                              TYPE
default_py3.6                 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12     020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt    069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6       09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12    09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9     0b848dd4-e681-5599-be41-b5f6fccc6471  base
```

```
xgboost_0.90-py3.6           4ff8d6c2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6       50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8          52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11   55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0              5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0               5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1            5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                   5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
autoai-kb_3.1-py3.7          632d4b22-10aa-5180-88f0-f52dfb6444d7  base
pytorch-onnx_1.7-py3.8       634d3cdc-b562-5bf9-a2d4-ea90a478456b  base
spark-mllib_2.3-r_3.6        6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c  base
tensorflow_2.4-py3.7         65e171d7-72d1-55d9-8ebb-f813d620c9bb  base
spss-modeler_18.2            687eddc9-028a-4117-b9dd-e57b36f1efa5  base
pytorch-onnx_1.2-py3.6       692a6a4d-2c4d-45ff-a1ed-b167ee55469a  base
spark-mllib_2.3-scala_2.11   7963efe5-bbec-417e-92cf-0574e21b4e8d  base
---------------------------  ------------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

```
]: software_spec_uid=client.software_specifications.get_uid_by_name("runtime-22.1-py3.9")
   software_spec_uid
```

```
]: '12b83a17-24d8-5082-900f-0ab31fbfd3cb'
```

```
]: model_details=client.repository.store_model(model='detect-fire-model_new.tgz',meta_props={
       client.repository.ModelMetaNames.NAME:"CNN",
       client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
       client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid})

   model_id= client.repository.get_model_uid(model_details)
```

```
This method is deprecated, please use get_model_id()
```

```
]: model_id
```

```
]: 'bb0cdbb8-7616-4627-8078-8d4f47944be2'
```