Final Code

Team Id: PNT2022TMID13815

Data Collection & Image Preprocessing:

```
import keras
from keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
import numpy as np
batch_size = 32

train_datagen = ImageDataGenerator(
    shear_range=0.2,
    rotation_range=180,
    zoom_range=0.2,
    horizontal_flip=True,
)

val_datagen = ImageDataGenerator(
    rescale=1./255
)
train_generator = train_datagen.flow_from_directory(
    'train_set/',
    target_size=(150, 150),
    batch_size=batch_size,
    class_mode='binary'
)

val_generator = val_datagen.flow_from_directory(
    'test_set/',
    target_size=(150, 150),
    batch_size=batch_size,
    class_mode='binary'
)
```

Output:

```
Found 435 images belonging to 2 classes.
Found 121 images belonging to 2 classes.
```

Model Creation and Summary :

```
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Activation
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers import Dense
model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(150,150,3)))
model.add(Activation('relu'))
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(150))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))
model.compile(
   loss='binary_crossentropy',
   optimizer='adam',
   metrics=['accuracy']
)
model.summary()model.fit(train_generator,steps_per_epoch=14,
        epochs=10,validation_data=val_generator
        ,validation_steps=4)
```

Output:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 148, 148, 32)      896
_________________________________________________________________
activation (Activation)      (None, 148, 148, 32)      0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 74, 74, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 175232)            0
_________________________________________________________________
dense (Dense)                (None, 150)               26284950
_________________________________________________________________
activation_1 (Activation)    (None, 150)               0
_________________________________________________________________
dropout (Dropout)            (None, 150)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 151
_________________________________________________________________
activation_2 (Activation)    (None, 1)                 0
=================================================================
Total params: 26,285,997
Trainable params: 26,285,997
Non-trainable params: 0
_________________________________________________________________
```

Video Analysis using CV2:

```
from keras.models import load_model
from keras.preprocessing import image
import numpy as np
import cv2
from PIL import Image, ImageOps
model=load_model("forest1.h5")
data=np.ndarray(shape=(1,150,150,3),dtype=np.float32)
class_name=['Fire','No_Fire']
img=image.load_img('train_set/forest/NoFire (1).bmp',target_size=(64,64))
img_array = image.img_to_array(img)
img_batch = np.expand_dims(img_array, axis=0)
# x=np.expand_dims(x,axis=0)
```

```
pred=model.predict(img_batch)
index=np.argmax(pred)
class_name[index]
```

Twilio Connection & Play Sound:

```
from twilio.rest import Client
from playsound import playsound
model=load_model('forest1.h5')
video=cv2.VideoCapture(0)
name=['forest','with fire']
account_sid='ACca0e8bb11699d2957d67c979ca84b68a'
auth_token='bcb5f3850ef4b7ed263f60efc9acecdb'
client =Client(account_sid,auth_token)
message=client.messages \
.create(
body='--------Forest Fire is detected,Stay Alert !!!--------',
   from_='+19457581434',to='+919943435141')
print(message.sid)
print("Alert Message sent")
```

Output:

```
SMb8a51eaeb987fbc8d5eced2dab56300a
Alert Message sent
```

Testing Model :

```
import cv2
import numpy as np
from keras.preprocessing import image
from keras.models import load_model
from twilio.rest import Client
from playsound import playsound
model=load_model('forest1.h5')
video=cv2.VideoCapture(0)
name=['forest','with fire']
while(True):
   ret,frame=video.read()
   cv2.imshow('frame',frame)
   cv2.imwrite('image.jpg',frame)
   img=image.load_img('train_set/forest/NoFire (1).bmp',target_size=(64,64))
   x=image.img_to_array(img)
   x=np.expand_dims(x,axis=0)
   pred=model.predict(x)
   index=np.argmax(pred)
   if index==0:
      account_sid='ACca0e8bb11699d2957d67c979ca84b68a'
      auth_token='bcb5f3850ef4b7ed263f60efc9acecdb'
      client =Client(account_sid,auth_token)
      message=client.messages \
```

```
        .create(body='-------Fire is detected,Stay Alert !!!--------',
                from_='+19457581434',to='+919943435141')
        print(message.sid)
        print('Fire detected')
        print("Alert Message sent!")
        playsound('tornado-siren.mp3')

    else:
        print('No Danger')
        cv2.imshow("image.jpg",frame)
        if cv2.waitkey(2)&0xff == ord('q'):
            break
video.release()
cv2.destroyAllWindows()
```

Final Output: