EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

MODEL BUILDING

PREDICTIONS

Date	04 November 2022
Team ID	PNT2022TMID10153
Project Name	Emerging Methods for Early Detection of Forest Fires

Importing The ImageDataGenerator Library

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

Define the parameters/arguments for ImageDataGenerator class

```
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rot
ati on_range=180,zoom_range=0.2, horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

Applying ImageDataGenerator functionality to trainset

```
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDriv
e/ Dataset/train_set',target_size=(128,128),batch_size=32,
class_mode='binary')
```

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

Applying ImageDataGenerator functionality to testset

```
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive
/ Dataset/test_set',target_size=(128,128),batch_size=32,
class_mode='binary')
```

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

Import model building libraries

```
#To define Linear initialisation import Sequential
from keras.models import Sequential
#To add layers import Dense
from keras.layers import Dense
#To create Convolution kernel import Convolution2D
from keras.layers import Convolution2D
#import Maxpooling layer
from keras.layers import MaxPooling2D
#import flatten layer
from keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')
```

Initializing the model

```
model=Sequential()
```

Add CNN Layer

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

```
#add maxpooling layer
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
#add flatten layer
```

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

Add Hidden Layer

```
  #add hidden layer
model.add(Dense(150,activation='relu'))
#add output layer
model.add(Dense(1,activation='sigmoid')
)
```

Configure the learning process

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

Train the model

```
model.fit_generator(x_train,steps_per_epoch=14,epochs=5,validation_data=x_tes
t,validation_steps=20)
```

```
Epoch 1/5
14/14 [==============================] - ETA: 0s - loss: 1.2125 - accuracy: 0.6972WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch * epochs` batches (in this case, 20 batches). You may need to use the repeat() function when building your dataset.
14/14 [==============================] - 123s 9s/step - loss: 1.2125 - accuracy: 0.6972 - val_loss: 0.2814 - val_accuracy: 0.9174
Epoch 2/5
14/14 [==============================] - 15s 1s/step - loss: 0.2566 - accuracy: 0.8945
Epoch 3/5
14/14 [==============================] - 15s 1s/step - loss: 0.1825 - accuracy: 0.9381
Epoch 4/5
14/14 [==============================] - 15s 1s/step - loss: 0.1477 - accuracy: 0.9312
Epoch 5/5
14/14 [==============================] - 16s 1s/step - loss: 0.1121 - accuracy: 0.9587
<keras.callbacks.History at 0x7ff99287ad50>
```

Save The Model

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