VIDEO ANALYSIS

OPEN CV FOR VIDEO PROCESSING

Date:10 November 2022

Team ID:PNT2022TMID50335

Project Name:Emerging Method For Early Detection Of Forest Fire

```
Importing Keras libraries
import keras
Importing ImageDataGenerator from Keras
from matplotlib import pyplot as plt
from keras.preprocessing.image import ImageDataGenerator
Defining the Parameters
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotati
on_range=180,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
Applying ImageDataGenerator functionality to train dataset
x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/NT/
Dataset/
train_set',target_size=(128,128),batch_size=32,class_mode='binary')
Found 676 images belonging to 2 classes.
Applying ImageDataGenerator functionality to test dataset
x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/NT/
Dataset/
test_set',target_size=(128,128),batch_size=32,class_mode='binary')
Found 121 images belonging to 2 classes.
Importing Model Building Libraries
#to define the linear Initialisation import sequential
from keras.models import Sequential
#to add layers import Dense
from keras.layers import Dense
#to create Convolutional 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()
Adding CNN Layers
model.add(Convolution2D(32,
(3,3),input_shape=(128,128,3),activation='relu'))
#add maxpooling layers
model.add(MaxPooling2D(pool_size=(2,2)))
#add faltten layer
```

```
model.add(Flatten())
Add Dense layers
#add hidden layers
model.add(Dense(150,activation='relu'))
#add output layer
model.add(Dense(1,activation='sigmoid'))
configuring the learning process
model.compile(loss='binary_crossentropy',optimizer="adam",metrics=["ac
curacy"])
Training the model
model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_da
ta=x_test,validation_steps=4)
Epoch 1/10
14/14 [==============================] - 75s 5s/step - loss: 2.0140 -
accuracy: 0.6548 - val_loss: 0.1757 - val_accuracy: 0.9339
Epoch 2/10
14/14 [==============================] - 36s 3s/step - loss: 0.2789 -
accuracy: 0.8762 - val_loss: 0.2360 - val_accuracy: 0.9091
Epoch 3/10
14/14 [==============================] - 28s 2s/step - loss: 0.2762 -
accuracy: 0.8817 - val_loss: 0.0724 - val_accuracy: 0.9669
Epoch 4/10
14/14 [==============================] - 26s 2s/step - loss: 0.2242 -
accuracy: 0.8929 - val_loss: 0.0770 - val_accuracy: 0.9752
Epoch 5/10
14/14 [==============================] - 25s 2s/step - loss: 0.2187 -
accuracy: 0.9129 - val_loss: 0.0646 - val_accuracy: 0.9752
Epoch 6/10
14/14 [==============================] - 25s 2s/step - loss: 0.2338 -
accuracy: 0.9143 - val_loss: 0.0597 - val_accuracy: 0.9669
Epoch 7/10
14/14 [==============================] - 24s 2s/step - loss: 0.2244 -
accuracy: 0.9040 - val_loss: 0.0851 - val_accuracy: 0.9587
Epoch 8/10
14/14 [==============================] - 25s 2s/step - loss: 0.2110 -
accuracy: 0.8996 - val_loss: 0.1276 - val_accuracy: 0.9504
Epoch 9/10
14/14 [==============================] - 25s 2s/step - loss: 0.1905 -
accuracy: 0.9214 - val_loss: 0.0768 - val_accuracy: 0.9752
Epoch 10/10
14/14 [==============================] - 25s 2s/step - loss: 0.1644 -
accuracy: 0.9330 - val_loss: 0.0548 - val_accuracy: 0.9835
model.save("forest.h5")
Predictions
#import load model from keras.model
from keras.models import load_model
#import image from keras
from tensorflow.keras.preprocessing import image
import numpy as np
#import cv2
```

```
import cv2
#load the saved model
model=load_model('forest.h5')
img=image.load_img('/content/drive/MyDrive/NT/Dataset/test_set/forest/
1170x500_Ireland_web.jpg')
x=image.img_to_array(img)
res=cv2.resize(x,dsize=(128,128),interpolation=cv2.INTER_CUBIC)
#expand the image shape
x=np.expand_dims(res,axis=0)
pred=model.predict(x)
1/1 [==============================] - 0s 149ms/step
pred
array([[0.]], dtype=float32)
Open cv2 for video processing
pip install twilio
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Collecting twilio
  Downloading twilio-7.15.1-py2.py3-none-any.whl (1.4 MB)
     |████████████████████████████████| 1.4 MB 5.1 MB/s
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-
packages (from twilio) (2022.6)
Requirement already satisfied: requests>=2.0.0 in
/usr/local/lib/python3.7/dist-packages (from twilio) (2.23.0)
Collecting PyJWT<3.0.0,>=2.0.0
  Downloading PyJWT-2.6.0-py3-none-any.whl (20 kB)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0-
>twilio) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio)
(2022.9.24)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio)
(3.0.4)
Requirement already satisfied: idna<3,>=2.5 in
/usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio)
(2.10)
Installing collected packages: PyJWT, twilio
Successfully installed PyJWT-2.6.0 twilio-7.15.1
pip install playsound
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Collecting playsound
  Downloading playsound-1.3.0.tar.gz (7.7 kB)
Building wheels for collected packages: playsound
  Building wheel for playsound (setup.py) ... done
  Created wheel for playsound: filename=playsound-1.3.0-py3-none-
any.whl size=7035
sha256=68e0d58bbbd0e88d87e2cbaeee4cb796e136a7c70ea6d9fecf2085f76e95dab
```

```
d
  Stored in directory:
/root/.cache/pip/wheels/ba/f8/bb/ea57c0146b664dca3a0ada4199b0ecb5f9dfc
b7b7e22b65ba2
Successfully built playsound
Installing collected packages: playsound
Successfully installed playsound-1.3.0
from logging import WARNING
#import opencv library
import cv2
#import numpy
import numpy as np
#import image function from keras
from keras.preprocessing import image
#import load_model from keras
from keras.models import load_model
#import client from twilio API
from twilio.rest import Client
#import playsound package
from playsound import playsound
WARNING:playsound:playsound is relying on another python subprocess.
Please use `pip install pygobject` if you want playsound to run more
efficiently.
#load the saved model
model=load_model('forest.h5')
#define video
video = cv2.VideoCapture(0)
#define the features
name = ['forest','with fire']
```