Importing Keras libraries

```
import keras
```

Importing ImageDataGenerator from Keras

```
from matplotlib import pyplot as plt
from keras.preprocessing.image import ImageDataGenerator
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.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='KXjEkgUBwu4dS1Lchix_OeLTtOdfWFcOzlhwXRqXtHro',
    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')

bucket = 'forestfire-donotdelete-pr-wmm56yysfedwtp'
object_key = 'Dataset.zip'

streaming_body_2 = cos_client.get_object(Bucket=bucket, Key=object_key)['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/
from io import BytesIO
import zipfile
unzip = zipfile.ZipFile(BytesIO(streaming_body_2.read()),'r')
file_paths = unzip.namelist()
for path in file_paths:
    unzip.extract(path)
pwd
'/home/wsuser/work'
import os
filenames = os.listdir('/home/wsuser/work/Dataset/train_set')
```

Defining the Parameters

```
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,shear_range=0.2,rotation_range=180,zoom_range=0.2,horizontal_flip=True)
```

Applying ImageDataGenerator functionality to train datasetx_train=train_datagen.flow_from_directory('/home/wsuser/work/Dataset/train_set',target_size=(64,64),batch_size=32,class_mode='binary')

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

Applying ImageDataGenerator functionality to test dataset

```
x_test=test_datagen.flow_from_directory('/home/wsuser/work/Dataset/test_set',target_size=(64,64),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=(64,64,3),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=["accuracy"])
```

Training the model

```
model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_test,validation_steps=4)
Epoch 1/10
14/14 [==============================] - 23s 2s/step - loss: 0.8269 - accuracy: 0.6835 - val_loss: 0.1792 -
 val_accuracy: 0.9504
Epoch 2/10
14/14 [==============================] - 23s 2s/step - loss: 0.2426 - accuracy: 0.8876 - val_loss: 0.1126 -
 val_accuracy: 0.9587
Epoch 3/10
14/14 [==============================] - 22s 2s/step - loss: 0.2107 - accuracy: 0.9128 - val_loss: 0.1256 -
 val_accuracy: 0.9421
Epoch 4/10
14/14 [==============================] - 22s 2s/step - loss: 0.2927 - accuracy: 0.8784 - val_loss: 0.1423 -
 val_accuracy: 0.9256
Epoch 5/10
14/14 [==============================] - 21s 1s/step - loss: 0.1980 - accuracy: 0.9151 - val_loss: 0.0976 -
 val_accuracy: 0.9669
Epoch 6/10
14/14 [==============================] - 21s 1s/step - loss: 0.1891 - accuracy: 0.9128 - val_loss: 0.0779 -
 val_accuracy: 0.9669
Epoch 7/10
14/14 [==============================] - 21s 2s/step - loss: 0.1688 - accuracy: 0.9381 - val_loss: 0.0945 -
 val_accuracy: 0.9421
Epoch 8/10
14/14 [==============================] - 22s 2s/step - loss: 0.1768 - accuracy: 0.9243 - val_loss: 0.0751 -
 val_accuracy: 0.9835
Epoch 9/10
14/14 [==============================] - 20s 1s/step - loss: 0.1583 - accuracy: 0.9312 - val_loss: 0.0522 -
 val_accuracy: 0.9917
```

```
Epoch 10/10
14/14 [==============================] - 21s 1s/step - loss: 0.1935 - accuracy: 0.9220 - val_loss: 0.0562 -
 val_accuracy: 0.9835
```

Save the model

```
model.save("forest.h5")
!tar -zcvf image-classification_new.tgz forest.h5
```

```
forest.h5
```

```
ls  -1
```

```
Dataset/
forest.h5
image-classification_new.tgz
```

```
!pip install watson-machine-learning-client --upgrade
```

```
Requirement already satisfied: watson-machine-learning-client in /opt/conda/envs/Python-3.9/lib/python3.9/site-pac
kages (1.0.391)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson
-machine-learning-client) (2.11.0)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-ma
chine-learning-client) (2.26.0)
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machi
ne-learning-client) (1.18.21)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-mach
ine-learning-client) (1.26.7)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-mac
hine-learning-client) (0.3.3)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machi
ne-learning-client) (2022.9.24)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-mac
hine-learning-client) (0.8.9)
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machi
ne-learning-client) (4.62.3)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-mac
hine-learning-client) (1.3.4)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (fr
om boto3->watson-machine-learning-client) (0.10.0)
Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-package
s (from boto3->watson-machine-learning-client) (1.21.41)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (f
rom boto3->watson-machine-learning-client) (0.5.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packag
es (from botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dat
eutil<3.0.0,>=2.1->botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (1.15.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages
 (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-pac
kages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from reques
ts->watson-machine-learning-client) (3.3)
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 panda
s->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 pan
das->watson-machine-learning-client) (1.20.3)
```

```
from ibm_watson_machine_learning import APIClient
```

```
wml_credentilas = {
                  "url":"https://us-south.ml.cloud.ibm.com",
                  "apikey":"hxe6koyIaU12_be6Qw-sQ8omzOrg9czDp9Ep11YppBs6"
}
client = APIClient(wml_credentilas)
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, 'Forestrecognition')
print("Space UID = " + space_uid)
Space UID = 2bae4b0b-57cd-4fd3-89ef-5fc4a44867a5
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
ai-function_0.1-py3.6         0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                    0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod  1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6             10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl     111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10       125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9            12b83a17-24d8-5082-900f-0ab31fbfd3cb  basePredictions
software_spec_uid = client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")
software_spec_uid
'acd9c798-6974-5d2f-a657-ce06e986df4d'
model_details = client.repository.store_model(model='image-classification_new.tgz',meta_props={
client.repository.ModelMetaNames.NAME:'CNN',
client.repository.ModelMetaNames.TYPE:"tensorflow_rt22.1",
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
'1baa1aab-07c5-4a4a-a297-9b4c3444d699'
#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('/home/wsuser/work/Dataset/test_set/with fire/forest_fire_2268729_1280.jpg')
x=image.img_to_array(img)
res=cv2.resize(x,dsize=(64,64),interpolation=cv2.INTER_CUBIC)
#expand the image shape
x=np.expand_dims(res,axis=0)
```

```
pred=model.predict(x)
pred = int(pred[0][0])
pred
int(pred)
1
if pred==1:
  print('Forest fire')
elif pred==0:
  print('No Fire')
Forest fire
```

Open cv for video processing

```
pip install twilio
Requirement already satisfied: twilio in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (7.15.2)
Requirement already satisfied: PyJWT<3.0.0,>=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (fro
m twilio) (2.4.0)
Requirement already satisfied: pytz in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from twilio) (2021.3
)
Requirement already satisfied: requests>=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from twil
io) (2.26.0)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from r
equests>=2.0.0->twilio) (2022.9.24)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from reques
ts>=2.0.0->twilio) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (fro
m requests>=2.0.0->twilio) (1.26.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages
 (from requests>=2.0.0->twilio) (2.0.4)
Note: you may need to restart the kernel to use updated packages.
pip install playsound
Requirement already satisfied: playsound in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.3.0)
Note: you may need to restart the kernel to use updated packages.
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
```

Creating An Account in Twilio Service

```
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.
```

```
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='KXjEkgUBwu4dS1Lchix_OeLTtOdfWFcOzlhwXRqXtHro',
    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')

bucket = 'forestfire-donotdelete-pr-wmm56yysfedwtp'
object_key = 'Vishwaroopam.mp3'

streaming_body_3 = cos_client.get_object(Bucket=bucket, Key=object_key)['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/
from twilio.rest import Client
from playsound import playsound
if pred==1:
  print('Forest fire')
  account_sid='AC34c4bee5e03df7bc7dba1eef29761275'
  auth_token='1fc522239435d0c251c1fd870d715295'
  client=Client(account_sid,auth_token)
  message=client.messages \
  .create(
      body='forest fire is detected,stay alert',
      #use twilio free number
      from_='+19803934024',
      #to number
      to='+919962082226')
  print(message.sid)
  print("Fire detected")
  print("SMS Sent!")
```

```
Forest fire
SM8520469cbcb2d1a83aba6aeaff9dbbca
Fire detected
SMS Sent!
```