

Train CNN Model On IBM

Train Image Classification Model

Team ID	PNT2022TMID42321
Project Name	Emerging Methods for Early Detection of Forest Fires

Project in Watson Studio:

The screenshot displays the IBM Watson Studio web interface. The browser address bar shows the URL: <https://dataplatform.cloud.ibm.com/projects/85599862-ccca-4d51-9f29-adc28f6f03f9/assets?context=cpdaas>. The interface includes a top navigation bar with the IBM Watson Studio logo, a search bar, and a 'Buy' button. Below the navigation bar, the project name 'CNN_model' is displayed. The main content area is divided into two sections: 'Assets' and 'Jobs'. The 'Assets' section is active, showing a list of assets. On the left, there is a sidebar with 'Asset types' including 'Data' (1) and 'Notebooks' (1). The main list of assets shows two items:

Name	Last modified
Image classification Notebook	8 hours ago Modified by you
Dataset-20221116T055616Z-001.zip application/x-zip-compressed	1 day ago Modified by you

At the bottom of the assets list, it indicates 'Items per page: 20' and '1-2 of 2 items'. On the right side of the interface, there is a panel titled 'Data in this project' with a dashed box containing the text: 'Drop data files here or browse for files to upload'.

Image Classification:

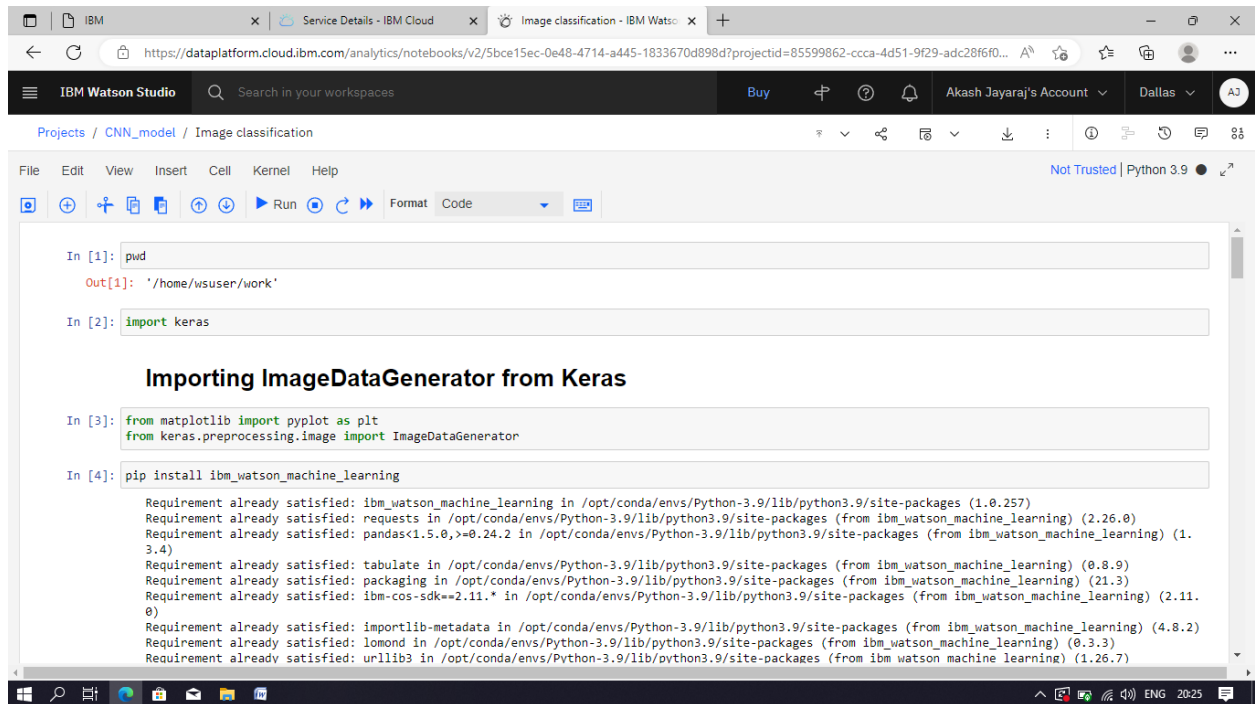


Image classification.ipynb:

```
In [1]:  
pwd  
Out[1]: '/home/wsuser/work'  
In [2]:  
import keras
```

Importing ImageDataGenerator from Keras

```
In [3]:  
from matplotlib import pyplot as plt  
from keras.preprocessing.image import ImageDataGenerator  
In [4]:  
pip install ibm_watson_machine_learning  
Requirement already satisfied: ibm_watson_machine_learning in  
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)  
Requirement already satisfied: requests in  
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from  
ibm_watson_machine_learning) (2.26.0) Requirement already  
satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-  
3.9/lib/python3.9/site-packages (from  
ibm_watson_machine_learning) (1.3.4) Requirement already
```

satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.8.9) Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (21.3) Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.11.0) Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (4.8.2) Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.3.3) Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.26.7) Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2022.9.24) Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (0.10.0) 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==2.11.*->ibm_watson_machine_learning) (2.11.0) Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.11.0) Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.8.2) Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (2021.3) Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (1.20.3) Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (1.15.0) Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm_watson_machine_learning) (2.0.4) Requirement

already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm_watson_machine_learning) (3.3) Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-metadata->ibm_watson_machine_learning) (3.6.0) Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm_watson_machine_learning) (3.0.4) Note: you may need to restart the kernel to use updated packages.

In [5]:

```
import ibm_watson_machine_learning
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "hxe6koyIaU12_be6Qw-
sQ8omzOrg9czDp9Ep11YppBs6"
}
client = APIClient(wml_credentials)
```

In [6]:

```
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'])
```

In [7]:

```
space_uid = guid_from_space_name(client, 'Forestrecognition')
print("Space UID = " + space_uid)
Space UID = 2bae4b0b-57cd-4fd3-89ef-5fc4a44867a5
```

In [8]:

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

Out[8]: 'SUCCESS'

In [9]:

```
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='bhtzDKXCeJ2Rko2lY0HoTEVMuI1P_Ds2Zmv5EwCoOnyp',
    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 = 'cnnmodel-donotdelete-pr-5jfgh2lpsrolaa'
object_key = 'Dataset-20221116T055616Z-001.zip'
```

```
streaming_body_1 = 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/
```

```
In [10]:
```

```
from io import BytesIO
import zipfile
unzip = zipfile.ZipFile(BytesIO(streaming_body_1.read()), 'r')
file_paths = unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

```
In [11]:
```

```
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, r
otation_range=180, zoom_range=0.2, horizontal_flip=True)
```

```
In [12]:
```

```
pwd
```

```
Out[12]: '/home/wsuser/work'
```

Applying ImageDataGenerator functionality to train dataset

```
In [13]:
```

```
x_train=train_datagen.flow_from_directory('/home/wsuser/work/Dat
aset/train_set', target_size=(64, 64), batch_size=32, class_mode='bi
nary')
```

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

Applying ImageDataGenerator functionality to test dataset

In [14]:

```
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

In [15]:

```
#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

In [16]:

```
model = Sequential()
```

Adding CNN Layers

In [17]:

```
model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu'))  
#add maxpooling layers  
model.add(MaxPooling2D(pool_size=(2,2)))  
#add flatten layer  
model.add(Flatten())
```

Add Dense layers

In [18]:

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

configuring the learning process

In [19]:

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

Training the model

In [20]:

```
model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_test,validation_steps=4)
```

```
Epoch 1/10 14/14 [=====] - 18s 1s/step  
- loss: 1.0992 - accuracy: 0.6353 - val_loss: 0.3017 -  
val_accuracy: 0.8926 Epoch 2/10 14/14  
[=====] - 16s 1s/step - loss: 0.3195 -  
accuracy: 0.8463 - val_loss: 0.1315 - val_accuracy: 0.9421 Epoch  
3/10 14/14 [=====] - 18s 1s/step -  
loss: 0.2219 - accuracy: 0.8968 - val_loss: 0.1086 -  
val_accuracy: 0.9587 Epoch 4/10 14/14  
[=====] - 16s 1s/step - loss: 0.1939 -  
accuracy: 0.9243 - val_loss: 0.0904 - val_accuracy: 0.9752 Epoch  
5/10 14/14 [=====] - 15s 1s/step -  
loss: 0.1744 - accuracy: 0.9106 - val_loss: 0.0827 -  
val_accuracy: 0.9752 Epoch 6/10 14/14  
[=====] - 16s 1s/step - loss: 0.1711 -  
accuracy: 0.9220 - val_loss: 0.0847 - val_accuracy: 0.9669 Epoch  
7/10 14/14 [=====] - 16s 1s/step -  
loss: 0.1884 - accuracy: 0.9151 - val_loss: 0.1547 -  
val_accuracy: 0.9174 Epoch 8/10 14/14  
[=====] - 16s 1s/step - loss: 0.1679 -  
accuracy: 0.9243 - val_loss: 0.0966 - val_accuracy: 0.9504 Epoch  
9/10 14/14 [=====] - 16s 1s/step -  
loss: 0.1589 - accuracy: 0.9358 - val_loss: 0.0740 -  
val_accuracy: 0.9752 Epoch 10/10 14/14  
[=====] - 15s 1s/step - loss: 0.1516 -  
accuracy: 0.9312 - val_loss: 0.0544 - val_accuracy: 0.9917
```

```
Out[20]: <keras.callbacks.History at 0x7f8cd1680ca0>
```

Save the model

In [21]:

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

In [22]:

```
#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/with fire (9).jpg')
x=image.img_to_array(img)
res=cv2.resize(x,dsizes=(64,64),interpolation=cv2.INTER_CUBIC)
#expand the image shape
x=np.expand_dims(res,axis=0)
```

Prediction

```
In [23]:
pred=model.predict(x)
pred = int(pred[0][0])
pred
int(pred)
```

```
Out[23]: 1
```

```
In [24]:
if pred==1:
    print('Forest fire')
elif pred==0:
    print('No Fire')
Forest fire
```

Open cv for video processing

```
In [25]:  
pip install twilio  
Collecting twilio Downloading twilio-7.15.3-py2.py3-none-any.whl  
(1.4 MB) | ████████████████████████████████████████ | 1.4 MB 24.5 MB/s eta  
0:00:01 Requirement already satisfied: pytz in  
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from  
twilio) (2021.3) Requirement already satisfied:  
PyJWT<3.0.0,>=2.0.0 in /opt/conda/envs/Python-  
3.9/lib/python3.9/site-packages (from twilio) (2.4.0)  
Requirement already satisfied: requests>=2.0.0 in  
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from  
twilio) (2.26.0) Requirement already satisfied:  
certifi>=2017.4.17 in /opt/conda/envs/Python-  
3.9/lib/python3.9/site-packages (from requests>=2.0.0->twilio)  
(2022.9.24) 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) Requirement already satisfied: urllib3<1.27,>=1.21.1 in  
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from  
requests>=2.0.0->twilio) (1.26.7) Requirement already satisfied:  
idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-  
packages (from requests>=2.0.0->twilio) (3.3) Installing  
collected packages: twilio Successfully installed twilio-7.15.3  
Note: you may need to restart the kernel to use updated  
packages.
```


In [26]:

```
pip install playsound
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=7037
sha256=0a843383e171bb37fde4f55b1c7a94c34b2f9b28a6ac76c864c10a469
fac6b53 Stored in directory:
/tmp/wsuser/.cache/pip/wheels/ba/39/54/c8f7ff9a88a644d3c58b4dec8
02d90b79a2e0fb2a6b884bf82 Successfully built playsound
Installing collected packages: playsound Successfully installed
playsound-1.3.0 Note: you may need to restart the kernel to use
updated packages.
```

In [27]:

```
import scipy
import IPython.display as ipd
```

In [28]:

```
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
playsound is relying on another python subprocess. Please use
`pip install pygobject` if you want playsound to run more
efficiently.
```

Sending Alert Message

In [29]:

```
from twilio.rest import Client
from playsound import playsound
if pred==1:
    print('Forest fire')
    account_sid='ACc0b32842aa3060ee6f4b2bfa1116247f'
    auth_token='c6a97bddc96eaf4803622bd76e8a96ff'
    client=Client(account_sid,auth_token)
    message=client.messages \
        .create(
            body='forest fire is detected,stay alert',
            #use twilio free number
```

```
from_='+14793974371',  
#to number  
to='+919025496556')  
print(message.sid)  
print("Fire detected")  
print("SMS Sent!")
```

Forest fire
SM9f80d788df7b59989488039a4e18a0f1
Fire detected SMS Sent!

Alert Received:

