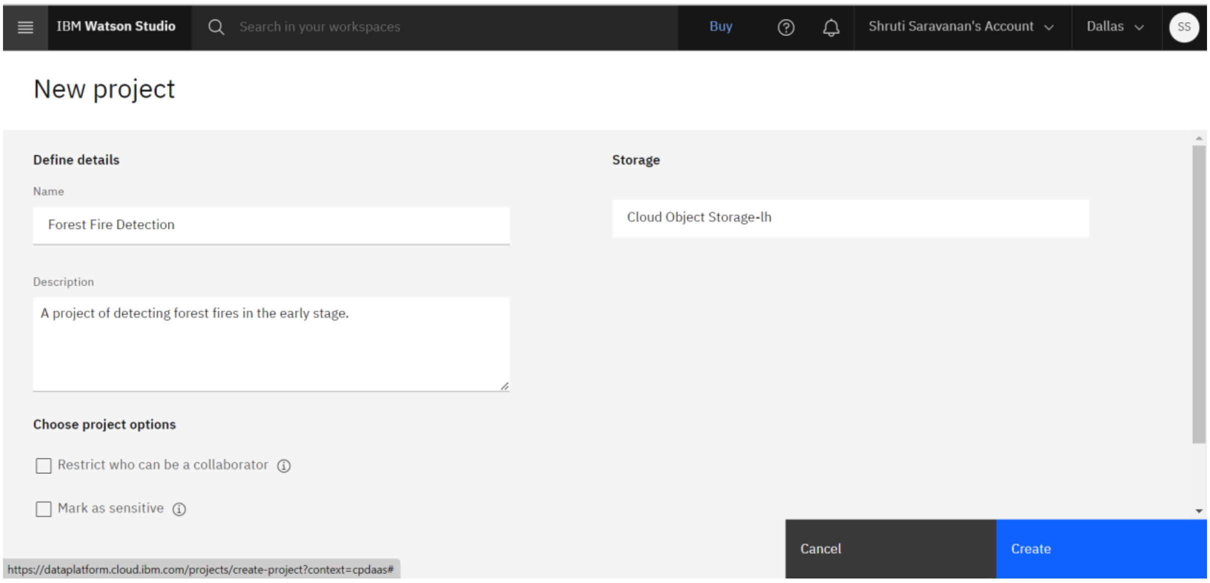
EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

Train CNN Model On IBM

Team ID: PNT2022TMID21552

Creating a new project in IBM Watson:



IBM Watson Studio
Search in your workspaces
Buy
Shruti Saravanan's Account
Dallas
SS

New project

Define details

Name: Forest Fire Detection

Description: A project of detecting forest fires in the early stage.

Storage: Cloud Object Storage-lh

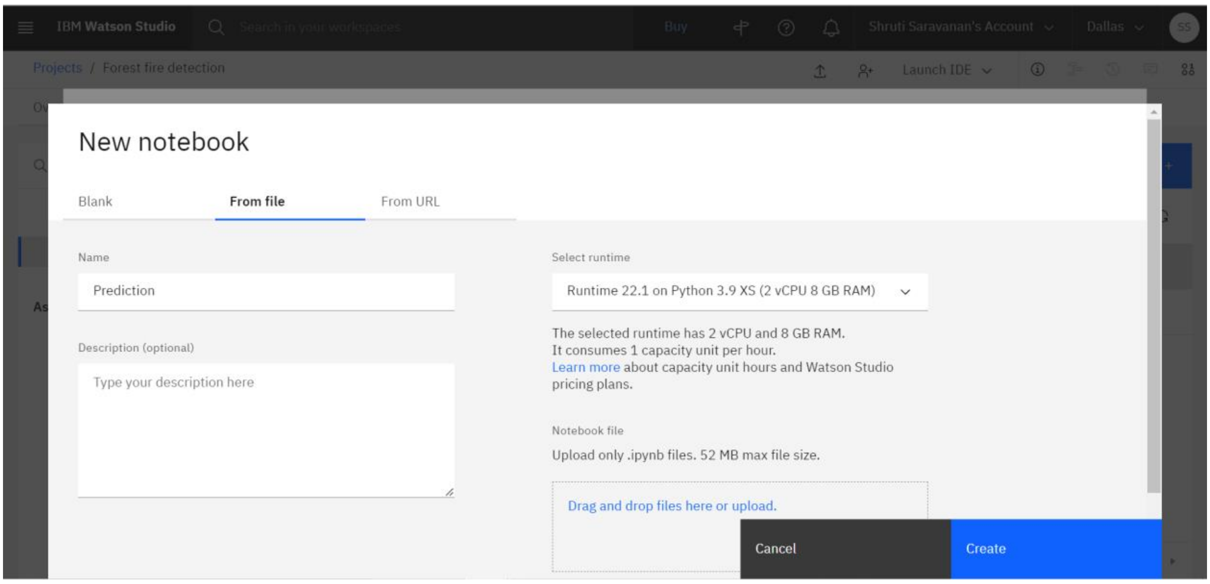
Choose project options

□ Restrict who can be a collaborator

□ Mark as sensitive

Cancel
Create
https://dataplatform.cloud.ibm.com/projects/create-project?context=cpdaas#

Starting a Jupyter notebook:



IBM Watson Studio
Search in your workspaces
Buy
Shruti Saravanan's Account
Dallas
SS
Projects / Forest fire detection
Launch IDE

New notebook

Name: Prediction

Description (optional): Type your description here

Select runtime: Runtime 22.1 on Python 3.9 XS (2 vCPU 8 GB RAM)

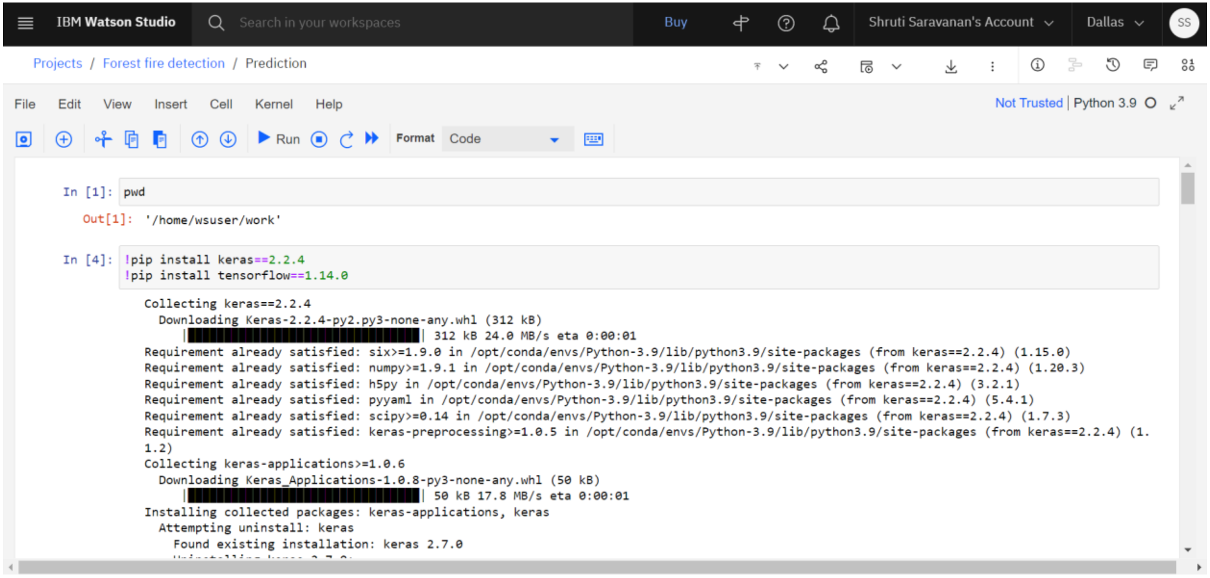
The selected runtime has 2 vCPU and 8 GB RAM.
It consumes 1 capacity unit per hour.
Learn more about capacity unit hours and Watson Studio pricing plans.

Notebook file

Upload only .ipynb files. 52 MB max file size.

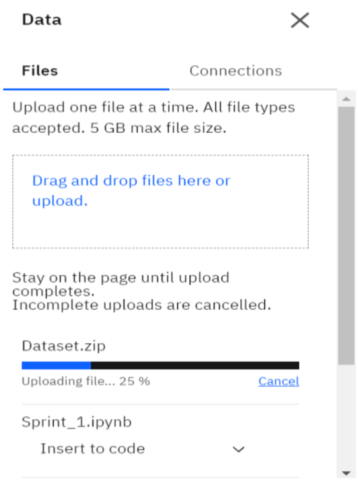
Drag and drop files here or upload.

Installing required libraries


IBM Watson Studio
Search in your workspaces
Buy
Shruti Saravanan's Account
Dallas
SS
Projects / Forest fire detection / Prediction
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.9
Run
Format
Code

```
In [1]: pwd

Out[1]: '/home/wsuser/work'

In [4]: !pip install keras==2.2.4
        !pip install tensorflow==1.14.0

        Collecting keras==2.2.4
          Downloading Keras-2.2.4-py2.py3-none-any.whl (312 kB)
            |████████████████████████████████| 312 kB 24.0 MB/s eta 0:00:01
        Requirement already satisfied: six>=1.9.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4) (1.15.0)
        Requirement already satisfied: numpy>=1.9.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4) (1.20.3)
        Requirement already satisfied: h5py in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4) (3.2.1)
        Requirement already satisfied: pyyaml in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4) (5.4.1)
        Requirement already satisfied: scipy>=0.14 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4) (1.7.3)
        Requirement already satisfied: keras-preprocessing>=1.0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from keras==2.2.4) (1.
        1.2)
        Collecting keras-applications>=1.0.6
          Downloading Keras_Applications-1.0.8-py3-none-any.whl (50 kB)
            |████████████████████████████████| 50 kB 17.8 MB/s eta 0:00:01
        Installing collected packages: keras-applications, keras
          Attempting uninstall: keras
            Found existing installation: keras 2.7.0
```

Uploading dataset and integrating to code:



Upload one file at a time. All file types accepted. 5 GB max file size.

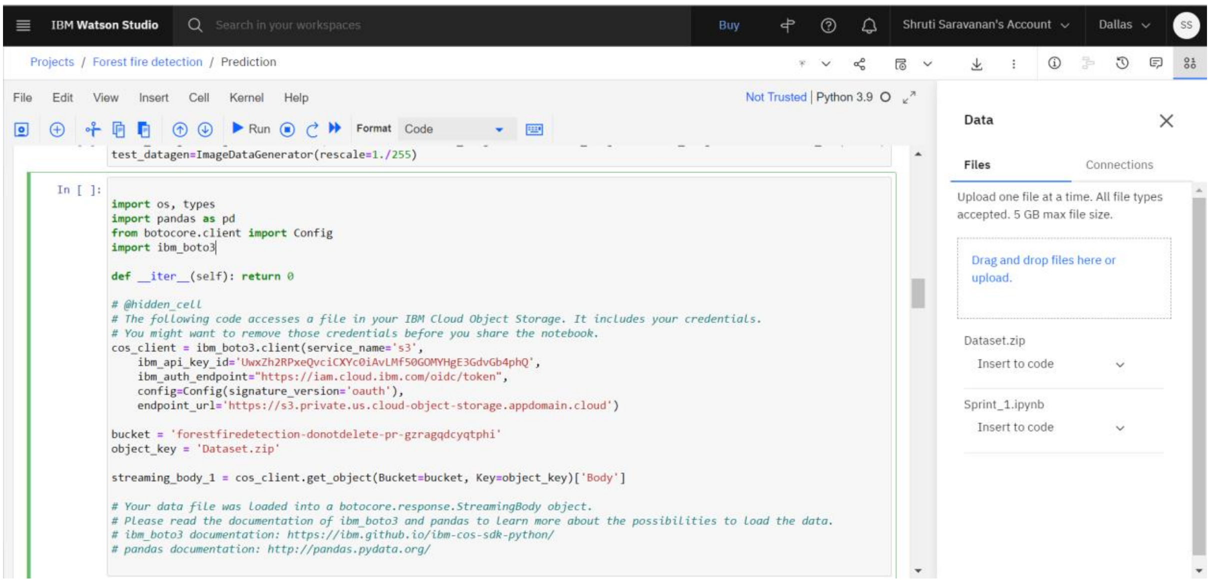
Drag and drop files here or upload.

Stay on the page until upload completes.
Incomplete uploads are cancelled.

Dataset.zip

Uploading file... 25 %

Sprint_1.ipynb


IBM Watson Studio
Search in your workspaces
Buy
Shruti Saravanan's Account
Dallas
SS
Projects / Forest fire detection / Prediction
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.9
Run
Format
Code
Data
Files
Connections
Insert to code
Insert to code

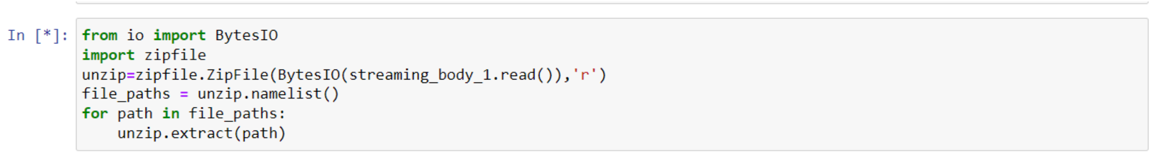
Upload one file at a time. All file types accepted. 5 GB max file size.

Drag and drop files here or upload.

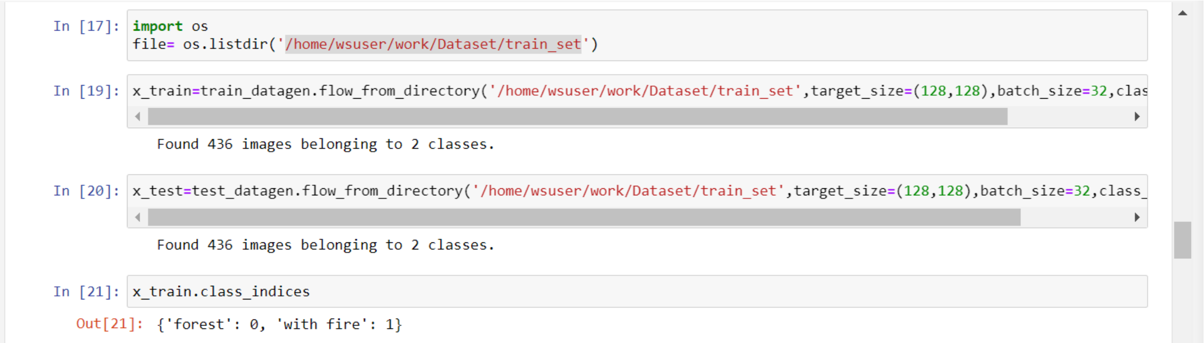
Dataset.zip

Sprint_1.ipynb

```
test_datagen=ImageDataGenerator(rescale=1./255)

In [ ]:
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='UwxZh2RPxeQvciCXYc0iAvLMf50GOMYHgE3GdvGb4phQ',
    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 = 'forestfiredetection-donotdelete-pr-gzragqdcyqtphi'
object_key = 'Dataset.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/
```

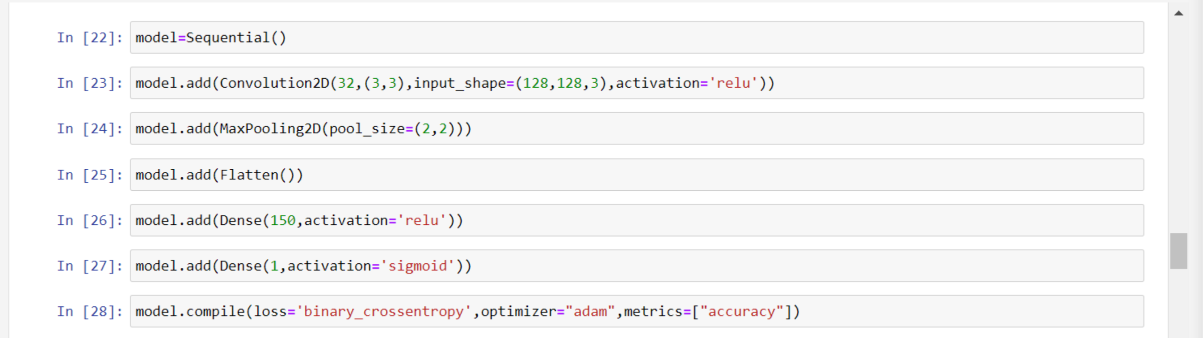


```
In [*]: from io import BytesIO
        import zipfile
        unzip=zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
        file_paths = unzip.namelist()
        for path in file_paths:
            unzip.extract(path)
```

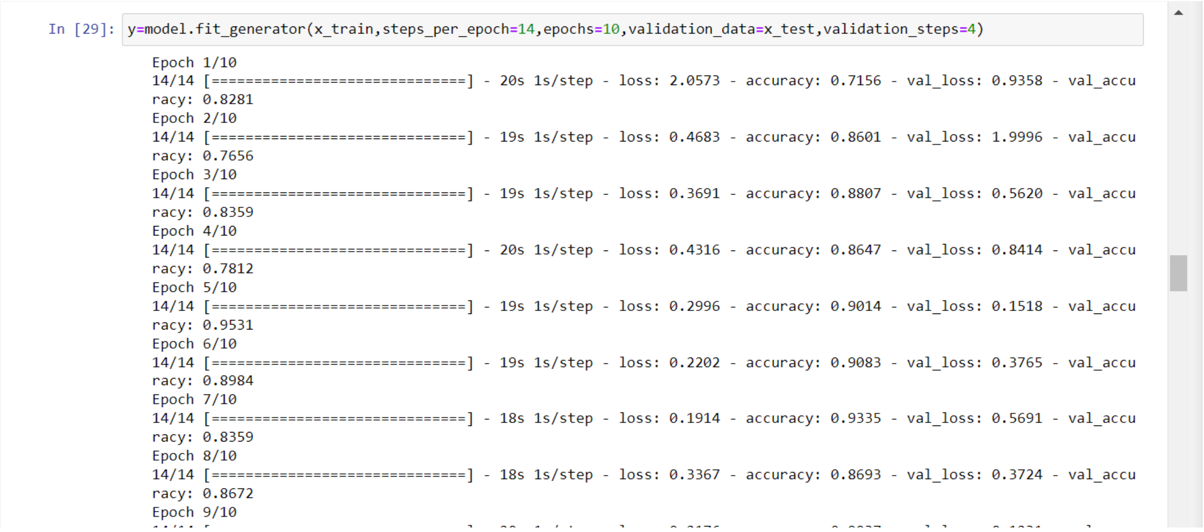


```
In [17]: import os
         file= os.listdir('/home/wsuser/work/Dataset/train_set')

In [19]: x_train=train_datagen.flow_from_directory('/home/wsuser/work/Dataset/train_set',target_size=(128,128),batch_size=32,clas

         Found 436 images belonging to 2 classes.

In [20]: x_test=test_datagen.flow_from_directory('/home/wsuser/work/Dataset/train_set',target_size=(128,128),batch_size=32,class_

         Found 436 images belonging to 2 classes.

In [21]: x_train.class_indices

Out[21]: {'forest': 0, 'with fire': 1}
```

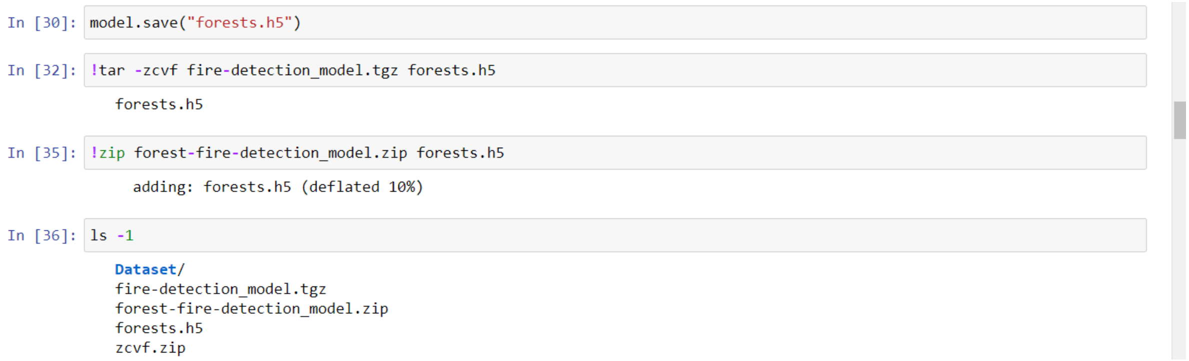
Building CNN model:



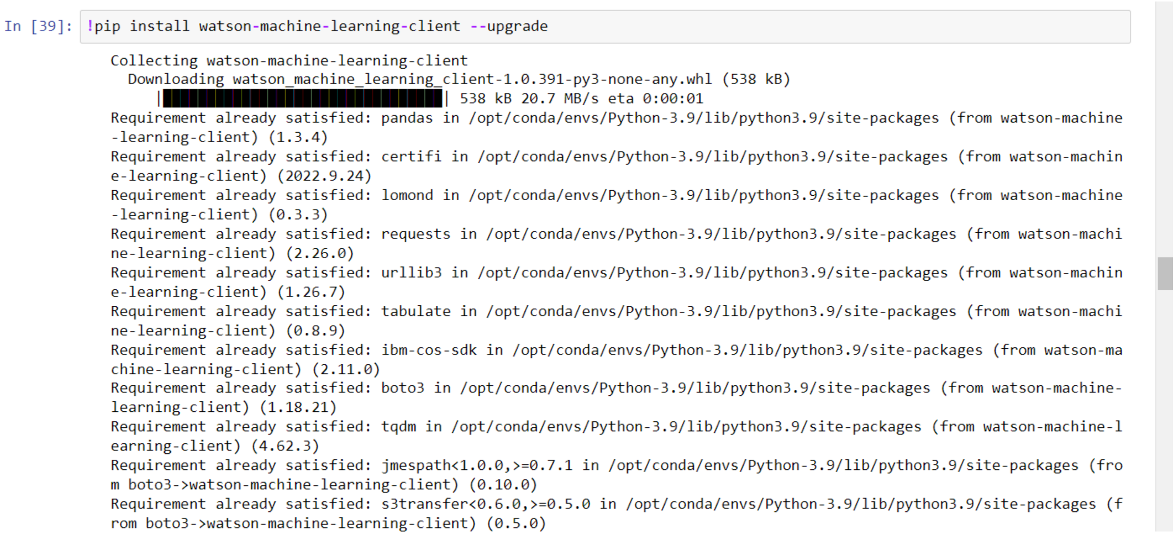
```
In [22]: model=Sequential()

In [23]: model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

In [24]: model.add(MaxPooling2D(pool_size=(2,2)))

In [25]: model.add(Flatten())

In [26]: model.add(Dense(150,activation='relu'))

In [27]: model.add(Dense(1,activation='sigmoid'))

In [28]: model.compile(loss='binary_crossentropy',optimizer="adam",metrics=["accuracy"])
```



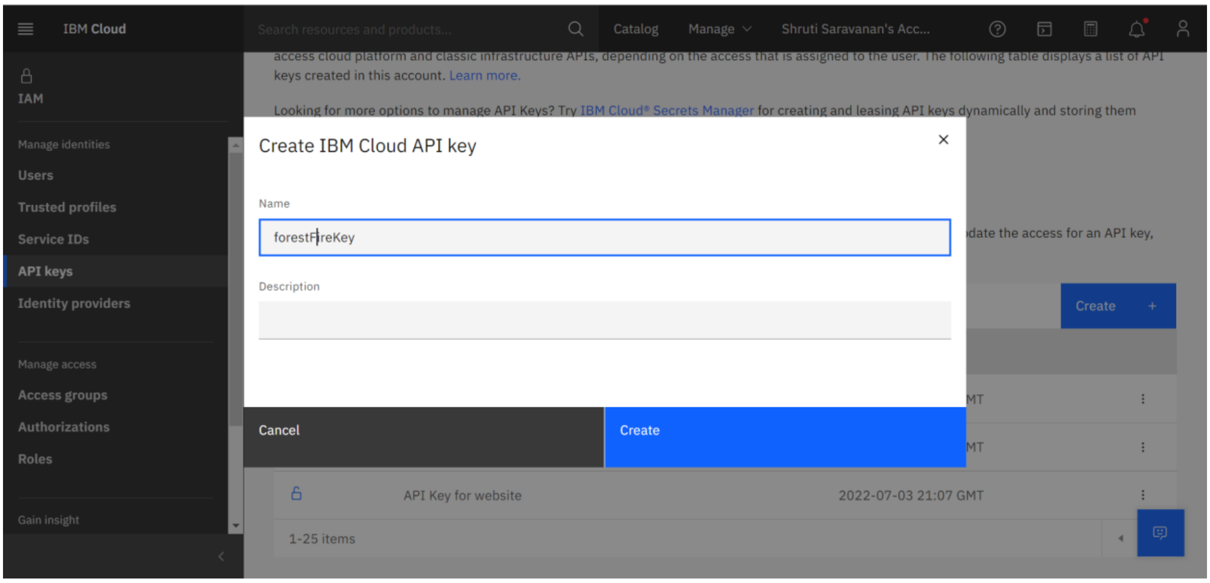
```
In [29]: y=model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_test,validation_steps=4)

         Epoch 1/10
         14/14 [==============================] - 20s 1s/step - loss: 2.0573 - accuracy: 0.7156 - val_loss: 0.9358 - val_accu
         racy: 0.8281
         Epoch 2/10
         14/14 [==============================] - 19s 1s/step - loss: 0.4683 - accuracy: 0.8601 - val_loss: 1.9996 - val_accu
         racy: 0.7656
         Epoch 3/10
         14/14 [==============================] - 19s 1s/step - loss: 0.3691 - accuracy: 0.8807 - val_loss: 0.5620 - val_accu
         racy: 0.8359
         Epoch 4/10
         14/14 [==============================] - 20s 1s/step - loss: 0.4316 - accuracy: 0.8647 - val_loss: 0.8414 - val_accu
         racy: 0.7812
         Epoch 5/10
         14/14 [==============================] - 19s 1s/step - loss: 0.2996 - accuracy: 0.9014 - val_loss: 0.1518 - val_accu
         racy: 0.9531
         Epoch 6/10
         14/14 [==============================] - 19s 1s/step - loss: 0.2202 - accuracy: 0.9083 - val_loss: 0.3765 - val_accu
         racy: 0.8984
         Epoch 7/10
         14/14 [==============================] - 18s 1s/step - loss: 0.1914 - accuracy: 0.9335 - val_loss: 0.5691 - val_accu
         racy: 0.8359
         Epoch 8/10
         14/14 [==============================] - 18s 1s/step - loss: 0.3367 - accuracy: 0.8693 - val_loss: 0.3724 - val_accu
         racy: 0.8672
         Epoch 9/10
```



```
In [30]: model.save("forests.h5")

In [32]: !tar -zcvf fire-detection_model.tgz forests.h5

         forests.h5

In [35]: !zip forest-fire-detection_model.zip forests.h5

           adding: forests.h5 (deflated 10%)

In [36]: ls -1

         Dataset/
         fire-detection_model.tgz
         forest-fire-detection_model.zip
         forests.h5
         zcvf.zip
```



```
In [39]: !pip install watson-machine-learning-client --upgrade

         Collecting watson-machine-learning-client
           Downloading watson_machine_learning_client-1.0.391-py3-none-any.whl (538 kB)
             |████████████████████████████████| 538 kB 20.7 MB/s eta 0:00:01
         Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine
         -learning-client) (1.3.4)
         Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machin
         e-learning-client) (2022.9.24)
         Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine
         -learning-client) (0.3.3)
         Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machi
         ne-learning-client) (2.26.0)
         Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machin
         e-learning-client) (1.26.7)
         Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machi
         ne-learning-client) (0.8.9)
         Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-ma
         chine-learning-client) (2.11.0)
         Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-
         learning-client) (1.18.21)
         Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-l
         earning-client) (4.62.3)
         Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (fro
         m boto3->watson-machine-learning-client) (0.10.0)
         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)
```



access cloud platform and classic infrastructure APIs, depending on the access that is assigned to the user. The following table displays a list of API keys created in this account. Learn more.

Looking for more options to manage API Keys? Try IBM Cloud® Secrets Manager for creating and leasing API keys dynamically and storing them

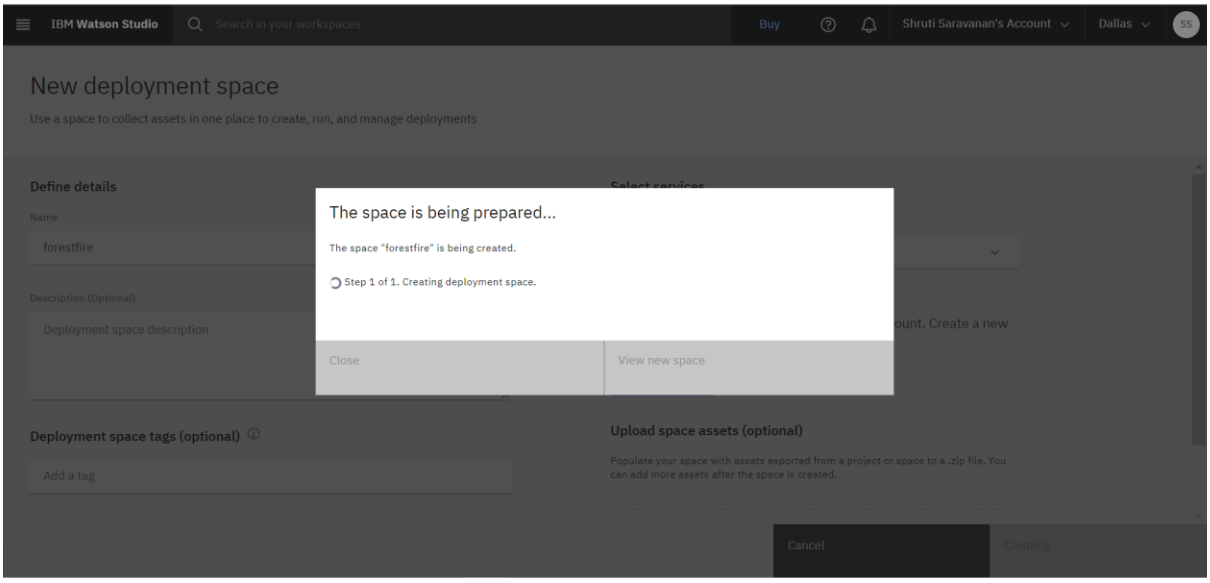
Create IBM Cloud API key

Name: forestFireKey

Description:

API Key for website 2022-07-03 21:07 GMT

1-25 items



New deployment space

Use a space to collect assets in one place to create, run, and manage deployments

Define details

Name: forestfire

Description (Optional): Deployment space description

Deployment space tags (optional)

Add a tag

Select services

ount. Create a new

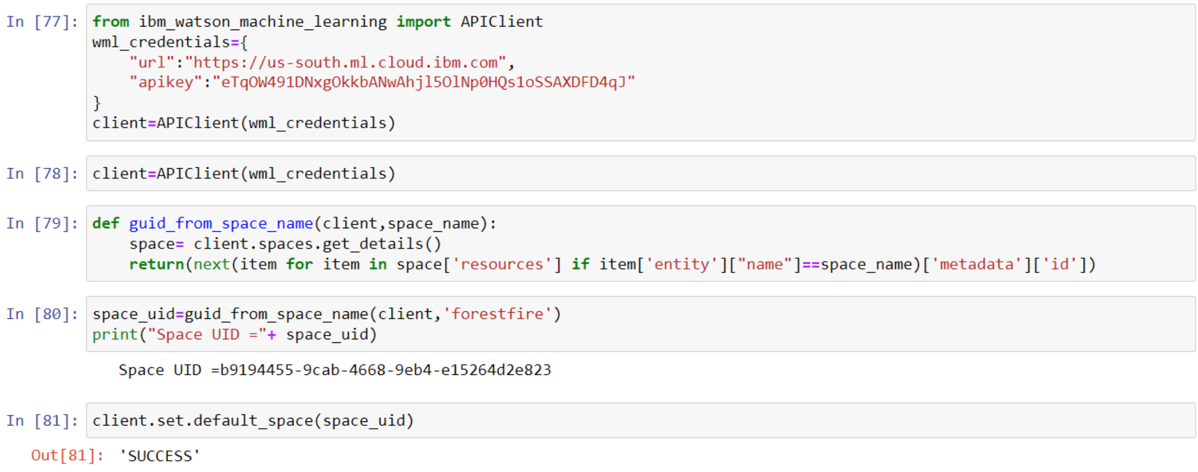
Upload space assets (optional)

Populate your space with assets exported from a project or space to a .zip file. You can add more assets after the space is created.

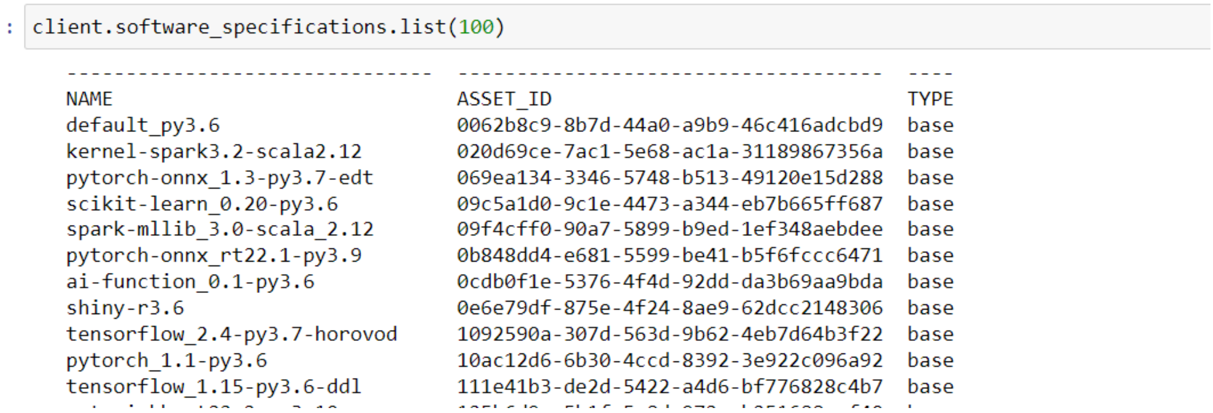
The space is being prepared...

The space "forestfire" is being created.

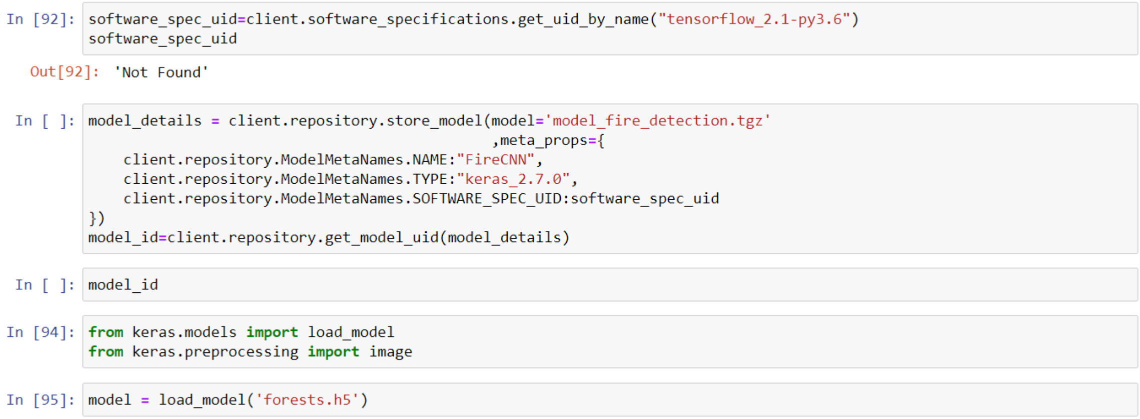
Step 1 of 1. Creating deployment space.



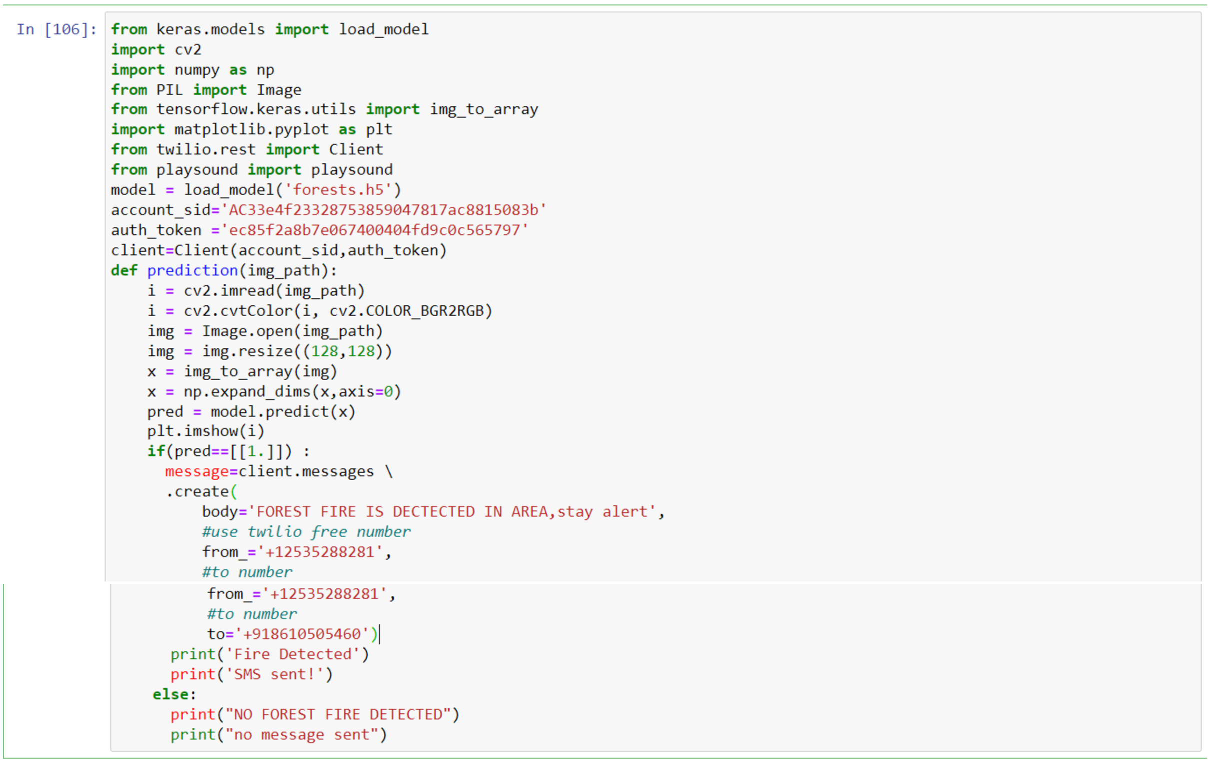
```
In [77]: from ibm_watson_machine_learning import APIClient
         wml_credentials={
             "url":"https://us-south.ml.cloud.ibm.com",
             "apikey":"eTqOW491DNxgOkkbANwAhjl5OlNp0HQs1oSSAXDFD4qJ"
         }
         client=APIClient(wml_credentials)

In [78]: client=APIClient(wml_credentials)

In [79]: 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 [80]: space_uid=guid_from_space_name(client,'forestfire')
         print("Space UID ="+ space_uid)

            Space UID =b9194455-9cab-4668-9eb4-e15264d2e823

In [81]: client.set.default_space(space_uid)

Out[81]: 'SUCCESS'
```



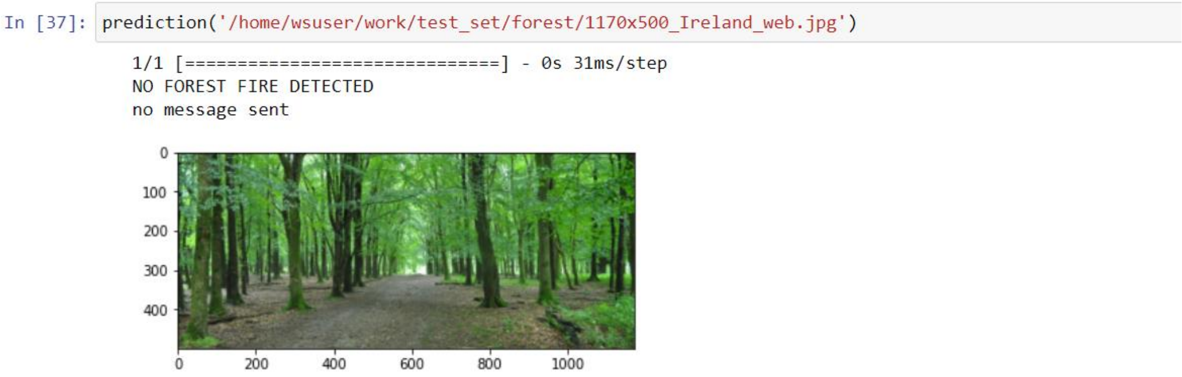
```
: client.software_specifications.list(100)

  -------------------------------  ------------------------------------  ----
  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
```



```
In [92]: software_spec_uid=client.software_specifications.get_uid_by_name("tensorflow_2.1-py3.6")
         software_spec_uid

Out[92]: 'Not Found'

In [ ]: model_details = client.repository.store_model(model='model_fire_detection.tgz'
                                                      ,meta_props={
             client.repository.ModelMetaNames.NAME:"FireCNN",
             client.repository.ModelMetaNames.TYPE:"keras_2.7.0",
             client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid
         })
         model_id=client.repository.get_model_uid(model_details)

In [ ]: model_id

In [94]: from keras.models import load_model
         from keras.preprocessing import image

In [95]: model = load_model('forests.h5')
```

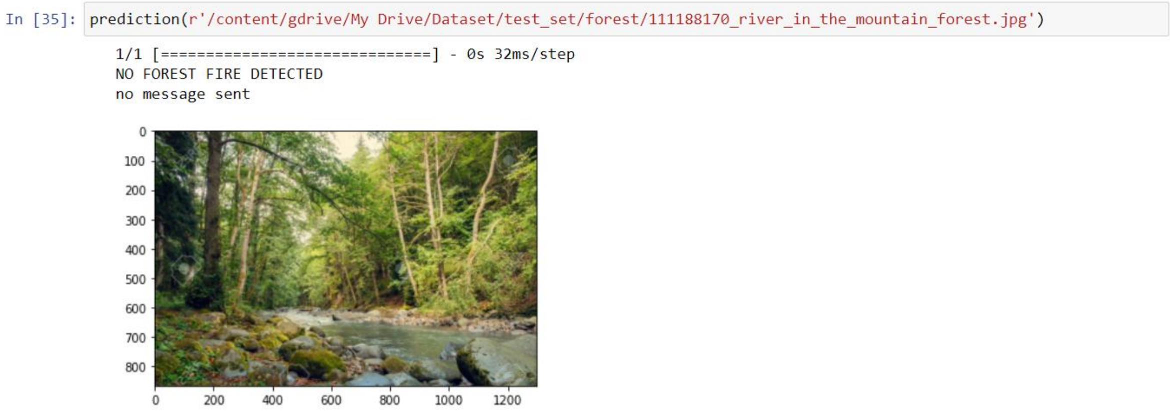


```
In [106]: from keras.models import load_model
          import cv2
          import numpy as np
          from PIL import Image
          from tensorflow.keras.utils import img_to_array
          import matplotlib.pyplot as plt
          from twilio.rest import Client
          from playsound import playsound
          model = load_model('forests.h5')
          account_sid='AC33e4f23328753859047817ac8815083b'
          auth_token ='ec85f2a8b7e067400404fd9c0c565797'
          client=Client(account_sid,auth_token)
          def prediction(img_path):
              i = cv2.imread(img_path)
              i = cv2.cvtColor(i, cv2.COLOR_BGR2RGB)
              img = Image.open(img_path)
              img = img.resize((128,128))
              x = img_to_array(img)
              x = np.expand_dims(x,axis=0)
              pred = model.predict(x)
              plt.imshow(i)
              if(pred==[[1.]]) :
                message=client.messages \
                .create(
                    body='FOREST FIRE IS DECTECTED IN AREA,stay alert',
                    #use twilio free number
                    from_='+12535288281',
                    #to number
                    from_='+12535288281',
                    #to number
                    to='+918610505460')
                print('Fire Detected')
                print('SMS sent!')
              else:
                print("NO FOREST FIRE DETECTED")
                print("no message sent")
```

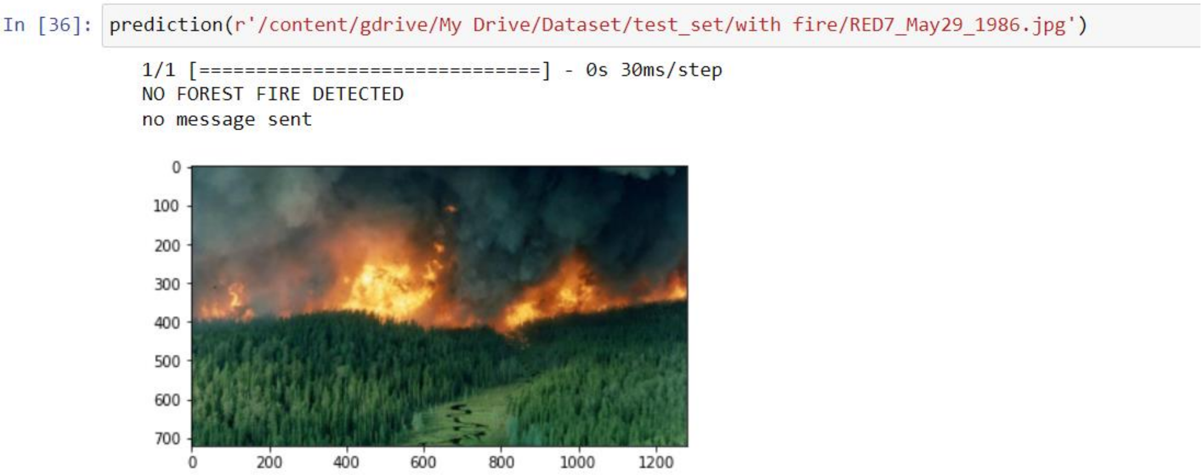


```
In [37]: prediction('/home/wsuser/work/test_set/forest/1170x500_Ireland_web.jpg')

1/1 [==============================] - 0s 31ms/step
NO FOREST FIRE DETECTED
no message sent
```

Using model locally:



```
In [35]: prediction(r'/content/gdrive/My Drive/Dataset/test_set/forest/111188170_river_in_the_mountain_forest.jpg')

1/1 [==============================] - 0s 32ms/step
NO FOREST FIRE DETECTED
no message sent
```



```
In [36]: prediction(r'/content/gdrive/My Drive/Dataset/test_set/with fire/RED7_May29_1986.jpg')

         1/1 [==============================] - 0s 30ms/step
         NO FOREST FIRE DETECTED
         no message sent
```