IBM CLOUD

|  |  |
| --- | --- |
| TEAM ID | PNT2022TMID40220 |
| PROJECT NAME | EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES |

{

"cells": [

{

"cell\_type": "markdown",

"metadata": {

"id": "5UKKDaCfUNCG"

},

"source": [

"#Importing Keras libraries"

]

},

{

"cell\_type": "code",

"execution\_count": 75,

"metadata": {

"id": "ipCWLeFzUXGa"

},

"outputs": [],

"source": [

"import keras"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "RNeZ0vrEUc55"

},

"source": [

"#Importing ImageDataGenerator from Keras"

]

},

{

"cell\_type": "code",

"execution\_count": 76,

"metadata": {

"id": "Mx2JxqSDUnPk"

},

"outputs": [],

"source": [

"from matplotlib import pyplot as plt\n",

"from keras.preprocessing.image import ImageDataGenerator"

]

},

{

"cell\_type": "code",

"execution\_count": 77,

"metadata": {},

"outputs": [],

"source": [

"\n",

"import os, types\n",

"import pandas as pd\n",

"from botocore.client import Config\n",

"import ibm\_boto3\n",

"\n",

"def \_\_iter\_\_(self): return 0\n",

"\n",

"# @hidden\_cell\n",

"# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.\n",

"# You might want to remove those credentials before you share the notebook.\n",

"cos\_client = ibm\_boto3.client(service\_name='s3',\n",

" ibm\_api\_key\_id='KXjEkgUBwu4dS1Lchix\_OeLTtOdfWFcOzlhwXRqXtHro',\n",

" ibm\_auth\_endpoint=\"https://iam.cloud.ibm.com/oidc/token\",\n",

" config=Config(signature\_version='oauth'),\n",

" endpoint\_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')\n",

"\n",

"bucket = 'forestfire-donotdelete-pr-wmm56yysfedwtp'\n",

"object\_key = 'Dataset.zip'\n",

"\n",

"streaming\_body\_2 = cos\_client.get\_object(Bucket=bucket, Key=object\_key)['Body']\n",

"\n",

"# Your data file was loaded into a botocore.response.StreamingBody object.\n",

"# Please read the documentation of ibm\_boto3 and pandas to learn more about the possibilities to load the data.\n",

"# ibm\_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/\n",

"# pandas documentation: http://pandas.pydata.org/\n"

]

},

{

"cell\_type": "code",

"execution\_count": 78,

"metadata": {},

"outputs": [],

"source": [

"from io import BytesIO\n",

"import zipfile\n",

"unzip = zipfile.ZipFile(BytesIO(streaming\_body\_2.read()),'r')\n",

"file\_paths = unzip.namelist()\n",

"for path in file\_paths:\n",

" unzip.extract(path)"

]

},

{

"cell\_type": "code",

"execution\_count": 79,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'/home/wsuser/work'"

]

},

"execution\_count": 79,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"pwd"

]

},

{

"cell\_type": "code",

"execution\_count": 80,

"metadata": {},

"outputs": [],

"source": [

"import os\n",

"filenames = os.listdir('/home/wsuser/work/Dataset/train\_set')"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "p7bB9Cd1b8kG"

},

"source": [

"Defining the Parameters"

]

},

{

"cell\_type": "code",

"execution\_count": 81,

"metadata": {

"id": "A1iR\_1R-cKY4"

},

"outputs": [],

"source": [

"train\_datagen=ImageDataGenerator(rescale=1./255,shear\_range=0.2,rotation\_range=180,zoom\_range=0.2,horizontal\_flip=True)\n",

"test\_datagen=ImageDataGenerator(rescale=1./255,shear\_range=0.2,rotation\_range=180,zoom\_range=0.2,horizontal\_flip=True)\n"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "pbYsUSI9rrd9"

},

"source": [

"#Applying ImageDataGenerator functionality to train dataset\n",

"\n"

]

},

{

"cell\_type": "code",

"execution\_count": 82,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "3ou6I0e9sR7o",

"outputId": "593301d8-eeba-4240-c203-a5cfef86b674"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Found 436 images belonging to 2 classes.\n"

]

}

],

"source": [

"x\_train=train\_datagen.flow\_from\_directory('/home/wsuser/work/Dataset/train\_set',target\_size=(64,64),batch\_size=32,class\_mode='binary')\n"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "943\_QP5BtsLF"

},

"source": [

"#Applying ImageDataGenerator functionality to test dataset"

]

},

{

"cell\_type": "code",

"execution\_count": 83,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "vIgeaeLht0Sn",

"outputId": "db6dfcce-f13f-4e1c-b077-19a1abd9487b"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Found 121 images belonging to 2 classes.\n"

]

}

],

"source": [

"x\_test=test\_datagen.flow\_from\_directory('/home/wsuser/work/Dataset/test\_set',target\_size=(64,64),batch\_size=32,class\_mode='binary')\n"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "ARyKyJLrVbK5"

},

"source": [

"#Importing Model Building Libraries"

]

},

{

"cell\_type": "code",

"execution\_count": 84,

"metadata": {

"id": "0sMPVWxFVg5P"

},

"outputs": [],

"source": [

"#to define the linear Initialisation import sequential\n",

"from keras.models import Sequential\n",

"#to add layers import Dense\n",

"from keras.layers import Dense\n",

"#to create Convolutional kernel import convolution2D\n",

"from keras.layers import Convolution2D\n",

"#import Maxpooling layer \n",

"from keras.layers import MaxPooling2D\n",

"#import flatten layer\n",

"from keras.layers import Flatten\n",

"import warnings\n",

"warnings.filterwarnings('ignore')"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "\_LVXcOT9eLM3"

},

"source": [

"#Initializing the model"

]

},

{

"cell\_type": "code",

"execution\_count": 85,

"metadata": {

"id": "TquY2jOfeRQF"

},

"outputs": [],

"source": [

"model = Sequential()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "CqOjIFO2eaKf"

},

"source": [

"#Adding CNN Layers"

]

},

{

"cell\_type": "code",

"execution\_count": 86,

"metadata": {

"id": "myoMJiBxeh3q"

},

"outputs": [],

"source": [

"model.add(Convolution2D(32,(3,3),input\_shape=(64,64,3),activation='relu'))\n",

"#add maxpooling layers\n",

"model.add(MaxPooling2D(pool\_size=(2,2)))\n",

"#add faltten layer\n",

"model.add(Flatten())"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "x67h-tMPfpHG"

},

"source": [

"#Add Dense layers"

]

},

{

"cell\_type": "code",

"execution\_count": 87,

"metadata": {

"id": "JodC\_f4Vfsn3"

},

"outputs": [],

"source": [

"#add hidden layers\n",

"model.add(Dense(150,activation='relu'))\n",

"#add output layer\n",

"model.add(Dense(1,activation='sigmoid'))"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "ELm-4mBdgfOv"

},

"source": [

"#configuring the learning process"

]

},

{

"cell\_type": "code",

"execution\_count": 88,

"metadata": {

"id": "evmGQQWXgo1t"

},

"outputs": [],

"source": [

"model.compile(loss='binary\_crossentropy',optimizer=\"adam\",metrics=[\"accuracy\"])"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "80s2C38ThYaj"

},

"source": [

"#Training the model"

]

},

{

"cell\_type": "code",

"execution\_count": 89,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "3Cy0ktrJhcfT",

"outputId": "dfd851cb-9f0a-490a-a9c7-40f21712d7c6"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Epoch 1/10\n",

"14/14 [==============================] - 23s 2s/step - loss: 0.8269 - accuracy: 0.6835 - val\_loss: 0.1792 - val\_accuracy: 0.9504\n",

"Epoch 2/10\n",

"14/14 [==============================] - 23s 2s/step - loss: 0.2426 - accuracy: 0.8876 - val\_loss: 0.1126 - val\_accuracy: 0.9587\n",

"Epoch 3/10\n",

"14/14 [==============================] - 22s 2s/step - loss: 0.2107 - accuracy: 0.9128 - val\_loss: 0.1256 - val\_accuracy: 0.9421\n",

"Epoch 4/10\n",

"14/14 [==============================] - 22s 2s/step - loss: 0.2927 - accuracy: 0.8784 - val\_loss: 0.1423 - val\_accuracy: 0.9256\n",

"Epoch 5/10\n",

"14/14 [==============================] - 21s 1s/step - loss: 0.1980 - accuracy: 0.9151 - val\_loss: 0.0976 - val\_accuracy: 0.9669\n",

"Epoch 6/10\n",

"14/14 [==============================] - 21s 1s/step - loss: 0.1891 - accuracy: 0.9128 - val\_loss: 0.0779 - val\_accuracy: 0.9669\n",

"Epoch 7/10\n",

"14/14 [==============================] - 21s 2s/step - loss: 0.1688 - accuracy: 0.9381 - val\_loss: 0.0945 - val\_accuracy: 0.9421\n",

"Epoch 8/10\n",

"14/14 [==============================] - 22s 2s/step - loss: 0.1768 - accuracy: 0.9243 - val\_loss: 0.0751 - val\_accuracy: 0.9835\n",

"Epoch 9/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.1583 - accuracy: 0.9312 - val\_loss: 0.0522 - val\_accuracy: 0.9917\n",

"Epoch 10/10\n",

"14/14 [==============================] - 21s 1s/step - loss: 0.1935 - accuracy: 0.9220 - val\_loss: 0.0562 - val\_accuracy: 0.9835\n"

]

},

{

"data": {

"text/plain": [

"<keras.callbacks.History at 0x7f2c91ec7bb0>"

]

},

"execution\_count": 89,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"model.fit\_generator(x\_train,steps\_per\_epoch=14,epochs=10,validation\_data=x\_test,validation\_steps=4)"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "CSmW5ypkizjF"

},

"source": [

"#Save the model"

]

},

{

"cell\_type": "code",

"execution\_count": 90,

"metadata": {

"id": "Q2DVxGYgi36I"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 91,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"forest.h5\r\n"

]

}

],

"source": [

"!tar -zcvf image-classification\_new.tgz forest.h5"

]

},

{

"cell\_type": "code",

"execution\_count": 92,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"\u001b[0m\u001b[01;34mDataset\u001b[0m/\r\n",

"forest.h5\r\n",

"image-classification\_new.tgz\r\n"

]

}

],

"source": [

"ls -1"

]

},

{

"cell\_type": "code",

"execution\_count": 93,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Requirement already satisfied: watson-machine-learning-client in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.391)\n",

"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)\n",

"Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)\n",

"Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)\n",

"Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)\n",

"Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)\n",

"Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)\n",

"Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)\n",

"Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)\n",

"Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)\n",

"Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.10.0)\n",

"Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (1.21.41)\n",

"Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.5.0)\n",

"Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (2.8.2)\n",

"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->botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (1.15.0)\n",

"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)\n",

"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->watson-machine-learning-client) (2.11.0)\n",

"Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client) (3.3)\n",

"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)\n",

"Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (2021.3)\n",

"Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (1.20.3)\n"

]

}

],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 94,

"metadata": {},

"outputs": [],

"source": [

"from ibm\_watson\_machine\_learning import APIClient\n",

"wml\_credentilas = {\n",

" \"url\":\"https://us-south.ml.cloud.ibm.com\",\n",

" \"apikey\":\"hxe6koyIaU12\_be6Qw-sQ8omzOrg9czDp9Ep11YppBs6\"\n",

"}\n",

"client = APIClient(wml\_credentilas)"

]

},

{

"cell\_type": "code",

"execution\_count": 95,

"metadata": {},

"outputs": [],

"source": [

"def guid\_from\_space\_name(client, space\_name):\n",

" space = client.spaces.get\_details()\n",

" return(next(item for item in space['resources'] if item['entity'][\"name\"] == space\_name)['metadata']['id'])"

]

},

{

"cell\_type": "code",

"execution\_count": 96,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Space UID = 2bae4b0b-57cd-4fd3-89ef-5fc4a44867a5\n"

]

}

],

"source": [

"space\_uid = guid\_from\_space\_name(client, 'Forestrecognition')\n",

"print(\"Space UID = \" + space\_uid)"

]

},

{

"cell\_type": "code",

"execution\_count": 97,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'SUCCESS'"

]

},

"execution\_count": 97,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"client.set.default\_space(space\_uid)"

]

},

{

"cell\_type": "code",

"execution\_count": 98,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"----------------------------- ------------------------------------ ----\n",

"NAME ASSET\_ID TYPE\n",

"default\_py3.6 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base\n",

"kernel-spark3.2-scala2.12 020d69ce-7ac1-5e68-ac1a-31189867356a base\n",

"pytorch-onnx\_1.3-py3.7-edt 069ea134-3346-5748-b513-49120e15d288 base\n",

"scikit-learn\_0.20-py3.6 09c5a1d0-9c1e-4473-a344-eb7b665ff687 base\n",

"spark-mllib\_3.0-scala\_2.12 09f4cff0-90a7-5899-b9ed-1ef348aebdee base\n",

"pytorch-onnx\_rt22.1-py3.9 0b848dd4-e681-5599-be41-b5f6fccc6471 base\n",

"ai-function\_0.1-py3.6 0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda base\n",

"shiny-r3.6 0e6e79df-875e-4f24-8ae9-62dcc2148306 base\n",

"tensorflow\_2.4-py3.7-horovod 1092590a-307d-563d-9b62-4eb7d64b3f22 base\n",

"pytorch\_1.1-py3.6 10ac12d6-6b30-4ccd-8392-3e922c096a92 base\n",

"tensorflow\_1.15-py3.6-ddl 111e41b3-de2d-5422-a4d6-bf776828c4b7 base\n",

"autoai-kb\_rt22.2-py3.10 125b6d9a-5b1f-5e8d-972a-b251688ccf40 base\n",

"runtime-22.1-py3.9 12b83a17-24d8-5082-900f-0ab31fbfd3cb base\n",

"scikit-learn\_0.22-py3.6 154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base\n",

"default\_r3.6 1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base\n",

"pytorch-onnx\_1.3-py3.6 1bc6029a-cc97-56da-b8e0-39c3880dbbe7 base\n",

"kernel-spark3.3-r3.6 1c9e5454-f216-59dd-a20e-474a5cdf5988 base\n",

"pytorch-onnx\_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f base\n",

"tensorflow\_2.1-py3.6 1eb25b84-d6ed-5dde-b6a5-3fbdf1665666 base\n",

"spark-mllib\_3.2 20047f72-0a98-58c7-9ff5-a77b012eb8f5 base\n",

"tensorflow\_2.4-py3.8-horovod 217c16f6-178f-56bf-824a-b19f20564c49 base\n",

"runtime-22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0-da66306ce658 base\n",

"do\_py3.8 295addb5-9ef9-547e-9bf4-92ae3563e720 base\n",

"autoai-ts\_3.8-py3.8 2aa0c932-798f-5ae9-abd6-15e0c2402fb5 base\n",

"tensorflow\_1.15-py3.6 2b73a275-7cbf-420b-a912-eae7f436e0bc base\n",

"kernel-spark3.3-py3.9 2b7961e2-e3b1-5a8c-a491-482c8368839a base\n",

"pytorch\_1.2-py3.6 2c8ef57d-2687-4b7d-acce-01f94976dac1 base\n",

"spark-mllib\_2.3 2e51f700-bca0-4b0d-88dc-5c6791338875 base\n",

"pytorch-onnx\_1.1-py3.6-edt 32983cea-3f32-4400-8965-dde874a8d67e base\n",

"spark-mllib\_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 base\n",

"spark-mllib\_2.4 390d21f8-e58b-4fac-9c55-d7ceda621326 base\n",

"autoai-ts\_rt22.2-py3.10 396b2e83-0953-5b86-9a55-7ce1628a406f base\n",

"xgboost\_0.82-py3.6 39e31acd-5f30-41dc-ae44-60233c80306e base\n",

"pytorch-onnx\_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-fb03b6f4fe12 base\n",

"pytorch-onnx\_rt22.2-py3.10 40e73f55-783a-5535-b3fa-0c8b94291431 base\n",

"default\_r36py38 41c247d3-45f8-5a71-b065-8580229facf0 base\n",

"autoai-ts\_rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 base\n",

"autoai-obm\_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 base\n",

"pmml-3.0\_4.3 493bcb95-16f1-5bc5-bee8-81b8af80e9c7 base\n",

"spark-mllib\_2.4-r\_3.6 49403dff-92e9-4c87-a3d7-a42d0021c095 base\n",

"xgboost\_0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 base\n",

"pytorch-onnx\_1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b base\n",

"autoai-ts\_3.9-py3.8 52c57136-80fa-572e-8728-a5e7cbb42cde base\n",

"spark-mllib\_2.4-scala\_2.11 55a70f99-7320-4be5-9fb9-9edb5a443af5 base\n",

"spark-mllib\_3.0 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9 base\n",

"autoai-obm\_2.0 5c2e37fa-80b8-5e77-840f-d912469614ee base\n",

"spss-modeler\_18.1 5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b base\n",

"cuda-py3.8 5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e base\n",

"autoai-kb\_3.1-py3.7 632d4b22-10aa-5180-88f0-f52dfb6444d7 base\n",

"pytorch-onnx\_1.7-py3.8 634d3cdc-b562-5bf9-a2d4-ea90a478456b base\n",

"----------------------------- ------------------------------------ ----\n",

"Note: Only first 50 records were displayed. To display more use 'limit' parameter.\n"

]

}

],

"source": [

"client.software\_specifications.list()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "G8KMppTojF\_\_"

},

"source": [

"#Predictions"

]

},

{

"cell\_type": "code",

"execution\_count": 99,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'acd9c798-6974-5d2f-a657-ce06e986df4d'"

]

},

"execution\_count": 99,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"software\_spec\_uid = client.software\_specifications.get\_uid\_by\_name(\"tensorflow\_rt22.1-py3.9\")\n",

"software\_spec\_uid"

]

},

{

"cell\_type": "code",

"execution\_count": 100,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"This method is deprecated, please use get\_model\_id()\n"

]

}

],

"source": [

"model\_details = client.repository.store\_model(model='image-classification\_new.tgz',meta\_props={ \n",

"client.repository.ModelMetaNames.NAME:'CNN',\n",

"client.repository.ModelMetaNames.TYPE:\"tensorflow\_rt22.1\",\n",

"client.repository.ModelMetaNames.SOFTWARE\_SPEC\_UID:software\_spec\_uid}\n",

" )\n",

"model\_id = client.repository.get\_model\_uid(model\_details)"

]

},

{

"cell\_type": "code",

"execution\_count": 101,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'1baa1aab-07c5-4a4a-a297-9b4c3444d699'"

]

},

"execution\_count": 101,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"model\_id"

]

},

{

"cell\_type": "code",

"execution\_count": 104,

"metadata": {

"id": "aZxNjrmJjImS"

},

"outputs": [],

"source": [

"#import load model from keras.model\n",

"from keras.models import load\_model\n",

"#import image from keras\n",

"from tensorflow.keras.preprocessing import image\n",

"import numpy as np\n",

"#import cv2\n",

"import cv2\n",

"#load the saved model\n",

"model=load\_model(\"forest.h5\")\n",

"img=image.load\_img('/home/wsuser/work/Dataset/test\_set/with fire/forest\_fire\_2268729\_1280.jpg')\n",

"x=image.img\_to\_array(img)\n",

"res=cv2.resize(x,dsize=(64,64),interpolation=cv2.INTER\_CUBIC)\n",

"#expand the image shape\n",

"x=np.expand\_dims(res,axis=0)"

]

},

{

"cell\_type": "code",

"execution\_count": 105,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "mR2jyaRCHMpM",

"outputId": "1c0c1544-60be-4bff-d3dc-a0f29d2bdac9"

},

"outputs": [

{

"data": {

"text/plain": [

"1"

]

},

"execution\_count": 105,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"pred=model.predict(x)\n",

"pred = int(pred[0][0])\n",

"pred\n",

"int(pred)"

]

},

{

"cell\_type": "code",

"execution\_count": 107,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "ZXIIcQnkcqHt",

"outputId": "8f780b37-25c5-4a15-a7f2-26abff48556f"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Forest fire\n"

]

}

],

"source": [

"if pred==1:\n",

" print('Forest fire')\n",

"elif pred==0:\n",

" print('No Fire')\n"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "gmgJxLoc47Xd"

},

"source": [

"#Open cv for video processing"

]

},

{

"cell\_type": "code",

"execution\_count": 108,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "gl3ITnbf5mWo",

"outputId": "e944bd46-2e61-4516-c0ff-aea25db57b85"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Requirement already satisfied: twilio in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (7.15.2)\n",

"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)\n",

"Requirement already satisfied: pytz in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from twilio) (2021.3)\n",

"Requirement already satisfied: requests>=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from twilio) (2.26.0)\n",

"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)\n",

"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)\n",

"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)\n",

"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)\n",

"Note: you may need to restart the kernel to use updated packages.\n"

]

}

],

"source": [

"pip install twilio"

]

},

{

"cell\_type": "code",

"execution\_count": 109,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "dDQfLrD66e1k",

"outputId": "bb0d2c10-344f-47bd-88f4-4aedd0186ef1"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Requirement already satisfied: playsound in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.3.0)\n",

"Note: you may need to restart the kernel to use updated packages.\n"

]

}

],

"source": [

"pip install playsound"

]

},

{

"cell\_type": "code",

"execution\_count": 112,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "L6CCxpf86y3g",

"outputId": "28cd53a1-b70f-40f9-a6ef-0a023eec68e7"

},

"outputs": [],

"source": [

"from logging import WARNING\n",

"#import opencv library\n",

"import cv2\n",

"#import numpy\n",

"import numpy as np\n",

"#import image function from keras\n",

"from keras.preprocessing import image\n",

"#import load\_model from keras\n",

"from keras.models import load\_model\n",

"#import client from twilio API\n",

"from twilio.rest import Client\n",

"#import playsound package\n",

"from playsound import playsound\n"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "CTN\_DehO9cjv"

},

"source": [

"#Creating An Account in Twilio Service"

]

},

{

"cell\_type": "code",

"execution\_count": 113,

"metadata": {},

"outputs": [],

"source": [

"\n",

"import os, types\n",

"import pandas as pd\n",

"from botocore.client import Config\n",

"import ibm\_boto3\n",

"\n",

"def \_\_iter\_\_(self): return 0\n",

"\n",

"# @hidden\_cell\n",

"# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.\n",

"# You might want to remove those credentials before you share the notebook.\n",

"cos\_client = ibm\_boto3.client(service\_name='s3',\n",

" ibm\_api\_key\_id='KXjEkgUBwu4dS1Lchix\_OeLTtOdfWFcOzlhwXRqXtHro',\n",

" ibm\_auth\_endpoint=\"https://iam.cloud.ibm.com/oidc/token\",\n",

" config=Config(signature\_version='oauth'),\n",

" endpoint\_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')\n",

"\n",

"bucket = 'forestfire-donotdelete-pr-wmm56yysfedwtp'\n",

"object\_key = 'Vishwaroopam.mp3'\n",

"\n",

"streaming\_body\_3 = cos\_client.get\_object(Bucket=bucket, Key=object\_key)['Body']\n",

"\n",

"# Your data file was loaded into a botocore.response.StreamingBody object.\n",

"# Please read the documentation of ibm\_boto3 and pandas to learn more about the possibilities to load the data.\n",

"# ibm\_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/\n",

"# pandas documentation: http://pandas.pydata.org/\n"

]

},

{

"cell\_type": "code",

"execution\_count": 117,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "Prz2Z8ffjOBh",

"outputId": "4f9e6e5b-d9e6-422b-a9e0-2427149817ba"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Forest fire\n",

"SM8520469cbcb2d1a83aba6aeaff9dbbca\n",

"Fire detected\n",

"SMS Sent!\n"

]

}

],

"source": [

"from twilio.rest import Client\n",

"from playsound import playsound\n",

"if pred==1:\n",

" print('Forest fire')\n",

" account\_sid='AC34c4bee5e03df7bc7dba1eef29761275'\n",

" auth\_token='1fc522239435d0c251c1fd870d715295'\n",

" client=Client(account\_sid,auth\_token)\n",

" message=client.messages \\\n",

" .create(\n",

" body='forest fire is detected,stay alert',\n",

" #use twilio free number\n",

" from\_='+19803934024',\n",

" #to number\n",

" to='+919962082226')\n",

" print(message.sid)\n",

" print(\"Fire detected\")\n",

" print(\"SMS Sent!\")"

]

}

],

"metadata": {

"colab": {

"collapsed\_sections": [],

"provenance": []

},

"kernelspec": {

"display\_name": "Python 3.9",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.9.13"

}

},

"nbformat": 4,

"nbformat\_minor": 1

}