{

"cells": [

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# IBM Project Name: Real-Time Communication System Powered by AI for Specially Abled\n",

"# TEAM ID: PNT2022TMID36138\n",

"# TEAM Member:Ganesh K M"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# IBM WATSON STUDIO DEPLOYMENT CODE "

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 1.]INSTALLING THE KERAS ,INSTALLING THE TENSORFLOW"

]

},

{

"cell\_type": "code",

"execution\_count": 97,

"metadata": {

"colab": {

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

},

"id": "rNxm\_bwDDmMj",

"outputId": "e55c8ab9-8150-4d3e-ce51-bd273575630e",

"scrolled": false

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Requirement already satisfied: Keras==2.2.4 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.2.4)\r\n",

"Requirement already satisfied: h5py in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from Keras==2.2.4) (3.2.1)\r\n",

"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)\r\n",

"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)\r\n",

"Requirement already satisfied: pyyaml in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from Keras==2.2.4) (5.4.1)\r\n",

"Requirement already satisfied: keras-applications>=1.0.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from Keras==2.2.4) (1.0.8)\r\n",

"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)\r\n",

"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)\r\n"

]

}

],

"source": [

"!pip install Keras==2.2.4\n",

"!pip install tensorflow==2.7"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\n",

"# 2.]IMPORTING LIBRARIES TO BUILD MODEL."

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "dutY\_IwFDl9g"

},

"outputs": [],

"source": [

"#library to train the model\n",

"import keras\n",

"import tensorflow\n",

"\n",

"\n",

"from tensorflow.keras.models import Sequential\n",

"from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D, Flatten"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 3.]IMPORTING LIBRARIES FOR IMAGE AUGMENTATION."

]

},

{

"cell\_type": "code",

"execution\_count": 99,

"metadata": {

"id": "3CSbv31FJuT1"

},

"outputs": [],

"source": [

"#image augmentation\n",

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

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

"test\_datagen=ImageDataGenerator(rescale=1./255)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 4.]ADDING STREAMING\_BODY\_OBJECT FOR DATASET.ZIP"

]

},

{

"cell\_type": "code",

"execution\_count": 100,

"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='aqprHZFuH38ECUn869hHk4qyvS\_iKJfrZAWUJJQ-mQKx',\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 = 'realtimecommunicationforspecially-donotdelete-pr-rfqndcvwgch6fu'\n",

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

"\n",

"streaming\_body\_4 = 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": 101,

"metadata": {

"colab": {

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

},

"id": "haMajSs9DliR",

"outputId": "7451541d-41e1-4ba8-9c28-cf00eea03b0b"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"\u001b[0m\u001b[01;34mDataset\u001b[0m/ \u001b[01;34mtest\_set\u001b[0m/ \u001b[01;34mtraining\_set\u001b[0m/\r\n"

]

}

],

"source": [

"ls"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "bqmiYuHxEe8t"

},

"source": [

"# 5.]UNZIPPING THE DATASET"

]

},

{

"cell\_type": "code",

"execution\_count": 102,

"metadata": {

"colab": {

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

"height": 130

},

"id": "R8H9cP5PEhbj",

"outputId": "ce76c8c9-6b37-4849-ea47-97dad4d231a7"

},

"outputs": [],

"source": [

"from io import BytesIO\n",

"import zipfile\n",

"unzip=zipfile.ZipFile(BytesIO(streaming\_body\_4.read()),'r')\n",

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

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

" unzip.extract(path)\n"

]

},

{

"cell\_type": "code",

"execution\_count": 103,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'/home/wsuser/work/Dataset'"

]

},

"execution\_count": 103,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"pwd"

]

},

{

"cell\_type": "code",

"execution\_count": 104,

"metadata": {

"colab": {

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

},

"id": "L1-M6zotIpLy",

"outputId": "2ae9caf5-a518-4b49-cb5b-e65f96842168"

},

"outputs": [],

"source": [

"#checking that the dataset is there are not\n",

"import os\n",

"filenamer = os.listdir('/home/wsuser/work/Dataset/training\_set')"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 6.]TRAINING AND TESTING IMAGES UNDER CLASSES"

]

},

{

"cell\_type": "code",

"execution\_count": 105,

"metadata": {

"colab": {

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

},

"id": "fWBTCyOVKp01",

"outputId": "c01cd057-5eae-429d-c737-541bb598118b"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Found 15750 images belonging to 9 classes.\n"

]

}

],

"source": [

"x\_train=train\_datagen.flow\_from\_directory(\"/home/wsuser/work/Dataset/training\_set\",target\_size=(64,64),class\_mode=\"categorical\",batch\_size=25)"

]

},

{

"cell\_type": "code",

"execution\_count": 106,

"metadata": {

"colab": {

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

},

"id": "yrG3iHrCKpRP",

"outputId": "148c6bdd-fa51-4729-91e1-9ca60d5c7b5a"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Found 2250 images belonging to 9 classes.\n"

]

}

],

"source": [

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

"class\_mode='categorical' , batch\_size=25)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 7.]TOTAL CLASSES UNDER TRAINING AND TESTING."

]

},

{

"cell\_type": "code",

"execution\_count": 107,

"metadata": {

"colab": {

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

},

"id": "vjoAMqLiL2BV",

"outputId": "2469327f-bc34-4811-9d6f-8d16e3ee57ff"

},

"outputs": [

{

"data": {

"text/plain": [

"{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}"

]

},

"execution\_count": 107,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x\_train.class\_indices"

]

},

{

"cell\_type": "code",

"execution\_count": 108,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}"

]

},

"execution\_count": 108,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x\_test.class\_indices"

]

},

{

"cell\_type": "code",

"execution\_count": 109,

"metadata": {

"id": "yke48DPEKGm3"

},

"outputs": [],

"source": [

"train\_datagen=ImageDataGenerator(rescale=1./255,zoom\_range=0.2,horizontal\_flip=True,vertical\_flip=False)"

]

},

{

"cell\_type": "code",

"execution\_count": 110,

"metadata": {

"id": "XglvnjXpKGTF"

},

"outputs": [],

"source": [

"test\_datagen=ImageDataGenerator(rescale=1./255)"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "Ska8jAKhMcjQ"

},

"source": [

"# 8.]MODEL BUILDING USING CNN"

]

},

{

"cell\_type": "code",

"execution\_count": 111,

"metadata": {

"id": "b9q-J6A0ME4K"

},

"outputs": [],

"source": [

"model=Sequential()"

]

},

{

"cell\_type": "code",

"execution\_count": 112,

"metadata": {

"id": "jMemaPnZNUHz"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 113,

"metadata": {

"id": "FC-UXn7wNT6x"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 114,

"metadata": {

"id": "Bib-ZohnNTet"

},

"outputs": [],

"source": [

"model.add(Flatten())"

]

},

{

"cell\_type": "code",

"execution\_count": 115,

"metadata": {

"colab": {

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

},

"id": "07-A3ymZNkOl",

"outputId": "4158a17e-898d-4dd1-e3b0-2ae5927c2ae0"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Model: \"sequential\_1\"\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"Layer (type) Output Shape Param # \n",

"=================================================================\n",

"conv2d\_1 (Conv2D) (None, 62, 62, 32) 896 \n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"max\_pooling2d\_1 (MaxPooling2 (None, 31, 31, 32) 0 \n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

"flatten\_1 (Flatten) (None, 30752) 0 \n",

"=================================================================\n",

"Total params: 896\n",

"Trainable params: 896\n",

"Non-trainable params: 0\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"

]

}

],

"source": [

"model.summary()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "QZoyo7TtNj9u"

},

"source": [

"# 9.]ADDING LAYERS FOR MODEL TRAINING."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "HLK-QpeFNwyz"

},

"source": [

"# HIDDEN LAYERS"

]

},

{

"cell\_type": "code",

"execution\_count": 117,

"metadata": {

"id": "fYWVG08rNjwG"

},

"outputs": [],

"source": [

"model.add(Dense(units = 300, activation='relu'))\n",

"#model.add(Dense(unit = 150,init = \"uniform\" activation='softmax'))"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "Qm4LWKnWN81\_"

},

"source": [

"# OUTPUT LAYERS"

]

},

{

"cell\_type": "code",

"execution\_count": 118,

"metadata": {

"id": "rCwPljf-NjgO"

},

"outputs": [],

"source": [

"model.add(Dense(units = 5, activation='softmax'))"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 10.]OPTIMIZING THE MODEL "

]

},

{

"cell\_type": "code",

"execution\_count": 119,

"metadata": {

"id": "TlnJKIOGOD6t"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 120,

"metadata": {

"colab": {

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

},

"id": "zLuzriYTODnO",

"outputId": "117cf1c3-97af-4d83-bc0d-42f5dfa28682"

},

"outputs": [

{

"data": {

"text/plain": [

"630"

]

},

"execution\_count": 120,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"\n",

"len(x\_train)"

]

},

{

"cell\_type": "code",

"execution\_count": 121,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"90"

]

},

"execution\_count": 121,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"len(x\_test)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 11.]FITTING THE MODEL"

]

},

{

"cell\_type": "code",

"execution\_count": 125,

"metadata": {

"colab": {

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

"height": 1000

},

"id": "dDjZmKsWOPlc",

"outputId": "989390bd-4c52-49c7-8408-ce22d8f4dfc3"

},

"outputs": [

{

"ename": "InvalidArgumentError",

"evalue": " logits and labels must be broadcastable: logits\_size=[25,5] labels\_size=[25,9]\n\t [[node categorical\_crossentropy/softmax\_cross\_entropy\_with\_logits\n (defined at /opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/backend.py:4889)\n]] [Op:\_\_inference\_train\_function\_2383]\n\nErrors may have originated from an input operation.\nInput Source operations connected to node categorical\_crossentropy/softmax\_cross\_entropy\_with\_logits:\nIn[0] categorical\_crossentropy/softmax\_cross\_entropy\_with\_logits/Reshape:\t\nIn[1] categorical\_crossentropy/softmax\_cross\_entropy\_with\_logits/Reshape\_1:\n\nOperation defined at: (most recent call last)\n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/runpy.py\", line 197, in \_run\_module\_as\_main\n>>> return \_run\_code(code, main\_globals, None,\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/runpy.py\", line 87, in \_run\_code\n>>> exec(code, run\_globals)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/\_\_main\_\_.py\", line 3, in <module>\n>>> app.launch\_new\_instance()\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/traitlets/config/application.py\", line 846, in launch\_instance\n>>> app.start()\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/kernelapp.py\", line 677, in start\n>>> self.io\_loop.start()\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tornado/platform/asyncio.py\", line 199, in start\n>>> self.asyncio\_loop.run\_forever()\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/asyncio/base\_events.py\", line 601, in run\_forever\n>>> self.\_run\_once()\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/asyncio/base\_events.py\", line 1905, in \_run\_once\n>>> handle.\_run()\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/asyncio/events.py\", line 80, in \_run\n>>> self.\_context.run(self.\_callback, \*self.\_args)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/kernelbase.py\", line 457, in dispatch\_queue\n>>> await self.process\_one()\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/kernelbase.py\", line 446, in process\_one\n>>> await dispatch(\*args)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/kernelbase.py\", line 353, in dispatch\_shell\n>>> await result\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/kernelbase.py\", line 648, in execute\_request\n>>> reply\_content = await reply\_content\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/ipkernel.py\", line 353, in do\_execute\n>>> res = shell.run\_cell(code, store\_history=store\_history, silent=silent)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/zmqshell.py\", line 533, in run\_cell\n>>> return super(ZMQInteractiveShell, self).run\_cell(\*args, \*\*kwargs)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/IPython/core/interactiveshell.py\", line 2914, in run\_cell\n>>> result = self.\_run\_cell(\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/IPython/core/interactiveshell.py\", line 2960, in \_run\_cell\n>>> return runner(coro)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/IPython/core/async\_helpers.py\", line 78, in \_pseudo\_sync\_runner\n>>> coro.send(None)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/IPython/core/interactiveshell.py\", line 3185, in run\_cell\_async\n>>> has\_raised = await self.run\_ast\_nodes(code\_ast.body, cell\_name,\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/IPython/core/interactiveshell.py\", line 3377, in run\_ast\_nodes\n>>> if (await self.run\_code(code, result, async\_=asy)):\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/IPython/core/interactiveshell.py\", line 3457, in run\_code\n>>> exec(code\_obj, self.user\_global\_ns, self.user\_ns)\n>>> \n>>> File \"/tmp/wsuser/ipykernel\_164/3808038373.py\", line 3, in <module>\n>>> model.fit\_generator(x\_train,steps\_per\_epoch=630,epochs=1,validation\_data=x\_test,validation\_steps=90)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py\", line 1966, in fit\_generator\n>>> return self.fit(\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py\", line 1189, in fit\n>>> tmp\_logs = self.train\_function(iterator)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py\", line 859, in train\_function\n>>> return step\_function(self, iterator)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py\", line 849, in step\_function\n>>> outputs = model.distribute\_strategy.run(run\_step, args=(data,))\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py\", line 842, in run\_step\n>>> outputs = model.train\_step(data)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py\", line 800, in train\_step\n>>> loss = self.compiled\_loss(\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/compile\_utils.py\", line 204, in \_\_call\_\_\n>>> loss\_value = loss\_obj(y\_t, y\_p, sample\_weight=sw)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/losses.py\", line 155, in \_\_call\_\_\n>>> losses = call\_fn(y\_true, y\_pred)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/losses.py\", line 259, in call\n>>> return ag\_fn(y\_true, y\_pred, \*\*self.\_fn\_kwargs)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/losses.py\", line 1679, in categorical\_crossentropy\n>>> return backend.categorical\_crossentropy(\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/backend.py\", line 4889, in categorical\_crossentropy\n>>> return nn.softmax\_cross\_entropy\_with\_logits\_v2(\n>>> ",

"output\_type": "error",

"traceback": [

"\u001b[0;31m---------------------------------------------------------------------------\u001b[0m",

"\u001b[0;31mInvalidArgumentError\u001b[0m Traceback (most recent call last)",

"\u001b[0;32m/tmp/wsuser/ipykernel\_164/1479672656.py\u001b[0m in \u001b[0;36m<module>\u001b[0;34m\u001b[0m\n\u001b[1;32m 1\u001b[0m \u001b[0;31m#model.fit\_generator(x\_train,steps\_per\_epoch=len(x\_train),validation\_data=x\_test,validation\_steps=len(x\_test),epochs=10)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 2\u001b[0m \u001b[0;31m# Fitting the Model Generator\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m----> 3\u001b[0;31m \u001b[0mmodel\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mfit\_generator\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mx\_train\u001b[0m\u001b[0;34m,\u001b[0m\u001b[0msteps\_per\_epoch\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0;36m630\u001b[0m\u001b[0;34m,\u001b[0m\u001b[0mepochs\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0;36m1\u001b[0m\u001b[0;34m,\u001b[0m\u001b[0mvalidation\_data\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mx\_test\u001b[0m\u001b[0;34m,\u001b[0m\u001b[0mvalidation\_steps\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0;36m90\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m 4\u001b[0m \u001b[0;31m#model.fit(x\_train, epochs=100, verbose=1)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;32m/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py\u001b[0m in \u001b[0;36mfit\_generator\u001b[0;34m(self, generator, steps\_per\_epoch, epochs, verbose, callbacks, validation\_data, validation\_steps, validation\_freq, class\_weight, max\_queue\_size, workers, use\_multiprocessing, shuffle, initial\_epoch)\u001b[0m\n\u001b[1;32m 1964\u001b[0m \u001b[0;34m'will be removed in a future version. '\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 1965\u001b[0m 'Please use `Model.fit`, which supports generators.')\n\u001b[0;32m-> 1966\u001b[0;31m return self.fit(\n\u001b[0m\u001b[1;32m 1967\u001b[0m \u001b[0mgenerator\u001b[0m\u001b[0;34m,\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 1968\u001b[0m \u001b[0msteps\_per\_epoch\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0msteps\_per\_epoch\u001b[0m\u001b[0;34m,\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;32m/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py\u001b[0m in \u001b[0;36mfit\u001b[0;34m(self, x, y, batch\_size, epochs, verbose, callbacks, validation\_split, validation\_data, shuffle, class\_weight, sample\_weight, initial\_epoch, steps\_per\_epoch, validation\_steps, validation\_batch\_size, validation\_freq, max\_queue\_size, workers, use\_multiprocessing)\u001b[0m\n\u001b[1;32m 1187\u001b[0m \_r=1):\n\u001b[1;32m 1188\u001b[0m \u001b[0mcallbacks\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mon\_train\_batch\_begin\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mstep\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m-> 1189\u001b[0;31m \u001b[0mtmp\_logs\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0mself\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mtrain\_function\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0miterator\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m 1190\u001b[0m \u001b[0;32mif\u001b[0m \u001b[0mdata\_handler\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mshould\_sync\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 1191\u001b[0m \u001b[0mcontext\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0masync\_wait\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;32m/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/util/traceback\_utils.py\u001b[0m in \u001b[0;36merror\_handler\u001b[0;34m(\*args, \*\*kwargs)\u001b[0m\n\u001b[1;32m 151\u001b[0m \u001b[0;32mexcept\u001b[0m \u001b[0mException\u001b[0m \u001b[0;32mas\u001b[0m \u001b[0me\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 152\u001b[0m \u001b[0mfiltered\_tb\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0m\_process\_traceback\_frames\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0me\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0m\_\_traceback\_\_\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m--> 153\u001b[0;31m \u001b[0;32mraise\u001b[0m \u001b[0me\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mwith\_traceback\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mfiltered\_tb\u001b[0m\u001b[0;34m)\u001b[0m \u001b[0;32mfrom\u001b[0m \u001b[0;32mNone\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m 154\u001b[0m \u001b[0;32mfinally\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 155\u001b[0m \u001b[0;32mdel\u001b[0m \u001b[0mfiltered\_tb\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;32m/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/eager/execute.py\u001b[0m in \u001b[0;36mquick\_execute\u001b[0;34m(op\_name, num\_outputs, inputs, attrs, ctx, name)\u001b[0m\n\u001b[1;32m 56\u001b[0m \u001b[0;32mtry\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 57\u001b[0m \u001b[0mctx\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mensure\_initialized\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m---> 58\u001b[0;31m tensors = pywrap\_tfe.TFE\_Py\_Execute(ctx.\_handle, device\_name, op\_name,\n\u001b[0m\u001b[1;32m 59\u001b[0m inputs, attrs, num\_outputs)\n\u001b[1;32m 60\u001b[0m \u001b[0;32mexcept\u001b[0m \u001b[0mcore\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0m\_NotOkStatusException\u001b[0m \u001b[0;32mas\u001b[0m \u001b[0me\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;31mInvalidArgumentError\u001b[0m: logits and labels must be broadcastable: logits\_size=[25,5] labels\_size=[25,9]\n\t [[node categorical\_crossentropy/softmax\_cross\_entropy\_with\_logits\n (defined at /opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/backend.py:4889)\n]] [Op:\_\_inference\_train\_function\_2383]\n\nErrors may have originated from an input operation.\nInput Source operations connected to node categorical\_crossentropy/softmax\_cross\_entropy\_with\_logits:\nIn[0] categorical\_crossentropy/softmax\_cross\_entropy\_with\_logits/Reshape:\t\nIn[1] categorical\_crossentropy/softmax\_cross\_entropy\_with\_logits/Reshape\_1:\n\nOperation defined at: (most recent call last)\n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/runpy.py\", line 197, in \_run\_module\_as\_main\n>>> return \_run\_code(code, main\_globals, None,\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/runpy.py\", line 87, in \_run\_code\n>>> exec(code, run\_globals)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/\_\_main\_\_.py\", line 3, in <module>\n>>> app.launch\_new\_instance()\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/traitlets/config/application.py\", line 846, in launch\_instance\n>>> app.start()\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/kernelapp.py\", line 677, in start\n>>> self.io\_loop.start()\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tornado/platform/asyncio.py\", line 199, in start\n>>> self.asyncio\_loop.run\_forever()\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/asyncio/base\_events.py\", line 601, in run\_forever\n>>> self.\_run\_once()\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/asyncio/base\_events.py\", line 1905, in \_run\_once\n>>> handle.\_run()\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/asyncio/events.py\", line 80, in \_run\n>>> self.\_context.run(self.\_callback, \*self.\_args)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/kernelbase.py\", line 457, in dispatch\_queue\n>>> await self.process\_one()\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/kernelbase.py\", line 446, in process\_one\n>>> await dispatch(\*args)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/kernelbase.py\", line 353, in dispatch\_shell\n>>> await result\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/kernelbase.py\", line 648, in execute\_request\n>>> reply\_content = await reply\_content\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/ipkernel.py\", line 353, in do\_execute\n>>> res = shell.run\_cell(code, store\_history=store\_history, silent=silent)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ipykernel/zmqshell.py\", line 533, in run\_cell\n>>> return super(ZMQInteractiveShell, self).run\_cell(\*args, \*\*kwargs)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/IPython/core/interactiveshell.py\", line 2914, in run\_cell\n>>> result = self.\_run\_cell(\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/IPython/core/interactiveshell.py\", line 2960, in \_run\_cell\n>>> return runner(coro)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/IPython/core/async\_helpers.py\", line 78, in \_pseudo\_sync\_runner\n>>> coro.send(None)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/IPython/core/interactiveshell.py\", line 3185, in run\_cell\_async\n>>> has\_raised = await self.run\_ast\_nodes(code\_ast.body, cell\_name,\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/IPython/core/interactiveshell.py\", line 3377, in run\_ast\_nodes\n>>> if (await self.run\_code(code, result, async\_=asy)):\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/IPython/core/interactiveshell.py\", line 3457, in run\_code\n>>> exec(code\_obj, self.user\_global\_ns, self.user\_ns)\n>>> \n>>> File \"/tmp/wsuser/ipykernel\_164/3808038373.py\", line 3, in <module>\n>>> model.fit\_generator(x\_train,steps\_per\_epoch=630,epochs=1,validation\_data=x\_test,validation\_steps=90)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py\", line 1966, in fit\_generator\n>>> return self.fit(\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py\", line 1189, in fit\n>>> tmp\_logs = self.train\_function(iterator)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py\", line 859, in train\_function\n>>> return step\_function(self, iterator)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py\", line 849, in step\_function\n>>> outputs = model.distribute\_strategy.run(run\_step, args=(data,))\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py\", line 842, in run\_step\n>>> outputs = model.train\_step(data)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py\", line 800, in train\_step\n>>> loss = self.compiled\_loss(\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/compile\_utils.py\", line 204, in \_\_call\_\_\n>>> loss\_value = loss\_obj(y\_t, y\_p, sample\_weight=sw)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/losses.py\", line 155, in \_\_call\_\_\n>>> losses = call\_fn(y\_true, y\_pred)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/losses.py\", line 259, in call\n>>> return ag\_fn(y\_true, y\_pred, \*\*self.\_fn\_kwargs)\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/losses.py\", line 1679, in categorical\_crossentropy\n>>> return backend.categorical\_crossentropy(\n>>> \n>>> File \"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/backend.py\", line 4889, in categorical\_crossentropy\n>>> return nn.softmax\_cross\_entropy\_with\_logits\_v2(\n>>> "

]

}

],

"source": [

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

"# Fitting the Model Generator\n",

"model.fit\_generator(x\_train,steps\_per\_epoch=630,epochs=1,validation\_data=x\_test,validation\_steps=90)\n",

"#model.fit(x\_train, epochs=100, verbose=1)"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "Av9oIJI2P-lD"

},

"source": [

"# 12.]SAVING THE MODEL"

]

},

{

"cell\_type": "code",

"execution\_count": 126,

"metadata": {

"id": "XEvO9YPmP08B"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"\u001b[0m\u001b[01;34mDataset\u001b[0m/ \u001b[01;34mtest\_set\u001b[0m/ \u001b[01;34mtraining\_set\u001b[0m/\r\n"

]

}

],

"source": [

"ls"

]

},

{

"cell\_type": "code",

"execution\_count": 127,

"metadata": {

"id": "T1BOK\_jHQIIF"

},

"outputs": [

{

"data": {

"text/plain": [

"'/home/wsuser/work/Dataset'"

]

},

"execution\_count": 127,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"pwd"

]

},

{

"cell\_type": "code",

"execution\_count": 128,

"metadata": {

"id": "OEcGdexzQL51"

},

"outputs": [],

"source": [

"model.save('Dataset.h5')"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 13.]CONVERTING ZIP FILE TO TAR FILE FOR LOCAL USE."

]

},

{

"cell\_type": "code",

"execution\_count": 134,

"metadata": {

"scrolled": true

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Dataset.h5\r\n"

]

}

],

"source": [

"#converting the model to tar\n",

"!tar -zcvf image.Classification.model\_new.tgz Dataset.h5"

]

},

{

"cell\_type": "code",

"execution\_count": 135,

"metadata": {

"scrolled": true

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

"Dataset.h5\r\n",

"image.Classification.model\_new.tgz\r\n",

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

"\u001b[01;34mtraining\_set\u001b[0m/\r\n"

]

}

],

"source": [

"ls -1"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 14.]INSTALLING WATSON MACHINE LEARNING CLIENT SOFTWARE"

]

},

{

"cell\_type": "code",

"execution\_count": 137,

"metadata": {

"scrolled": false

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Collecting watson\_machine\_learning\_client\n",

" Downloading watson\_machine\_learning\_client-1.0.391-py3-none-any.whl (538 kB)\n",

"\u001b[K |████████████████████████████████| 538 kB 23.9 MB/s eta 0:00:01\n",

"\u001b[?25hRequirement 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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",

"Installing collected packages: watson-machine-learning-client\n",

"Successfully installed watson-machine-learning-client-1.0.391\n"

]

}

],

"source": [

"#installing the machine learning repository\n",

"!pip install watson\_machine\_learning\_client --upgrade"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 15.]IMPORTING APICLIENT FOR DEPLOYING."

]

},

{

"cell\_type": "code",

"execution\_count": 138,

"metadata": {},

"outputs": [],

"source": [

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

"url\_credentials = {\n",

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

" \"apikey\": \"sqLVTXSP3nnAKfzJ1rKRKCpNzS\_XZ8\_HXa9FRwV7BvOP\"\n",

"}\n",

"client = APIClient(url\_credentials)\n"

]

},

{

"cell\_type": "code",

"execution\_count": 139,

"metadata": {},

"outputs": [],

"source": [

"client = APIClient(url\_credentials)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 16.]CREATING API\_CLIENT SPACE ID. "

]

},

{

"cell\_type": "code",

"execution\_count": 140,

"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": 143,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"space UID = d90f421e-9169-47e7-a58c-0e7bb0e65685\n"

]

}

],

"source": [

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

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

]

},

{

"cell\_type": "code",

"execution\_count": 145,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'SUCCESS'"

]

},

"execution\_count": 145,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 147,

"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": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": [

"software\_spec\_uid = client.software\_specifications.get\_uid\_by\_name(\"tensorflow\")\n",

"software\_spec\_uid"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 17.]STORING THE MODEL\_ID FOR DATASET.H5"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": [

"#store the model\n",

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

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

" client.repository.ModelMetaNames.TYPE:\"keras\_2.2.4\",\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": null,

"metadata": {},

"outputs": [],

"source": [

"model\_id"

]

},

{

"cell\_type": "code",

"execution\_count": 171,

"metadata": {},

"outputs": [],

"source": [

"model.save('Dataset.h5')"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 18.]DOWNLOADING THE TAR FILE ON CLIENT REPOSITORY"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": [

"client.repository.download(model\_id, 'my\_model.tar.gz')"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "9T68YyFGQvZH"

},

"source": [

"# 19.]TEST THE MODEL"

]

},

{

"cell\_type": "code",

"execution\_count": 186,

"metadata": {

"id": "\_HAKckWyQu5C"

},

"outputs": [],

"source": [

"import numpy as np\n",

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

"from keras.preprocessing import image"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 20.]LOADING THE DATASET"

]

},

{

"cell\_type": "code",

"execution\_count": 187,

"metadata": {

"id": "69LLKetXRCPW"

},

"outputs": [],

"source": [

"#Load the model\n",

"model=load\_model('Dataset.h5')"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 21.]ADDING STREAMING\_BODY FOR TEST IMAGE."

]

},

{

"cell\_type": "code",

"execution\_count": 188,

"metadata": {},

"outputs": [],

"source": [

"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='aqprHZFuH38ECUn869hHk4qyvS\_iKJfrZAWUJJQ-mQKx',\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 = 'realtimecommunicationforspecially-donotdelete-pr-rfqndcvwgch6fu'\n",

"object\_key = '1.png'\n",

"\n",

"streaming\_body\_5 = 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",

"\n"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# 22.]TESTING ON SEVERAL TESTING IMAGES"

]

},

{

"cell\_type": "code",

"execution\_count": 189,

"metadata": {},

"outputs": [

{

"ename": "TypeError",

"evalue": "expected str, bytes or os.PathLike object, not StreamingBody",

"output\_type": "error",

"traceback": [

"\u001b[0;31m---------------------------------------------------------------------------\u001b[0m",

"\u001b[0;31mTypeError\u001b[0m Traceback (most recent call last)",

"\u001b[0;32m/tmp/wsuser/ipykernel\_164/365554034.py\u001b[0m in \u001b[0;36m<module>\u001b[0;34m\u001b[0m\n\u001b[0;32m----> 1\u001b[0;31m \u001b[0mimg\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0mimage\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mload\_img\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mstreaming\_body\_5\u001b[0m\u001b[0;34m,\u001b[0m\u001b[0mtarget\_size\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;36m64\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0;36m64\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m 2\u001b[0m \u001b[0;31m#img=image.load\_img(\"/home/wsuser/work/1\",target\_size=(64,64))\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;32m/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/keras/preprocessing/image.py\u001b[0m in \u001b[0;36mload\_img\u001b[0;34m(path, grayscale, color\_mode, target\_size, interpolation)\u001b[0m\n\u001b[1;32m 311\u001b[0m \u001b[0;31m`\u001b[0m\u001b[0;31m`\u001b[0m\u001b[0;31m`\u001b[0m\u001b[0mpython\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 312\u001b[0m \u001b[0;34m(\u001b[0m\u001b[0mx\_train\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0my\_train\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0;34m(\u001b[0m\u001b[0mx\_test\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0my\_test\u001b[0m\u001b[0;34m)\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0mcifar10\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mload\_data\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m--> 313\u001b[0;31m \u001b[0my\_train\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0mnp\_utils\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mto\_categorical\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0my\_train\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mnum\_classes\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m 314\u001b[0m \u001b[0my\_test\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0mnp\_utils\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mto\_categorical\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0my\_test\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mnum\_classes\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 315\u001b[0m \u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;32m/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/keras\_preprocessing/image/utils.py\u001b[0m in \u001b[0;36mload\_img\u001b[0;34m(path, grayscale, color\_mode, target\_size, interpolation)\u001b[0m\n\u001b[1;32m 111\u001b[0m raise ImportError('Could not import PIL.Image. '\n\u001b[1;32m 112\u001b[0m 'The use of `load\_img` requires PIL.')\n\u001b[0;32m--> 113\u001b[0;31m \u001b[0;32mwith\u001b[0m \u001b[0mopen\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mpath\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0;34m'rb'\u001b[0m\u001b[0;34m)\u001b[0m \u001b[0;32mas\u001b[0m \u001b[0mf\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m 114\u001b[0m \u001b[0mimg\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0mpil\_image\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mopen\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mio\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mBytesIO\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mf\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mread\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 115\u001b[0m \u001b[0;32mif\u001b[0m \u001b[0mcolor\_mode\u001b[0m \u001b[0;34m==\u001b[0m \u001b[0;34m'grayscale'\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;31mTypeError\u001b[0m: expected str, bytes or os.PathLike object, not StreamingBody"

]

}

],

"source": [

"img = image.load\_img(streaming\_body\_5,target\_size=(64, 64))\n",

"#img=image.load\_img(\"/home/wsuser/work/1\",target\_size=(64,64))"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "IlQxA5dJRB3Q"

},

"outputs": [],

"source": [

"ls"

]

},

{

"cell\_type": "code",

"execution\_count": 181,

"metadata": {

"id": "kAWQrtBwRBym"

},

"outputs": [

{

"ename": "FileNotFoundError",

"evalue": "[Errno 2] No such file or directory: '/content/drive/MyDrive/IBM\_PROJECT/Dataset/training\_set/A/1.png'",

"output\_type": "error",

"traceback": [

"\u001b[0;31m---------------------------------------------------------------------------\u001b[0m",

"\u001b[0;31mFileNotFoundError\u001b[0m Traceback (most recent call last)",

"\u001b[0;32m/tmp/wsuser/ipykernel\_164/1035932264.py\u001b[0m in \u001b[0;36m<module>\u001b[0;34m\u001b[0m\n\u001b[0;32m----> 1\u001b[0;31m \u001b[0mimg\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mimage\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mload\_img\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34mr\"/content/drive/MyDrive/IBM\_PROJECT/Dataset/training\_set/A/1.png\"\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m",

"\u001b[0;32m/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/preprocessing/image.py\u001b[0m in \u001b[0;36mload\_img\u001b[0;34m(path, grayscale, color\_mode, target\_size, interpolation)\u001b[0m\n\u001b[1;32m 311\u001b[0m \u001b[0mValueError\u001b[0m\u001b[0;34m:\u001b[0m \u001b[0;32mif\u001b[0m \u001b[0minterpolation\u001b[0m \u001b[0mmethod\u001b[0m \u001b[0;32mis\u001b[0m \u001b[0;32mnot\u001b[0m \u001b[0msupported\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 312\u001b[0m \"\"\"\n\u001b[0;32m--> 313\u001b[0;31m return image.load\_img(path, grayscale=grayscale, color\_mode=color\_mode,\n\u001b[0m\u001b[1;32m 314\u001b[0m target\_size=target\_size, interpolation=interpolation)\n\u001b[1;32m 315\u001b[0m \u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;32m/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/keras\_preprocessing/image/utils.py\u001b[0m in \u001b[0;36mload\_img\u001b[0;34m(path, grayscale, color\_mode, target\_size, interpolation)\u001b[0m\n\u001b[1;32m 111\u001b[0m raise ImportError('Could not import PIL.Image. '\n\u001b[1;32m 112\u001b[0m 'The use of `load\_img` requires PIL.')\n\u001b[0;32m--> 113\u001b[0;31m \u001b[0;32mwith\u001b[0m \u001b[0mopen\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mpath\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0;34m'rb'\u001b[0m\u001b[0;34m)\u001b[0m \u001b[0;32mas\u001b[0m \u001b[0mf\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m 114\u001b[0m \u001b[0mimg\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0mpil\_image\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mopen\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mio\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mBytesIO\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mf\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mread\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 115\u001b[0m \u001b[0;32mif\u001b[0m \u001b[0mcolor\_mode\u001b[0m \u001b[0;34m==\u001b[0m \u001b[0;34m'grayscale'\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;31mFileNotFoundError\u001b[0m: [Errno 2] No such file or directory: '/content/drive/MyDrive/IBM\_PROJECT/Dataset/training\_set/A/1.png'"

]

}

],

"source": [

"img=image.load\_img(r\"/home/wsuser/work/Dataset/test\_set/A/1.png\")\n"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "LeiulNjOSwmD"

},

"outputs": [],

"source": [

"img"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "BY0zVMqnSw--"

},

"outputs": [],

"source": [

"img1=image.load\_ing(r\"/home/wsuser/work/Dataset/test\_set/C/1.png\")"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "uUXt\_ZQWRBtm"

},

"outputs": [],

"source": [

"img1"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "BuWcxXfKRBie"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "l10rMIDJRBYA"

},

"outputs": [],

"source": [

"x"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "-anXa0TFRA5O"

},

"outputs": [],

"source": [

"x1=np.expand\_dims(x,axis=1)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "naRcte\_mXUh6"

},

"outputs": [],

"source": [

"x1"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "KiRWlqpqXVLZ"

},

"outputs": [],

"source": [

"y=np.argmax(model.predoct(x),axis=1)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "btSYV89FXVqy"

},

"outputs": [],

"source": [

"y"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "M3U9UhQFXgf1"

},

"outputs": [],

"source": [

"x\_train.class\_indices"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "KxO8yCNDXiAN"

},

"outputs": [],

"source": [

"index=['A','B','C','D','E','F','G','H','I']"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "1FIK2U5oXhvO"

},

"outputs": [],

"source": [

"index[y[0]]"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "NMZj0zboXhKu"

},

"outputs": [],

"source": [

"img=image.load\_img(r\"/home/wsuser/work/Dataset/test\_set/A/90.png\",target\_size=(64,64))\n",

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

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

"y=fnp.argmax(model.predict(x),axis=1)\n",

"index=['A','B','C','D','E','F','G','H','I']\n",

"index[y[0]]]"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "EvUOmI7eYRn8"

},

"outputs": [],

"source": [

"img=image.load\_img( \"/home/wsuser/work/Dataset/test\_set/D/1.png\",target\_size=(64,64))\n",

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

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

"y=np.argmax(model.predict(x)\n",

"index=['A','B','C','D','E','F','G','H','I']\n",

"index[y[0]]"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "9HRrjTYCYRTC"

},

"outputs": [],

"source": [

"img=image.load\_img(r\"/content/drive/MyDrive/IBM\_PROJECT/Dataset/test\_set/G/1.png\",target\_size=(64,64))\n",

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

"x=np.expand\_dims(x,axisme)\n",

"y=np.argmax(model.predict(x), axis=1)\n",

"index=['A','B','C','D','E','F','G','H','I']\n",

"index[y[0]]"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "HR3o6fisYQOv"

},

"outputs": [],

"source": [

"img=image.load\_img(r\"/content/drive/MyDrive/IBM\_PROJECT/Dataset/test\_set/D/1.png\",target\_size=(64,64))\n",

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

"x=np.expand\_dims(x,axisme)\n",

"y=np.argmax(model.predict(x), axis=1)\n",

"index=['A','B','C','D','E','F','G','H','I']\n",

"index[y[0]]"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "DG\_fGnHhZXJx"

},

"outputs": [],

"source": [

"!tar -zcvf Dataset-classification-model.tgz specially.h5"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "2sB\_7ubnZW7p"

},

"outputs": [],

"source": [

"import tensorflow as tf\n",

"tf .\_\_ \_version\_"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "td9zCLyDb\_mJ"

},

"outputs": [],

"source": [

"!pip install keras == 2.2.4"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "bUx7C1jKcRDk"

},

"source": [

"# 23.]IBM DEPLOYMENT"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "0nFP\_MzMcVlE"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "I5FaOmgGca5s"

},

"outputs": [],

"source": [

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

"wml\_credentials={\n",

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

"\"apikey\":\"x91CJTUTrrIfLvrXsKf8yLyI1KHb3JV0Y7Qrwy1zilb2\"\n",

"}\n",

"client=APIClient(wml\_credentials)"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "LWiFTStydPNe"

},

"source": [

"# CLIENT"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "KRfT3nwkcjqB"

},

"outputs": [],

"source": [

"def guid\_space\_name(client,animal\_deploy):\n",

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

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "ToZHLNDicrmf"

},

"outputs": [],

"source": [

"space\_uid-guid\_space\_name(client,'animal\_deploy\")\n",

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "viITQa6edWZv"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "gk83aFHUdYcA"

},

"outputs": [],

"source": [

"client,software specifications.list(200)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "8\_AJilmkdnFS"

},

"outputs": [],

"source": [

"software\_space\_uid=client.software\_specifications.get\_uid\_by\_name('tensorflow\_rt22.1-py3.9¹)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "xeAmvLnydm6h"

},

"outputs": [],

"source": [

"software\_space\_uid"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "tJFzLvBodmrl"

},

"outputs": [],

"source": [

"model\_details=client.repository.store\_model(model='Dataset.tgz',meta\_props={\n",

"client.repository.ModelMetaNames.NAME: \"CNN Model Building\",\n",

"client.repository.ModelMetaNames.TYPE: 'tensorflow\_2.7',\n",

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

"})"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "ELRBCgMMdvkp"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "x1S3mF-UeqS1"

},

"outputs": [],

"source": [

"model\_id"

]

}

],

"metadata": {

"colab": {

"provenance": []

},

"kernelspec": {

"display\_name": "Python 3.10.0 64-bit",

"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": "3.10.0"

},

"vscode": {

"interpreter": {

"hash": "26de051ba29f2982a8de78e945f0abaf191376122a1563185a90213a26c5da77"

}

}

},

"nbformat": 4,

"nbformat\_minor": 1

}