**Download-Model.ipynb**

{

"cells": [

{

"cell\_type": "code",

"execution\_count": 1,

"metadata": {

"colab": {

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

},

"id": "YOAcNmuPQaGp",

"outputId": "946ba046-2f3e-4342-d528-1c8e6f6d4c0b"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/\n",

"Requirement already satisfied: ibm\_watson\_machine\_learning in /usr/local/lib/python3.7/dist-packages (1.0.257)\n",

"Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (0.8.10)\n",

"Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (2.23.0)\n",

"Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (4.13.0)\n",

"Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (2022.9.24)\n",

"Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (1.3.5)\n",

"Requirement already satisfied: lomond in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (0.3.3)\n",

"Requirement already satisfied: ibm-cos-sdk==2.7.\* in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (2.7.0)\n",

"Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (1.24.3)\n",

"Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from ibm\_watson\_machine\_learning) (21.3)\n",

"Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /usr/local/lib/python3.7/dist-packages (from ibm-cos-sdk==2.7.\*->ibm\_watson\_machine\_learning) (0.10.0)\n",

"Requirement already satisfied: ibm-cos-sdk-s3transfer==2.7.0 in /usr/local/lib/python3.7/dist-packages (from ibm-cos-sdk==2.7.\*->ibm\_watson\_machine\_learning) (2.7.0)\n",

"Requirement already satisfied: ibm-cos-sdk-core==2.7.0 in /usr/local/lib/python3.7/dist-packages (from ibm-cos-sdk==2.7.\*->ibm\_watson\_machine\_learning) (2.7.0)\n",

"Requirement already satisfied: docutils<0.16,>=0.10 in /usr/local/lib/python3.7/dist-packages (from ibm-cos-sdk-core==2.7.0->ibm-cos-sdk==2.7.\*->ibm\_watson\_machine\_learning) (0.15.2)\n",

"Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.7/dist-packages (from ibm-cos-sdk-core==2.7.0->ibm-cos-sdk==2.7.\*->ibm\_watson\_machine\_learning) (2.8.2)\n",

"Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (from pandas<1.5.0,>=0.24.2->ibm\_watson\_machine\_learning) (1.21.6)\n",

"Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas<1.5.0,>=0.24.2->ibm\_watson\_machine\_learning) (2022.6)\n",

"Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.7.0->ibm-cos-sdk==2.7.\*->ibm\_watson\_machine\_learning) (1.15.0)\n",

"Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->ibm\_watson\_machine\_learning) (3.0.4)\n",

"Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->ibm\_watson\_machine\_learning) (2.10)\n",

"Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->ibm\_watson\_machine\_learning) (4.1.1)\n",

"Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->ibm\_watson\_machine\_learning) (3.10.0)\n",

"Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->ibm\_watson\_machine\_learning) (3.0.9)\n"

]

}

],

"source": [

"!pip install ibm\_watson\_machine\_learning"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"metadata": {

"id": "0GnEnen7QkGa"

},

"outputs": [],

"source": [

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

"\n",

"wml\_credentials={\n",

" \n",

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

" \"apikey\":\"xhYfr2sw7BoEM8unaHSrXivomwlh4Cqyk0F7MzxcloSa\"\n",

"}"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"metadata": {

"colab": {

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

},

"id": "\_h\_nRm3RQ0ri",

"outputId": "3353e56e-89e5-401c-a00f-028c8b4d726f"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Python 3.7 and 3.8 frameworks are deprecated and will be removed in a future release. Use Python 3.9 framework instead.\n"

]

}

],

"source": [

"client = APIClient(wml\_credentials)"

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"metadata": {

"colab": {

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

},

"id": "0Yjf02PoRBIL",

"outputId": "3536b034-40ee-45ea-e230-9cbc6a4f8671"

},

"outputs": [

{

"data": {

"text/plain": [

"<ibm\_watson\_machine\_learning.client.APIClient at 0x7fafa7d62790>"

]

},

"execution\_count": 4,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"client"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"metadata": {

"id": "X\_00r5WfRLPx"

},

"outputs": [],

"source": [

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

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

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

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"metadata": {

"colab": {

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

},

"id": "ECgng7TbRMOD",

"outputId": "1839d297-8271-456e-8bcb-a5a4b4ef67c1"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Space UID = 9930b49f-907b-4a1f-b5f0-e146c7d081b1\n"

]

}

],

"source": [

"space\_uid=guid\_space\_name(client,'gesture-recognition')\n",

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

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"metadata": {

"colab": {

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

"height": 36

},

"id": "MVsOojLiRoTS",

"outputId": "68894d45-e055-4f62-f84c-86e294a060f2"

},

"outputs": [

{

"data": {

"application/vnd.google.colaboratory.intrinsic+json": {

"type": "string"

},

"text/plain": [

"'SUCCESS'"

]

},

"execution\_count": 7,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"metadata": {

"colab": {

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

},

"id": "F4w3oSJkSk-M",

"outputId": "7c4099bd-cb5d-4599-f347-e4988b19bbc3"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Mounted at /content/drive\n",

"Change successful.\n"

]

}

],

"source": [

"import os\n",

"from google.colab import drive\n",

"drive.mount('/content/drive',force\_remount=True)\n",

"os.chdir('/content/drive/My Drive')\n",

"print(\"Change successful.\")"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"metadata": {

"colab": {

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

"height": 53

},

"id": "hB\_MZ3F3RppM",

"outputId": "6bd29aac-f71a-4a53-8c93-e7815aa83f4f"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Successfully saved model content to file: 'gesture-model.tar.gz'\n"

]

},

{

"data": {

"application/vnd.google.colaboratory.intrinsic+json": {

"type": "string"

},

"text/plain": [

"'/content/drive/MyDrive/gesture-model.tar.gz'"

]

},

"execution\_count": 9,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"client.repository.download(\"b551a542-6d4f-43d1-a1df-815fe17e7e8c\",\"gesture-model.tar.gz\")"

]

}

],

"metadata": {

"colab": {

"provenance": []

},

"kernelspec": {

"display\_name": "Python 3.9.0 ('venv': venv)",

"language": "python",

"name": "python3"

},

"language\_info": {

"name": "python",

"version": "3.9.0"

},

"vscode": {

"interpreter": {

"hash": "5f31d7f7db7bde69e73362a1cdfafb650c813f8be944ec1b58fbeb2af0618ef9"

}

}

},

"nbformat": 4,

"nbformat\_minor": 0

}

**Hand-Gesture-Classification-v3.ipynb**

{

"cells": [

{

"cell\_type": "code",

"execution\_count": 1,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'/home/wsuser/work'"

]

},

"execution\_count": 1,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"pwd"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'2.7.2'"

]

},

"execution\_count": 2,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"import tensorflow as tf\n",

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

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'2.7.0'"

]

},

"execution\_count": 3,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"import keras\n",

"keras.\_\_version\_\_"

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"metadata": {},

"outputs": [],

"source": [

"import os\n",

"import numpy as np\n",

"import pandas as pd"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"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='NhA5XAu3PupIxjK1ipy7QqayZVZ4XXNlHfuJv-CTURwF',\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 = 'gesturerecognition-donotdelete-pr-bqiugumhf2mrwn'\n",

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

"\n",

"streaming\_body\_1 = 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": 6,

"metadata": {},

"outputs": [],

"source": [

"from io import BytesIO\n",

"import zipfile\n",

"unzip=zipfile.ZipFile(BytesIO(streaming\_body\_1.read()),'r')\n",

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

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

" unzip.extract(path)"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'/home/wsuser/work'"

]

},

"execution\_count": 7,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"pwd"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"metadata": {},

"outputs": [],

"source": [

"#Checks if the dataset got unzipped properly\n",

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

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"metadata": {

"id": "YW8TYMnT8T7E"

},

"outputs": [],

"source": [

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

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

"from tensorflow.keras import layers\n",

"from tensorflow.keras.layers import Dense, Flatten\n",

"from tensorflow.keras.layers import Conv2D, MaxPooling2D\n",

"from keras.preprocessing.image import ImageDataGenerator"

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"metadata": {

"id": "7OOBj4myLq2t"

},

"outputs": [],

"source": [

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

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

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"metadata": {

"colab": {

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

},

"id": "pwp5XVzSMnMJ",

"outputId": "14caf2a4-1f78-445e-8e0a-aebb5c4d0cff"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Found 594 images belonging to 6 classes.\n",

"Found 30 images belonging to 6 classes.\n"

]

}

],

"source": [

"x\_train = train\_datagen.flow\_from\_directory('/home/wsuser/work/Dataset/train/', target\_size=(64, 64), batch\_size=5, color\_mode='grayscale', class\_mode='categorical')\n",

"x\_test = test\_datagen.flow\_from\_directory('/home/wsuser/work/Dataset/test/', target\_size=(64, 64), batch\_size=5, color\_mode='grayscale', class\_mode='categorical')"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"{'0': 0, '1': 1, '2': 2, '3': 3, '4': 4, '5': 5}"

]

},

"execution\_count": 12,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x\_train.class\_indices"

]

},

{

"cell\_type": "code",

"execution\_count": 13,

"metadata": {

"id": "Sm\_wLYdONXGy"

},

"outputs": [],

"source": [

"classifier = Sequential()"

]

},

{

"cell\_type": "code",

"execution\_count": 14,

"metadata": {

"id": "3w\_s3wNXNiVy"

},

"outputs": [],

"source": [

"classifier.add(Conv2D(32, (3, 3), input\_shape=(64, 64, 1) ,activation='relu'))\n",

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

"classifier.add(Conv2D(32, (3, 3), activation='relu'))\n",

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

"classifier.add(Flatten())"

]

},

{

"cell\_type": "code",

"execution\_count": 15,

"metadata": {

"id": "Ri0SYCWYNrzH"

},

"outputs": [],

"source": [

"classifier.add(Dense(units=128, activation='relu'))\n",

"classifier.add(Dense(units=6, activation='softmax'))"

]

},

{

"cell\_type": "code",

"execution\_count": 16,

"metadata": {

"colab": {

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

},

"id": "vg5aGy1mNz9Y",

"outputId": "9a45162e-f58c-4686-ab90-776041162f03"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

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

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

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

" conv2d (Conv2D) (None, 62, 62, 32) 320 \n",

" \n",

" max\_pooling2d (MaxPooling2D (None, 31, 31, 32) 0 \n",

" ) \n",

" \n",

" conv2d\_1 (Conv2D) (None, 29, 29, 32) 9248 \n",

" \n",

" max\_pooling2d\_1 (MaxPooling (None, 14, 14, 32) 0 \n",

" 2D) \n",

" \n",

" flatten (Flatten) (None, 6272) 0 \n",

" \n",

" dense (Dense) (None, 128) 802944 \n",

" \n",

" dense\_1 (Dense) (None, 6) 774 \n",

" \n",

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

"Total params: 813,286\n",

"Trainable params: 813,286\n",

"Non-trainable params: 0\n",

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

]

}

],

"source": [

"classifier.summary()"

]

},

{

"cell\_type": "code",

"execution\_count": 17,

"metadata": {

"id": "32uuVOGTN7lY"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 18,

"metadata": {

"colab": {

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

},

"id": "f\_gh1voON9wD",

"outputId": "276fd9cb-6208-411e-af45-ce2d4188fae9"

},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"/tmp/wsuser/ipykernel\_4012/2617134232.py:1: UserWarning: `Model.fit\_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.\n",

" classifier.fit\_generator(\n"

]

},

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Epoch 1/20\n",

"119/119 [==============================] - 5s 42ms/step - loss: 1.4090 - accuracy: 0.4259 - val\_loss: 0.6351 - val\_accuracy: 0.8000\n",

"Epoch 2/20\n",

"119/119 [==============================] - 5s 38ms/step - loss: 0.7041 - accuracy: 0.7003 - val\_loss: 0.5916 - val\_accuracy: 0.8000\n",

"Epoch 3/20\n",

"119/119 [==============================] - 5s 41ms/step - loss: 0.5020 - accuracy: 0.7963 - val\_loss: 0.6768 - val\_accuracy: 0.7333\n",

"Epoch 4/20\n",

"119/119 [==============================] - 5s 41ms/step - loss: 0.3939 - accuracy: 0.8519 - val\_loss: 0.4509 - val\_accuracy: 0.8667\n",

"Epoch 5/20\n",

"119/119 [==============================] - 5s 40ms/step - loss: 0.3088 - accuracy: 0.8889 - val\_loss: 0.3790 - val\_accuracy: 0.8667\n",

"Epoch 6/20\n",

"119/119 [==============================] - 5s 40ms/step - loss: 0.2642 - accuracy: 0.8973 - val\_loss: 0.4686 - val\_accuracy: 0.8667\n",

"Epoch 7/20\n",

"119/119 [==============================] - 5s 40ms/step - loss: 0.1891 - accuracy: 0.9343 - val\_loss: 0.3799 - val\_accuracy: 0.9333\n",

"Epoch 8/20\n",

"119/119 [==============================] - 5s 41ms/step - loss: 0.1654 - accuracy: 0.9360 - val\_loss: 0.6095 - val\_accuracy: 0.8667\n",

"Epoch 9/20\n",

"119/119 [==============================] - 5s 40ms/step - loss: 0.1182 - accuracy: 0.9579 - val\_loss: 0.4162 - val\_accuracy: 0.9333\n",

"Epoch 10/20\n",

"119/119 [==============================] - 5s 40ms/step - loss: 0.1253 - accuracy: 0.9680 - val\_loss: 0.4763 - val\_accuracy: 0.9000\n",

"Epoch 11/20\n",

"119/119 [==============================] - 5s 39ms/step - loss: 0.1078 - accuracy: 0.9646 - val\_loss: 0.5120 - val\_accuracy: 0.9000\n",

"Epoch 12/20\n",

"119/119 [==============================] - 5s 39ms/step - loss: 0.0657 - accuracy: 0.9764 - val\_loss: 0.2290 - val\_accuracy: 0.9667\n",

"Epoch 13/20\n",

"119/119 [==============================] - 5s 41ms/step - loss: 0.1008 - accuracy: 0.9680 - val\_loss: 0.2593 - val\_accuracy: 0.9667\n",

"Epoch 14/20\n",

"119/119 [==============================] - 5s 41ms/step - loss: 0.0969 - accuracy: 0.9663 - val\_loss: 0.2971 - val\_accuracy: 0.9667\n",

"Epoch 15/20\n",

"119/119 [==============================] - 5s 40ms/step - loss: 0.0698 - accuracy: 0.9731 - val\_loss: 0.2917 - val\_accuracy: 0.9667\n",

"Epoch 16/20\n",

"119/119 [==============================] - 5s 40ms/step - loss: 0.0492 - accuracy: 0.9832 - val\_loss: 0.2443 - val\_accuracy: 0.9333\n",

"Epoch 17/20\n",

"119/119 [==============================] - 5s 40ms/step - loss: 0.0212 - accuracy: 0.9949 - val\_loss: 0.2986 - val\_accuracy: 0.9667\n",

"Epoch 18/20\n",

"119/119 [==============================] - 5s 42ms/step - loss: 0.0190 - accuracy: 0.9933 - val\_loss: 0.1804 - val\_accuracy: 0.9333\n",

"Epoch 19/20\n",

"119/119 [==============================] - 5s 40ms/step - loss: 0.0799 - accuracy: 0.9646 - val\_loss: 0.2960 - val\_accuracy: 0.9667\n",

"Epoch 20/20\n",

"119/119 [==============================] - 5s 40ms/step - loss: 0.0567 - accuracy: 0.9848 - val\_loss: 0.2684 - val\_accuracy: 0.9667\n"

]

},

{

"data": {

"text/plain": [

"<keras.callbacks.History at 0x7fb22655a880>"

]

},

"execution\_count": 18,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"classifier.fit\_generator(\n",

" generator=x\_train, steps\_per\_epoch=len(x\_train),\n",

" epochs=20, validation\_data=x\_test, validation\_steps=len(x\_test)\n",

")"

]

},

{

"cell\_type": "code",

"execution\_count": 20,

"metadata": {

"id": "2NIQFVo4OId7"

},

"outputs": [],

"source": [

"classifier.save('gesture.h5')"

]

},

{

"cell\_type": "code",

"execution\_count": 21,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"gesture.h5\r\n"

]

}

],

"source": [

"!tar -zcvf gesture-classifier.tgz gesture.h5"

]

},

{

"cell\_type": "code",

"execution\_count": 22,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"\u001b[0m\u001b[01;34mDataset\u001b[0m/ gesture-classifier.tgz gesture.h5\r\n"

]

}

],

"source": [

"ls"

]

},

{

"cell\_type": "code",

"execution\_count": 23,

"metadata": {

"id": "uN-EJRl7OVda"

},

"outputs": [],

"source": [

"model\_json = classifier.to\_json()\n",

"with open(\"model-bw.json\", \"w\") as f:\n",

" f.write(model\_json)"

]

},

{

"cell\_type": "code",

"execution\_count": 24,

"metadata": {},

"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 24.5 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: 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: 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: 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: 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: 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: 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: 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: 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": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 25,

"metadata": {},

"outputs": [],

"source": [

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

"\n",

"wml\_credentials={\n",

" \n",

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

" \"apikey\":\"xhYfr2sw7BoEM8unaHSrXivomwlh4Cqyk0F7MzxcloSa\"\n",

"}"

]

},

{

"cell\_type": "code",

"execution\_count": 26,

"metadata": {},

"outputs": [],

"source": [

"client = APIClient(wml\_credentials)"

]

},

{

"cell\_type": "code",

"execution\_count": 27,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<ibm\_watson\_machine\_learning.client.APIClient at 0x7fb1eec10760>"

]

},

"execution\_count": 27,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"client"

]

},

{

"cell\_type": "code",

"execution\_count": 28,

"metadata": {},

"outputs": [],

"source": [

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

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

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

]

},

{

"cell\_type": "code",

"execution\_count": 29,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Space UID = 9930b49f-907b-4a1f-b5f0-e146c7d081b1\n"

]

}

],

"source": [

"space\_uid=guid\_space\_name(client,'gesture-recognition')\n",

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

]

},

{

"cell\_type": "code",

"execution\_count": 30,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'SUCCESS'"

]

},

"execution\_count": 30,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 31,

"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",

"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",

"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",

"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",

"spark-mllib\_2.3-r\_3.6 6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c base\n",

"tensorflow\_2.4-py3.7 65e171d7-72d1-55d9-8ebb-f813d620c9bb base\n",

"spss-modeler\_18.2 687eddc9-028a-4117-b9dd-e57b36f1efa5 base\n",

"pytorch-onnx\_1.2-py3.6 692a6a4d-2c4d-45ff-a1ed-b167ee55469a base\n",

"spark-mllib\_2.3-scala\_2.11 7963efe5-bbec-417e-92cf-0574e21b4e8d base\n",

"spark-mllib\_2.4-py37 7abc992b-b685-532b-a122-a396a3cdbaab base\n",

"caffe\_1.0-py3.6 7bb3dbe2-da6e-4145-918d-b6d84aa93b6b base\n",

"pytorch-onnx\_1.7-py3.7 812c6631-42b7-5613-982b-02098e6c909c base\n",

"cuda-py3.6 82c79ece-4d12-40e6-8787-a7b9e0f62770 base\n",

"tensorflow\_1.15-py3.6-horovod 8964680e-d5e4-5bb8-919b-8342c6c0dfd8 base\n",

"hybrid\_0.1 8c1a58c6-62b5-4dc4-987a-df751c2756b6 base\n",

"pytorch-onnx\_1.3-py3.7 8d5d8a87-a912-54cf-81ec-3914adaa988d base\n",

"caffe-ibm\_1.0-py3.6 8d863266-7927-4d1e-97d7-56a7f4c0a19b base\n",

"spss-modeler\_17.1 902d0051-84bd-4af6-ab6b-8f6aa6fdeabb base\n",

"do\_12.10 9100fd72-8159-4eb9-8a0b-a87e12eefa36 base\n",

"do\_py3.7 9447fa8b-2051-4d24-9eef-5acb0e3c59f8 base\n",

"spark-mllib\_3.0-r\_3.6 94bb6052-c837-589d-83f1-f4142f219e32 base\n",

"cuda-py3.7-opence 94e9652b-7f2d-59d5-ba5a-23a414ea488f base\n",

"nlp-py3.8 96e60351-99d4-5a1c-9cc0-473ac1b5a864 base\n",

"cuda-py3.7 9a44990c-1aa1-4c7d-baf8-c4099011741c base\n",

"hybrid\_0.2 9b3f9040-9cee-4ead-8d7a-780600f542f7 base\n",

"spark-mllib\_3.0-py38 9f7a8fc1-4d3c-5e65-ab90-41fa8de2d418 base\n",

"autoai-kb\_3.3-py3.7 a545cca3-02df-5c61-9e88-998b09dc79af base\n",

"spark-mllib\_3.0-py39 a6082a27-5acc-5163-b02c-6b96916eb5e0 base\n",

"runtime-22.1-py3.9-do a7e7dbf1-1d03-5544-994d-e5ec845ce99a base\n",

"default\_py3.8 ab9e1b80-f2ce-592c-a7d2-4f2344f77194 base\n",

"tensorflow\_rt22.1-py3.9 acd9c798-6974-5d2f-a657-ce06e986df4d base\n",

"kernel-spark3.2-py3.9 ad7033ee-794e-58cf-812e-a95f4b64b207 base\n",

"autoai-obm\_2.0 with Spark 3.0 af10f35f-69fa-5d66-9bf5-acb58434263a base\n",

"default\_py3.7\_opence c2057dd4-f42c-5f77-a02f-72bdbd3282c9 base\n",

"tensorflow\_2.1-py3.7 c4032338-2a40-500a-beef-b01ab2667e27 base\n",

"do\_py3.7\_opence cc8f8976-b74a-551a-bb66-6377f8d865b4 base\n",

"spark-mllib\_3.3 d11f2434-4fc7-58b7-8a62-755da64fdaf8 base\n",

"autoai-kb\_3.0-py3.6 d139f196-e04b-5d8b-9140-9a10ca1fa91a base\n",

"spark-mllib\_3.0-py36 d82546d5-dd78-5fbb-9131-2ec309bc56ed base\n",

"autoai-kb\_3.4-py3.8 da9b39c3-758c-5a4f-9cfd-457dd4d8c395 base\n",

"kernel-spark3.2-r3.6 db2fe4d6-d641-5d05-9972-73c654c60e0a base\n",

"autoai-kb\_rt22.1-py3.9 db6afe93-665f-5910-b117-d879897404d9 base\n",

"tensorflow\_rt22.1-py3.9-horovod dda170cc-ca67-5da7-9b7a-cf84c6987fae base\n",

"autoai-ts\_1.0-py3.7 deef04f0-0c42-5147-9711-89f9904299db base\n",

"tensorflow\_2.1-py3.7-horovod e384fce5-fdd1-53f8-bc71-11326c9c635f base\n",

"default\_py3.7 e4429883-c883-42b6-87a8-f419d64088cd base\n",

"do\_22.1 e51999ba-6452-5f1f-8287-17228b88b652 base\n",

"autoai-obm\_3.2 eae86aab-da30-5229-a6a6-1d0d4e368983 base\n",

"do\_20.1 f686cdd9-7904-5f9d-a732-01b0d6b10dc5 base\n",

"scikit-learn\_0.19-py3.6 f963fa9d-4bb7-5652-9c5d-8d9289ef6ad9 base\n",

"tensorflow\_2.4-py3.8 fe185c44-9a99-5425-986b-59bd1d2eda46 base\n",

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

]

}

],

"source": [

"client.software\_specifications.list(100)"

]

},

{

"cell\_type": "code",

"execution\_count": 32,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

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

]

},

"execution\_count": 32,

"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": 34,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

]

},

{

"name": "stderr",

"output\_type": "stream",

"text": [

"/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ibm\_watson\_machine\_learning/repository.py:1453: UserWarning: This method is deprecated, please use get\_model\_id()\n",

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

]

}

],

"source": [

"model\_details = client.repository.store\_model(model='gesture-classifier.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",

"\n",

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

]

},

{

"cell\_type": "code",

"execution\_count": 35,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'b551a542-6d4f-43d1-a1df-815fe17e7e8c'"

]

},

"execution\_count": 35,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"model\_id"

]

},

{

"cell\_type": "code",

"execution\_count": 36,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Successfully saved model content to file: 'gesture-model.tar.gz'\n"

]

},

{

"data": {

"text/plain": [

"'/home/wsuser/work/gesture-model.tar.gz'"

]

},

"execution\_count": 36,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"client.repository.download(model\_id, 'gesture-model.tar.gz')"

]

}

],

"metadata": {

"accelerator": "GPU",

"colab": {

"collapsed\_sections": [],

"provenance": []

},

"gpuClass": "standard",

"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": "3.9.0"

},

"vscode": {

"interpreter": {

"hash": "67f1dc6f6f712f7142079021955b91e049abb319dcfdc9eed010dd73dd4d845d"

}

}

},

"nbformat": 4,

"nbformat\_minor": 1

}