import sys

import os

import json

sys.path.append("/Users/manfrednde/Library/Python/3.9/lib/python/site-packages")

import requests

from urllib.parse import urlparse

from io import BytesIO

from PIL import Image, ImageDraw

import matplotlib.pyplot as plt

import os, time, uuid

from azure.cognitiveservices.vision.customvision.training import CustomVisionTrainingClient

from azure.cognitiveservices.vision.customvision.prediction import CustomVisionPredictionClient

from azure.cognitiveservices.vision.customvision.training.models import ImageFileCreateBatch, ImageFileCreateEntry, Region

from msrest.authentication import ApiKeyCredentials

def show\_image\_in\_cell(img\_url):

response = requests.get(img\_url)

img = Image.open(BytesIO(response.content))

plt.figure(figsize=(20,10))

plt.imshow(img)

plt.show()

TRAINING\_ENDPOINT = "https://customvisionboardingkiosk.cognitiveservices.azure.com"

training\_key = "cc665eb582794c84ae65e45209c6b243"

training\_resource\_id = '/subscriptions/1d697ec8-0d3c-4a0f-b9b2-a422ba575a26/resourceGroups/rg-boarding-kiosk/providers/Microsoft.CognitiveServices/accounts/customvisionboardingkiosk'

PREDICTION\_ENDPOINT = 'https://customvisionboardingkiosk-prediction.cognitiveservices.azure.com'

prediction\_key = "0092025e801246bea8129015b03dd65e"

prediction\_resource\_id = "/subscriptions/1d697ec8-0d3c-4a0f-b9b2-a422ba575a26/resourceGroups/rg-boarding-kiosk/providers/Microsoft.CognitiveServices/accounts/customvisionboardingkiosk-Prediction"

# Instantiate and authenticate the training client with endpoint and key

training\_credentials = ApiKeyCredentials(in\_headers={"Training-key": training\_key})

trainer = CustomVisionTrainingClient(TRAINING\_ENDPOINT, training\_credentials)

trainer.api\_version

prediction\_credentials = ApiKeyCredentials(in\_headers={"Prediction-key": prediction\_key})

predictor = CustomVisionPredictionClient(PREDICTION\_ENDPOINT, prediction\_credentials)

predictor.api\_version

# Find the object detection domain

obj\_detection\_domain = next(domain for domain in trainer.get\_domains() if domain.type == "ObjectDetection" and domain.name == "General")

# Todo: create a new project

print ("Your Object Detection Training project has been created. Please move on.")

project\_name = uuid.uuid4()

project = trainer.create\_project(project\_name, domain\_id=obj\_detection\_domain)

# project = {

# "id": "e859fe51-5814-4266-b10b-0793e8d8b35d"

# }

# Getting Project Details as collective information

project.as\_dict()

project.status

# Todo: add tags based on training requirements

tag\_1 = trainer.create\_tag(project.id, "lighter")

iteration = trainer.train\_project(project.id)

# We will keep checking every 10 seconds during the training progress

while (iteration.status != "Completed"):

iteration = trainer.get\_iteration(project.id, iteration.id)

print ("Training status: " + iteration.status)

print ("Waiting 10 seconds...")

time.sleep(10)

iteration.as\_dict()

iteration\_list = trainer.get\_iterations(project.id)

for iteration\_item in iteration\_list:

print(iteration\_item)

# Todo: check the preformance

model\_perf = trainer.get\_iteration\_performance(project.id, iteration\_list[0].id)

model\_perf.as\_dict()

# Todo: set the Iteration Name.

publish\_iteration\_name = "first-iteration-classes-object-detection-custom"

# Todo: publish it to the project endpoint

trainer.publish\_iteration(project.id, iteration.id, publish\_iteration\_name, prediction\_resource\_id)

print ("Done!")