import cv2

import numpy as np

import wiotp.sdk.device

import playsound

import random

import time

import datetime

import ibm\_boto3

from ibm\_botocore.client import Cofig, ClientError

#CloudantDB

from cloudant.client import Cloudent

from cloudant.error import CloudantException

from cloudant.result import Result, ResultByKey

from clarifai\_grpc.channel.clarifai\_channel import Clarifaichannel

from clarifai\_grpc.grpc.api import service\_pb2\_grpc

stub = service\_pb2\_grpc.V2stub(Clarifaichannel.get\_grpc\_channel())

from clarifai\_grpc\_grpc.api import service\_pb2, resources\_pb2

from clarifai\_grpc.grpc.api.status import status\_code\_pb2

# This is how you authenticaate.

metadata = (('authorization' 'Key bc885e5165d74ef8f8f42f6f6a2c9eb87'),)

cos\_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud" # Current list avaiable at https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints

cos\_API\_KEY\_ID - "f6Ap-ct18m07S9UzL7xpbAF170omePLLUQPzqmnAzb" # eg W00YiRnLW4a3fTjMB-odB-2ysfTrFBIQQWamc--p3byk"

COS\_AUTH\_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"

COS\_RESOURCE\_CRN ="crn:v1:bluemix:public:cloud-object-storage:global:a/bc85725531d4401488896ce505c7487d:90dd0fe6-11ac-484f-875c-247e68122e32::"

clientdb = Cloudant("apikey-v2-28j97wn6imki3og0g05cshoyuss464vvvr6x3muktxwg" ,url= "https://apikey-v2-28j97wn6imki3og0g05cshoyuss464vvvr6x3muktxwg:9277efda7e7b899ab71cdfa2a2264a0c@727f0654-0ba0-4464-bb6c-92da01a3fca4-bluemix.cloudantnosqldb.appdomain.cloud",

auth\_token = "9277efda7e7b899ab71cdfa2a2264a0c")

clientdb.connect()

# Create resource

cos = ibm\_boto3.resource("s3",

ibm\_api\_key\_id=COS\_API\_KEY\_IC,

ibm\_service\_instance\_id=COS\_RESOOURCE\_CRN,

ibm\_auth\_endpoint=COS\_AUTH\_ENDPOINT,

config=Cofig(signature\_version="oauth"),

endpoint\_url=cos\_ENDPOINT

)

def multi\_part\_upload(bucket\_name, item\_name,file\_path):

try:

print("Starting file transfer for (0) to bucket: {1}\n".format(item\_name, bucket\_name))

# set 5 MB chunks

part\_size = 1024 \* 5

# set threadhold to 15 MB

file\_threshold = 1024 \* 15

#set the transfer threshold and chunk size

transfer\_config = ibm\_boto3.transfer.TransferConfig(

multipart\_threshold=file\_threshold,

multipart\_chuksize=part\_size

)

# the upload\_fileobj mrthod will automatically execute a multi-part upload

# in 5 MB chuks for all files over 15 MB

with open(file\_path, "rb") as file\_data:

cod.Object(bucker\_name, item\_name).upload\_fileobj(

Fileobj=file\_data,

Config=transfer\_config

)

print("Transfer for {0} Complete!\n".format(item\_name))

except ClientError as be:

print("CLIENT ERROR: {0}\n".format(be))

except Exception as e:

print("Unable to complete multi-part upload:{0}".format(e))

def myCommandCallback(cmd):

print("Command received:%s" % cmd.data)

command=cmd.data['command']

print(command)

if(command=='lighton'):

print('lighton')

elif(command=='lightoff'):

print('lightoff')

elif(command=='motoron'):

print('motoron')

elif(command=='motoroff'):

print('motoroff')

myConfig = {

"identity": {

"orgId": "hj5fmy",

"typeId": "NodeMOU",

"deviceId":"12345"

},

"auth": {

"token": "12345678"

}

}

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)

client.connect()

database\_name = "samplel"

my\_database = clientdb.create\_database(database\_name)

if my\_database.exists():

print(f"'{database\_name}' successfully created.")

cap=cv2.VideoCapture('garden.mp4')

if(cap.isOpened()==True):

print('File opened')

else:

print('File not found')

while(cap.isOpened()):

ret,frame = cap.read()

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

ims = cv2.resize(frame, (960, 540))

cv2.imwrite('ex.jpg',ims)

with open("ex.jpg", "rb") as f:

file\_bytes = f.read()

#This is the model ID of a publicly available General model. You may use any other public or custom model ID.

request = service\_pb2.PostModelOutputsRequest(

model\_id='aaa03c23b3724a16a56b629203edc62c',

inputs=[resources\_pb2.Input(data=resources\_pb2.Data(image=resources\_pb2.Image(base64=file\_byres))

)])

response = stub.PostModelOutputs(request, metadata=metadata)

if response.status.code != status\_code\_pb2.SUCCESS:

raise Exception("Request failed, status code:" + str(response.status.code))

detect=False

for concept in response.outputs[0].data.concepts:

#print('%12s: %.2f' % (concept.name, concept.value))

if(concept.value>0.98):

#print(concept.name)

if(concept.name=="animal"):

print("Alert! Alert! animal detected")

playsound.playsound('alert.mp3')

picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")

cv2.imwrite(picname+'.jpg',frame)

multi\_part\_upload('gnaneshwar', picname+'.jpg', picname+'.jpg')

json\_document={"link":COS\_ENDPOINT+'/'+'gnaneshwar'+'/'+picname+'.jpg'}

new\_document= my\_database.create\_document(jsom\_document)

if new\_document.exists():

print(f"Document successfully created.")

time.sleep(5)

detect=True

moist=random.randint(0,1000)

hunidity=random.randint(0,100)

myData={'Animal': detect,'moisture':moist,'humidity':humidity}

print(muData)

if(humidity!=None):

client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,onPublish=None)

client.commandcCallback = myCommandCallback

cv2.imshow('frame',imS)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

client.disconnect()

cap.release()

cv2.desteoyAllWindows()