import datetime

import ibm\_boto3

from ibm\_botocore.client import Config, ClientError

import cv2

import numpy as np

import sys

import ibmiotf.application

import ibmiotf.device

import random

import time

from cloudant.client import Cloudant

from cloudant.error import CloudantException

from cloudant.result import Result, ResultByKey

organization = "bb2bpw"

deviceType = "RaspberryPi"

deviceId = "24102001"

authMethod = "token"

authToken = "raspberry"

def myCommandCallback(cmd):

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

print(cmd.data['command'])

if cmd.data['command']=="sirenon":

print("SIREN ON")

if cmd.data['command']=="sirenoff":

print("SIREN OFF")

if cmd.data['command']=="ledon":

print("BLINKING LED ON")

if cmd.data['command']=="ledoff":

print("BLINKING LED OFF")

if cmd.data['command']=="motoron":

print("MOTOR ON")

if cmd.data['command']=="motoroff":

print("MOTOR OFF")

try:

deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}

deviceCli = ibmiotf.device.Client(deviceOptions)

except Exception as e:

print("Caught exception connecting device: %s" % str(e))

sys.exit()

deviceCli.connect()

animal\_classifier=cv2.CascadeClassifier("haar-animal.xml")

video=cv2.VideoCapture(0)

COS\_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"

COS\_API\_KEY\_ID = "ffU9G4WuxXvsAV0muEVv-iAi2x3oS\_dcS5Q8qceZ2ZXA"

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

COS\_RESOURCE\_CRN = "crn:v1:bluemix:public:cloud-object-storage:global:a/8899ffc5103f4b6c824747890ea97e9f:a7ddbd11-4325-464e-82b5-d07c670c2642::"

client = Cloudant("apikey-v2-3r2lgzf2d6tor5oq125zsnodnm119qtsqnorjqjaff4","2481c47aba0e16cdeca03d4a11c6deca", url="https://apikey-v2-3r2lgzf2d6tor5oq125zsnodnm119qtsqnorjqjaff4:2481c47aba0e16cdeca03d4a11c6deca@ebc43d84-877b-439c-be4f-3f20160e4b30-bluemix.cloudantnosqldb.appdomain.cloud")

client.connect()

database\_name = "securitycamera"

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

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

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

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

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

endpoint\_url=COS\_ENDPOINT

)

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

try:

part\_size = 1024 \* 1024 \* 5

file\_threshold = 1024 \* 1024 \* 15

transfer\_config = ibm\_boto3.s3.transfer.TransferConfig(

multipart\_threshold=file\_threshold,

multipart\_chunksize=part\_size

)

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

cos.Object(bucket\_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))

while True:

check,frame=video.read()

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

animal=animal\_classifier.detectMultiScale(gray,1.3,5)

for(x,y,w,h) in animal:

cv2.rectangle(frame, (x,y), (x+y,y+h), (124,255,0), 2)

cv2.imshow('Animal Detection', frame)

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

picname=picname+".jpg"

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

cv2.imwrite(picname,frame)

mammal=1

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

multi\_part\_upload("cloud-object-storage-wb-cos-standard-kcg",picname,pic+".jpg")

if my\_database.exists():

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

json\_document = {

"\_id": pic,

"link":COS\_ENDPOINT+"/cloud-object-storage-wb-cos-standard-kcg/"+picname

}

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

if new\_document.exists():

print("Document '(new\_document)' successfully created.")

time.sleep(1)

t=26

h=63

m=38

data = {"d":{ 'temperature': t, 'humidity': h, 'soilmoisture': m, 'mammal': mammal}}

def myOnPublishCallback():

print ("Published data to IBM Watson")

success = deviceCli.publishEvent("data", "json", data, qos=0, on\_publish=myOnPublishCallback)

if not success:

print("Not connected to IoTF")

time.sleep(1)

mammal=0

deviceCli.commandCallback = myCommandCallback

Key=cv2.waitKey(1)

if Key==ord('q'):

video.release()

deviceCli.disconnect()

cv2.destroyAllWindows()

break