#libraries needed for taking pictures and naming files

from picamera import PiCamera

import datetime

import time

from time import sleep

#libraries needed to send emails from the R-Pi

import smtplib,ssl

from email.mime.multipart import MIMEMultipart

from email.mime.base import MIMEBase

from email.mime.text import MIMEText

from email.utils import formatdate

from email import encoders

#libraries needed for object detection

import numpy as np

import argparse

import cv2

#library used to delete pictures later

import os

date = datetime.datetime.now().strftime("%m\_%d\_%Y %H:%M") #find the current date and time

camera = PiCamera() #create a PiCamera object

sleep(5)

camera.capture("/home/pi/Desktop/"+date+".jpg") #take a photo and name it according to the date and time

camera.close()

del camera #delete PiCamera object

ap = argparse.ArgumentParser() #construct the argument parse for the command line

ap.add\_argument("-i", "--image", default="/home/pi/Desktop/"+date+".jpg", #image that we want to pass through the neural network

 help="path to input image")

ap.add\_argument("-p", "--prototxt", required=True,

 help="path to Caffe 'deploy' prototxt file")

ap.add\_argument("-m", "--model", required=True,

 help="path to Caffe pre-trained model")

ap.add\_argument("-c", "--confidence", type=float, default=0.2, #set the minimum confidence level.

 help="minimum probability to filter weak detections")

args = vars(ap.parse\_args())

CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat", #list of objects that this neural network is trained to detect

 "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",

 "dog", "horse", "motorbike", "person", "pottedplant", "sheep",

 "sofa", "train", "tvmonitor"]

COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3)) #assign a random color to each object

net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"]) #load the model

image = cv2.imread(args["image"]) #load image from command line

(h, w) = image.shape[:2] #resize image to 300x300 pixels

blob = cv2.dnn.blobFromImage(cv2.resize(image, (300, 300)), 0.007843, (300, 300), 127.5) #create "blob" from image

net.setInput(blob) #feed blob through network

detections = net.forward() #calculate detected objects

Objects=[] #array that will store names of detected objects for use in email later

Count=[] #array that will count number of detected objects for use in email later

for i in np.arange(0, detections.shape[2]): #for each detection:

 confidence = detections[0, 0, i, 2] #find the confidence level for each detection

 if confidence > args["confidence"]: #remove detections with confidence levels lower than the set minimum confidence level

 idx = int(detections[0, 0, i, 1]) #identify name of object

 Objects.append(idx) #add object to array "Objects"

 box = detections[0, 0, i, 3:7] \* np.array([w, h, w, h] #create bounding box for object

 (startX, startY, endX, endY) = box.astype("int")

 label = "{}: {:.2f}%".format(CLASSES[idx], confidence \* 100) #display detection on image

 print("{}".format(label))

 cv2.rectangle(image, (startX, startY), (endX, endY), #display bounding box on image

 COLORS[idx], 2)

 y = startY - 15 if startY - 15 > 15 else startY + 15

 cv2.putText(image, label, (startX, y), #display name and confidence of object

 cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, COLORS[idx], 2)

os.remove("/home/pi/Desktop/"+date+".jpg") #delete image from R-Pi

def send\_an\_email(): #sending an email with relevant information

 toaddr = “email@gmail.com” #email address you want to send data to

 me = "email@gmail.com" #email address you are sending from

 subject = "Object Detection :%s" %(date) #subject of email

 msg = MIMEMultipart()

 msg['Subject'] = subject

 msg['From'] = me

 msg['To'] = toaddr

 msg.preamble = "test"

 ObjectsNum = np.array(Objects) #this array will be used as the text in the email

 for g in range(len(CLASSES)): #for each type of object:

 Count.append(np.count\_nonzero(ObjectsNum == g))

 if Count[g] != 0: #if there was more than 0 counts of this object type detected:

 body = "Object: %s Count: %d \n"%(CLASSES[g],Count[g]) #print the name of the object and how many times it was detected

 msg.attach(MIMEText(body,'plain')) #attach information to email

 try:

 s = smtplib.SMTP('smtp.gmail.com', 587)

 s.ehlo()

 s.starttls()

 s.ehlo()

 s.login(user = 'email@gmail.com', password = 'password') #provide email and password of sending email

 s.sendmail(me, toaddr, msg.as\_string())

 s.quit()

 except smtplib.SMTPException:

 print ("Error") #print “error” in case of an exception

send\_an\_email()

**Important code notes:**

 Confidence level

 Classes

 Enter your email address and password