

yolo object recognition

By calling yolov4-tiny through opencv for object recognition detection, most object names can be recognized.

1. Main code

Code path:

```
/root/Dofbot/6.AI_Visual/8.Object_recognition.ipynb
```

2. File configuration

First, we need to import the yolov4-tiny network model structure cfg file, network weight weights file, and txt file of the classification name of the COCO data set. (The official data set and model of yolo4 are used directly here)

3. Target detection

3.1 Build model network structure

First, use the **cv2.dnn.readNet()** function to construct the CSPDarknet53 network structure, and pass in the model structure cfg file and the network weight weights file. opencv provides several methods to support image classification, detection, and segmentation for the neural network module, and automatically implements pre-processing and post-processing of input images. The target detection module **cv2.dnn_DetectionModel()** is used here to pass in the network model.

```
self.net = cv2.dnn.readNet('yolov4-tiny.cfg', 'yolov4-tiny.weights')
self.model = cv2.dnn_DetectionModel(self.net)
```

3.2 Target detection method

```
classIds, scores, bboxes = self.model.detect(image, confThreshold, numsThreshold)
```

parameter:

frame: Input image

confThreshold: The confidence threshold used to filter the selection box, the minimum confidence level for target detection

numsThreshold: Custom threshold in non-maximum suppression

return value:

classIds: category index

confidences: Confidence, the probability that the detection frame belongs to a certain category

boxes: Detection box information, upper left corner coordinates (x, y), box width and height (w, h)

3.3 Set model input parameters

```
self.model.setInputParams(size=(320,320), scale=1/255)
```

size means scaling the input image to the specified size. The larger the size, the better the detection effect, but the slower the detection speed. **scale** represents the scale size of the pixel value.

4. Use testing

The following code content needs to be executed according to the actual step, or you can click the Run All button. The specific code can be viewed in the code path.

Import various libraries and model files

```
#!/usr/bin/env python
# coding: utf-8
import Arm_Lib
import cv2 as cv
import threading
from time import sleep
import ipywidgets as widgets
from IPython.display import display
from Object_recognition import Object_recognition_identify
```

object recognition function

```
def detect_image(self, image):

    classids, scores, bboxes = self.model.detect(image, 0.5, 0.3)

    for class_id, self.score, bbox in zip(classids, scores, bboxes):
        self.x, self.y, self.w, self.h = bbox
        self.class_name = self.classes[class_id]

        cv2.rectangle(image, (self.x,self.y), (self.x+self.w,self.y+self.h),
(255,255,0), 2)

        cv2.putText(image, self.class_name, (self.x,self.y+self.h+20),
cv2.FONT_HERSHEY_COMPLEX, 1, (0,255,0), 2)

        cv2.putText(image, str(int(self.score*100))+'%', (self.x,self.y-5),
cv2.FONT_HERSHEY_COMPLEX, 1, (0,255,255), 2)

    return image
```

List of object names:

```
1 person
2 bicycle
3 car
4 motorbike
5 aeroplane
6 bus
7 train
8 truck
9 boat
10 traffic light
11 fire hydrant
12 stop sign
13 parking meter
14 bench
15 bird
16 cat
17 dog
18 horse
19 sheep
20 cow
21 elephant
22 bear
23 zebra
24 giraffe
25 backpack
26 umbrella
27 handbag
28 tie
29 suitcase
30 frisbee
31 skis
32 snowboard
33 sports ball
34 kite
35 baseball bat
36 baseball glove
37 skateboard
38 surfboard
39 tennis racket
40 bottle
41 wine glass
42 cup
43 fork
```

main thread

```
def camera():
    # Open camera Open camera
    capture = cv.VideoCapture(0)
    # Loop execution when the camera is turned on normally
    while capture.isOpened():
        try:
            _, img = capture.read()
            img = cv.resize(img, (640, 480))
            img = ob_re.detect_image(img)
            if model == 'Exit':
```

```
cv.destroyAllWindows()
capture.release()
break
imgbox.value = cv.imencode('.jpg', img)[1].tobytes()
except KeyboardInterrupt:capture.release()
```

After the program block is run, you can see the following interface to identify the corresponding object.

