

```

import cv2
import numpy as np
import matplotlib.pyplot as plt

yolo = cv2.dnn.readNet("/yolov3-tiny.weights", "/yolov3.cfg")

classes = []
with open("/coco.names", "r") as f:
    classes = f.read().splitlines()

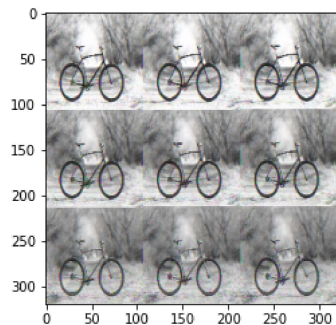
img = cv2.imread("/SKS08671.jpg")

blob = cv2.dnn.blobFromImage(img, 1/255, (320,320), (0,0,0), swapRB=True, crop=False)

width, height = 320, 320

i = blob[0].reshape(320,320,3)
plt.imshow(i)
plt.show()

```



```

yolo.setInput(blob)

outer_layer = yolo.getUnconnectedOutLayersNames()
layeroutput = yolo.forward(outer_layer)

```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

```

class_ids = []

for output in layeroutput:
    for detection in output:
        score = detection[5:]
        class_id = np.argmax(score)
        conf = score[class_id]
        if(conf > 0.7):
            cen_x = int(detection[0]*width)
            cen_y = int(detection[1]*height)

            w = int(detection[2] * width)
            h = int(detection[3]*height)

            x = int(cen_x - w/2)
            y = int(cen_y - h/2)

            box.append([x,y,w,h])
            confs.append(float(conf))
            class_ids.append(class_id)

```

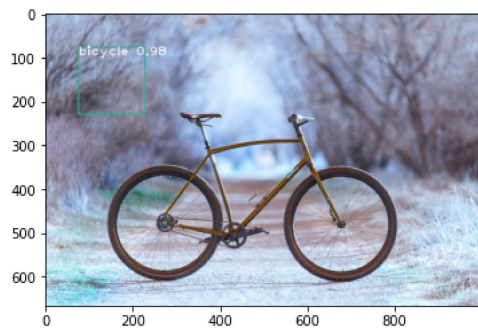
```
len(box)
```

```
indexes = cv2.dnn.NMSBoxes(box,confs,0.5, 0.4)
font = cv2.FONT_HERSHEY_PLAIN
colors = np.random.uniform(0, 255, size = (len(box),3))

for i in indexes.flatten():
    x,y,w,h = box[i]
    label = str(classes[class_ids[i]])
    confi = str(round(confs[i],2))
    color = colors[i]
    cv2.rectangle(img, (x,y), (x+w, y+h), color, 2)
    cv2.putText(img, label+" "+confi, (x,y+20), font, 2, (255,255,255), 2)

plt.imshow(img)
```

<matplotlib.image.AxesImage at 0x7f52d40a4a60>



Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

[Colab paid products](#) - [Cancel contracts here](#)