

GRIP THE SPARKS FOUNDATION INTERNSHIP

COMPUTER VISION AND IOT

NAME:- ANUJA JADHAV

Task 1: IMPLEMENT AN OBJECT DETECTOR WHICH IDENTIFIES THE CLASSES OF THE OBJECTS IN AN IMAGE OR VIDEO

```
In [3]: import cv2
```

```
In [4]: import matplotlib.pyplot as plt
```

```
In [5]: config_file = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
frozen_model = 'frozen_inference_graph.pb'
```

```
In [6]: model = cv2.dnn_DetectionModel(frozen_model,config_file)
```

```
In [7]: classLabels = []
file_name = 'Labels.txt'
with open(file_name, 'rt') as fpt:
    classLabels = fpt.read().rstrip('\n').split('\n')
```

```
In [8]: print(classLabels)
```

```
['person', 'bicycle', 'car', 'motorbike', 'aeroplane', 'bus', 'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'back pack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard', 'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'sofa', 'pottedplant', 'bed', 'diningtable', 'toilet', 'tvmonitor', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone', 'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush']
```

```
In [9]: print(len(classLabels))
```

```
80
```

```
In [10]: model.setInputSize(320,320)
model.setInputScale(1.0/127.5)
model.setInputMean((127.5,127.5,127.5))
model.setInputSwapRB(True)
```

```
Out[10]: <dnn_Model 000001A010A07850>
```

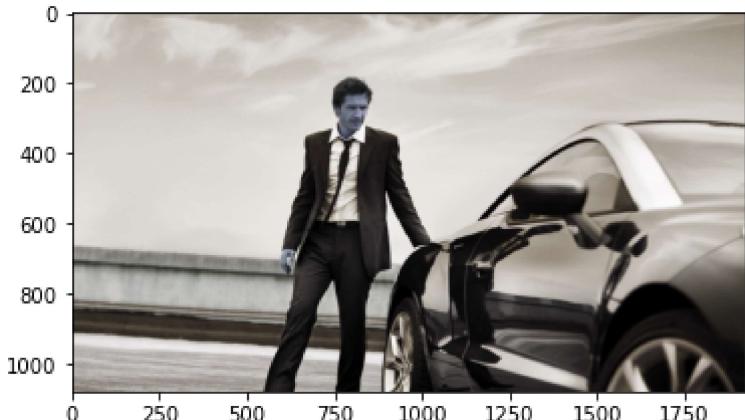
Image Detection

```
In [14]:
```

```
img = cv2.imread('man_tuxedo_car_style_27329_1920x1080.jpg')
```

In [15]: `plt.imshow(img) ##bgr`

Out[15]: <matplotlib.image.AxesImage at 0x1a0129c24f0>



In [16]: `plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))`

Out[16]: <matplotlib.image.AxesImage at 0x1a012a1d760>



In [17]: `ClassIndex, confidence, bbox = model.detect(img, confThreshold=0.5)`

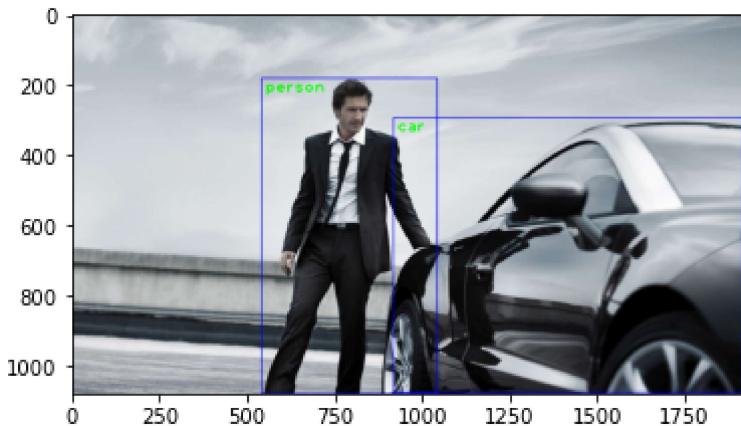
In [18]: `print(ClassIndex)`

[3 1]

In [19]: `font_scale = 3
font = cv2.FONT_HERSHEY_PLAIN
for ClassInd, conf, boxes in zip(ClassIndex.flatten(), confidence.flatten(), bbox):
 cv2.rectangle(img, boxes, (255, 0, 0), 2)
 cv2.putText(img, classLabels[ClassInd-1], (boxes[0]+10, boxes[1]+40), font, fontScale=font_scale)`

In [20]: `plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))`

Out[20]: <matplotlib.image.AxesImage at 0x1a012a7a940>



Video Detection

In [21]:

```
cap = cv2.VideoCapture("City street time lapse.mp4")

if not cap.isOpened():
    cap = cv2.VideoCapture(0)
if not cap.isOpened():
    raise IOError("Cannot open video")

font_scale = 3
font = cv2.FONT_HERSHEY_PLAIN

while True:
    ret,frame = cap.read()

    ClassIndex, confidence, bbox = model.detect(frame,confThreshold=0.55)

    print(ClassIndex)
    if (len(ClassIndex)!=0):
        for ClassInd, conf, boxes in zip(ClassIndex.flatten(), confidence.flatten(),
                                         bbox):
            if (ClassInd<=80):
                cv2.rectangle(frame,boxes,(255, 0, 0), 2 )
                cv2.putText(frame,classLabels[ClassInd-1],(boxes[0]+10,boxes[1]+40),fontScale=font_scale,fontFace=font)

    cv2.imshow('Object Detection Tutorial',frame)

    if cv2.waitKey(2) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

```
[ 3  3  1  3  3  1 10  3  1]
[ 3  3  3  1  1  3 10  3  1  3  8  3]
[ 3  3  3  1  1  3 10  3  1  3  8  3]
[ 3  3  3  3 10  1  3  1  3  1  3  3]
[ 3  3  3  3  1  1  3  3 10  3  3  6]
[ 3  3  3  8  1 10  3  1  1]
[ 3  3  1  1  3 10]
[ 3  3  3  3  1  1  3 10  3]
[ 3  3  3  3  3 10  3  1  1  3]
[ 3  3  3  3  3 10  3  3  1  3  1  3]
[ 1  3  3  3  3  3 10  1  1  3  6  3]
[ 3 1 3 3 3 1 1 3 3]
[ 1 8 3 1 3 1 3 3 3]
```

```
[ 3 1 3 3 1 3 1 3 3 10]
[ 3 1 1 1 3 3 3 3 1 3 3 3 1 10]
[ 3 3 1 3 3 1 3 3 3 1 10 3]
[ 3 1 3 1 3 3 1 3 3 10 3]
[ 3 3 1 1 3 1 3 1 3 10]
[ 3 1 1 3 1 3 1 1 3 10 3 3]
[ 3 1 3 1 3 1 3 3 3 10]
[ 3 3 3 1 1 3 1 3 3 3 1]
[ 3 3 3 3 1 3 3 3 3 1 10]
[ 3 3 3 3 3 1 3 1 3 10 10 3]
[ 3 3 1 3 10 3 1 10 3 3]
[ 3 3 1 1 10 3 1 3 10 3 3]
[ 3 3 1 1 3 1 3 1 10 3 3 3]
[ 3 3 3 3 1 1 1 3 3 1]
[ 3 1 3 3 1 3 3 1 3 3 3 10]
[ 1 3 1 3 3 1 3 3 3 3 10]
[ 1 3 1 3 3 3 1 3 3 3 3 10]
[ 3 1 3 3 3 3 3 1 3 3 1 10 3]
[ 3 3 3 1 3 3 1 3 3 3 3]
[ 3 3 1 3 3 3 3 3 3 1]
[ 3 3 3 3 1 3 3 3 3 3 3 1 1]
[ 3 3 3 3 3 1 3 3 3 3 3 1]
[ 3 3 3 3 3 1 3 3 3 1]
[ 3 1 3 3 3 3 3 3 3 3 3]
[ 3 3 1 3 3 3 3 3 3 3]
[ 3 3 1 3 3 3 3 3 3 3]
[ 3 3 3 3 1 3 3 3 1 3]
[ 3 1 3 3 3 3 3 3 3 3 3]
[ 3 1 3 3 3 3 3 3 3 3]
[ 3 3 3 3 3 3 3 3 3 3]
[ 3 3 3 3 3 3 1 1 1]
[ 3 3 3 1 3 3 3 1 3 3 3]
[ 3 3 3 3 1 3 3 1]
[ 3 3 3 1 8 1 3 3 1 3 3 3]
[ 8 3 1 3 3 3 1 3 3 1 3 10]
[ 3 8 3 3 3 3 3 3 1 3]
[ 3 6 3 3 3 3 3]
[ 6 3 3 3 3 3]
[ 3 1 3 3 3 3 3 10 3 3]
[ 1 3 3 10 1 3 3 3 3 3]
[ 3 1 3 3 3 3 10 1 1 3]
[ 3 3 1 1 3 10 1 3]
[ 3 3 3 1 1 3 3 3 3 10 1 3]
[ 3 3 3 1 3 3 3 3 3 3 1]
[ 3 3 3 3 3 3 3 3 3 1 1]
[ 3 1 3 1 3 3 1 1 3 3 3 3 1 3]
[ 3 3 3 1 3 3 3 1 3 1 1 1]
[ 3 1 3 3 3 1 1 1 1 1 1 3 3 3 3 10]
[ 3 3 1 3 1 1 1 1 1 1 3 3]
[ 3 1 3 3 3 1 1 1 1 1 1 1 10 3]
[ 3 1 3 3 3 1 1 1 10 3 3 3]
[ 3 3 1 3 1 1 3 10 3 3]
[ 3 3 3 3 3 1 1 3 3 3 10 3]
[ 3 1 3 3 3 3 3 1 10]
[ 3 3 3 3 1 1 3 6 3 3 3 1]
[ 3 1 3 1 3 3 3 3 1 3 3 3 6 3]
[ 3 3 3 3 1 3 3 3 8 1 3 1 3]
[ 3 3 3 3 3 3 3 3 3 3 3]
[ 3 3 3 3 3 3 3 3 3 3 3]
[ 3 3 1 3 3 3 3 3 1 1 3]
[ 3 3 1 3 3 3 3 3 3 3 3]
[ 3 1 3 3 3 3 3 1 3 3 3 3 6]
[ 3 1 3 3 3 3 1 3]
[ 3 3 3 3 1 3 3 3 1 3 10 3]
[ 3 1 3 3 3 3 3 1 10 3]
[ 3 3 3 1 3 1 3]
[ 3 3 1 3 3 10 1]
[ 1 3 3 3 3 1 10 3 3 1]
[ 1 1 3 3 3 1 3 1 6 3 10]
```

```
[ 3  1  1  3  1  1  1  3  10  3]
[ 1  1  3  3  1  3  10  3]
[ 1  3  3  1  1  1  3  3  10  3  3]
[3 1 3 1 1 1 3 3 1 3]
[1 1 3 3 1 1 3 1 3 3 3 3]
[3 1 1 3 3 1 3 3 3 1]
[3 1 3 1 3 3 3 3 1 3 3]
[1 3 1 3 3 3 1 3 3 3 3]
[3 3 3 3 3 3 1 1 1 3 3]
[3 1 3 3 3 3 3 1 3 3]
[3 3 3 3 1 3 3]
[ 3  3  1  1  10  1  3  3]
[ 3  3  1  10  3]
[ 3  1  3  1  3  10  1]
[ 3  1  3  1  10  3  3]
[ 3  1  1  1  3  10]
[1 3 3 3]
[1 3 3 3 3 1]
[3 3 3 3 3]
[3 3 3 3 3 3]
[ 3  3  3  3  3  3  3  10  3]
[ 3  3  3  3  3  3  3  10]
[ 3  3  3  3  3  3  3  10]
[ 3  3  3  3  3  3  3  3  10]
[ 1  3  3  3  3  3  3  3  3  10]
[1 3 3 3 3 1 3 3 3]
[ 3  3  3  3  3  3  1  10  3  3]
[ 3  3  1  3  3  3  3  3  3  3  3  10]
[ 3  3  1  3  3  3  3  3  3  3  3  10]
[ 3  3  3  3  1  3  3  3  3  3  3  10]
[ 3  3  3  3  3  1  3  3  3  3  3  10]
[3 3 3 3 3 3 3 3 3 1]
[ 3  3  3  3  3  3  3  10  3  3  3  3]
[ 3  1  3  3  3  3  3  3  10  3  3]
[ 3  1  3  3  3  3  3  8  3  10  3]
[ 3  3  3  3  3  3  3  10  6  3]
[3 3 3 3 3 1 3 3]
[ 3  3  3  3  3  3  3  10  3  3  8]
[ 3  3  3  3  3  3  1  3  3  10  3  3]
[ 3  3  3  3  3  3  1  3  1  3  3  3  10]
[ 3  3  8  3  3  3  1  3  3  3  10  3  1]
[ 3  3  3  3  3  3  1  1  1  3  3  3  10]
[3 3 3 3 3 1 3 1 3 3]
[3 3 3 1 3 1 3 3 3]
[ 3  3  3  3  1  1  3  3  3  10]
[ 3  3  3  3  1  3  10  3  3]
[3 3 3 3 1 3 3 3 3 1]
[ 3  3  1  3  1  1  1  3  3  3  3  10  3]
[3 3 1 3 3 3 3 3 3]
[ 3  3  3  1  3  3  3  3  3  3  3  10]
[ 3  3  3  1  3  10  3  3]
[3 1 3 3 1 3 3 3 3]
[3 3 3 3 3 1 3 1]
[ 3  3  3  3  3  3  3  1  1  3  1  3  1  1  10  3]
[3 8 3 3 3 1 1 3 1]
[3 3 3 1 3 1]
[3 1 8 3 3]
[8 3 3 1 3 3 3]
[8 3 3 3 3 3 3 1 1]
[8 3 3 3 3 1 3]
[ 8  1  3  3  3  3  1  10  3]
[ 8  3  10  3  3  1  1  1  3  10  3  10]
[ 3  8  3  3  3  3  10  10  3  3  3]
[3 3 3 3 3 3 3 8 3]
[ 8  3  3  3  3  3  3  3  3  3  10]
[ 3  3  3  3  3  3  3  3  3  10]
[3 3 3 3 3 3 3 3 3]
[ 3  3  3  3  3  3  3  3  10]
[3 3 3 3 3]
```

```
[3 3 3 3 3 8]
[3 3 3 3 3 3 3]
[1 6 3 8 3 3 3 3 3]
[8 3 3 3 3 3 1 3 3]
[ 1 3 3 3 10 3]
[ 1 3 1 3 3 3 10 3]
[1 1 3 3 3 3 3 3]
[ 3 1 3 1 3 3 10 3]
[1 6 3 3 3 3 3 3 3]
[ 6 3 3 3 8 3 10 3 1 6 3]
[6 3 3 3 1 3 3]
[6 1 3 3 3 3]
[3 3 1 1 3 3 1 1]
[ 3 1 1 3 3 6 3 1 3 10]
[3 3 3 3 3 3 1]
[ 3 3 3 3 1 10 3 3]
[ 3 3 3 3 10 3 3 3]
[ 3 3 3 3 10 3 3]
[ 3 3 3 3 10 3 3]
[ 3 3 3 3 10 3 3]
[ 3 3 3 3 10 3 3]
[ 3 3 8 3 3 3 10 3]
[ 3 3 3 3 3 10 6 3]
[ 3 3 3 3 10 3 3]
[ 3 3 3 3 3 10 3]
[ 3 3 3 3 8 3 3 10 3]
[ 3 3 3 3 10 3]
[ 3 3 3 3 10 3]
[ 3 3 3 1 10 3 3]
[ 3 1 3 3 3 10 3 8 3 3]
[ 1 3 3 10 3 3 3 3]
[ 3 3 3 10 1 3 1 3 3]
[ 3 3 3 3 10 3 3 3 10]
[ 3 3 3 3 3 10 3 3]
[ 3 3 3 3 3 10 3]
[ 3 3 3 3 10 3 3]
[ 3 3 3 3 10 3 3]
[ 3 3 3 3 10 8 3 3]
[ 3 3 3 3 8 3 10 3 3]
[ 3 3 3 3 3 10 3]
[ 3 3 3 3 3 10 3 3]
[ 3 3 3 3 3 10 3 3 10]
[ 3 3 3 3 3 10 3 3]
[ 3 3 3 3 10 3 3 1]
[ 1 3 3 3 1 10 3]
[ 3 1 3 3 3 10 3 3 1]
[ 3 3 3 3 10 3 3 1]
[ 3 3 3 10 3 3 3]
[ 3 3 3 10 3 1 3]
[ 1 3 3 3 1 10 3]
[ 3 1 3 3 3 10 3 3 1]
[ 3 3 3 3 10 3 3 1]
[ 3 3 3 10 3 1 3]
[ 3 3 3 3 3 10 3 3]
[ 3 3 3 3 3 10 3]
[ 3 1 3 3 1 10 3 1]
[ 3 3 3 3 3 10 3 1 3 3 1 1]
[ 3 3 3 3 3 3 3 10 3 3]
[ 3 3 3 3 3 3 3 3 10 1]
[ 3 3 3 3 3 8 3 3 3 3 10]
[ 1 3 3 3 3 3 3 10 1 1 3]
[ 3 1 3 3 3 3 3 3 3 10]
[ 3 8 3 3 3 1 6 10 3 10 3]
[3 3 3 3 3 1 3 1]
[ 3 3 1 3 1 3 3 3 3 1 1 10 1]
[ 3 1 1 3 3 3 3 3 3 1 10 3]
[ 3 3 3 3 3 3 1 3 3 10 1 1]
[ 3 3 1 3 3 3 3 1 3 3 3 10 1 1]
[ 3 3 3 3 3 3 3 8 3 3 10]
```

```
[ 3 3 3 3 3 3 3 3 3 10 8]
[ 3 3 3 3 3 3 3 3 10 3]
[ 3 3 3 3 3 3 1 3 3 10 8 3 3]
[3 3 6 3 3 3 3 3 1]
[ 3 3 3 3 6 3 1 3 3 10]
[ 3 3 3 3 3 3 3 1 3 10]
[3 3 3 3 3 3 3 1]
[ 3 3 3 3 3 3 1 3 3 10 8]
[ 3 3 3 3 3 3 3 3 10 3 1]
[ 3 3 3 3 1 10]
[3 3 3 3 3 3]
[3 3 3 3 3 1]
[3 3 3]
[ 3 1 3 3 1 3 3 3 10 3 3 1 3]
[ 1 3 3 3 3 1 1 3 3 10 3 1 3 3]
[ 3 3 1 1 1 3 3 1 3 3 1 10 1]
[ 3 3 3 3 3 3 3 1 1 10 1 3]
[3 3 1 3 3 3 3 3 3 1 3 1 3]
[3 3 3 3 1 3 1 1]
[3 1 1 3 3 1 1 3 3 3 1 3 1 3]
[3 1 1 1 3 3 3 3 3 1 3 3]
[1 3 3 3 1 1 3]
[ 3 1 3 3 3 1 3 3 3 1 10]
[3 3 3 1 3 3 1 3 1 3]
[1 3 1 3 3 1 3 3 3]
[3 1 1 3 3 3 3]
[3 3 1 3 3 3]
[3 1 3 3 1 3 3 3 3 3 3]
[3 3 1 3 1 3 3 3 3]
[3 3 1 3 3 3 3]
[1 3 3 3 1 3 3 8 3 1]
[3 3 1 1 3 3 3 8 3 3 3 3]
[3 1 3 1 3 3 3 3 3 3 3 1]
[3 1 1 3 1 3 3]
[3 3 1 3 3 1 1 1 1 3]
[3 3 1 3 3 1 1 1 3]
[ 3 3 3 1 1 3 1 3 3 1 3 3 1 3 10]
[3 1 3 3 1 3 1 3 3 3 1]
[3 3 1 3 1 1 3 3 3 3 1 1]
[3 3 3 1 3 1 1 3 1 1 3]
[ 3 3 3 1 1 1 3 3 1 1 3 1 3 1 1 10 1]
[ 3 3 3 3 1 1 3 1 3 1 1 10 3 3 1 1 1]
[ 3 3 1 3 1 3 1 3 1 1 3 3 3 1 1 1 1 3 1 10]
[ 3 3 3 1 1 1 3 3 1 3 1 1 1 1 3 3 3 10 1]
[ 3 3 3 3 3 1 3 1 3 1 1 1 1 1 10 1]
[ 3 3 3 3 1 3 1 3 1 3 1 3 10 1 1 3]
[ 3 3 3 1 3 3 3 1 3 1 3 1 3 10 3 10 3 3 3]
[ 3 3 3 3 1 3 1 3 3 3 1 1 10 10 3]
[ 3 3 3 1 3 3 3 1 1 10 1 3 3 3 1 6 1 3 10]
[ 3 3 3 3 3 3 3 1 1 3 1 3 1 1 3 6 3 10 3 3 3]
[ 3 3 3 3 3 1 3 1 3 1 3 3 3 10 3]
[ 3 3 3 3 3 1 3 1 3 1 3 3 3 10 3]
[ 3 3 3 3 1 3 1 3 1 3 3 3 10 3]
[ 3 3 3 3 1 3 1 3 1 3 3 3 10 3]
[ 3 1 3 1 3 1 3 3 3 3 10 3]
[ 3 3 1 3 3 1 3 1 3 1 10 3]
[ 3 1 3 3 1 3 1 3 1 10 3]
[ 3 3 1 3 3 1 3 1 3 1 10 3]
[ 3 3 3 1 3 1 3 1 3 1 10 3]
[ 3 3 3 1 3 1 3 1 3 1 10 3]
[ 3 3 3 1 3 1 3 1 3 1 10 3]
[ 3 1 1 3 3 3 3 3 3 1 1 10 3]
[ 1 1 3 3 3 1 3 1 3 1 10 3]
[ 1 3 3 3 1 3 1 3 1 10 3]
[ 1 3 3 1 3 1 3 1 3 1 10 3]
[ 1 3 3 1 3 1 3 1 3 1 10 3]
[ 1 3 3 1 3 1 3 1 3 1 10 3]
```

```
[ 3 3 3 3 3 10 3 3]
[ 3 3 3 3 3 3 3 10 3]
[ 3 3 3 3 3 10 3 ]
[ 3 3 3 3 3 3 10 3 3 3]
[ 3 3 3 10 3 3 3 3 ]
[ 3 3 3 10 3 3 3 3 ]
[ 3 3 3 3 3 10 3 3 ]
[ 3 3 3 3 3 10 3 ]
[ 3 3 3 8 3 3 10 3]
[ 3 3 3 3 10 8 3 3 3]
[ 3 3 3 10 3 3 8 3]
[ 3 3 3 10 3 3 ]
[ 3 3 3 10 3 ]
[ 1 3 3 3 1 1 10 3 3 3]
[ 3 3 10 3 3 10 3 1 1]
[ 1 1 1 3 3 10 1 1 1 3 3 3]
[ 3 1 3 1 10 1 3 3 3 1 3]
[ 1 3 1 3 1 1 3 3 10 1 3 1 1]
[ 3 1 3 3 1 10 1 3 1 3 3 3 1]
[ 3 1 3 1 1 3 3 10 1 3 3 3 1]
[ 1 3 1 3 3 3 1 3 10 3 1 3 3 3]
[ 1 3 3 1 3 3 3 10 1 3 3 3 6]
[ 1 3 3 3 1 10 1 3 3 3 ]
[ 1 1 3 1 3 3 3 10 3 3 3 1 6]
[ 1 3 3 3 1 1 1 10 3 3 3 1 3]
[ 3 1 3 3 1 3 10 3 3 1 1 3]
[ 1 3 3 3 3 10 1 3 3 3 ]
[ 3 1 1 3 1 3 3 3 10 3 3 3]
[ 1 3 3 3 1 8 3 6 3 3 10 3 1 3 8]
[ 1 3 3 3 3 3 3 10 3 3 ]
[ 3 1 3 8 1 3 3 1 8 10 3 6 6]
[ 3 3 3 3 1 1 6 1 10 3 1 ]
[ 3 3 3 3 3 1 10 1 10 3 1 3]
[ 3 3 1 1 1 3 3 1 10 3 1 ]
[ 3 1 3 1 3 3 1 3 10 3]
[ 3 1 3 1 3 3 1 1 10 3]
[ 3 1 3 3 1 3 3 1 1 1 10 3]
[ 3 3 1 3 3 3 1 1 3 10 3 1 1 1]
[ 3 3 3 3 3 1 1 3 10 ]
[ 3 1 3 3 3 1 1 1 3 3 3 10 ]
[ 3 3 3 3 1 3 10 ]
[ 3 3 3 3 1 3 1 1 3 3 3 10 ]
[ 3 3 3 1 3 3 3 1 3 ]
[ 3 1 1 1 3 3 3 3 3 6]
[ 3 1 3 3 1 3 3 3 1]
[ 1 3 3 3 3 1 3 1 1 3 3]
[ 1 3 3 3 3 1 1 1 3]
[ 1 3 3 3 1 1 3 3 1]
[ 1 3 3 1 3 1 3 3]
[ 1 1 3 3 3 3 3 1 3]
[ 3 3 3 3 10 3 3 1 3 1 3 3]
[ 3 8 3 3 3 1 3 6 1 3]
[ 3 3 3 1 3 3 8 1 3 1]
[ 3 3 3 3 3 10 3 3 1 1 3 1 1]
[ 3 3 3 3 3 10 8 6]
[ 3 3 3 3 3 3 1 1]
[ 3 3 3 3 1 3 1]
[ 3 3 3 1 3 3 1]
[ 3 3 3 1 3 1 1 1]
[ 3 3 3 3 3 1 3 1 1 3]
[ 3 3 1 3 1 3 1 1 3]
[ 3 3 3 3 3 1 3]
[ 3 3 3 3 3 1]
[ 3 3 3 3 3 1]
[ 3 3 1 3 3 1]
[ 3 3 3 1 3 3 1]
[ 3 1 3 3 3 1 1 3 3]
[ 3 3 1 3 3]
[ 3 3 3 3 1 1]
```

```
[3 3 1 3 1 3 6]
[1 3 3 3 1 1 8]
[3 3 1 1 1 3 3 1]
[1 3 3 1 1 3 1 8 1]
[1 3 3 1 1 8 3 1 3 3 3 3]
[3 3 1 1 1 1 1 3 1 3 3 8 3]
[3 3 3 1 3 3 3 3 3 8 1]
[3 3 1 3 1 1 3 3 3]
[3 3 3 3 1 3 3 3 3 3 1 3 3 10 1 3]
[3 3 3 3 2 1 3 1 1 3 10 1 8]
[3 3 3 1 1 1 3 8 1 3 10 1]
[3 3 3 3 8 3 3 3]
[3 3 3 3 3 3 8 1 6]
[3 3 3 6 3 8 3 8 3]
[3 3 3 8 3 3 6]
[3 3 3 8 3 3]
[3 3 8 3 3 3]
[3 3 3 1 3 3 8 3 3]
[3 3 1 3 3 8 3 3 3]
[3 3 3 1 3 3 3 3 8 1 1]
[3 3 3 1 8 3 1 1 3 6 3]
[3 1 6 3 8 1 3 3 3 3 3 3]
[3 3 1 3 1 3 3 8 3 3 3]
[3 3 3 1 1 3 6 3 3 3]
[3 3 3 3 3 1 3]
[3 3 3 3 1 3 3 3]
[3 3 3 3 3]
[3 3 3 3 3 3 8 3 3 3]
[3 3 3 3 3 1 3 1 3 3 3 6]
[3 3 3 3 3 3 3 6 3 3]
[3 3 3 1 3 3 3]
[3 3 3 3 3 3 3 10 3 3 3]
[3 3 3 3 3 10]
[3 8 3 3 3 3 3]
[3 3 3 3 3 3 10 3 3]
[3 3 3 3 3 3 10]
[3 3 3 3 3 10]
[3 3 3 3 10 3 3]
[3 1 3 3 3 3 10]
[3 1 1 3 3 3 10]
[3 1 3 3 3 10 3]
[3 1 3 3 3 3 10]
[3 3 1 3 3]
[3 1 3 1 3 3 3]
[3 3 3 1 3 3 3 3 1 1]
[3 3 3 3 3 3 3 10 3 1 3 3 8 6]
[3 3 1 1 1 3 3 3 3 10 3 8]
[3 1 3 3 1 3 1 3 1 10 1 3 3]
[3 3 1 1 3 3 3 1 1 3 3 10 1 3 3 3 1]
[1 3 3 3 3 3 1 1 10]
[3 3 1 1 3 3 3 10 1]
[3 3 3 1 1 3 3 10 8 1]
[3 3 3 1 1 3 3 3 10]
[3 3 3 3 1 1 3 3 3]
[3 3 3 1 3 1 1 3 3 1]
[1 3 3 3 3 1 1 3 3 3]
[3 1 3 1 3 3 1 3 1 3 1 3 3]
[3 1 3 1 3 3 1 1 3 1 3 1 3]
[1 3 3 3 3 1 1 3 1 3 1]
[3 1 3 8 3 3 1 3 1 1 3 3 1]
[3 3 1 3 3 3 1 8 3 1]
[1 3 3 3 3 3 1 1 3 3]
[3 3 1 3 1 3 1 3 3 3]
[3 3 1 3 3 1 1 3 8 3 3]
[3 3 3 1 3 1 1 3 1 3 3]
```

```
[3 3 3 1 1 3 3 3]
[3 3 3 3 3 1 3 3 1]
[3 3 3 3 1 3 1 3 3]
[3 3 3 1 1 1 3 1 3 3 3 3]
[3 3 3 1 1 1 1 3 1 3 3 1 3]
[3 3 1 3 3 1 3 3 1 1 3 3 3 8]
[3 3 3 1 1 3 3 1 3 1 1 3 3 1 1]
[3 3 3 1 1 3 3 3 1 3 1 1 3 3 1 3]
[3 3 3 3 1 1 3 3 1 3 1 3 1 3 1 3 1 3]
[3 3 3 1 3 3 1 1 3 1 3 1 3 1 3 1 3 1]
[3 3 1 3 1 3 3 1 3 3 1 1 3 1 1 3 1 3]
[3 1 3 1 3 3 1 3 3]
[1 3 3 3 1 3 1 3 1]
[ 3 1 1 3 3 3 3 10 1 3]
[ 3 3 3 1 10 3 1]
[ 3 3 3 3 3 10]
[ 3 3 3 3 10 3 1]
[ 3 3 8 3 3 10]
[ 3 3 3 10 3]
[ 3 3 10]
[ 3 10]
[ 3 3 3 10]
[ 3 3 10]
[ 3 3 1 10]
[ 3 3 1 1 3 10]
[ 3 1 3 3 1 3 10 1 1 3]
[ 3 1 3 1 3 1 1 10 3]
[ 3 3 3 1 10 3 1 1 1 3 3 1 3]
[ 3 3 1 3 1 3 1 3 10 3 1 3]
[ 3 3 1 3 1 10 3 1 1 3]
[ 3 3 3 10 3 1]
[ 3 1 3 1 1 10 3 3 1]
[ 3 3 3 1 10 3 1 3 3 1]
[ 3 3 1 3 1 10 3 3 1 1 10]
[ 3 1 3 3 1 1 3 3 10 3 1]
[ 3 3 1 1 1 3 10 3 3]
[ 3 3 1 1 3 1 3 1 1 3 10]
[ 3 3 3 1 3 1 1 3 1 3]
[ 3 3 3 3 1 3 3 3 3]
[ 3 1 3 3 3 1 1 3 10 3]
[ 3 1 3 3 3 1 3 1 3 10 1 3]
[ 3 3 3 1 1 3 3 3 1 3 10 3 3]
[ 1 3 3 3 3 1 3 1 3 10 1 1 3 1 3]
[ 3 3 1 3 1 3 1 3 10 3 3]
[ 3 3 3 3 1 1 1 3 10 3 1 3]
[ 3 3 3 3 1 1 1 3 10 1 1 1 1]
[ 3 3 3 3 3 3 3 3 10 1 1 1 1]
[ 3 3 3 1 3 3 3 3 3]
[ 3 3 1 3 1 3 1 3]
[ 3 3 3 1 3 3 10 1]
[ 3 3 3 10 3 3]
[ 3 3 3 1 1 3 3 3 3 1 3]
[ 3 3 1 3 3 3 1 1 3 1 10]
[ 3 3 3 3 3 1 1 1 3 1 3 10]
[ 3 3 1 1 3 3 3 1 1]
[ 3 1 3 3 3 1 3]
[ 3 3 3 1 1 3 1 3 3 3 8 1]
[ 3 3 3 3 3 3 1 3 1 8]
[ 3 3 1 8 3 1 3 3]
[ 3 3 8 1 3 3 3 1]
[ 3 3 8 3 3 1 1 3 3 3 1]
[ 3 8 3 3 3 3 1 1]
[ 3 3 3 8 3 3 1]
```

```
[3 3 8 3 3 3 3 3 1]
[3 8 3 3 3 3 3 1]
[3 8 3 3 1 3 3]
[3 3 1 3 3 3 3 3]
[ 3 1 3 3 1 3 1 1 3 1 3 1 10]
[8 3 1 3 6 3 1 3 3 3 1 1]
[3 3 1 3 8 1 3 3]
[ 3 3 1 10 1 3 1 1 3 3]
[ 3 3 1 3 10 3 1 1]
[ 3 3 3 3 1 10 1 1 3]
[ 1 3 3 3 10 1]
[3 3 3 3 1 1]
[ 1 3 1 1 3 3 3 1 10]
[1 1 3 3 1 1]
[3 1 1 1 3 3 3 1 3]
[3 1 1 1 1 3 1 3 3 1 1]
[1 3 3 1 3 3 1 3]
[1 3 1 3 1 1 3 1 1 3]
[3 1 1 3 3]
[3 3 1 1 1]
[3 3 3 8]
[3 3 1 3]
[3 3 1 3 1 1]
[3 1 1 1]
[3 3 3 1 3]
[3 3 1 1 1]
[3 1 1 3 1 3 1]
[3 1 3 1 1 3 3 3]
[3 1 3 1 1 1 1 3]
[3 3 1 3 3 3 1 1 1]
[1 3 3 3 3 3 1 1 1]
[3 3 1 3 3 1 1 1]
[1 3 3 1 1 3 3 1 1]
[3 1 3 1 1 3 1 1]
[3 1 1 1 3 1 1 3 3 1 1]
[ 1 3 1 3 3 3 3 1 10 1]
[3 1 3 3 1 3]
[3 1 3 1 3 3 3 1]
[3 1 3 1 3 3 3 3 1]
[3 3 1 3 3 3]
[3 3 3 3 3 1 3 3]
[3 3 3 3 3]
[3 1 3 1 3]
[3 1 1 3 3 1 1]
[3 1 1 3 3]
[1 1 3]
[ 1 3 1 1 1 10]
[1 1 3 3 1 3 1 1 3 1 3]
[3 1 1 1 1 1 3 3 3 3]
[1 1 3 3 1 1 3 3 3 3]
[ 1 1 10 3 1 1 3 3 1 3 1 1 3 3 3]
[ 1 3 1 1 3 3 3 1 1 3 1 1 1 10 3]
[3 3 3 1 3 1 1 3 1]
[3 1 1 3 3 3 3 1]
[ 3 3 1 3 3 3 3 1 1 10 1]
[ 3 3 1 3 1 3 1 3 10 1 10]
[3 3 3 1 3 1 1 3]
[1 3 3 1 1 3]
[3 1 1 3 1 3 3]
[3 1 3 3 3]
[3 3 3 3 1 8 1]
[3 3 1 3 3 3 1 3]
[1 3 3 1 1 3 3]
[3 3 3 1 1 3 3 1]
[3 1 3 1 1 3 3 3 3]
[ 3 1 1 1 3 3 3 10 3]
[ 3 3 1 1 3 1 3 1 3 1 10]
[ 3 3 1 1 1 3 1 10 3 3]
```

```
[ 3  3  1  3 10  1  3]
[ 3  3  1  3  1 10]
[3 3 1 1 3 3 1]
[ 3  3  1  1  3  3 10]
[ 3  3  1  1  1 10  3]
[ 1  3  3  3  3 10  1  1]
[ 3  1  3  1  1  3  3 10  1  1]
[ 1  3  1  3  1  3  1  3 10  1]
[ 3  1  3  1  3  3  1  1 10]
[3 1 3 1 1 1 3 3]
[1 1 3 3 1 1 1 3 3]
[ 3  1  1  3  1  1  1 10  3]
[ 3  3  1  1  1  1  1 10  3  3]
```

```
error                                         Traceback (most recent call last)
<ipython-input-21-95bec21b5e85> in <module>
    14     ret,frame = cap.read()
    15
--> 16     ClassIndex, confidence, bbox = model.detect(frame,confThreshold=0.55)
    17
    18     print(ClassIndex)

error: OpenCV(4.5.4-dev) D:\a\opencv-python\opencv-python\opencv\modules\imgproc\src\resize.cpp:4051: error: (-215:Assertion failed) !ssize.empty() in function 'cv::resize'
```

In []:

In []: