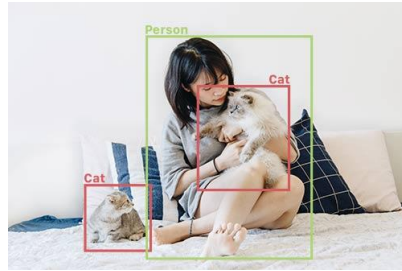


APVC – Exercício – Detecção de objetos



Este exercício envolve a deteção de objetos em imagens fixas, com o intuito de avaliar subjetivamente a qualidade das localizações dos objetos detetados (*bounding boxes*). O trabalho a desenvolver é composto pelas seguintes etapas:

- 1) Obter um pequeno conjunto de imagens (≈ 10), cujo conteúdo incida numa das classes de objetos detetada pelo modelo Yolo. Para tal, selecione uma classe à sua escolha, mas dentro das que são detetadas pelo Yolo. Poderá escolher por exemplo *Person*, *Car*, *Dog*, *Cat*, *Bird*, *Airplane*, etc. (veja a lista em anexo). Use algumas imagens mais “difíceis” com as seguintes características:

- Imagens onde o objeto da classe escolhida está parcialmente ocluído (escondido);
- imagens onde estejam presentes múltiplos objetos da classe escolhida;
- imagens onde estejam objetos da classe escolhida e objetos de outras classes que o Yolo também possa detetar (por exemplo, uma pessoa com um cão).

- 2) Adaptar o código para deteção de objetos com o Yolo do script `objectDetection.py`, de forma a conseguir aplicá-lo às imagens que selecionou. Visualize os resultados das predições usando `results[0].plot()`, onde `results` é o resultado devolvido pelo modelo. Caso detete tudo bem, volte ao ponto 1 e procure imagens mais difíceis.

- 3) Processar as *bounding boxes* de uma forma mais controlada, de forma a conseguir mostrar **apenas** as *bounding boxes* relativas aos **objetos da classe que escolheu**. Para este passo poderá ser necessário:

- consultar documentação do Yolo para aferir a forma como os resultados devolvidos pelo modelo estão organizados. Veja mais informações (incluindo um vídeo) em <https://docs.ultralytics.com/modes/predict/>;
- saber os IDs de cada classe. Para isso basta consultar o anexo ou então usar o atributo `names` do modelo devolvido pela função `YOLO(...)`:

```
model = YOLO('yolo11n.pt')  
print(model.names)
```

- utilizar a função `cv2.rectangle(...)` do OpenCV. Por exemplo, a instrução:

```
cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 1)
```

iria desenhar um retângulo na imagem `img`, entre os cantos com coordenadas `(x1,y1)` e `(x2,y2)`; o retângulo desenhado teria uma cor dada pelo BGR `(0,255,0)`, pelo que neste caso seria verde; a espessura da linha que delimita o retângulo é dada pelo último parâmetro da função – neste caso seria `1` pixel.

4) Com base nos resultados, indique:

- quantos objetos foram bem detetados? I.e., quantos *verdadeiros positivos* (TP)?
- quantos objetos presentes não foram detetados? I.e., quantos *falsos negativos* (FN)?
- quantos objetos foram devolvidos pelo detetor, sem correspondência com o que estava de facto presente na imagem (ou com um baixo grau de sobreposição com o objeto correspondente)? I.e. quantos *falsos positivos* (FP)?

5) Utilizando as contagens da alínea anterior, calcule a *precision*, *recall* e *F1-Score* no seu pequeno conjunto de imagens. Recorde que:

- $Precision = TP / (TP + FP)$
- $Recall = TP / (TP + FN)$
- $F1-Score = 2 \times Precision \times Recall / (Precision + Recall) = TP / (TP + FP/2 + FN/2)$

Anexo – lista de classes detetadas pelos modelos Yolo mais recentes

0: person	30: skis	60: dining table
1: bicycle	31: snowboard	61: toilet
2: car	32: sports ball	62: tv
3: motorcycle	33: kite	63: laptop
4: airplane	34: baseball bat	64: mouse
5: bus	35: baseball glove	65: remote
6: train	36: skateboard	66: keyboard
7: truck	37: surfboard	67: cell phone
8: boat	38: tennis racket	68: microwave
9: traffic light	39: bottle	69: oven
10: fire hydrant	40: wine glass	70: toaster
11: stop sign	41: cup	71: sink
12: parking meter	42: fork	72: refrigerator
13: bench	43: knife	73: book
14: bird	44: spoon	74: clock
15: cat	45: bowl	75: vase
16: dog	46: banana	76: scissors
17: horse	47: apple	77: teddy bear
18: sheep	48: sandwich	78: hair drier
19: cow	49: orange	79: toothbrush
20: elephant	50: broccoli	
21: bear	51: carrot	
22: zebra	52: hot dog	
23: giraffe	53: pizza	
24: backpack	54: donut	
25: umbrella	55: cake	
26: handbag	56: chair	
27: tie	57: couch	
28: suitcase	58: potted plant	
29: frisbee	59: bed	