

```
In [1]: from ultralytics import YOLO
import cv2
import math
```

```
In [3]: # start webcam
cap = cv2.VideoCapture(0)
cap.set(3, 640)
cap.set(4, 480)
```

Out[3]: True

```
In [4]: # model
model = YOLO("yolo-Weights/yolov8n.pt")
```

```
In [5]: # object classes
classNames = ["person", "bicycle", "car", "motorbike", "aeroplane", "bus", "tr
            "traffic light", "fire hydrant", "stop sign", "parking meter", "
            "dog", "horse", "sheep", "cow", "elephant", "bear", "zebra", "gi
            "handbag", "tie", "suitcase", "frisbee", "skis", "snowboard", "s
            "baseball glove", "skateboard", "surfboard", "tennis racket", "b
            "fork", "knife", "spoon", "bowl", "banana", "apple", "sandwich",
            "carrot", "hot dog", "pizza", "donut", "cake", "chair", "sofa",
            "diningtable", "toilet", "tvmonitor", "laptop", "mouse", "remote
            "microwave", "oven", "toaster", "sink", "refrigerator", "book",
            "teddy bear", "hair drier", "toothbrush"
            ]
```

```
In [*]: while True:
    success, img = cap.read()
    results = model(img, stream=True)

    # coordinates
    for r in results:
        boxes = r.boxes

        for box in boxes:
            # bounding box
            x1, y1, x2, y2 = box.xyxy[0]
            x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2) # convert to int

            # put box in cam
            cv2.rectangle(img, (x1, y1), (x2, y2), (255, 0, 255), 3)

            # confidence
            confidence = math.ceil((box.conf[0]*100))/100
            print("Confidence --->", confidence)

            # class name
            cls = int(box.cls[0])
            print("Class name -->", classNames[cls])

            # object details
            org = [x1, y1]
            font = cv2.FONT_HERSHEY_SIMPLEX
            fontScale = 1
            color = (255, 0, 0)
            thickness = 2

            cv2.putText(img, classNames[cls], org, font, fontScale, color, thickness)
            cv2.imshow('Webcam', img)
            if cv2.waitKey(1) == ord('q'):
                break

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

```
class name      / scissors
```

```
0: 480x640 1 person, 1 scissors, 1 toothbrush, 143.2ms
```

```
Speed: 0.0ms preprocess, 143.2ms inference, 0.0ms postprocess per image at
```

```
'image' (1280 1000 1000)
```

In []: