

الدرس الثاني: كيفية التعامل مع نتائج YOLOv8

مرحبًا بكم في الدرس الثاني حول استخدام YOLOv8 في التعرف على العناصر، في هذا الدرس، سنركز على كيفية التعامل مع النتائج التي نحصل عليها بعد تطبيق النموذج على الصور. سنقوم بفهم تفاصيل النتائج خطوة بخطوة وكيفية استخراج المعلومات المفيدة منها مثل إحداثيات الكائنات المكتشفة، مستويات الثقة، والتصنيفات.

الهدف من الدرس

في هذا الدرس، سنبدأ بطباعة المتغير `results` لمعرفة شكل البيانات التي نحصل عليها من النموذج. سنتعرف على وجود عنصر مهم داخل النتائج يسمى `boxes`، وسنستعرض محتوياته بالتفصيل مثل إحداثيات الصناديق المحيطة، مستويات الثقة، والفئات المكتشفة.

الكود [Download from GitHub](#)

إليك الكود الذي سنستخدمه في هذا الدرس:

```
from ultralytics import YOLO
import cv2
import numpy as np
# تحميل النموذج المدرب مسبقًا
model = YOLO('yolov8n.pt')
# قراءة الصورة
img = cv2.imread("00.jpg")
img = cv2.resize(img, (800, 600))
# تطبيق النموذج على الصورة
results = model(img, verbose=False)
print('----- results -----')
print(results)
print('-----')
print('----- Start results[0] -----')
print(results[0].boxes)
print('-----')
# استخراج إحداثيات الصناديق المحيطة
boxes = results[0].boxes.xyxy.tolist()
# استخراج إحداثيات المركز والعرض والارتفاع
xywhs = results[0].boxes.xywh.tolist()
# استخراج مستويات الثقة
```

```
confidences = results[0].boxes.conf.tolist()
```

```
# استخراج الفئات المكتشفة
```

```
classes = results[0].boxes.cls.tolist()
```

```
# استخراج أسماء الفئات
```

```
names = model.names
```

```
print('----- Start boxes -----')
```

```
print(boxes)
```

```
print('----- Start confidences -----')
```

```
print(confidences)
```

```
print('----- Start classes -----')
```

```
print(classes)
```

```
print('----- names[1] -----')
```

```
print(names[1])
```

```
print('-----')
```

```
# رسم الصناديق المحيطة والتسميات على الصورة
```

```
for xywh, box, conf, cls in zip(xywhs, boxes, confidences, classes):
```

```
    x1, y1, x2, y2 = map(int, box) # تحويل إحداثيات الصندوق إلى أعداد صحيحة
```

```
    xc, yc, w, h = map(int, xywh) # إحداثيات المركز والعرض والارتفاع
```

```
    confidence = conf
```

```
    detected_class = cls
```

```
    name = names[int(cls)] # الحصول على اسم الفئة من قائمة الأسماء
```

```
    # إذا كان الكائن هو شخص
```

```
    if cls == 0:
```

```
        # رسم دائرة في المركز
```

```
        cv2.circle(img, (xc, yc), 4, (255, 0, 0), -1)
```

```
        # رسم صندوق باستخدام إحداثيات المركز
```

```
        cv2.rectangle(img, (xc - w // 2, yc - h // 2), (xc + w // 2, yc + h // 2), (0, 255, 255), 2)
```

```
        # إضافة تسميات
```

```
        cv2.putText(img, f'{name} {confidence:.2f}', (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
```

```
        # طباعة معلومات الكائن المكتشف
```

```
        print(f"الفئة: {name}, الثقة: {confidence:.2f}, الإحداثيات: ({x1},{y1},{x2},{y2})")
```

```
# عرض الصورة مع الصناديق المحيطة
```

```
cv2.imshow("كشف الكائنات باستخدام YOLOv8", img)
```

```
if cv2.waitKey(0) & 0xFF == 27:
```

```
    cv2.destroyAllWindows()
```

شرح الكود خطوة بخطوة

١. طباعة النتائج الأولية:

○ أول خطوة نقوم بها هي طباعة المتغير `results` حتى نرى ما الذي نحصل عليه من النموذج بعد تطبيقه على الصورة حيث `results` يحتوي على جميع المعلومات التي نحتاجها عن الكائنات المكتشفة.

٢. استكشاف كائن `boxes`:

○ نقوم بطباعة `results[0].boxes` والذي يحتوي على الصناديق المحيطة (bounding boxes)، ومستويات الثقة، والفئات المكتشفة.

○ `boxes` يتضمن تفاصيل متعددة مثل إحداثيات الصناديق، مستويات الثقة، والفئات، وهي معلومات أساسية نحتاجها للتعامل مع الكائنات المكتشفة.

○ سوف نجد ان `results` عبارة `list` داخلها `object called boxes`

```
>>> %Run main.py
```

```
----- results -----
```

```
[ultralytics.engine.results.Results object with attributes:
```

```
boxes: ultralytics.engine.results.Boxes object
```

```
keypoints: None
```

```
masks: None
```

```
names: {0: 'person', 1: 'bicycle', 2: 'car', 3: 'motorcycle', 4: 'airp ...
```

```
obb: None
```

```
orig_img: array([[144, 159, 166],  
                 [144, 160, 167],  
                 [144, 160, 167]])
```

```
----- Start results[0] -----
```

```
ultralytics.engine.results.Boxes object with attributes:
```

```
cls: tensor([ 0., 26.])
```

```
conf: tensor([0.9159, 0.3638])
```

```
data: tensor([[3.8722e+00, 0.0000e+00],
```

```
          [1.4736e+02, 2.6000e+01]])
```

```
id: None
```

```
is_track: False
```

orig_shape: (600, 800)

shape: torch.Size([2, 6])

xywh: tensor([[64.4074, 343.1084, 121.0705, 355.1733],

[185.4653, 310.0549, 76.2165, 134.9742]])

xywhn: tensor([[0.0805, 0.5718, 0.1513, 0.5920],

[0.2318, 0.5168, 0.0953, 0.2250]])

xyxy: tensor([[3.8722, 165.5218, 124.9426, 520.6951],

[147.3571, 242.5678, 223.5735, 377.5420]])

xyxyn: tensor([[0.0048, 0.2759, 0.1562, 0.8678],

[0.7302, 0.1830, 0.9087, 0.9894],

[0.1842, 0.4043, 0.2795, 0.6292]])

٣. استخراج البيانات المهمة:

- نقوم باستخراج الإحداثيات باستخدام:

```
results[0].boxes.xyxy.tolist()
```

- ومستويات الثقة باستخدام:

```
results[0].boxes.conf.tolist()
```

- والفئات باستخدام:

```
results[0].boxes.cls.tolist()
```

٤. عرض النتائج على الصورة:

- في هذه الخطوة، نقوم برسم الصناديق المحيطة بالكائنات على الصورة، مع إضافة تسميات توضح اسم الكائن ومستوى الثقة.