

الدرس الأول: التعرف على الكائنات باستخدام YOLOv8 و OpenCV

مرحبًا بكم في درسنا الأول حول استخدام YOLOv8، وهو أحد أحدث النماذج في مجال التعرف على الكائنات. سنبدأ اليوم بتعريفكم على YOLOv8 وكيفية استخدامه في تطبيق بسيط.

ما هو YOLOv8 ؟

YOLOv8 هو نموذج متقدم للتعرف على الكائنات يستخدم في تحليل الصور والفيديوهات لتحديد وتصنيف الكائنات الموجودة فيها. يتميز هذا النموذج بدقته وسرعته العالية في معالجة الصور، مما يجعله مثاليًا لتطبيقات مختلفة مثل المراقبة الأمنية، القيادة الذاتية، والروبوتات. وغيرها من التطبيقات.

سوف نستخدم برنامج Thonny لكتابة الاكواد وتطبيق الدروس

ما هو برنامج Thonny ؟

Thonny هو بيئة تطوير متكاملة (IDE) بسيطة ومناسبة للمبتدئين في برمجة Python. يوفر Thonny واجهة مستخدم سهلة الاستخدام مع أدوات مدمجة لتصحيح الأخطاء وتجربة الأكواد، مما يجعله خيارًا ممتازًا لمشاريع تعلم البرمجة.

كيفية تحميل وتثبيت Thonny

١. تحميل: Thonny

- اذهب إلى موقع Thonny الرسمي: [Thonny Download](#)
- اختر نظام التشغيل المناسب لجهازك (Windows ، macOS ، أو Linux) وقم بتحميل الملف التنفيذي.

٢. تثبيت: Thonny

- افتح الملف الذي قمت بتحميله واتبع التعليمات المعروضة لتثبيت البرنامج على جهازك.

٣. تشغيل: Thonny

- بعد التثبيت، قم بفتح برنامج Thonny من قائمة التطبيقات أو من سطح المكتب.

إعداد بيئة العمل في Thonny

١. تثبيت المكتبات الضرورية:

- افتح Thonny ، ثم اذهب إلى قائمة "Tools" (الأدوات) واختر ("Manage packages") إدارة الحزم.
- في نافذة إدارة الحزم، ابحث عن `opencv-python` و `ultralytics` وقم بتثبيتهما.

ما سنقوم به في هذا الدرس

في هذا الدرس، سنقوم بإنشاء تطبيق بسيط يستخدم YOLOv8 مع مكتبة OpenCV لعرض النتائج على فيديو. سوف نتعرف على كيفية تحميل النموذج، قراءة الفيديو، وتطبيق النموذج على كل إطار من الفيديو، ثم عرض النتائج مع الإطار المعالج.

١. إعداد المشروع:

○ أنشئ ملفاً جديداً في Thonny وضع فيه الكود التالي: [download from GitHub](#)

```
import cv2
import numpy as np
from ultralytics import YOLO

# تعريف النموذج المدرب مسبقاً
model = YOLO("yolov8s.pt")

# تعريف مصدر الفيديو
cap = cv2.VideoCapture("video2.mp4")

while True:
    success, frame = cap.read()
    if success:
        frame = cv2.resize(frame, (1020, 500)) # تغيير حجم الإطار
        results = model.predict(frame) # تطبيق النموذج على الإطار

        # عرض النتائج على الإطار
        annotated_frame = results[0].plot()
        cv2.imshow("نتائج YOLOv8", annotated_frame)
    else:
        break

    if cv2.waitKey(1) & 0xFF == 27: # التوقف عند الضغط على مفتاح ESC
        break

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

شرح الكود

١. استيراد المكتبات: نقوم أولاً باستيراد المكتبات الضرورية، مثل cv2 (OpenCV) و YOLO من مكتبة ultralytics.
٢. تحميل النموذج: نقوم بتحميل النموذج المدرب مسبقاً باستخدام YOLO("yolov8s.pt"). هذا النموذج يحتوي على خوارزميات التعرف على الكائنات.
٣. فتح الفيديو: نستخدم cv2.VideoCapture لفتح ملف الفيديو الذي سنعمل عليه.
٤. قراءة وعرض الإطارات: في حلقة لا نهائية، نقرأ كل إطار من الفيديو، نقوم بتغيير حجمه، ثم نطبق النموذج على هذا الإطار. نقوم بعد ذلك بعرض الإطار المعالج مع الرسوم البيانية للكائنات المميزة.
٥. إيقاف التنفيذ: نستخدم cv2.waitKey حتى يتم الضغط على مفتاح ESC، والذي سيوقف تشغيل الفيديو ويغلق جميع النوافذ.

تشغيل التطبيق

شغل الكود وسترى نافذة تعرض الفيديو مع الكائنات التي تم اكتشافها. ستلاحظ أن الإطار يحتوي على مستطيلات حول الكائنات مع تسميات توضح نوع كل كائن.

في الدروس القادمة، سنتناول كيفية التعامل مع المتغيرات والنتائج في قوائم وبيانات إطارية (DataFrames) تابعونا لمزيد من التفاصيل!

أرجو أن يكون هذا الدرس واضحًا.

دمتم بخير وإلى اللقاء في الدرس القادم.