



VISIÓN ARTIFICIAL

Proyecto. Detección de objetos

Ingeniería en Mecatrónica
6to semestre

Mtro. Mauricio Alejandro Cabrera Arellano
Alana Michelle Cantón Moreno - 22310155

Detección de objetos

La detección de objetos es una de las aplicaciones más importantes de la visión artificial moderna, ya que permite identificar y localizar objetos dentro de imágenes o secuencias de video en tiempo real. Para esta tarea, se utilizan redes neuronales convolucionales especializadas como YOLO (You Only Look Once), que destacan por su rapidez y precisión.

En esta práctica se implementó un sistema de detección de objetos en video en tiempo real, utilizando el modelo YOLOv8 y la biblioteca OpenCV en Python.

Objetivo

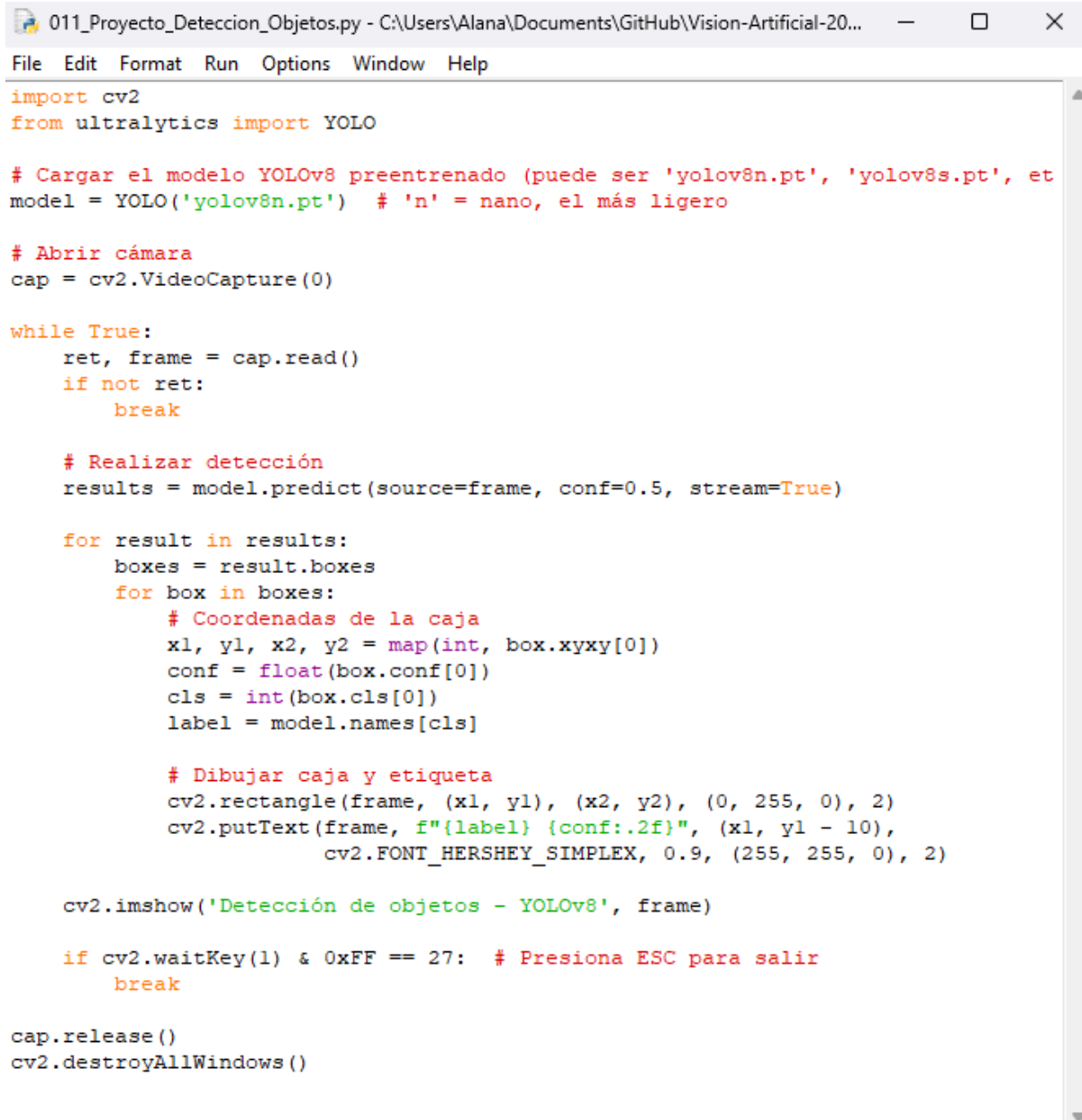
Implementar un programa de detección de objetos en tiempo real utilizando el modelo YOLOv8 preentrenado, procesando imágenes capturadas desde una cámara web, mostrando las cajas delimitadoras y etiquetas de los objetos detectados.

Procedimiento

1. Se instaló la librería **ultralytics** y **opencv-python** mediante `pip install ultralytics opencv-python`.
2. Se importaron las librerías necesarias: `cv2` y `YOLO` desde **ultralytics**.
3. Se cargó el modelo YOLOv8 preentrenado (versión `nano` por su ligereza y rapidez).
4. Se activó la cámara web mediante `cv2.VideoCapture(0)`.
5. Se creó un bucle `while` para capturar cada frame de la cámara.
6. Cada imagen se procesó con el modelo para obtener predicciones de los objetos detectados.
7. Se dibujaron cajas delimitadoras y etiquetas sobre los objetos detectados en cada frame.

8. Se visualizó el video con las detecciones en una ventana de OpenCV.
9. El bucle finalizó al presionar la tecla **ESC**.
10. Se liberó la cámara y se cerraron las ventanas.

Este es el código que se desarrolló:



```
011_Proyecto_Deteccion_Objetos.py - C:\Users\Alana\Documents\GitHub\Vision-Artificial-20...
File Edit Format Run Options Window Help

import cv2
from ultralytics import YOLO

# Cargar el modelo YOLOv8 preentrenado (puede ser 'yolov8n.pt', 'yolov8s.pt', et
model = YOLO('yolov8n.pt') # 'n' = nano, el más ligero

# Abrir cámara
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    if not ret:
        break

    # Realizar detección
    results = model.predict(source=frame, conf=0.5, stream=True)

    for result in results:
        boxes = result.boxes
        for box in boxes:
            # Coordenadas de la caja
            x1, y1, x2, y2 = map(int, box.xyxy[0])
            conf = float(box.conf[0])
            cls = int(box.cls[0])
            label = model.names[cls]

            # Dibujar caja y etiqueta
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.putText(frame, f"{label} {conf:.2f}", (x1, y1 - 10),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.9, (255, 255, 0), 2)

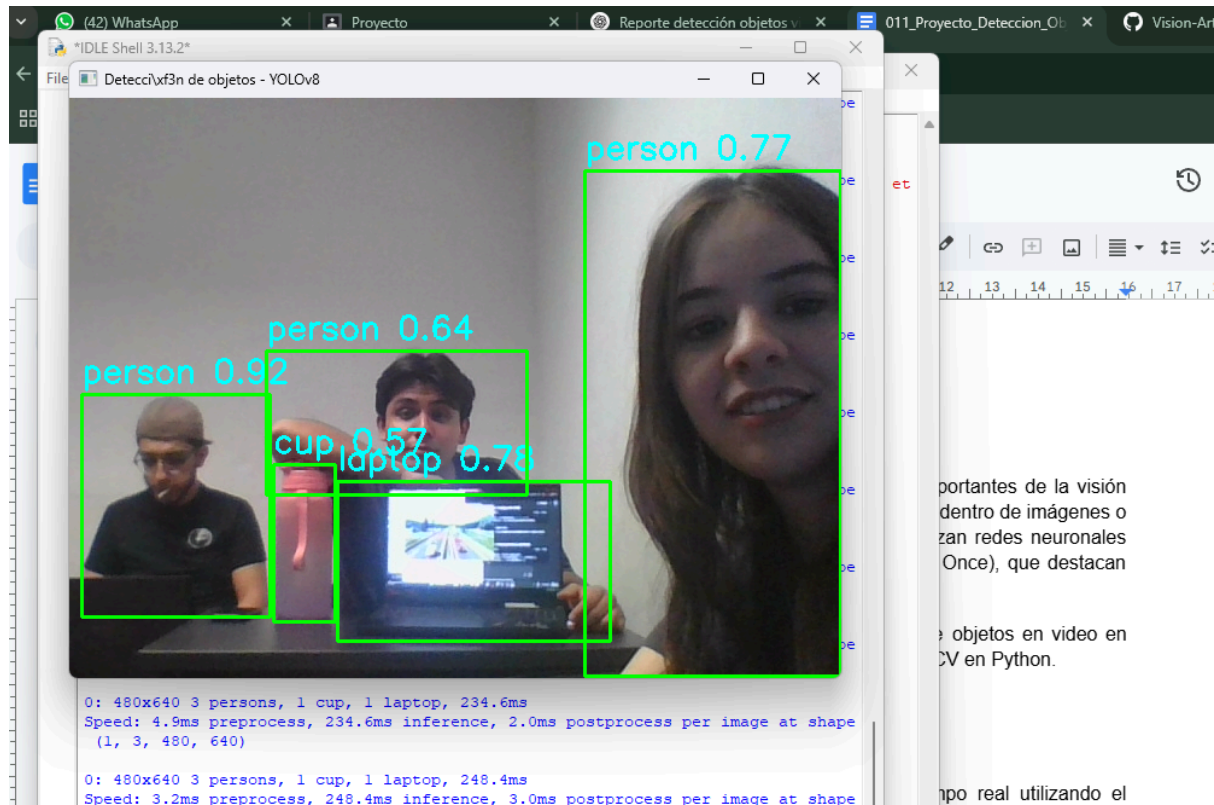
        cv2.imshow('Detección de objetos - YOLOv8', frame)

    if cv2.waitKey(1) & 0xFF == 27: # Presiona ESC para salir
        break

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

Resultados

Se logró implementar exitosamente un sistema de detección de objetos en tiempo real. Al ejecutar el programa, la cámara web se activó y mostró un video en el que aparecieron cajas verdes con etiquetas y valores de confianza sobre los objetos detectados, tales como “person”, “cell phone”, “cup”, entre otros, dependiendo de lo que la cámara captara en ese momento.



Conclusiones

La práctica permitió comprender el funcionamiento de un modelo de detección de objetos basado en aprendizaje profundo, utilizando YOLOv8 preentrenado. Se logró una detección en tiempo real eficiente y estable, evidenciando el potencial de estas herramientas para aplicaciones prácticas en seguridad, automatización y análisis de video.

El uso de la versión **nano** de YOLOv8 resultó adecuado para la capacidad de procesamiento del equipo, manteniendo una buena velocidad de detección sin comprometer significativamente la precisión.

Link al proyecto en GitHub:

https://github.com/AlanaMich/Vision-Artificial-2025/tree/0c9a6edd595f965c305a14661f3cae2381e87785/011_Proyecto_Deteccion_Objeto