



Centro de Enseñanza Técnica Industrial

Diego Sebastián Becerril Vizcaya

22310275

6°G

Proyecto Final

Visión artificial

Mtro. Mauricio Alejandro Cabrera Arellano

20 de junio de 2025

Objetivos del Proyecto

- Implementar un modelo de detección de objetos utilizando la arquitectura YOLO.
- Utilizar un dataset compatible con YOLO desde Hugging Face para entrenamiento o inferencia.
- Comprender el flujo de trabajo en la visión por computadora basada en deep learning.
- Probar el funcionamiento del modelo en imágenes o video, identificando y etiquetando objetos.
- Analizar posibles errores durante la ejecución y documentar soluciones.

Descripción del Proyecto

El proyecto consiste en utilizar un modelo preentrenado basado en la arquitectura YOLO (You Only Look Once) para realizar detección de objetos en imágenes o videos. El código se desarrolló en Python utilizando bibliotecas como ultralytics, opencv-python, y torch. Se utilizó un dataset alojado en la plataforma **Hugging Face**, con anotaciones en formato YOLO, lo que facilitó la implementación sin necesidad de convertir formatos.

Se eligió YOLO por su balance entre **velocidad** y **precisión**, ideal para aplicaciones en tiempo real.

Estructura y Código

```
import cv2
```

```
from ultralytics import YOLO
```

```
model = YOLO('yolov8n.pt') # 'n' = nano, el más ligero
```

```
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()

Pruebas y Resultados

- Se realizaron pruebas sobre imágenes y videos locales, así como con la cámara en tiempo real.
- El modelo detectó con éxito objetos como personas, vehículos y animales (según el dataset).
- La precisión aumentó en la versión entrenada respecto al modelo base.

Errores y Soluciones

| Problema | Causa | Solución |
|-------------------------------------|-------------------------------|--|
| Dataset no reconocido | Formato incorrecto | Se adaptaron las rutas y el data.yaml |
| Error No module named 'ultralytics' | Falta de instalación | Ejecutar pip install ultralytics |
| Baja precisión | Dataset pequeño o mal anotado | Aumentar cantidad de datos o hacer limpieza manual |