



Universidad Autónoma de San Luis Potosí
Facultad de ingeniería
Inteligencia Artificial Aplicada
Práctica: 9
Nombre Práctica: Detección de Objetos
Nombre del Alumno: Manuel Ramírez Galván



Fecha: 09/04/2025

Procedimiento

9.1.- Abrir el Notebook “object_detection.ipynb” y seguir los pasos indicados.

Ultralytics proporciona un modelo de YOLOv8 pre entrenado el cual podemos utilizar para diversas tareas

```
[ ] from ultralytics import YOLO

#Creamos el modelo yolov8 pre entrenado
model = YOLO("yolov8n.pt")
```

Imagen 1.- Importar y Crear Modelo

Entrenaremos el modelo con el dataset de [COCO](#).

Este paso nos permite adaptar nuestro modelo a un dataset personalizado.

```
[ ] results = model.train(data='coco128.yaml', epochs=5)
```

Imagen 2.- Entrenar Modelo

Despues de haber entrenado el modelo es importante validar los datos para asegurarnos de que las predicciones tienen un bajo porcentaje de error.

```
results = model.val()
```

Imagen 3.- Validar Modelo

```
[ ] from google.colab import files

uploaded = files.upload()
```

Imagen 4.- Cargar Imágenes

```
[ ] import cv2
import matplotlib.pyplot as plt

image_path = list(uploaded.keys())[0]

image = cv2.imread(image_path)

plt.imshow(image[:, :, ::-1])
```

Imagen 5.- Mostrar imagen Cargada

```

results = model.predict(source=image, conf=0.5)

results[0].save("result.png")

result = cv2.imread("result.png")

plt.imshow(result[:, :, :-1])

```

Imagen 6.- Mostrar imagen con Predicción

Resultados

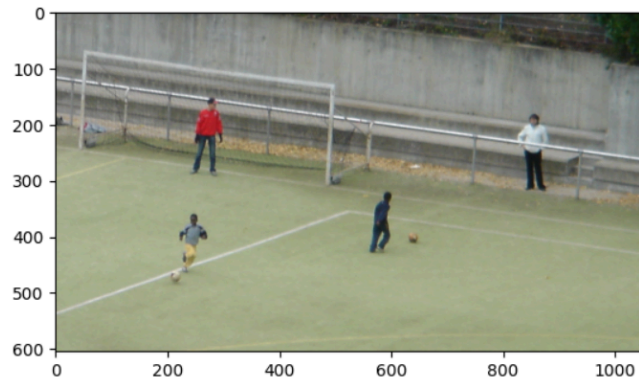


Imagen 7.- Imagen

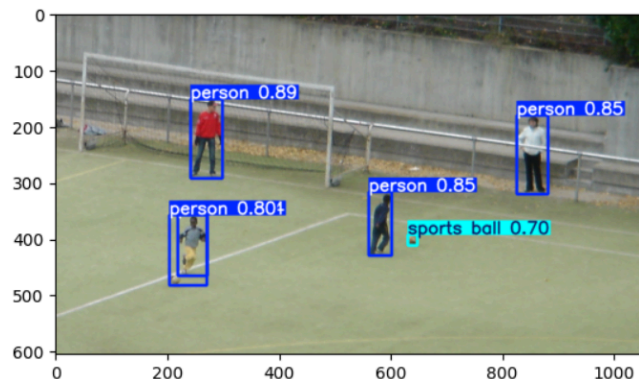


Imagen 8.- Imagen con Predicción

Procedimiento

9.2.- Abrir el Notebook “custom_object_detection.ipynb” y seguir los pasos indicados para entrenar un modelo personalizado de detección de objetos.

En el enlace del conjunto de datos, haz clic en `Download`, como formato selecciona `YOLOv8` que es la versión de Ultralytics que utilizaremos en este cuaderno y selecciona `Show Download Code`.

En la pestaña `Jupyter` copia el código de descarga y ejecútalo en una celda de código. Debe verse similar al siguiente código:

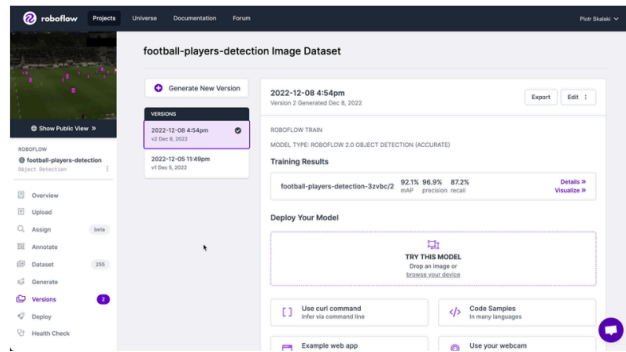


Imagen 9.- Descarga de Dataset

```
from roboflow import Roboflow
rf = Roboflow(api_key="RiMLBe8qPf6pC02iGZGr")
project = rf.workspace("brad-dwyer").project("pklot-1tros")
version = project.version(4)
dataset = version.download("yolov8")
```

Imagen 10.- Cargar Dataset

```
from ultralytics import YOLO

#Creamos el modelo yolov8 pre entrenado
model = YOLO("yolov8n.pt")
```

Imagen 11.- Crear Modelo

```
data= "/content/data1.yaml"
model.train(data=data, epochs=5)
```

Imagen 12.- Entrenar Modelo

```
results = model.val()
```

Imagen 13.- Validar Modelo

```
[ ] from google.colab import files

uploaded = files.upload()
```

Imagen 14.- Cargar Imágenes

```
[ ] import cv2
import matplotlib.pyplot as plt

image_path = list(uploaded.keys())[0]

image = cv2.imread(image_path)

plt.imshow(image[:,:,:-1])
```

Imagen 15.- Mostrar imagen Cargada

```
results = model.predict(source=image, conf=0.5)

results[0].save("result.png")

result = cv2.imread("result.png")

plt.imshow(result[:,:,:-1])
```

Imagen 16.- Mostrar imagen con Predicción

Resultados

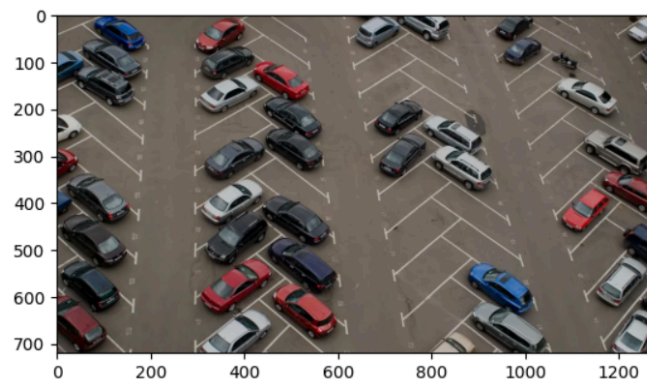


Imagen 17.- Imagen

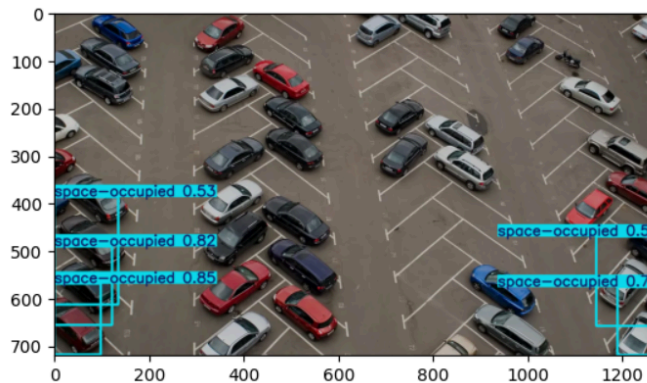


Imagen 8.- Imagen con Predicción

Comprensión

1. Mencione 2 arquitecturas populares utilizadas en detección de objetos.

YOLO (You Only Look Once) es una arquitectura de detección de objetos en tiempo real que trata la detección como un problema de regresión. Divide la imagen en una cuadrícula y predice directamente las cajas delimitadoras y las clases en una sola pasada por la red.

SSD (Single Shot MultiBox Detector) es otra arquitectura de detección rápida que detecta objetos en una sola pasada por la red pero con una estrategia diferente: utiliza múltiples mapas de características para detectar objetos de diferentes tamaños.

2. ¿Qué son las cajas delimitadoras? Y ¿Cómo se representan?

Una caja delimitadora es un rectángulo que encierra un objeto detectado en una imagen. Se utiliza en tareas de detección de objetos para indicar la ubicación y el tamaño de cada objeto dentro de la imagen.

Una caja delimitadora se representa generalmente con cuatro valores numéricos, que definen la posición y el tamaño del rectángulo. Hay dos formas comunes de representación:

- Por coordenadas de esquina (x_min, y_min, x_max, y_max)
- Por centro y dimensiones (x_center, y_center, width, height)

3. Explique las métricas se utilizan en detección de objetos

Métrica	¿Qué mide?	¿Por qué es útil?
IoU	Superposición entre cajas	Evalúa precisión espacial
Precisión	Exactitud de detecciones	Minimiza falsos positivos
Recall	Cobertura de detecciones	Minimiza objetos no detectados
mAP	Precisión promedio	Métrica estándar en benchmarks
F1-Score	Balance entre precisión y recall	Análisis equilibrado
FPS	Velocidad	Crucial en aplicaciones en tiempo real

4. ¿Qué es Ultralytics y qué soluciones de IA ofrece?

Es una plataforma y librería de código abierto basada en Python que permite entrenar, probar e implementar modelos de IA para tareas como, detección de objetos, segmentación de imágenes, clasificación de imágenes, seguimiento de objetos en video. Todo esto con una interfaz muy accesible y fácil de usar.

Soluciones que ofrece:

- **YOLOv5 y YOLOv8** (modelos de detección de objetos altamente optimizados).
- **Detección de objetos** (detecta y ubica múltiples objetos en una imagen con cajas delimitadoras y etiquetas).
- **Segmentación de instancias** (más precisa que la detección clásica: dibuja el contorno exacto de cada objeto, no sólo cajas).
- **Clasificación de imágenes** (asigna una etiqueta de clase a una imagen completa).
- **Seguimiento de objetos** (Realiza seguimiento continuo de objetos detectados en video).
- **Despliegue de modelos** (Deployment)

5. ¿Qué es Roboflow y qué soluciones de IA ofrece?

Roboflow es una plataforma completa basada en la nube que facilita el desarrollo de modelos de visión por computadora desde cero.

Soluciones que ofrece:

- Gestión y anotación de datasets
- Aumento de datos (Data Augmentation)
- Entrenamiento de modelos
- Despliegue de modelos (Deployment)

6. ¿Qué es Edge Impulse y qué soluciones de IA ofrece?

Edge Impulse es una plataforma especializada en el desarrollo de modelos de inteligencia artificial para dispositivos embebidos y de IoT.

Su objetivo es facilitar la creación, entrenamiento e implementación de modelos de aprendizaje automático on the Edge, directamente en microcontroladores, sensores, placas como ESP32, STM32, Arduino, Raspberry Pi, Jetson Nano, entre otros.

Soluciones que ofrece:

- Entrenamiento de modelos con sensores reales
- Clasificación y detección de movimiento
- Reconocimiento de audio y voz
- Visión por computadora en el borde
- Despliegue optimizado para hardware embebido
- Interfaz web fácil de usar más herramientas de autoentrenamiento

Conclusiones

La detección de objetos permite no solo identificar qué objetos hay en una imagen, sino también dónde están ubicados mediante el uso de cajas delimitadoras.

Gracias a arquitecturas avanzadas como YOLO, es posible realizar detecciones con alta precisión y velocidad, incluso en tiempo real y en dispositivos embebidos. Estas soluciones son útiles en, seguridad y monitoreo, automatización industrial, robótica, accesibilidad y asistencia y aplicaciones móviles inteligentes.

La detección de objetos impulsa el desarrollo de sistemas inteligentes capaces de entender su entorno visual, y sirve en la creación de soluciones innovadoras que conectan el mundo físico con la inteligencia artificial.