



Universidad Autónoma de San Luis Potosí
Facultad de ingeniería
Inteligencia Artificial Aplicada
Practica 9
Detección de objetos
Ana Sofía Medina Martínez



Fecha 16/11/2024

Objetivo

Que el alumno aprenda los conceptos principales de detección de objetos y a aprenda a utilizar y entrenar modelos de detección de objetos.

Procedimiento

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

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

Resultados

✖ Paso 2. Cargar el modelo en Python

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

```
[ ] from ultralytics import YOLO

#Creamos el modelo yolo v8 pre entrenado
model = YOLO("yolov8n.pt")
```

Creating new Ultralytics Settings v0.0.6 file ☒
View Ultralytics Settings with 'yolo settings' or at '/root/.config/Ultralytics/settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see <https://docs.ultralytics.com/quickstart/#ultralytics-settings>.
Downloading <https://github.com/ultralytics/assets/releases/download/v8.3.0/yolov8n.pt> to 'yolov8n.pt'...
100% |██████████| 6.25M/6.25M [00:00<00:00, 100MB/s]

✖ Paso 3. Entrenar el 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)
```

Epoch 2/5 GPU_mem 1.19G box_loss 1.418 cls_loss 1.233 dfl_loss 1.247 Instances 249 Size 640: 100% |██████████| 8/8 [02:09<00:00, 16.19s/it] mAP50 mAP50-95: 100% |██████████| 4/4 [00:48<00:00, 12.22s/it] all 128 929 0.659 0.
Class Images Instances Box(P) R
Epoch 3/5 GPU_mem 1.142G box_loss 1.378 cls_loss 1.247 dfl_loss 1.247 Instances 190 Size 640: 100% |██████████| 8/8 [02:06<00:00, 15.83s/it] mAP50 mAP50-95: 100% |██████████| 4/4 [00:42<00:00, 10.74s/it] all 128 929 0.687 0.
Class Images Instances Box(P) R
Epoch 4/5 GPU_mem 1.171G box_loss 1.381 cls_loss 1.231 dfl_loss 1.222 Instances 222 Size 640: 100% |██████████| 8/8 [02:08<00:00, 16.12s/it] mAP50 mAP50-95: 100% |██████████| 4/4 [00:46<00:00, 11.51s/it] all 128 929 0.678 0.
Class Images Instances Box(P) R

✖ Paso 3. Validar


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()
```

Ultralytics 8.3.28 Python-3.10.12 torch-2.5.0+cu121 CPU (Intel Xeon 2.20GHz)
Model summary (fused): 168 layers, 3,151,904 parameters, 0 gradients, 8.7 GFLOPs
val: Scanning /content/datasets/coco128/labels/train2017.cache... 126 images, 2 backgrounds, 0 corrupt: 100% |██████████| 128/128 [00:00<?, ?it/s]
Class Images Instances Box(P) R mAP50 mAP50-95: 100% |██████████| 8/8 [00:33<00:00, 4.15s/it]
all 128 929 0.683 0.587 0.648 0.483
person 61 254 0.805 0.681 0.778 0.547
bicycle 3 6 0.61 0.333 0.328 0.273
car 12 46 0.708 0.217 0.301 0.18
motorcycle 4 5 0.685 0.877 0.938 0.754
airplane 5 6 0.725 0.833 0.894 0.685
bus 5 7 0.678 0.714 0.735 0.644
train 3 3 0.553 0.667 0.83 0.781
truck 5 12 1 0.308 0.494 0.309
boat 2 6 0.397 0.333 0.428 0.238
traffic light 4 14 0.734 0.2 0.201 0.138
stop sign 2 2 1 0.99 0.995 0.717
bench 5 9 0.758 0.556 0.667 0.452
bird 2 16 1 0.804 0.967 0.665
cat 4 4 0.853 1 0.995 0.816
dog 9 9 0.666 0.888 0.862 0.63
horse 1 2 0.523 1 0.995 0.497
elephant 4 17 0.891 0.824 0.928 0.68
bear 1 1 0.611 1 0.995 0.995
zebra 2 4 0.858 1 0.995 0.965
giraffe 4 9 0.747 0.983 0.968 0.765
backpack 4 6 0.6 0.333 0.385 0.232
umbrella 4 18 0.714 0.554 0.681 0.47
handbag 9 19 0.912 0.105 0.224 0.128
tie 6 7 0.663 0.714 0.603 0.414
suitcase 2 4 0.619 1 0.849 0.593
frisbee 5 5 0.647 0.8 0.733 0.633
skis 1 1 0.736 1 0.995 0.497
snowboard 2 7 0.64 0.714 0.712 0.488
sports ball 6 6 0.598 0.333 0.448 0.245
kite 2 10 0.787 0.5 0.602 0.245

▼ Paso 4. Realizar predicciones con el modelo

```
[ ] from google.colab import files
    uploaded = files.upload()
```

 Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving aparcamientos-disuasorios-ciudades.webp to aparcamientos-disuasorios-ciudades.webp
Saving image_via_parking_industry.jpg to image_via_parking_industry.jpg
Saving parking-in-philadelphia.jpg to parking-in-philadelphia.jpg
Saving Parkings-640x321.jpg to Parkings-640x321.jpg

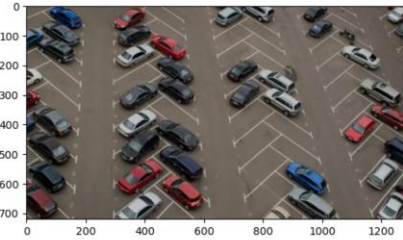
```
[ ] import cv2
import matplotlib.pyplot as plt

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

image = cv2.imread(image_path)

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

 <matplotlib.image.AxesImage at 0x7c414e543fa8>

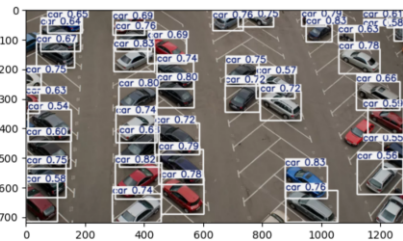


```
results = model.predict(source=image, conf=0.5)
results[0].save("result.png")

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

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

 0: 384x640 41 cars, 245.5ms
Speed: 6.8ms preprocess, 245.5ms inference, 8.7ms postprocess per image at shape (1, 3, 384, 640)
<matplotlib.image.AxesImage at 0x7c414e5f5180>



▼ Paso 6. Exportar el modelo

Ultralytics permite exportar el modelo para utilizarlo en distintas aplicaciones.

Puedes revisar más sobre como exportar el modelo en la documentación oficial.

▼ Paso 6. Exportar el modelo

Ultralytics permite exportar el modelo para utilizarlo en distintas aplicaciones.


Puedes revisar más sobre como exportar el modelo en la documentación oficial.

```
 #Exportar el modelo
model.export()
```

 Ultralytics 8.3.28 Python-3.10.12 torch-2.5.0+cu121 CPU (Intel Xeon 2.20GHz)

PyTorch: starting from 'runs/detect/train/weights/best.pt' with input shape (1, 3, 640, 640) BCHW and output shape(s) (1, 84, 8400) (6.2 MB)

TorchScript: starting export with torch 2.5.0+cu121...

TorchScript: export success  3.0s, saved as 'runs/detect/train/weights/best.torchscript' (12.5 MB)

Export complete (4.5s)

Results saved to /content/runs/detect/train/weights

Predict: yolo predict task=detect model=runs/detect/train/weights/best.torchscript imgsz=640

Validate: yolo val task=detect model=runs/detect/train/weights/best.torchscript imgsz=640 data=/usr/local/lib/python3.10/dist-packages/ultralytics/cfg/datasets/coco128.yaml

Visualize: <https://netron.app>

'runs/detect/train/weights/best.torchscript'

```
[ ] #Descargar el modelo
files.download("yolov8n.pt")
```

El notebook custom_objet_detector.ipynb no funcionó y por eso mismo no se colocan las capturas de pantalla.

Comprensión

- 1. Mencione 2 arquitecturas populares utilizadas en detección de objetos.**
YOLO (You Only Look Once)
Faster R-CNN
- 2. ¿Qué son las cajas delimitadoras? Y ¿Cómo se representan?**
Son rectángulos que rodean objetos detectados en imágenes para definir su posición y tamaño. Se representan mediante coordenadas (x, y, ancho, alto).
- 3. Explique las métricas se utilizan en detección de objetos**
Precisión: Proporción de predicciones correctas frente a las totales.
Recall (Sensibilidad): Capacidad de detectar objetos reales.
IoU (Intersección sobre Unión): Mide la superposición entre la predicción y el objeto real.
- 4. ¿Qué es Ultralytics y que soluciones de IA ofrece?**
Es una empresa que desarrolla herramientas para visión por computadora, como modelos optimizados de YOLO para detección de objetos y segmentación.
- 5. ¿Qué es Roboflow y que soluciones de IA ofrece?**
Es una plataforma que ayuda en el etiquetado, preparación y entrenamiento de modelos para visión por computadora. Ofrece flujos de trabajo completos para datos visuales.
- 6. ¿Qué es Edge Impulse y que soluciones de IA ofrece?**
Es una herramienta para diseñar y entrenar modelos de aprendizaje automático optimizados para dispositivos de borde (Edge AI), como sensores y cámaras.

Conclusiones

Las tecnologías actuales como YOLO, Faster R-CNN y herramientas como Ultralytics, Roboflow y Edge Impulse están impulsando la detección de objetos con enfoques innovadores. Estas soluciones permiten la creación de modelos efectivos, desde procesamiento en la nube hasta implementaciones en dispositivos pequeños, ampliando las posibilidades de uso en diferentes industrias.

