



Universidad Autónoma de San Luis Potosí
Facultad de ingeniería
Inteligencia Artificial Aplicada



Práctica: 11

Nombre Práctica: Segmentación de Objetos
y Estimación de Pose

Nombre del Alumno: Manuel Ramírez Galván

Fecha: 07/05/2025

Procedimiento

11.1.- Sigue las instrucciones del archivo "yolo_seg_app.ipynb" para entrenar un modelo de segmentación.

```
import cv2
import matplotlib.pyplot as plt

from ultralytics import YOLO
```

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

Imagen 1.- Importar Librerías y Modelo

```
# Cargar el modelo preentrenado para segmentación
model = YOLO('yolov8n-seg.pt')

# Configuración del entrenamiento
model.train(
    data='coco8-seg.yaml', # Archivo YAML con información del dataset
    epochs=2, # Reducido para propósitos de demostración
)
```

| | | | | |
|----------|----------|----------|----------|----------|
| 0.6967, | 0.6977, | 0.6987, | 0.6997, | 0.7007, |
| 0.70871, | 0.70971, | 0.71071, | 0.71171, | 0.71271, |
| 0.71371, | 0.72072, | 0.72172, | 0.72272, | 0.72372, |
| 0.72472, | 0.73273, | 0.73373, | 0.73473, | 0.73574, |
| 0.73674, | 0.73774, | 0.74474, | 0.74575, | 0.74675, |
| 0.74775, | 0.74875, | 0.75676, | 0.75776, | 0.75876, |
| 0.75976, | 0.76076, | 0.76176, | 0.76877, | 0.76977, |
| 0.77077, | 0.77177, | 0.77277, | 0.78178, | 0.78278, |
| 0.78378, | 0.78478, | 0.78579, | 0.79279, | 0.79379, |
| 0.79479, | 0.79578, | 0.79678, | 0.8048, | 0.80581, |
| 0.80681, | 0.80781, | 0.80881, | 0.81682, | 0.81782, |
| 0.81882, | 0.81982, | 0.82082, | 0.82883, | 0.82983, |
| 0.83083, | 0.83183, | 0.83283, | 0.83383, | 0.84084, |
| 0.84184, | 0.84284, | 0.84384, | 0.84484, | 0.85285, |
| 0.85385, | 0.85485, | 0.85586, | 0.85686, | 0.85786, |
| 0.86486, | 0.86587, | 0.86687, | 0.86787, | 0.86887, |
| 0.87688, | 0.87788, | 0.87888, | 0.87988, | 0.88088, |
| 0.88188, | 0.88889, | 0.88989, | 0.89089, | 0.89189, |
| 0.89289, | 0.9009, | 0.9019, | 0.9029, | 0.9039, |
| 0.9049, | 0.90591, | 0.91291, | 0.91391, | 0.91491, |
| 0.91592, | 0.91692, | 0.92492, | 0.92593, | 0.92693, |
| 0.92793, | 0.92893, | 0.92993, | 0.93694, | 0.93794, |
| 0.93894, | 0.93994, | 0.94094, | 0.94895, | 0.94995, |
| 0.95095, | 0.95195, | 0.95295, | 0.95395, | |

Imagen 2.- Cargar y Entrenar Modelo

```
results = model.predict(source='datasets/coco8-seg/images/val/000000000042.jpg') #Puedes cargar tu propia imagen
```

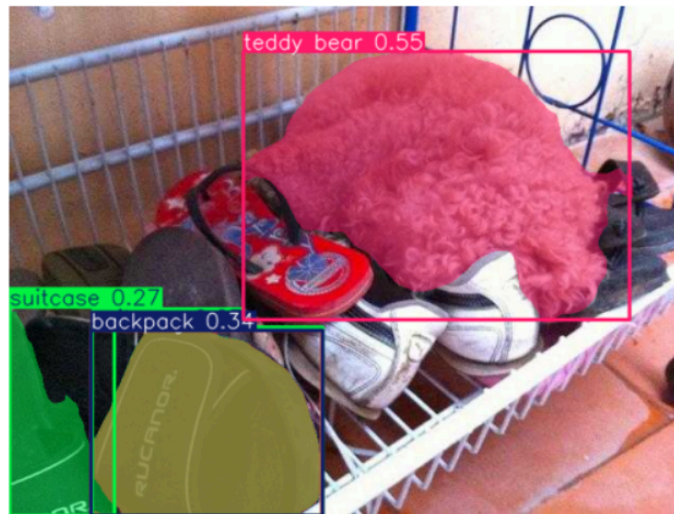
image 1/1 /content/datasets/coco8-seg/images/val/000000000042.jpg: 480x640 1 backpack, 1 handbag, 2 suitcases, 1 teddy bear, 737.5ms
Speed: 14.1ms preprocess, 737.5ms inference, 60.5ms postprocess per image at shape (1, 3, 480, 640)

Imagen 3.- Cargar Imagen

```
for result in results:  
    annotated_frame = result.plot()  
    plt.imshow(annotated_frame[:, ::-1])  
    plt.axis('off')  
    plt.show()
```

Imagen 4.- Mostrar Imagen con Predicción

Resultados



Imágen 5.- Imagen con Predicción

Procedimiento

11.2.- Sigue las instrucciones del archivo “yolo_pose_app.ipynb” para entrenar un modelo de estimación de pose.

```
import cv2  
import matplotlib.pyplot as plt  
  
from ultralytics import YOLO
```


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

Imagen 9.- Descarga de Dataset

```
# Cargar el modelo preentrenado para segmentación
model = YOLO('yolov8n-pose.pt')

# Configuración del entrenamiento
model.train(
    data='coco8-pose.yaml', # Archivo YAML con información del dataset
    epochs=2, # Reducido para propósitos de demostración
)
```

Downloading <https://github.com/ultralytics/assets/releases/download/v8.3.0/yolov8n-pose.pt> to 'yolov8n-pose.pt'...
100%|██████████| 6.52M/6.52M [00:00<00:00, 86.4MB/s]
Ultralytics 8.3.128 Python-3.11.12 torch-2.6.0+cu124 CPU (Intel Xeon 2.20GHz)
engine/trainer: agnostic_nms=False, amp=True, augment=False, auto_augment=randaugument, batch=16, bgr=0.0, box=7.5,
WARNING ⚠ Dataset 'coco8-pose.yaml' images not found, missing path '/content/datasets/coco8-pose/images/val'
Downloading <https://ultralytics.com/assets/coco8-pose.zip> to '/content/datasets/coco8-pose.zip'...
100%|██████████| 334k/334k [00:00<00:00, 15.0MB/s]
Unzipping /content/datasets/coco8-pose.zip to /content/datasets/coco8-pose...: 100%|██████████| 27/27 [00:00<00:00,

Downloading <https://ultralytics.com/assets/Arial.ttf> to '/root/.config/Ultralytics/Arial.ttf'...
100%|██████████| 755k/755k [00:00<00:00, 19.9MB/s]

| | from | n | params | module | arguments |
|---|------|---|--------|----------------------------------|---------------------|
| 0 | -1 | 1 | 464 | ultralytics.nn.modules.conv.Conv | [3, 16, 3, 2] |
| 1 | -1 | 1 | 4672 | ultralytics.nn.modules.conv.Conv | [16, 32, 3, 2] |
| 2 | -1 | 1 | 7360 | ultralytics.nn.modules.block.C2f | [32, 32, 1, True] |
| 3 | -1 | 1 | 18560 | ultralytics.nn.modules.conv.Conv | [32, 64, 3, 2] |
| 4 | -1 | 2 | 49664 | ultralytics.nn.modules.block.C2f | [64, 64, 2, True] |
| 5 | -1 | 1 | 73984 | ultralytics.nn.modules.conv.Conv | [64, 128, 3, 2] |
| 6 | -1 | 2 | 197632 | ultralytics.nn.modules.block.C2f | [128, 128, 2, True] |
| 7 | -1 | 1 | 295424 | ultralytics.nn.modules.conv.Conv | [128, 256, 3, 2] |

Imagen 10.- Cargar y Entrenar Modelo

```
results = model.predict(source='datasets/coco8-pose/images/val/000000000110.jpg') #Puedes cargar tu propia imagen
```

image 1/1 /content/datasets/coco8-pose/images/val/000000000110.jpg: 480x640 8 persons, 327.6ms
Speed: 3.7ms preprocess, 327.6ms inference, 1.8ms postprocess per image at shape (1, 3, 480, 640)

Imagen 11.- Cargar Imagen

Resultados

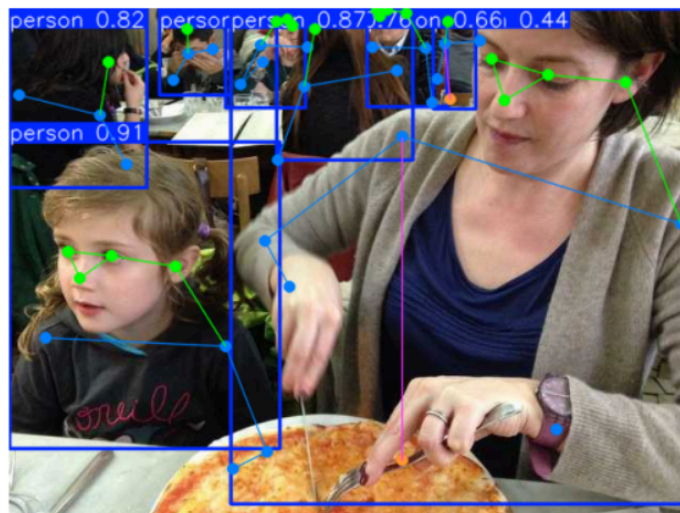


Imagen 17.- Imágen

Comprensión

1. ¿Cuáles son los 3 tipos de segmentación y en qué consisten cada uno?

Segmentación semántica

Clasifica cada píxel de la imagen según su clase (por ejemplo: persona, auto, árbol). No distingue entre diferentes instancias del mismo objeto (todas las personas se ven como una sola clase).

Segmentación por instancia

Similar a la semántica, pero además distingue cada objeto individualmente aunque sea de la misma clase.

Segmentación panóptica

Combina la segmentación semántica y la segmentación por instancia. Asigna una clase y un ID único a cada píxel.

2. ¿En qué aplicaciones es recomendable utilizar modelos de segmentación?

- **Inspección industrial:** Detectar defectos o formas irregulares en piezas.
- **Conducción autónoma:** Diferenciar carriles, peatones, autos y señales.
- **Medicina:** Delimitar tumores u órganos en imágenes médicas.
- **Reconocimiento de objetos complejos:** Separar objetos que están muy juntos o parcialmente ocultos.
- **Agricultura de precisión:** Contar y segmentar frutas, hojas o cultivos.

3. ¿En qué aplicaciones es recomendable utilizar modelos de estimación de pose?

- **Reconocimiento de gestos y lenguaje de señas:** Detecta posiciones de manos, brazos y dedos.
- **Seguimiento de actividad física o ejercicio:** Analiza posturas y movimientos.
- **Control por movimiento:** Juegos o interfaces controladas por el cuerpo.
- **Interacción hombre-robot:** Permite a los robots adaptarse a la postura humana.
- **Análisis de biomecánica:** Estudia el movimiento en deportes, rehabilitación o ergonomía.

Conclusiones

La segmentación de objetos permite identificar la forma y los límites de los objetos en una imagen, siendo útil para tareas donde se requiere un entendimiento visual detallado, como inspección industrial, medicina o conducción autónoma.

La estimación de pose se enfoca en localizar puntos clave del cuerpo u objetos, facilitando aplicaciones relacionadas con reconocimiento de gestos, actividad física, interfaces por movimiento y análisis biomecánico.

Ambas técnicas son aplicables en la visión por computadora y amplían significativamente las capacidades de los sistemas inteligentes para comprender e interactuar con su entorno de forma precisa y dinámica.