La tarea consiste en conocer y usar el modelo de detección de objetos YOLO (You Only Look Once). Para ello nos guiaremos de los dos documentos PDF subidos a la plataforma.

Toda la documentación está disponible en: https://docs.ultralytics.com/es/

(3 Puntos) **1 Script de iniciación: (YOLO 1.pdf)**

Instalamos ultralytics y con el fichero yolov8n lo ponemos en nuestro entorno de trabajo.

1.1 Desacargar archivo Yolov8n → https://docs.ultralytics.com/models/yolov8/#performance-metrics

1.2 Abrir visual Code Studio e instalar Ultralytics.

1.3 Copiamos el código y lo ejecutamos.

```
from ultralytics import YOLO
# Instancia del modelo YOLO con los pesos especificados
pesos_yolo = "C:/Users/Usuario/Desktop/YoloV8/yolov8n.pt"
model = YOLO(pesos_yolo)
try:
        # Realiza la detección de objetos en el flujo de video de la cámara del computador
         results = model(source=0, show=True, conf=0.3, save=True)
except Exception as e:
        print("Ocurrió un error al realizar la detección de objetos:", e)
```

(1 Punto) **2 Script de traking:**

https://docs.ultralytics.com/es/modes/track/
https://docs.ultralytics.com/es/datasets/detect/coco/#sample-imagesand-annotations

```
from ultralytics import YOLO
# Configure the tracking parameters and run the tracker
model = YOLO('yolov8n.pt')
results = model.track(source="0", conf=0.3, iou=0.5, show=True)
```

El script muestra el rastreo de una persona, piensa que ha sido entrenado con la lista Coco.txt.

2.1 Modifica el script para que rastrease otro objeto de dicha lista.

(6 Puntos) **3 Script entrenado el modelo. (YOLO 2.pdf)**

3.1 Preparando el Dataset.
3.2 CVAT (Computer Vision Annotation Tool)
3.3 Preparando los archivos de imagen .txt
3.4 Creamos nuevo proyecto (etiquetar y exportar Dataset).
3.5 Entrenado el modelo en Google Colab.
3.6 Predicción.

```
import cv2
from ultralytics import YOLO
import numpy as np
# Cargar el modelo YOLO preentrenado
model = YOLO("C:/Users/Usuario/Desktop/YoloV8/4.Predict/last_Naranjas.pt")
# Realizar la predicción en una imagen
results = model("C:/Users/Usuario/Desktop/YoloV8/4.Predict/imagen3.jpg", conf=0.3,iou=0.5)

# Iterar sobre cada resultado en la lista
for result in results:
        # Obtener las coordenadas de los bounding boxes
        boxes = result.boxes[0].xyxy
         # Obtener el diccionario de nombres de clases
        names_dict = result.namesç
        # Cargar la imagen
        image = cv2.imread("C:/Users/Usuario/Desktop/YoloV8/4.Predict/imagen3.jpg")
        # Dibujar los bounding boxes en la imagen
        for box in boxes:
                print("Box:", box) # Imprimir el contenido de box
                xmin, ymin, xmax, ymax = box[:4] # Obtener las coordenadas del bounding box
                xmin, ymin, xmax, ymax = int(xmin), int(ymin), int(xmax), int(ymax)
                 cv2.rectangle(image, (xmin, ymin), (xmax, ymax), (0,255,0), 2)
        # Mostrar la imagen
         cv2.imshow('Image', image)
         cv2.waitKey(0)
        cv2.destroyAllWindows()
```

3.7 Hacer un script similar, mostrando todos los bounding box y sus etiquetas.

https://www.youtube.com/watch?v=m9fH9OWn8YM&ab_channel=Computervisionengineer
https://www.youtube.com/watch?v=ZeLg5rxLGLg&list=RDCMUCLALGa2fSkx0MI88eALnILw&index=2&ab_channel=Computervisionengineer

---

Se subirá un único archivo PDF con índice a cada apartado, las capturas de ejecución y código fuente, junto con las explicaciones de lo realizado.