DETECCIÓN DE OBJETOS CON YOLO



Adrián Yared Armas de la Nuez

**Contenido**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

[**1. Objetivo 2**](#_2c35568x6wg8)

[**2. Actividad 2**](#_gdnmctyhxy83)

[**2.1 Script de iniciación: 2**](#_2eety28995jz)

[**2.1.1 Descarga de YOLO 2**](#_xf6uuy6tn37)

[**2.1.2 Instalo ultralytics 2**](#_aig92oe7k2m0)

[**2.1.2 Main.py 2**](#_e9uqlcl56mu)

[**2.1.2.1 Código 2**](#_s6g3v9ondidv)

[**2.1.2.2 Ejecución 3**](#_fj7wtj3vkkwc)

[**2.1.2.3 Prueba del programa 4**](#_okhf5lw065hg)

[**3. Tracking 4**](#_lj5qufa75x5p)

[**3.1.1 Código 4**](#_yprvf3yz6py5)

[**3.1.2 Ejecución (personas) 5**](#_ac17ks12usau)

[**4. Script entrenando el modelo 5**](#_8ea6zwdq3yj2)

[**4.1.1 Preparando el Dataset. 5**](#_lfrjoo6kbsp9)

[**4.1.2 CVAT (Computer Vision Annotation Tool) 6**](#_5j87vhy80f8t)

[**4.1.3 Creamos nuevo proyecto (etiquetar y exportar Dataset). 6**](#_bryl1pmi3sx1)

[**4.1.4 Preparando los archivos de imagen .txt 8**](#_y3gltqyhg8rk)

[**4.1.5 Entrenado el modelo en Google Colab. 9**](#_pzci0ky9erdg)

[**4.1.5.0 Recoger el proyecto de drive 9**](#_zcv3tbwmnq74)

[**4.1.5.1 Código 9**](#_udu4t6ytnt83)

[**4.1.5.2 Ejecución 10**](#_uyzrbchutbot)

[**4.1.6 Predicción. 10**](#_xc9w1uw3dcfg)

[**5. Bibliografía 12**](#_gdswomysh4g7)

[**6. Github y Colab 13**](#_lmdngf8dlin3)

## 

## 

## 

## **1. Objetivo**

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.

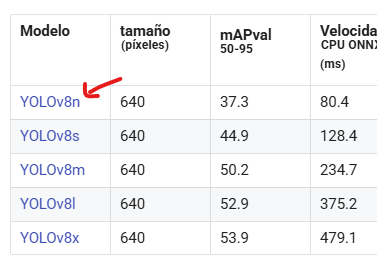
Toda la documentación está disponible en: <https://docs.ultralytics.com/es/>

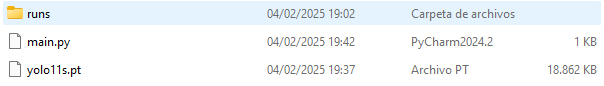
## **2. Actividad**

### **2.1 Script de iniciación:**

#### **2.1.1 Descarga de YOLO**

<https://docs.ultralytics.com/es/models/yolo11/#performance-metrics>

Descargo la versión:  


Añado el archivo al entorno del programa:  


#### **2.1.2 Instalo ultralytics**



#### **2.1.2 Main.py**

##### **2.1.2.1 Código**

from ultralytics import YOLO

def initial():

try:

# Instancia del modelo YOLO con los pesos especificados

model = YOLO("yolov11s.pt") # Asegúrate de que el nombre del archivo de pesos sea correcto

# Realiza la detección de objetos en el flujo de video de la cámara del computador

model.predict(source=0, show=True, conf=0.3, save=True)

except Exception as e:

print("Ocurrió un error al realizar la detección de objetos:", e)

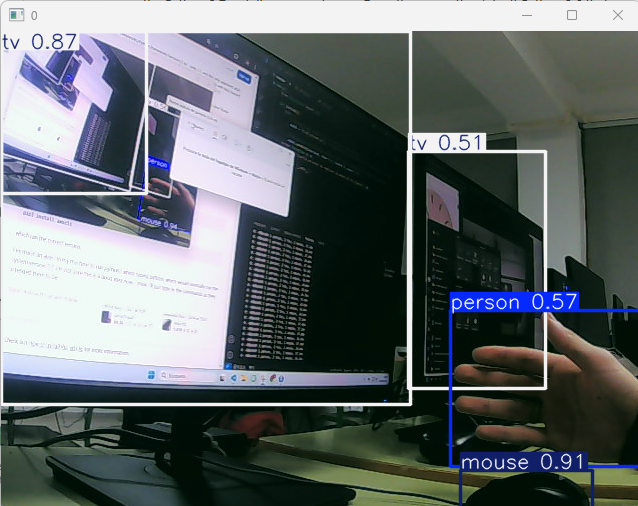
if \_\_name\_\_ == "\_\_main\_\_":

initial()

##### **2.1.2.2 Ejecución**



##### **2.1.2.3 Prueba del programa**



## **3. Tracking**

<https://docs.ultralytics.com/es/modes/track/>

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

#### **3.1.1 Código**

En vez de iniciar initial() inicio tracking() para que detecte **personas**, ya que es lo másfacil que considero tener a mano

from ultralytics import YOLO

import os

os.environ["KMP\_DUPLICATE\_LIB\_OK"] = "TRUE" # crash duplication error solution

model = YOLO("yolo11s.pt")

def initial():

try:

model.predict(source=0, show=True, conf=0.3, save=True)

except Exception as e:

print("Ocurrió un error al realizar la detección de objetos:", e)

def tracking():

try:

results = model.track(source="0", conf=0.3, iou=0.5, show=True, classes=[0]) #id 2 = car

except Exception as e:

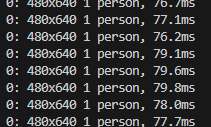
print(e)

if \_\_name\_\_ == "\_\_main\_\_":

#initial()

tracking()

#### **3.1.2 Ejecución (personas)**





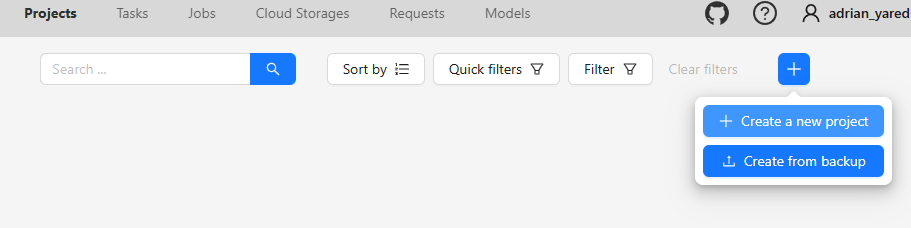
## **4. Script entrenando el modelo**

#### **4.1.1 Preparando el Dataset.**

Descargo el dataset de bolígrafos de:  
<https://universe.roboflow.com/yolov5-h3oo1/pen-zyj9j/dataset/1>

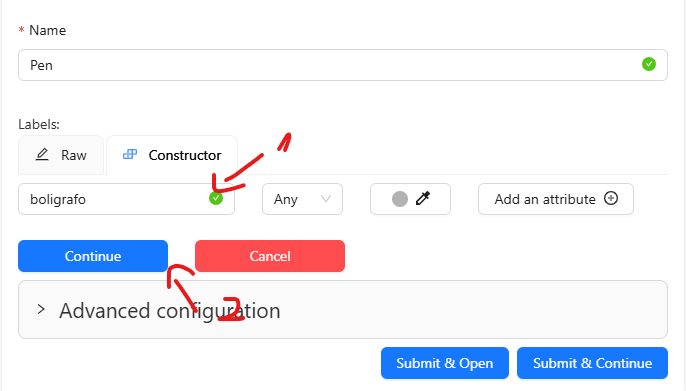
#### **4.1.2 CVAT (Computer Vision Annotation Tool)**

Me registro en <https://app.cvat.ai/>

Y creo un nuevo proyecto:  


#### **4.1.3 Creamos nuevo proyecto (etiquetar y exportar Dataset).**

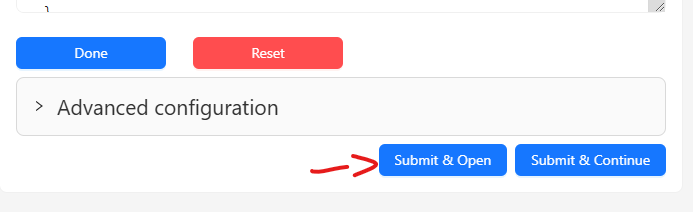
Creo el proyecto y le añado la clase pen:

****

Resultado:

****

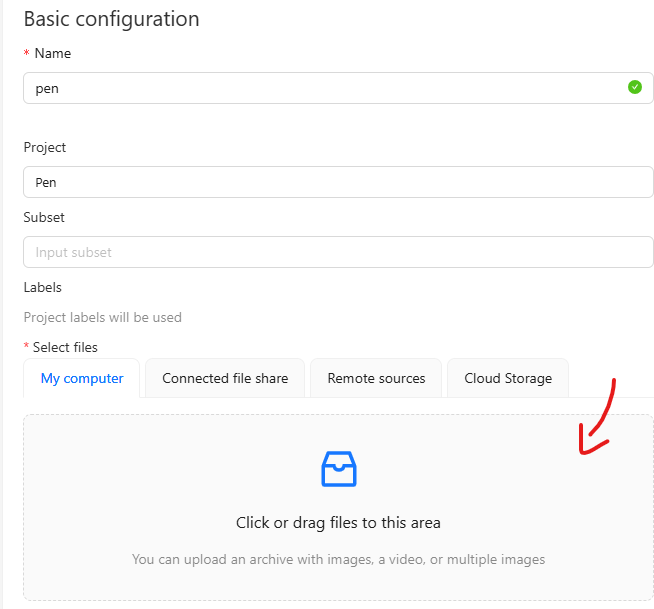
Voy al siguiente apartado:

****

Creo una task:

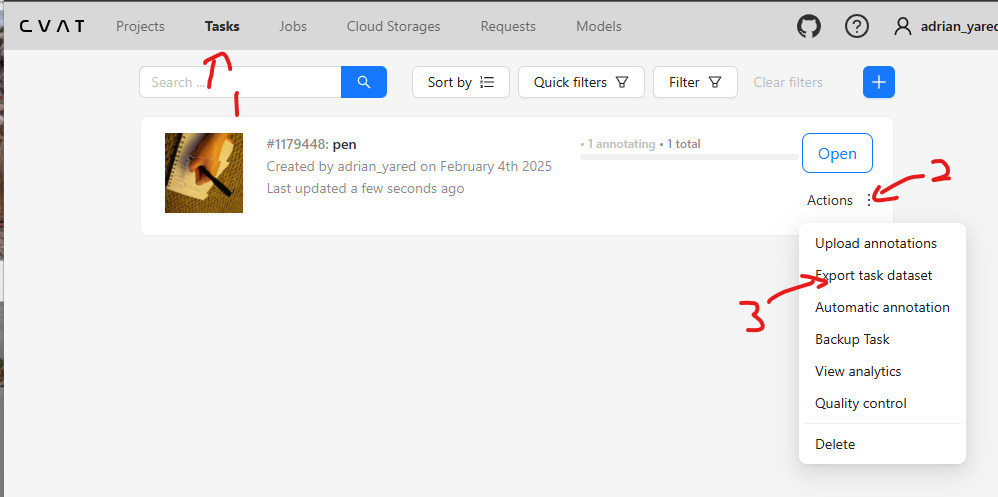
****

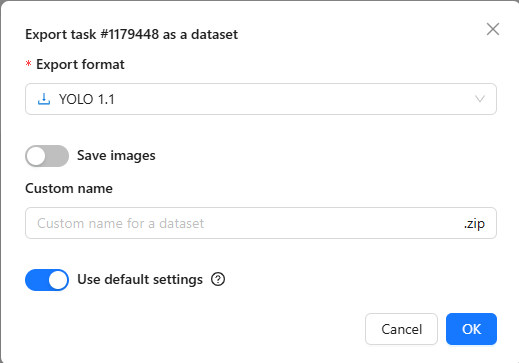
Subo todas las imagenes:

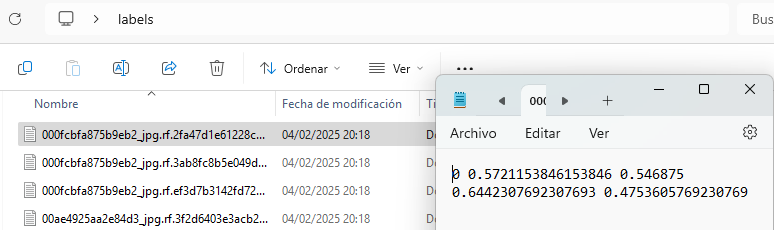


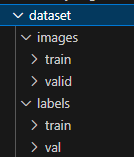
Procedo a la limpieza de datos, yendo imagen por imagen y seleccionando el contorno de los bolígrafos:  


#### **4.1.4 Preparando los archivos de imagen .txt**

Descarga de la task de imagenes a txt, tras la limpieza de imagenes:  


Lo guardo en modo Yolo1.1:  


Labels descargados:  


Estructura de carpetas:  


Estructura de carpetas en el archivo .yaml

train: /content/Yolo/dataset/images/train

val: /content/Yolo/dataset/images/val

nc: 1

names: ['Pen']

roboflow:

workspace: yolov5-h3oo1

project: pen-zyj9j

version: 1

license: CC BY 4.0

url: https://universe.roboflow.com/yolov5-h3oo1/pen-zyj9j/dataset/1

#### **4.1.5 Entrenado el modelo en Google Colab.**

##### **4.1.5.0 Recoger el proyecto de drive**

from google.colab import drive

drive.mount('/content/drive')

!pip install gdown

!gdown --fuzzy "https://drive.google.com/uc?id=1iLsgaOKUwRtI0vnDostX9cYHHwkZ1v8p"

import zipfile

import os

file\_path = "/content/Yolo.zip" # Cambia esto si el nombre es diferente

extract\_path = "/content" # Carpeta donde se extraerán los archivos

with zipfile.ZipFile(file\_path, 'r') as zip\_ref:

zip\_ref.extractall(extract\_path)

print("Archivo descomprimido en:", extract\_path)

!pip install ultralytics

##### **4.1.5.1 Código**

from ultralytics import YOLO

# Cargar el modelo YOLO preentrenado

model = YOLO('/content/Yolo/yolo11s.pt')

# Ruta del archivo de configuración YAML

data\_yaml = "/content/Yolo/dataset/data.yaml" # Asegúrate de que el archivo existe y tiene la estructura correcta

# Entrenar el modelo

model.train(

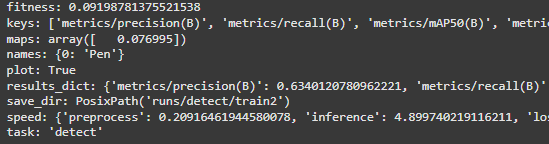
data=data\_yaml,

epochs=50,

imgsz=640,

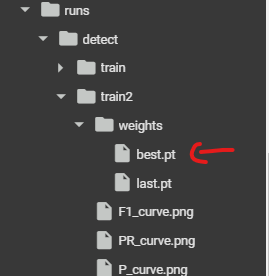
)

##### **4.1.5.2 Ejecución**

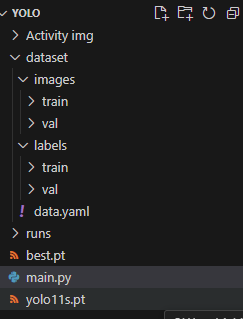


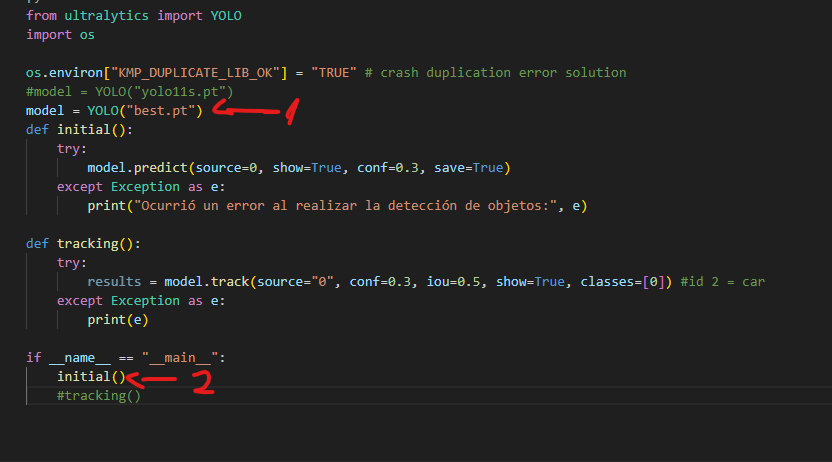
#### **4.1.6 Predicción.**

Descargo el archivo best.pt para usarlo en mi predicción local



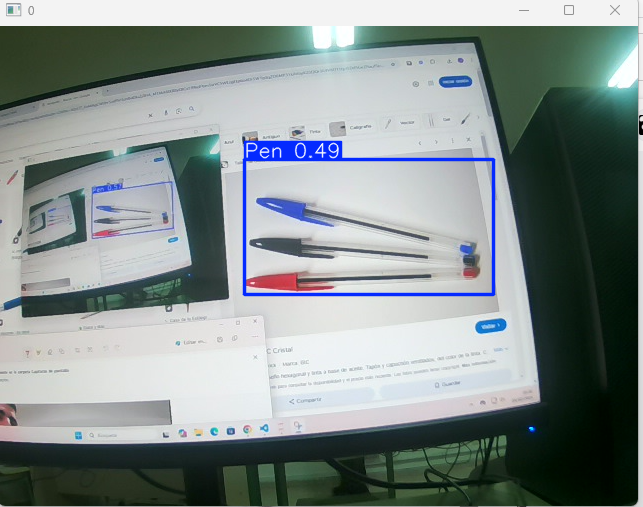
Lo añado a la carpeta de entorno de trabajo local:



Digo que utilice mi archivo e initial:  


**Pruebas de predicción:**  
Otros labels:  


Bolígrafos:

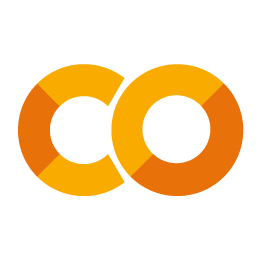


## **5. Bibliografía**

He utilizado una herramienta comunitaria al estilo de kaggle que contiene datasets preparados para entrenar con yolo

<https://universe.roboflow.com/yolov5-h3oo1/pen-zyj9j/dataset/1>

## **6. Github y Colab**

[](https://colab.research.google.com/drive/1xCwKneDzhzrQ_IdjtYmXbG0_QO3wZgRk?usp=sharing)