

Práctica 2

Detectar objetos con YOLOv5

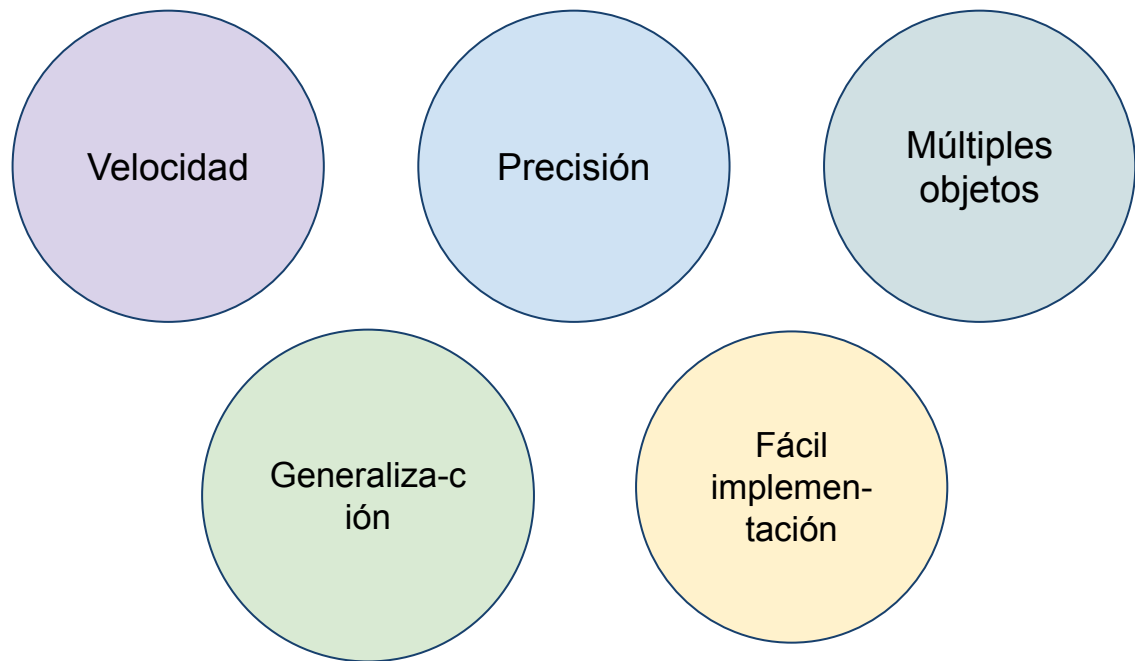


Profesor:
David Campoy Miñarro



¿Qué es YOLOv5?

YOLO, acrónimo de "*You Only Look Once*" (Solo Mira una Vez), es un algoritmo de detección de objetos en imágenes y videos que revolucionó la forma de detectar objetos en el campo del procesamiento de imágenes y visión por ordenador.



Es fundamental utilizar esta tecnología con responsabilidad y tener en cuenta sus implicaciones éticas y legales.



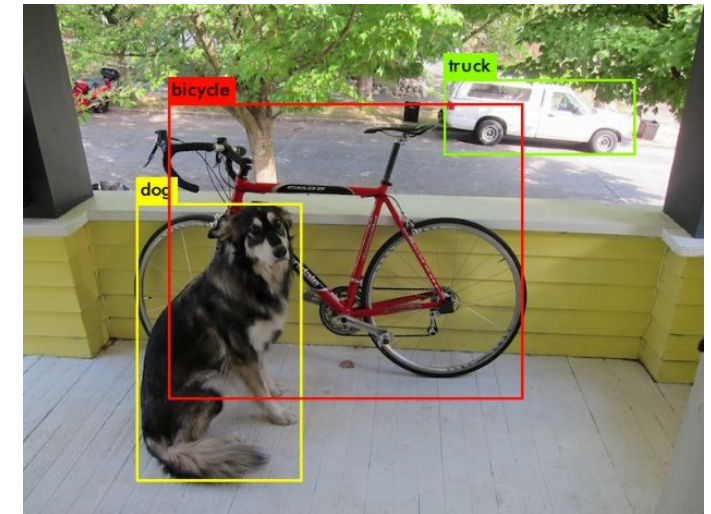
YOLO

YOLO es una biblioteca de Python desarrollada para tareas de reconocimiento de objetos:

<https://github.com/ultralytics/yolov5>

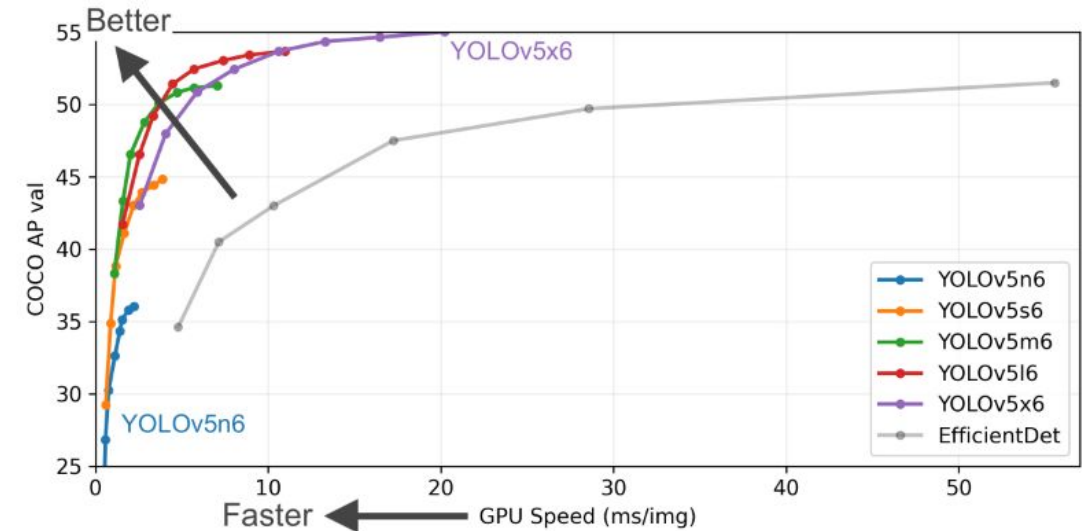
Licencia AGPL-3.0: Esta licencia de código abierto aprobada por OSI es ideal para estudiantes.

Licencia Enterprise: Diseñada para uso comercial, esta licencia permite la integración sin problemas del software y modelos de IA de Ultralytics en bienes y servicios comerciales, evitando los requisitos de código abierto de la licencia AGPL-3.0.



Redes entrenadas de YOLO

Model	size (pixels)	mAP ^{val} 50-95	mAP ^{val} 50	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	45	6.3	0.6	1.9	4.5
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7
YOLOv5n6	1280	36.0	54.4	153	8.1	2.1	3.2	4.6
YOLOv5s6	1280	44.8	63.7	385	8.2	3.6	12.6	16.8
YOLOv5m6	1280	51.3	69.3	887	11.1	6.8	35.7	50.0
YOLOv5l6	1280	53.7	71.3	1784	15.8	10.5	76.8	111.4
YOLOv5x6	1280	55.0	72.7	3136	26.2	19.4	140.7	209.8
+ TTA	1536	55.8	72.7	-	-	-	-	-



Comando YOLO

```
python detect.py --weights yolov5s.pt --img 640 --conf 0.25 --source data/images
```

```
python detect.py --weights yolov5s.pt --source 0                # webcam
                                     img.jpg                     # image
                                     vid.mp4                    # video
                                     screen                      # screenshot
                                     path/                      # directory
                                     list.txt                   # list of images
                                     list.streams              # list of streams
                                     'path/*.jpg'              # glob
                                     'https://youtu.be/LNwODJXcvt4' # YouTube
                                     'rtsp://example.com/media.mp4' # RTSP, RTMP, HTTP stream
```


Paso 1

Vamos a instalar la librería YOLO para que detecte objetos en tiempo real capturando vídeo de la webcam del ordenador.



Esta actividad no funciona en Google Colab, necesitarás probar el código en tu propio ordenador. ¿Por qué?

```
•[14]: !git clone https://github.com/ultralytics/yolov5
      %cd yolov5
      %pip install -qr requirements.txt comet_ml
```



Es posible que necesites actualizar la librería CV2, si no detecta la webman deberás actualizar la librería CV2 con:

```
!pip install opencv-python --upgrade
import cv2
```

Paso 2

Lo primero es detectar en qué dispositivo de entrada tenemos conectada la cámara web.

```
for i in range(10):  
    cap = cv2.VideoCapture(i)  
    if not cap.isOpened():  
        break  
    cap.release()  
    print(f"La webcam {i} está disponible.")
```

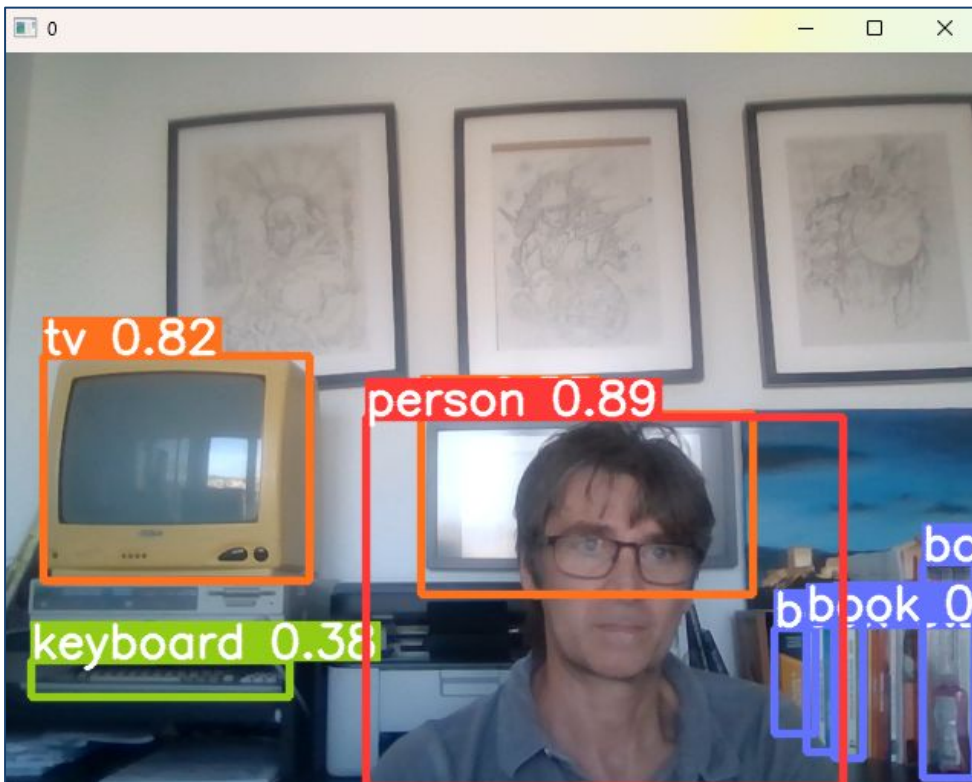
indica que tengo una webcam en la entrada cero.

La webcam 0 está disponible.

Paso 3

Comenzamos el reconocimiento de objetos a través de la webcam.

```
!python ./detect.py --source 0
```



Realiza distintas pruebas con otros modelos entrenados:

```
!python ./detect.py --weights yolov5s.pt --source 0
```

Model	size (pixels)	mAP ^{val} 50-95	mAP ^{val} 50	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	45	6.3	0.6	1.9	4.5
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1




```
#!/git clone https://github.com/ultralytics/yolov5
%cd yolov5
#%pip install -qr requirements.txt comet_ml
#!pip install opencv-python --upgrade
import cv2

for i in range(10):
    cap = cv2.VideoCapture(i)
    if not cap.isOpened():
        break
    cap.release()
    print(f"La webcam {i} está disponible.")

!python ./detect.py --weights yolov5s.pt --source 0
```

Paso 4

Ahora vamos a reconocer los objetos de una imagen estática.
Para esta actividad puedes utilizar tanto *Google Colab* como Jupyter Notebook.

```
▶ #!git clone https://github.com/ultralytics/yolov5 # clone
  #%cd yolov5
  #%pip install -qr requirements.txt comet_ml # install

import torch
import utils
display = utils.notebook_init()

!python detect.py --weights yolov5s.pt --img 640 --conf 0.25 --source data/images


display.Image(filename='runs/detect/exp/zidane.jpg', width=600)
```

source data/images Va a procesar todas las imágenes que coloques en data/images

El resultado del reconocimiento se guardará como imagen en “runs/detect/exp/”



Ahora detecta los objetos de un vídeo de Youtube:

```
 #!git clone https://github.com/ultralytics/yolov5 # clone
#%cd yolov5
#%pip install -qr requirements.txt comet_ml # install

import torch
import utils
display = utils.notebook_init()

!python detect.py --weights yolov5s.pt --img 640 --conf 0.25 --source "https://youtube.com/watch?v=82Ab-0Xi3FQ"
```

```
0: 384x640 2 persons, 1 pizza, 2 books, 263.7ms
0: 384x640 3 persons, 1 cell phone, 1 book, 267.4ms
0: 384x640 3 persons, 1 cell phone, 1 book, 381.0ms
0: 384x640 3 persons, 1 cell phone, 1 book, 428.9ms
0: 384x640 3 persons, 1 pizza, 1 cell phone, 1 book, 438.1ms
0: 384x640 3 persons, 1 pizza, 1 cell phone, 1 book, 410.0ms
0: 384x640 3 persons, 1 pizza, 1 cell phone, 425.8ms
0: 384x640 2 persons, 1 dining table, 1 cell phone, 1 book, 431.2ms
0: 384x640 3 persons, 1 donut, 1 laptop, 1 cell phone, 452.6ms
0: 384x640 2 persons, 1 donut, 1 dining table, 1 cell phone, 430.1ms
0: 384x640 2 persons, 1 dining table, 1 cell phone, 2 books, 430.7ms
0: 384x640 3 persons, 1 cell phone, 1 book, 410.7ms
0: 384x640 3 persons, 1 laptop, 1 cell phone, 1 book, 430.9ms
0: 384x640 3 persons, 1 pizza, 1 dining table, 2 books, 405.9ms
0: 384x640 3 persons, 1 pizza, 1 dining table, 2 books, 416.3ms
0: 384x640 3 persons, 1 pizza, 1 dining table, 2 books, 421.1ms
```



“Como tecnólogo, veo cómo la IA y la cuarta revolución industrial afectarán todos los aspectos de la vida de las personas”.

Fei-Fei Li, profesora de ciencias de la computación en la Universidad de Stanford.

