

::: Detección de objetos con YOLO en Raspberry Pi :::

By Freddy Alcarazo

19 de Abril del 2025 – Perú

¿Qué es YOLO?

You Only Look Once (YOLO) es un algoritmo de detección de objetos en tiempo real de última generación introducido en 2015 por Joseph Redmon, Santosh Divvala, Ross Girshick y Ali Farhadi en su famoso trabajo de investigación "You Only Look Once: Detección de Objetos Unificada y en Tiempo Real".

Los autores enmarcan el problema de la detección de objetos como un problema de regresión en lugar de una tarea de clasificación, separando espacialmente las cajas delimitadoras y asociando probabilidades a cada una de las imágenes detectadas mediante una única red neuronal convolucional (CNN).

Pasos para ejecutar Yolo en la Raspberry Pi:

1. Actualizar sistema

```
$ sudo apt update && sudo apt upgrade -y
```

2. Crear carpeta "yolo"

3. Moverse a la carpeta "yolo"

4. Crear un environment llamado "venv" dentro de la carpeta "yolo"

```
# Crear el environment:
```

```
python3 -m venv --system-site-packages venv
```

```
# Activar el environment:
```

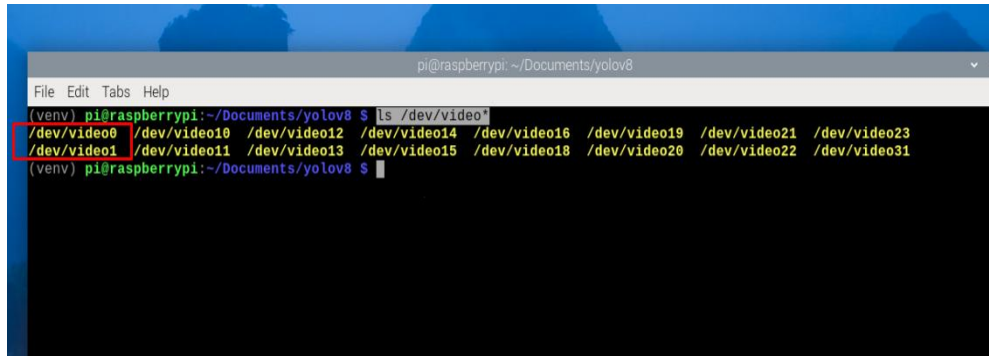
```
source venv/bin/activate
```

5. Instalar dependencias

```
$ pip install -r requirements.txt
```

6. Conectar webcam
7. Verificar la lista de webcams

```
$ ls /dev/video*
```



```
pi@raspberrypi: ~/Documents/yolov8
File Edit Tabs Help
(venv) pi@raspberrypi:~/Documents/yolov8 $ ls /dev/video*
/dev/video0 /dev/video10 /dev/video12 /dev/video14 /dev/video16 /dev/video19 /dev/video21 /dev/video23
/dev/video1 /dev/video11 /dev/video13 /dev/video15 /dev/video18 /dev/video20 /dev/video22 /dev/video31
(venv) pi@raspberrypi:~/Documents/yolov8 $
```

8. Configurar el Modelo Yolo

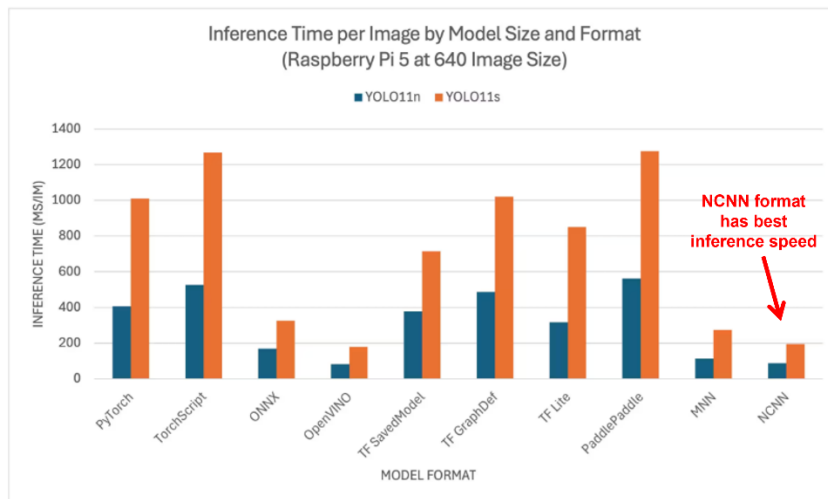
Descargue un modelo YOLO listo para usar (off-the-shelf YOLO model)

Para ello, usaremos Ultralytics el cual facilita la descarga y el uso de un modelo YOLO estándar. Estos modelos se entrenan con el conjunto de datos COCO y pueden detectar 80 objetos comunes, como "persona", "coche", "silla", etc. Para descargar un modelo de detección YOLO11n, ejecute:

```
yolo detect predict model=yolo11n.pt
```

Esto descargará automáticamente un archivo de modelo "yolo11n.pt" y lo ejecutará en un par de imágenes de prueba. Si deseamos usar un modelo diferente, simplemente reemplazamos "yolo11n.pt" por otro modelo, como "yolo11s.pt" o "yolov8n.pt".

9. Exportar el modelo YOLO al formato NCNN



Reemplazar la palabra `your_model.pt` por el nombre del modelo.

```
yolo export model=your_model.pt format=ncnn
```

Quedaría así:

```
yolo export model=yolo11n.pt format=ncnn
```

Una vez convertido, usaremos este para la inferencia.

10. Ejecutar el modelo Yolo

Parámetros para ejecutar el script “`yolo_detect.py`”

https://www.ejtech.io/code/yolo_detect.py

Descargarlo desde el navegador o bien usando `wget`:

```
wget https://ejtech.io/code/yolo_detect.py
```

Parámetros para ejecutar el script:

- **--model:** Path to the trained model weights (e.g. "runs/detect/train/weights/yolo11n_ncnn_model")
- **--source:** Path to an image file ("test_img.jpg"), a folder of images ("img_dir"), a video file ("test_vid.mp4"), or the index for a connected USB camera ("usb0").
- **--resolution** (optional): Resolution in WxH to display inference results at. If not specified, the program will match the source resolution.

Ejecutar el siguiente comando para ejecutar la detección con el modelo "yolo11n_ncnn_model" usando una cámara USB conectada con una resolución de 640x480. Si usa un modelo personalizado, reemplace "yolo11n_ncnn_model" con el nombre de la carpeta del modelo NCNN personalizado (por ejemplo, "custom_ncnn_model").

```
# Ejecutar a una resolución de 640x480
```

```
python yolo_detect.py --model=yolo11n_ncnn_model --source=usb0 --  
resolution=640x480
```

```
# Ejecutar usando una resolución de 1280x720
```

```
python yolo_detect.py --model=yolo11n_ncnn_model --source=usb0 --  
resolution=1280x720
```

Luego de ejecutar una Raspberry Pi 4B (4GB RAM) va muy lento o bajos FPS (1.5). En la fuente de este recurso lo ejecutaron en una Raspberry Pi 5 con una resolución de 1280x720 y le fue muy fluido con 6 FPS, aquí se obtuvo 1.5FPS.

A continuación, se muestran otros ejemplos de comandos que muestran cómo ejecutar el script en un archivo de vídeo o una imagen:

```
# Run on video file named "test_video.mp4"
python yolo_detect.py --model=yolo11n_ncnn_model --source=test_video.mp4

# Run on folder of images named "img_dir" en un modelo custom.
python yolo_detect.py --model=custom_ncnn_model --source=img_dir
```

11. Fuentes

- <https://www.ejtech.io/learn/yolo-on-raspberry-pi>
- <https://www.youtube.com/watch?v=z70ZrSZNi-8>