

Yolov8

1. Setup your webcam

A. If using raspberry pi camera

i. Connect it and confirm it works by going to the raspberry icon in upper left>Accessories>Terminal and type this

```
libcamera-hello
```

ii. If you see an image great, if not perhaps your camera is not fully connected or you might need to reboot the Pi

B. If using an external USB webcam

i. Connect it and confirm it works by going to the raspberry icon in upper left>Accessories>Terminal and install some relevant software

```
sudo apt install fswebcam
```

ii. Capture an image with raspistill also in terminal put this command (if your webcam doesn't support this resolution change the "-r 1920x1080" portion to "-r 1280x720")

```
fswebcam -r 1920x1080 --no-banner ~/usb-webcam-img001.jpg
```

iii. Open the image by going to the two folder icon in the upper left 2 icons down from the raspberry icon and double click on the new image called usb-webcam-img001.jpg

iv. If the image doesn't appear or raspistill doesn't detect it you may need to reboot or check that your webcam is detected with this command from the terminal

```
lsusb
```

2. Initialize python environment and install yolo v8

A. Install a virtual python environment and python requirements

i. Make a new directory and enter it, go to the terminal app and enter the following two commands

```
mkdir ultralytatics-yolov8
```

```
cd ultralytatics-yolov8
```

ii. Create a new virtual python environment (this keeps your libraries and other configurations organized in case you have multiple projects) by typing this in the terminal

```
python3 -m venv ~/ultralytatics-yolov8/
```

iii. Install yolo v8 and dependent libraries by typing this in the terminal

```
~/ultralytatics-yolov8/bin/pip3 install ultralytatics
```

3. Run yolo against the default model

A. Start python

i. Go to the terminal and enter the following command

```
~/ultralytatics-yolov8/bin/python
```

B. Start yolo with the default model and streaming webcam

i. Enter the following commands in python REPL console, note that the first one may take a moment (do not add any leading spaces)

```
from ultralytatics import YOLO
```

```
model = YOLO("yolov8n.pt")
```

```
results = model(source=0, show=True, conf=0.4, save=True)
```

ii. You should see a window pop-up with your webcam image and outlines of any common objects the yolo model thinks are present with a certainty level (0.8 or 80% certainty is a very strong indicator)

C. Want to train your own model to detect empty shelf space, flying robots or anything else you have an extensive set of example images for? Try it with roboflow

- <https://blog.roboflow.com/how-to-deploy-a-yolov8-model-to-a-raspberry-pi/>

Next level (scratchpad, not fully baked) - Faster new generation YOLO-NAS, broken dependencies with python library installations

<https://github.com/Deci-AI/super-gradients/blob/master/YOLONAS.md>

Yolov8 (or yolo-nas)

1. Setup your webcam

A. If using raspberry pi camera

i. Connect it and confirm it works by going to the raspberry icon in upper left>Accessories>Terminal and type this

```
libcamera-hello
```

ii. If you see an image great, if not perhaps your camera is not fully connected or you might need to reboot the Pi

B. If using an external USB webcam

i. Connect it and confirm it works by going to the raspberry icon in upper left>Accessories>Terminal and install some relevant software

```
sudo apt install fswebcam
```

ii. Capture an image with raspistill also in terminal put this command (if your webcam doesn't support this resolution change the "-r 1920x1080" portion to "-r 1280x720")

```
fswebcam -r 1920x1080 --no-banner ~/usb-webcam-img001.jpg
```

iii. Open the image by going to the two folder icon in the upper left 2 icons down from the raspberry icon and double click on the new image called usb-webcam-img001.jpg

iv. If the image doesn't appear or raspistill doesn't detect it you may need to reboot or check that your webcam is detected with this command from the terminal

```
lsusb
```

2. Download and install YOLO-NAS and software compilation tool cmake, meson and language libraries

A. In the terminal type this command

```
git clone https://github.com/Deci-AI/super-gradients.git
```

```
sudo apt install cmake meson libevdev-dev gfortran libatlas-base-dev  
protobuf-compiler -y
```

B. Install a virtual python environment and python requirements

i. Go into the directory with the github python code in it by typing this into the terminal

```
cd super-gradients
```

ii. Create a new virtual python environment (this keeps your libraries and other configurations organized in case you have multiple projects) by typing this in the terminal

```
python3 -m venv ~/super-gradients/
```

iii. Install required python libraries and yolo-nas by typing this command in the terminal

```
~/super-gradients/bin/pip3 install super-gradients
```

iii(ALT). Alternatively you can try to compile directly

```
~/super-gradients/bin/python -m pip install numpy scipy matplotlib ipython
```

```
jupyter pandas sympy nose
```

```
~/super-gradients/bin/pip3 install cython
```

```
~/super-gradients/bin/pip3 install scipy  
~/super-gradients/bin/pip3 install -r requirements.txt
```