

# Roboflow

[ROBOFLOW](#)

<https://roboflow.com/#WatchDemo>

<https://docs.roboflow.com/>

## Welcome! Let's get started.

Create your first workspace to house all of your projects and collaborate with teammates.

Name your workspace:

Trimaran Workspace

Choose your plan:

### Starter Trial

For startups and businesses.

**Free 14 day trial, no credit card required,**

\$249 / mo to continue.

- Your Datasets and Models are Private
- Commercial Deployment License
- Outsource Labeling, Accurate Train, Model Evaluation, and Community Plan Features.

**50 projects, 10 base training credits + 5 more / mo, 10,000 monthly hosted inference API calls.\***

\*You can always customize and add more limits later.



### Public Plan

For hobbyists, students, and personal use.

**Free,**

with public data and limited features.

- Your Datasets and Models are Public

- No Commercial Deployment License

- Model-Assisted Labeling, Image Preprocessing and Augmentations, Dataset Health Check.

**3 projects, 3 base training credits, 1,000 monthly hosted inference API calls.**



## Invite teammates.

Add collaborators to help with labeling, upload data, train models, and more.

Invite Teammates via Email:

dev@trimaran.com

Role: Admin ▾

## You are on a Starter Plan trial!

The Starter Plan comes with premium features, higher limits, and priority support.

After 14 days, you will have the option of remaining on the Starter Plan for \$249 per month. Email us at [starter-plan@roboflow.com](mailto:starter-plan@roboflow.com) with any questions!

**Continue!**

# Let's create your project.

Trimaran Workspace > [New Private Project](#)

Project Name

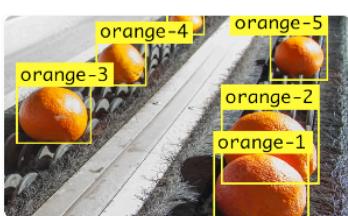
Trimaran Project

Visibility [?](#)

Private



Project Type

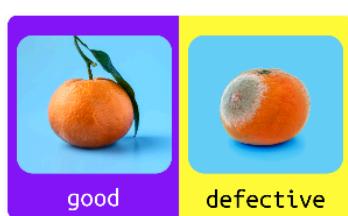


## Object Detection

Identify objects and their positions with bounding boxes.

Best For

# Counting Tracking

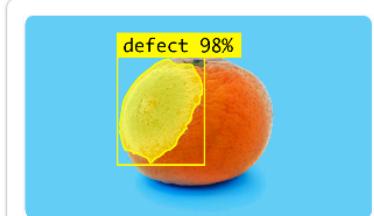


## Classification

Assign labels to the entire image.

Best For

Filtering Content Moderation



## Instance Segmentation

Detect multiple objects and their actual shape.

Best For

Measurements Odd Shapes

Show More

Annotation Group [?](#)

sailboat

Cancel

Create Private Project

**Object Detection:** Find the location of objects in an image.

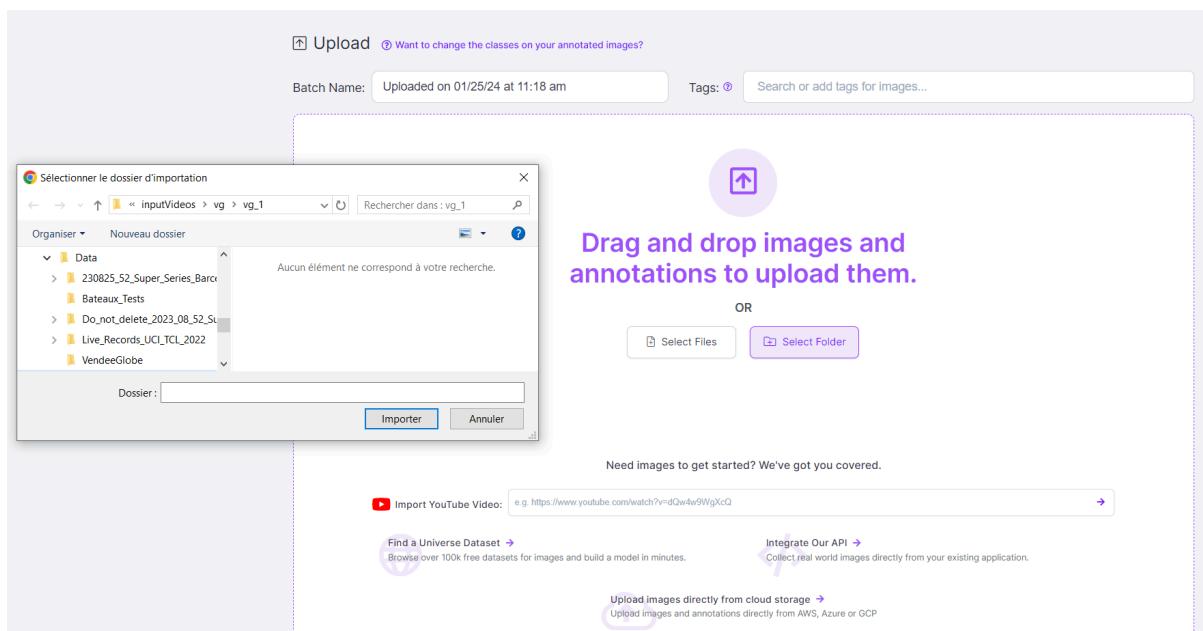
**Single-Label Classification:** Given a limited set of categories, assign a label to an image.

**Multi-Label Classification:** Given a limited set of categories, assign an arbitrary number of labels that are relevant to the image.

**Instance Segmentation:** To the pixel level, find the location of objects in an image.

**Semantic Segmentation:** To the pixel level, find the location of objects in an image and create unique references for each object found.

**Keypoint Detection:** Find the location of objects and their keypoints in an image. Commonly used for determining the pose of an object.



**Upload** Want to change the classes on your annotated images?

Batch Name: Vendée Globe - 2023/10/12 Tags:

All Images 4 Annotated 0 Not Annotated 4

**Drag and drop images and annotations.**

in .jpg, .png, .bmp in 26 formats >> in .mov, .mp4, .avi

0451.jpg 0441.jpg 0011.jpg 0001.jpg

TOTAL IMAGES TO ASSIGN: 4 / 4

Assign Images to Teammates

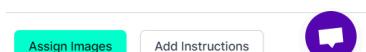
Choose teammates to label images. Images will be evenly divided between selected teammates.

You can assign specific images on the [Unassigned images tab](#)

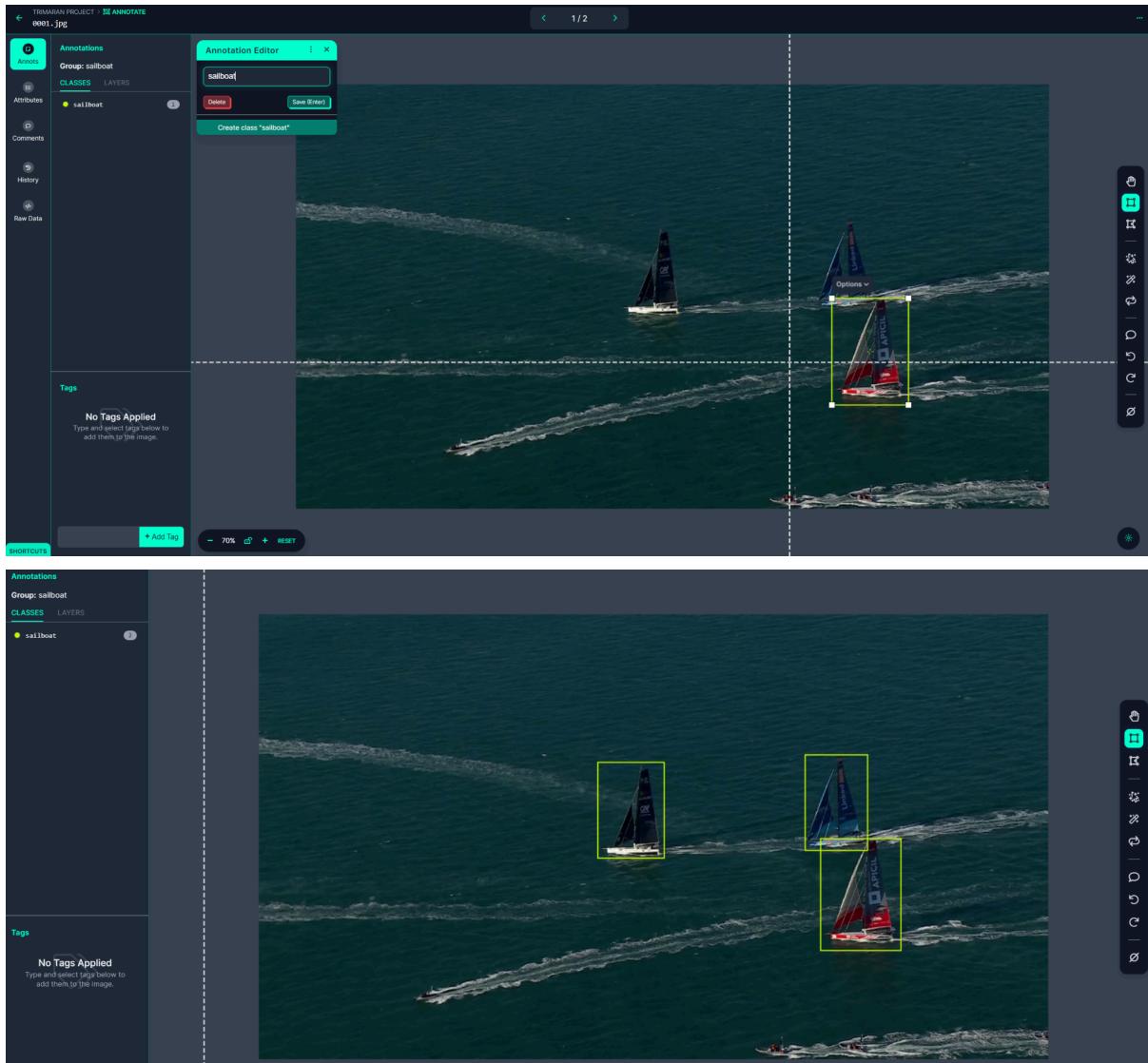
LLVD laura.llvd@free.fr 2 images

dev@trimaran.com 2 images

Labeling support



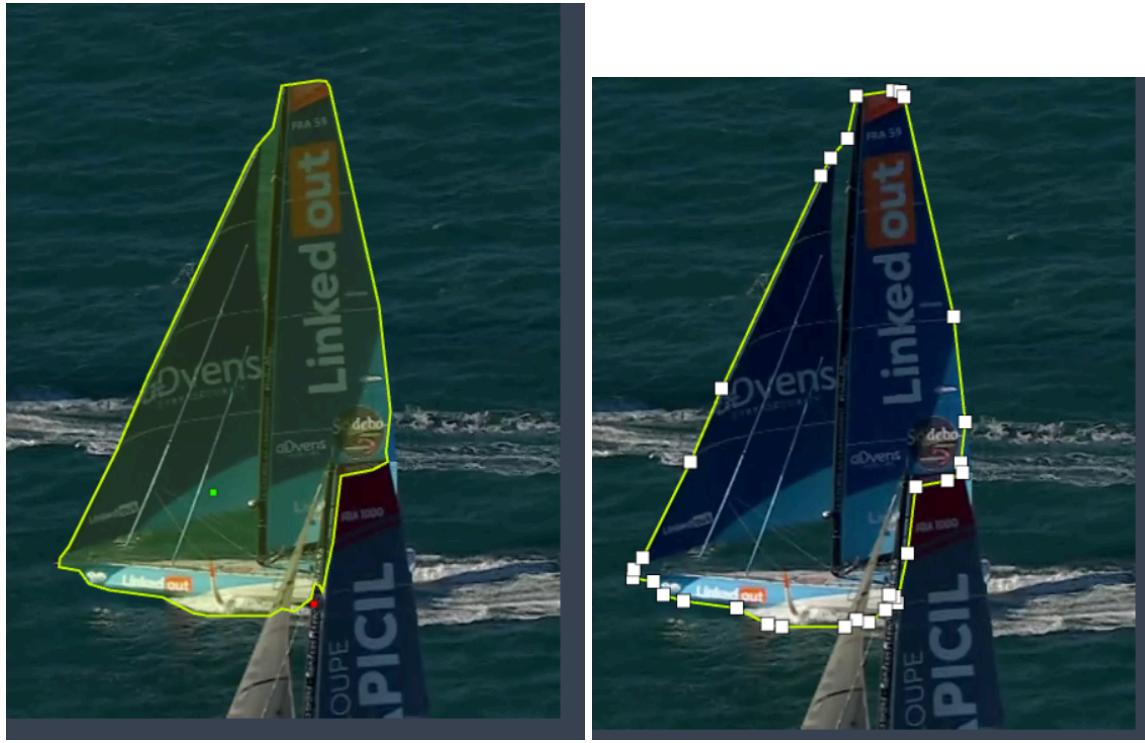
## Annotation : rectangle bounding box



## Annotation : segmentation with smart polygone

Left click outside the green zone : expand the zone to this point and to similar pixels around  
Left click inside the green zone : limits the zone to get out this point and similar pixels around





Enter + click on white points in the line to move the selection borders

Enter x2 → capture the selection as an object of the class specified in the left top corner

### Annotations

Group: sailboat

**CLASSES** LAYERS

- sailboat

3

### Annotation Editor

sailboat\_corner

Delete Save (Enter)

Create class "sailboat\_corner"

Annotations

Group: sailboat

**CLASSES** LAYERS

- sailboat
- sailboat\_corner

No Tags Applied

Type and select tags below to add them to the image.

Mark Null : indicates that there is no object to detect on the image. Very useful to the learning process to have many views of what is not one of the objects to detect

To use an existing model to help labeling process, use one public model, or train a custom model on parts of your annotated dataset. If you want to use a model trained on another project or account, the model may need to be deployed and considered as public.

### Label Assist

Use the predictions from a public model or a model trained by [Roboflow Train](#) as a starting point for annotating new images in this dataset.

[Your Models](#) [Public Models](#)

You haven't trained any custom models yet.

To use label assist with a custom model, annotate a few dozen images by hand, add them to your dataset, create a version, and train a model. Then it will show up here.

[Cancel](#) [Continue](#)

### Label Assist

Use the predictions from a public model or a model trained by [Roboflow Train](#) as a starting point for annotating new images in this dataset.

[Your Models](#) [Public Models](#)

PROJECT [MS COCO](#)

MODEL [v1 - Best \(Common Objects, 55.8% mAP\)](#)

★ Star public models on [Universe](#) to use them in Label Assist.

[Cancel](#) [Continue](#)

### Add 2 images to dataset

[? What's Train, Valid, Test?](#)

Method

Split Images Between Train/Valid/Test

[Add 2 images to Dataset](#)

Train 70%      Valid 20%      Test 10%

Image Distribution

Train: 1 images  
Valid: 1 images  
Test: 0 images

You are about to add 2 images to the dataset  
0 images will be sent back as part of a new job

[+ Create New Version](#)

[Add Images](#)

4

## Augmentation

[?](#) What can augmentation do?

Create new training examples for your model to learn from by generating augmented versions of each image in your training set.

Add Augmentation Step

Continue

### Augmentation Options

X

Augmentations create new training examples for your model to learn from.

#### IMAGE LEVEL AUGMENTATIONS



#### BOUNDING BOX LEVEL AUGMENTATIONS [?](#)



Cancel

5

## Create

Review your selections and select a version size to create a moment-in-time snapshot of your dataset with the applied transformations.

Larger versions take longer to train but often result in better model performance. [See how this is calculated »](#)

### Maximum Version Size

6 images (5x)

**Create**

- 3 images (2x)
  - 4 images (3x)
  - 5 images (4x)
  - 6 images (5x)**
  - 7 images (6x - Requires Upgrade)
  - 8 images (7x - Requires Upgrade)
  - 9 images (8x - Requires Upgrade)
  - 10 images (9x - Requires Upgrade)
  - 11 images (10x - Requires Upgrade)
  - 12 images (11x - Requires Upgrade)
  - 13 images (12x - Requires Upgrade)
  - 14 images (13x - Requires Upgrade)
  - 15 images (14x - Requires Upgrade)
  - 16 images (15x - Requires Upgrade)
  - 17 images (16x - Requires Upgrade)
  - 18 images (17x - Requires Upgrade)
  - 19 images (18x - Requires Upgrade)
  - 20 images (19x - Requires Upgrade)
  - 21 images (20x - Requires Upgrade)
  - 22 images (21x - Requires Upgrade)
- 6 images (5x)

**Create**



**Trimaran Project Image Dataset**

**Trimaran Project** Object Detection

**Create New Version**

**v1** 2024-01-25 11:40am Generated on Jan 25, 2024

**Export Dataset** **⋮**

**Versions**

2024-01-25 11:40am v1 Jan 25, 2024

**This version doesn't have a model.**

Train an optimized, state of the art model with Roboflow or upload a custom trained model to use features like Label Assist and Model Evaluation and deployment options like our auto-scaling API and edge device support.

**Train with Roboflow** **Custom Train and Upload**

Available Credits: 18

**6 Total Images** **View All Images →**

**Dataset Split**

TRAIN SET	5 Images
VALID SET	1 Images
TEST SET	0 Images

**Preprocessing** Auto-Orient: Applied  
Resize: Stretch to 640x640

**2024-01-25 11:40am**

Generated on Jan 25, 2024

**Export**

**Export** **X**

**This version doesn't have a model.**

Train an optimized, state of the art model with Roboflow or upload a custom trained model to use features like Label Assist and Model Evaluation and deployment options like our auto-scaling API and edge device support.

**Custom Train and Upload**

**Format**

YOLOv8

TXT annotations and YAML config used with **YOLOv8**.

download zip to computer  show download code

**Cancel** **Continue**

**View All Images →**

ra > Dev > LowerPythonEnv > datasets > roboflow

Nom	Modifié le	Type	Taille
📁 train	25/01/2024 12:08	Dossier de fichiers	
📁 valid	25/01/2024 12:08	Dossier de fichiers	
📄 data.yaml	25/01/2024 11:44	Fichier source Yaml	1 Ko
📝 README.roboflow.txt	25/01/2024 11:44	Fichier TXT	2 Ko

# Train a model

<https://docs.ultralytics.com/fr/>

<https://pypi.org/project/ultralytics/>

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    COMMENTS

Microsoft Windows [version 10.0.19045.3930]
(c) Microsoft Corporation. Tous droits réservés.

(.venv) D:\LEVRAUDLaura\Dev\LowerPythonEnv>python
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

(.venv) D:\LEVRAUDLaura\Dev\LowerPythonEnv>pip3 install ultralytics
Requirement already satisfied: ultralytics in d:\levraudlaura\dev\lowerpythonenv\.venv\lib\site-packages (8.0.199)
```

Device is determined automatically. If a GPU is available then it will be used, otherwise training will start on CPU.

Python    **CLI**

```
# Build a new model from YAML and start training from scratch
yolo detect train data=coco128.yaml model=yolov8n.yaml epochs=100 imgs=640

# Start training from a pretrained *.pt model
yolo detect train data=coco128.yaml model=yolov8n.pt epochs=100 imgs=640

# Build a new model from YAML, transfer pretrained weights to it and start training
yolo detect train data=coco128.yaml model=yolov8n.yaml pretrained=yolov8n.pt epochs=100 imgs=640
```

coco128.yaml → Dataset COCO used to train yolov8n (.yaml / .pt / .onnx). Common objects to replace with the custom dataset : data.yaml

```
(.venv) D:\LEVRAUDLaura\Dev\LowerPythonEnv\datasets\roboflow>yolo detect train
```

```
data=data.yaml model=yolov8n.yaml pretrained=yolov8n.pt epochs=100 imgs=640
```

```
(.venv) D:\LEVRAUDLaura\Dev\LowerPythonEnv\datasets\roboflow>yolo detect train data=data.yaml model=yolov8n.yaml pretrained=yolov8n.pt epochs=100 imgs=640

      from   n   params   module           arguments
0        -1   1       464   ultralytics.nn.modules.conv.Conv      [3, 16, 3, 2]
1        -1   1      4672  ultralytics.nn.modules.conv.Conv      [16, 32, 3, 2]
2        -1   1      7360  ultralytics.nn.modules.block.C2f      [32, 32, 1, True]
3        -1   1      18568 ultralytics.nn.modules.conv.Conv      [32, 64, 3, 2]
4        -1   2      49664 ultralytics.nn.modules.block.C2f      [64, 64, 2, True]
5        -1   1      73984 ultralytics.nn.modules.conv.Conv      [64, 128, 3, 2]
6        -1   2      197632 ultralytics.nn.modules.block.C2f      [128, 128, 2, True]
7        -1   1      295424 ultralytics.nn.modules.conv.Conv      [128, 256, 3, 2]
8        -1   1      460288 ultralytics.nn.modules.block.C2f      [256, 256, 1, True]
9        -1   1     164608 ultralytics.nn.modules.block.SPPF      [256, 256, 5]
10       -1   1       0   torch.nn.modules.upsampling.Upsample      [None, 2, 'nearest']
11      [-1, 6]  1       0   ultralytics.nn.modules.conv.Concat      [1]
12       -1   1     148224 ultralytics.nn.modules.block.C2f      [384, 128, 1]
13       -1   1       0   torch.nn.modules.upsampling.Upsample      [None, 2, 'nearest']
14      [-1, 4]  1       0   ultralytics.nn.modules.conv.Concat      [1]
15       -1   1     37248 ultralytics.nn.modules.block.C2f      [192, 64, 1]
16       -1   1     36992 ultralytics.nn.modules.conv.Conv      [64, 64, 3, 2]
17      [-1, 12]  1       0   ultralytics.nn.modules.conv.Concat      [1]
18       -1   1     123648 ultralytics.nn.modules.block.C2f      [192, 128, 1]
19       -1   1     147712 ultralytics.nn.modules.conv.Conv      [128, 128, 3, 2]
20      [-1, 9]  1       0   ultralytics.nn.modules.conv.Concat      [1]
21       -1   1     493056 ultralytics.nn.modules.block.C2f      [384, 256, 1]
22     [15, 18, 21]  1     897664 ultralytics.nn.modules.head.Detect      [80, [64, 128, 256]]
YOLOv8n summary: 225 layers, 3157200 parameters, 3157184 gradients, 8.9 GFLOPs

Downloading https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8n.pt to 'yolov8n.pt'...
```

```

Epoch    GPU_mem    box_loss    cls_loss    dfl_loss    Instances    Size
87/100      0G      0.9851      1.121      1.154      32          640: 100% |██████████| 1/1 [00:00<00:00,  2.21it/s]
              Class   Images   Instances   Box(P      R      mAP50  mAP50-95): 100% |██████████| 1/1 [00:00<00:00, 15.15it/s]
                  all       1         3      0.746      0.5      0.621      0.278

Epoch    GPU_mem    box_loss    cls_loss    dfl_loss    Instances    Size
88/100      0G      1.042      1.091      1.107      49          640: 100% |██████████| 1/1 [00:00<00:00,  2.05it/s]
              Class   Images   Instances   Box(P      R      mAP50  mAP50-95): 100% |██████████| 1/1 [00:00<00:00, 15.38it/s]
                  all       1         3      0.746      0.5      0.538      0.311
Stopping training early as no improvement observed in last 50 epochs. Best results observed at epoch 38, best model saved as best.pt.
To update EarlyStopping(patience=50) pass a new patience value, i.e. `patience=300` or use `patience=0` to disable EarlyStopping.

88 epochs completed in 0.017 hours.
Optimizer stripped from runs\detect\train\weights\last.pt, 6.3MB
Optimizer stripped from runs\detect\train\weights\best.pt, 6.3MB

Validating runs\detect\train\weights\best.pt...
Ultralytics YOLOv8.0.199 Python-3.11.6 torch-2.1.0+cpu CPU (AMD Ryzen 9 7950X 16-Core Processor)
YOLOv8n summary (fused): 168 layers, 3006038 parameters, 0 gradients, 8.1 GFLOPs
              Class   Images   Instances   Box(P      R      mAP50  mAP50-95): 100% |██████████| 1/1 [00:00<00:00, 18.51it/s]
                  all       1         3      0.82      0.5      0.912      0.862
                  sailboat     1         2      0.639      1      0.828      0.729
                  sailboat_corner     1         1      1         0      0.995      0.995
Speed: 1.0ms preprocess, 43.0ms inference, 0.0ms loss, 5.0ms postprocess per image
Results saved to runs\detect\train
Learn more at https://docs.ultralytics.com/modes/train

```

## Resume the training if interrupted (last epoch completed < total epochs ordered) :

<https://docs.ultralytics.com/fr/modes/train/>

Python      CLI

```
# Resume an interrupted training
yolo train resume model=path/to/last.pt
```

## Retrain with new dataset :

Select the last best weights and start a new training on the new dataset.

```
yolo detect train data=..\..\inputTrain15\data.yaml model=.\runs\detect\train\weights\best.pt epochs=100
```

## BE CAREFUL !

Through the epochs of a model training process, the goal of the model evolution is to perform with the lowest number of errors on the dataset the model is CURRENTLY training. Imagine you have trained model 1, whose best weights are best1.pt, on dataset 1. You want to keep training this model with new data, stored in dataset 2. You use this command :

```
yolo detect train data=data2.yaml model=best1.pt epochs=100 imgs=640
```

Then you will create a new model, initialized with best1.pt weights, that will evolve during the next epochs as last2.pt. The goal of this training process is to save best2.pt, the weights that perform the lowest error rate on the dataset 2.

It may produce very different results from retraining best1.pt on the whole dataset formed by dataset 1 + dataset 2.

With the command :

```
yolo detect train data=data1+2.yaml model=best1.pt epochs=100 imgsz=640
```

It may take longer to train, but the model accuracy on the whole dataset (1 + 2) will be the first concern of the process. So you should retrain your model on your whole dataset, or at least on dataset based on parts of dataset 1 + whole dataset 2, from time to time. Except if you judge that the new data is more representative of the objects you want to detect now, than the old data that may become outdated.

So be careful of the choice of your dataset for the training process : it may lead to the creation of biased models.

## Test your model in python :

<https://docs.ultralytics.com/fr/modes/predict/>

```
Return a list with stream=False      Return a generator with stream=True

from ultralytics import YOLO

# Load a model
model = YOLO('yolov8n.pt') # pretrained YOLOv8n model

# Run batched inference on a list of images
results = model(['im1.jpg', 'im2.jpg']) # return a list of Results objects

# Process results list
for result in results:
    boxes = result.boxes # Boxes object for bbox outputs
    masks = result.masks # Masks object for segmentation masks outputs
    keypoints = result.keypoints # Keypoints object for pose outputs
    probs = result.probs # Probs object for classification outputs
```

## Use model.predict arguments :

```
from ultralytics import YOLO

# Load a pretrained YOLOv8n model
model = YOLO('yolov8n.pt')

# Run inference on 'bus.jpg' with arguments
model.predict('bus.jpg', save=True, imgsz=320, conf=0.5)
```

Name	Type	Default	Description
source	str	'ultralytics/assets'	source directory for images or videos
conf	float	0.25	object confidence threshold for detection
iou	float	0.7	intersection over union (IoU) threshold for NMS
imgsz	int or tuple	640	image size as scalar or (h, w) list, i.e. (640, 480)
half	bool	False	use half precision (FP16)
device	None or str	None	device to run on, i.e. cuda device=0/1/2/3 or device=cpu
max_det	int	300	maximum number of detections per image
vid_stride	bool	False	video frame-rate stride
stream_buffer	bool	False	buffer all streaming frames (True) or return the most recent frame (False)
visualize	bool	False	visualize model features
augment	bool	False	apply image augmentation to prediction sources
agnostic_nms	bool	False	class-agnostic NMS
classes	list[int]	None	filter results by class, i.e. classes=0, or classes=[0,2,3]

Name	Type	Default	Description
show	bool	False	show predicted images and videos if environment allows
save	bool	False	save predicted images and videos
save_frames	bool	False	save predicted individual video frames
save_txt	bool	False	save results as .txt file
save_conf	bool	False	save results with confidence scores
save_crop	bool	False	save cropped images with results
show_labels	bool	True	show prediction labels, i.e. 'person'
show_conf	bool	True	show prediction confidence, i.e. '0.99'
show_boxes	bool	True	show prediction boxes
line_width	None or int	None	line width of the bounding boxes. Scaled to image size if None.

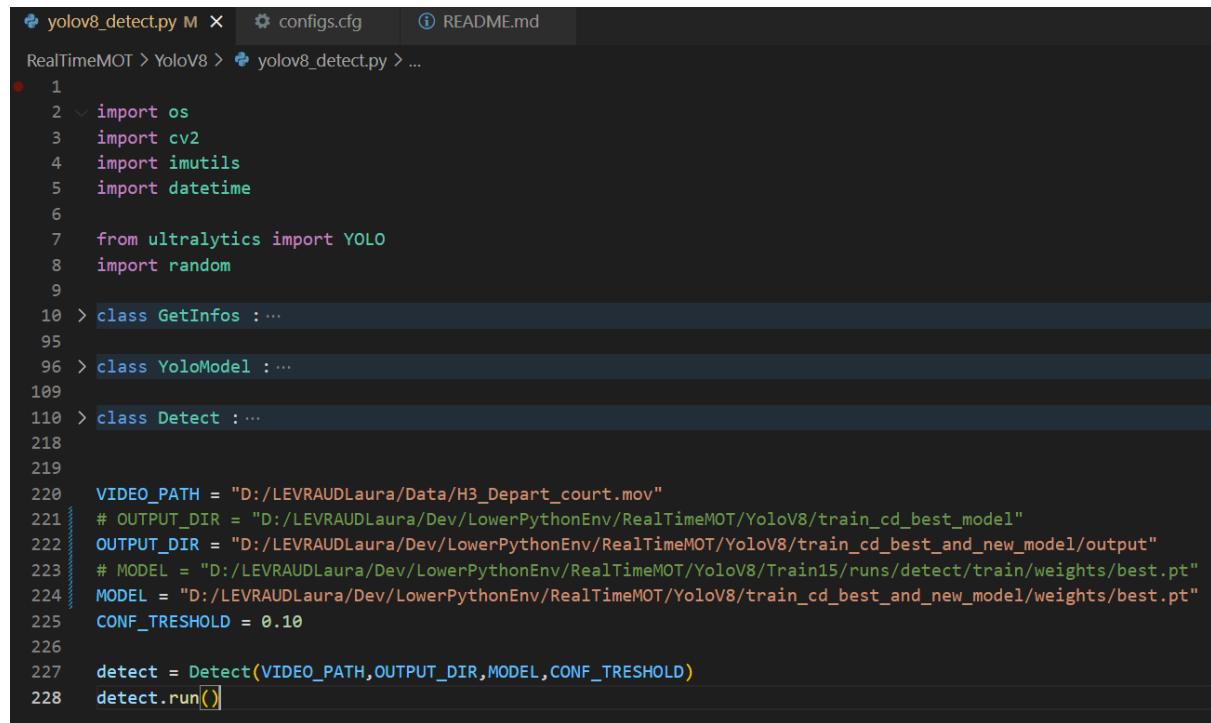
The below table contains valid Ultralytics image formats.

<b>Image Suffixes</b>	<b>Example Predict Command</b>	<b>Reference</b>
.bmp	yolo predict source=image.bmp	<a href="#">Microsoft BMP File Format</a>
.dng	yolo predict source=image.dng	<a href="#">Adobe DNG</a>
.jpeg	yolo predict source=image.jpeg	<a href="#">JPEG</a>
.jpg	yolo predict source=image.jpg	<a href="#">JPEG</a>
.mpo	yolo predict source=image.mpo	<a href="#">Multi Picture Object</a>
.png	yolo predict source=image.png	<a href="#">Portable Network Graphics</a>
.tif	yolo predict source=image.tif	<a href="#">Tag Image File Format</a>
.tiff	yolo predict source=image.tiff	<a href="#">Tag Image File Format</a>
.webp	yolo predict source=image.webp	<a href="#">WebP</a>
.pfm	yolo predict source=image.pfm	<a href="#">Portable FloatMap</a>

The below table contains valid Ultralytics video formats.

<b>Video Suffixes</b>	<b>Example Predict Command</b>	<b>Reference</b>
.ASF	yolo predict source=video.asf	<a href="#">Advanced Systems Format</a>
.avi	yolo predict source=video.avi	<a href="#">Audio Video Interleave</a>
.gif	yolo predict source=video.gif	<a href="#">Graphics Interchange Format</a>
.m4v	yolo predict source=video.m4v	<a href="#">MPEG-4 Part 14</a>
.mkv	yolo predict source=video.mkv	<a href="#">Matroska</a>
.mov	yolo predict source=video.mov	<a href="#">QuickTime File Format</a>
.mp4	yolo predict source=video.mp4	<a href="#">MPEG-4 Part 14 - Wikipedia</a>
.mpeg	yolo predict source=video.mpeg	<a href="#">MPEG-1 Part 2</a>
.mpg	yolo predict source=video.mpg	<a href="#">MPEG-1 Part 2</a>
.ts	yolo predict source=video.ts	<a href="#">MPEG Transport Stream</a>
.wmv	yolo predict source=video.wmv	<a href="#">Windows Media Video</a>
.webm	yolo predict source=video.webm	<a href="#">WebM Project</a>

## Test your model on a video with [yolov8\\_detect.py](#) :



```
 1 import os
 2 import cv2
 3 import imutils
 4 import datetime
 5
 6 from ultralytics import YOLO
 7 import random
 8
 9
10 > class GetInfos :
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
```

```
VIDEO_PATH = "D:/LEVRAUDLaura/Data/H3_Depart_count.mov"
# OUTPUT_DIR = "D:/LEVRAUDLaura/Dev/LowerPythonEnv/RealTimeMOT/YoloV8/train_cd_best_model"
OUTPUT_DIR = "D:/LEVRAUDLaura/Dev/LowerPythonEnv/RealTimeMOT/YoloV8/train_cd_best_and_new_model/output"
# MODEL = "D:/LEVRAUDLaura/Dev/LowerPythonEnv/RealTimeMOT/YoloV8/Train15/runs/detect/train/weights/best.pt"
MODEL = "D:/LEVRAUDLaura/Dev/LowerPythonEnv/RealTimeMOT/YoloV8/train_cd_best_and_new_model/weights/best.pt"
CONF_TRESHOLD = 0.10
detect = Detect(VIDEO_PATH,OUTPUT_DIR,MODEL,CONF_TRESHOLD)
detect.run()
```

Change VIDEO\_PATH, OUTPUT\_DIR, MODEL and CONF\_THRESHOLD parameters.  
Each prediction is given by the model with a confidence score. CONF\_THRESHOLD allows only the detections with higher confidence to appear in the video output.