YOLO GUIDE

Last Update: June 2025

Table of Contents

1 PREREQUISITES

1.1 PYTHON

Install Python 3.8+ *(I used python 3.10 (windows) and 3.11 (raspberry py))*

Install python here

1.1.1 Check your python version

In the command prompt, run `python --version` to check which version you have installed.

1.2 VIRTUAL ENVIRONMENT

Navigate to a folder where you will run all yolo code and create a virtual environment. The virtual environment is a safe place for installing packages and running yolo.

If you skip this step, you may break your system packages and have issues later on.

1.2.1 Windows

dir – command to print the contents of the current directory

cd – command to change to other directories

Create a virtual environment

Make sure you are in a folder you can locate later, the virtual environment will be created locally

```
python -m venv environment_name
        NOTE: Add “--system-site-packages” option to give venv access to globals
```

Open the virtual environment

First, navigate to the folder where the venv was created

```
.\environment_name\Scripts\activate
```

Close the virtual environment

```
deactivate
```

1.2.2 Linux

ls – command to print the contents of the current directory

cd – command to change to other directories

Create a virtual environment

Make sure you are in a folder you can locate later, the virtual environment will be created locally

```
python -m venv environment_name
       NOTE: Add “--system-site-packages” option to give venv access to globals
```

Open the virtual environment

First, navigate to the folder where the venv was created

```
source environment_name/bin/activate
```

Close the virtual environment

```
deactivate
```

1.3 PACKAGES AND SOFTWARE REQUIREMENTS

1.3.1 Windows

Pip

Install pip if you do not already have it…

Update it if you do have it:

```
python -m pip install --upgrade pip
```

Ultralytics

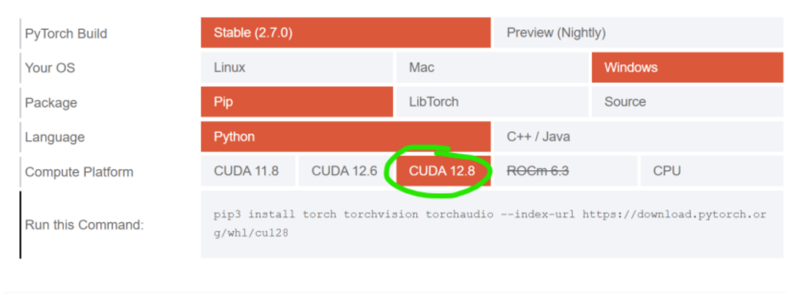
```
pip install ultralytics
```

PyTorch

Use this website https://pytorch.org/get-started/locally/ to get the correct command.

```
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu128
```

I selected version 12.8 because my cuda drivers require it later on…



PyTorch Build	[✓ Stable (2.7.0)]	Preview (Nightly)			
Your OS	Linux	Mac	[✓ Windows]		
Package	[✓ Pip]	LibTorch	Source		
Language	[✓ Python]	C++ / Java			
Compute Platform	CUDA 11.8	CUDA 12.6	[✓ CUDA 12.8]	~~ROCm 6.3~~	CPU

Run this Command:

```
pip3 install torch torchvision torchaudio --index-url https://download.pytorch.or
g/whl/cu128
```

1.3.2 Linux

Pip

Install pip if you do not already have it…

Update it if you do have it:

```
python -m pip install --upgrade pip
```

Ultralytics

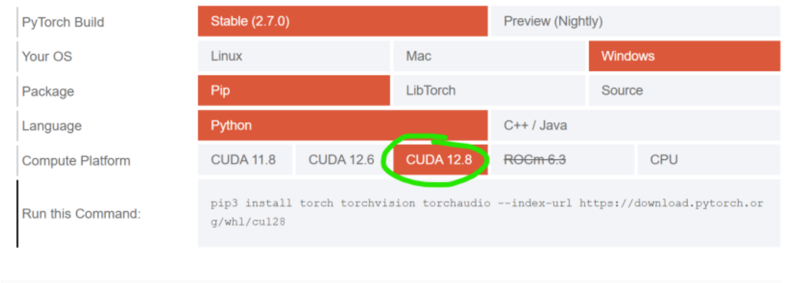
```
pip install ultralytics
```

PyTorch

Use this website https://pytorch.org/get-started/locally/ to get the correct command.

```
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu128
```

I selected version 12.8 because my cuda drivers require it later on…



PyTorch Build	[✓ Stable (2.7.0)]	Preview (Nightly)			
Your OS	Linux	Mac	[✓ Windows]		
Package	[✓ Pip]	LibTorch	Source		
Language	[✓ Python]	C++ / Java			
Compute Platform	CUDA 11.8	CUDA 12.6	[✓ CUDA 12.8]	~~ROCm 6.3~~	CPU

Run this Command:

```
pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu128
```

1.3.3 Cuda Toolkit

Install the Cuda Toolkit here https://developer.nvidia.com/cuda-downloads

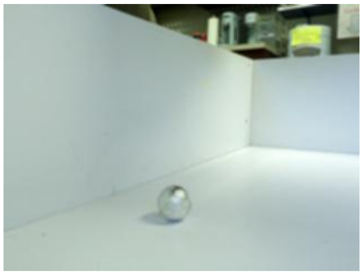
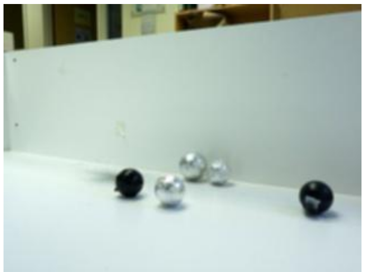
NOTE: you may require an older version if you have an older gpu

2 Training Data

The training data consists of images with bounding rectangles around the objects in the image you wish to identify. No knowledge of computer vision required.

2.1 Image Collection

I used my robot’s camera to collect around 100 images of the Evac Zone from a few different angles. The images contain a mix of one or more of the objects.

The images can contain full or partial views of the objects.

Hint: this can be done quickly with a program that saves an image when you press space bar

2.2 RoboFlow

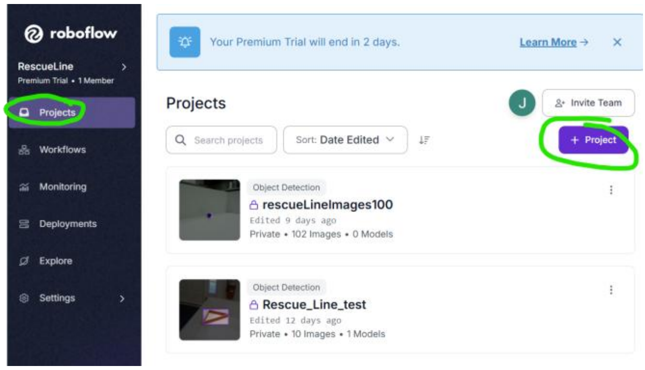
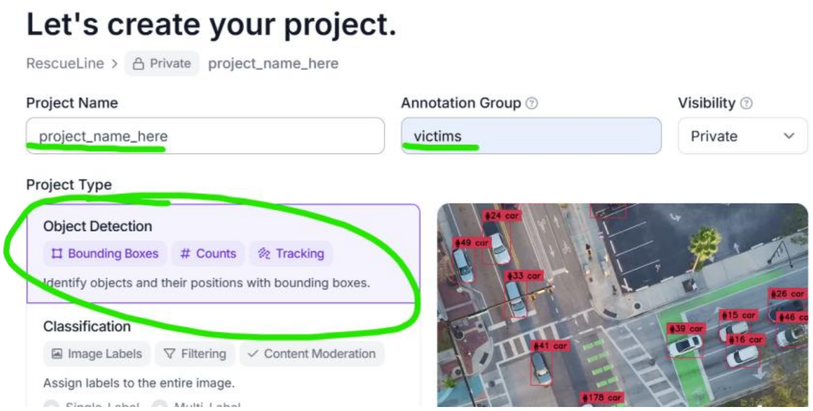
Per Ultralytics suggestion, I used roboflow to create the training data.

I previously wrote a program to create the training data manually, but RCJ does not require development anymore so I will be showcasing a much simpler method.

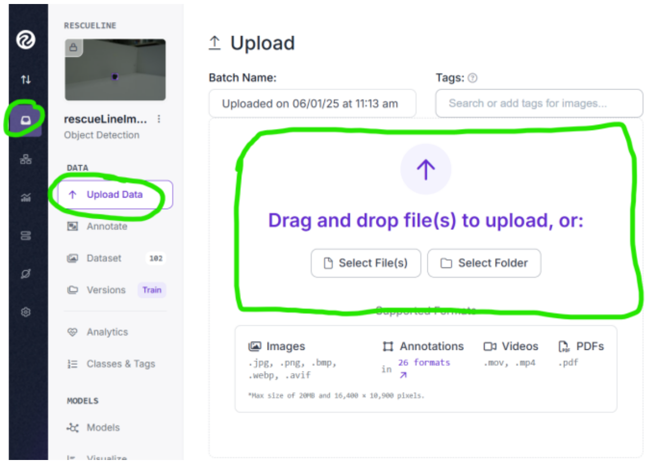
2.2.1 Create an account

Create an account here on roboflow

2.2.2 Create a project

 roboflow RescueLine Premium Trial • 1 Member Projects Workflows Monitoring Deployments Explore Settings Your Premium Trial will end in 2 days. Learn More → Projects J Invite Team Search projects Sort: Date Edited + Project Object Detection rescueLineImages100 Edited 9 days ago Private • 102 Images • 0 Models Object Detection Rescue_Line_test Edited 12 days ago Private • 10 Images • 1 Models	Create a new project in the Project tab
 Let's create your project. RescueLine > Private project_name_here Project Name: project_name_here Annotation Group: victims Visibility: Private Project Type Object Detection Bounding Boxes # Counts Tracking Identify objects and their positions with bounding boxes. Classification Image Labels Filtering Content Moderation Assign labels to the entire image. Single Label Multi Label	Select **Object Detection** as the project type. Name the project and give a name to the group of items you can trying to detect (i.e. *victims*)

2.2.3 Add images to the project

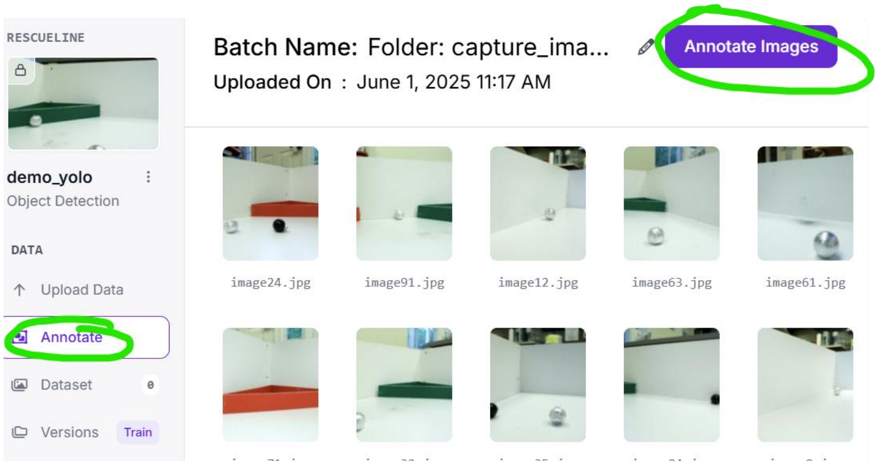
 RESCUELINE rescueLineIm... Object Detection DATA Upload Data Annotate Dataset 102 Versions Train Analytics Classes & Tags MODELS Models Upload Batch Name: Tags: Search or add tags for images... Drag and drop file(s) to upload, or: Select File(s) Select Folder Supported Formats Images .jpg, .png, .bmp, .webp, .avif Annotations in 26 formats Videos .mov, .mp4 PDFs .pdf *Max size of 20MB and 16,400 x 10,900 pixels. Batch Name: Uploaded on 06/01/25 at 11:13 am	Upload the folder with all the images captured on from your robot

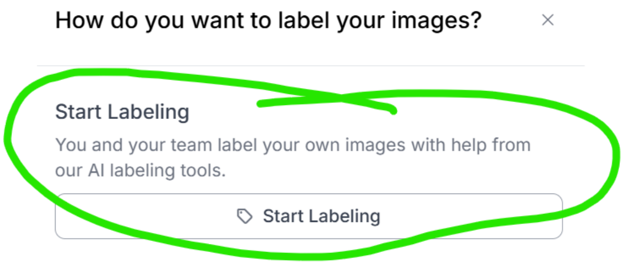
2.2.4 Annotate your images

Annotating your images is the process of identifying the objects in your uploaded images. All you need to do is draw bounding boxes around your images and specify the class if belongs to.

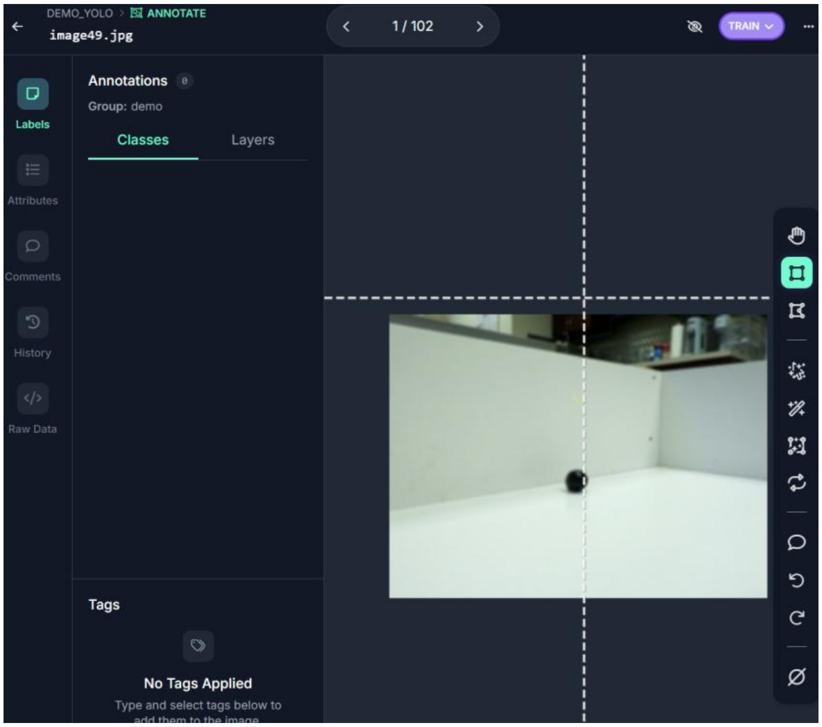
I will share some helpful notes for annotating.

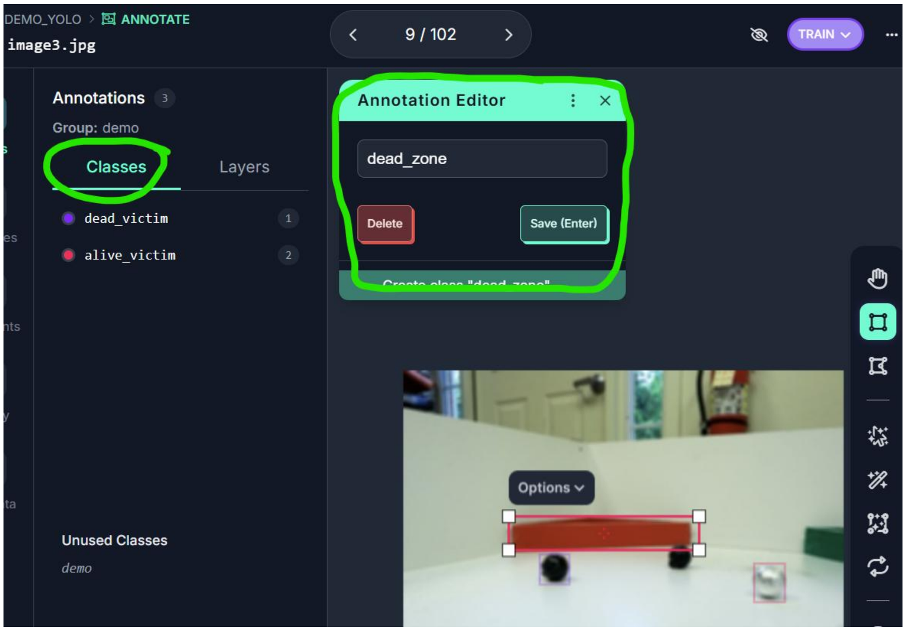
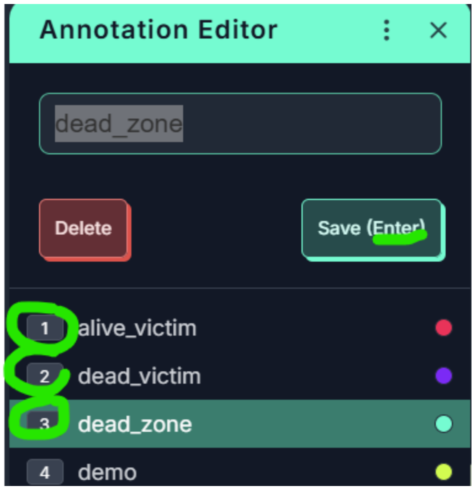
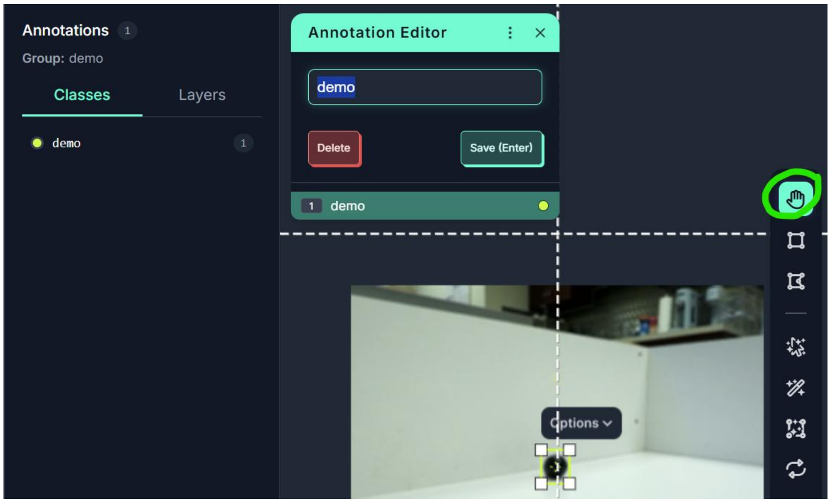
How to get started

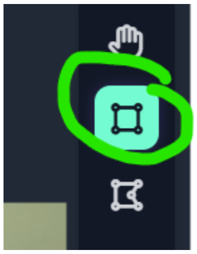
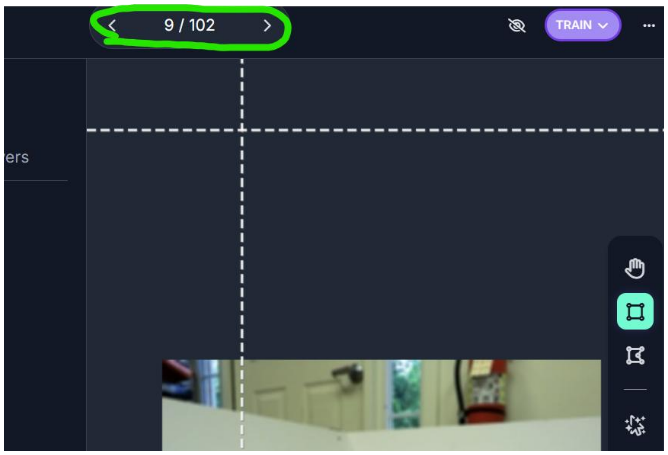
 RESCUELINE demo_yolo Object Detection DATA Upload Data Annotate Dataset 0 Versions Train Annotate Images image24.jpg image91.jpg image12.jpg image63.jpg image61.jpg Batch Name: Folder: capture_ima... Uploaded On : June 1, 2025 11:17 AM	Navigate to the **annotate** page and select **annotate images** in the top right

 How do you want to label your images? Start Labeling You and your team label your own images with help from our AI labeling tools. Start Labeling	When prompted, select to annotate the images manually. This is the easiest way to annotate the first time

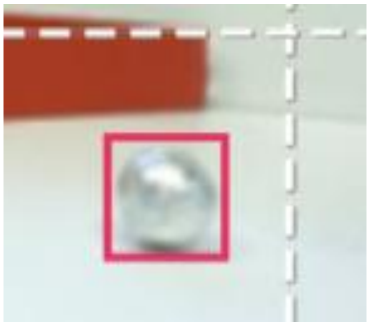
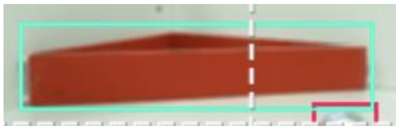
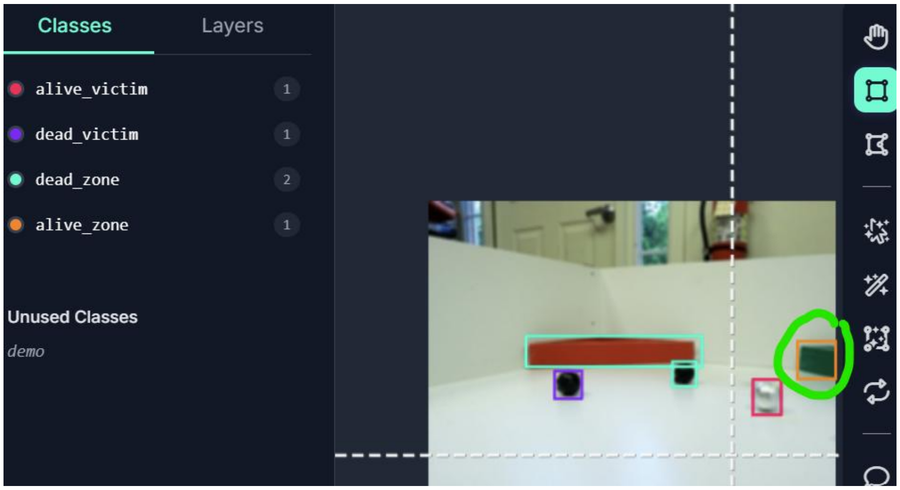
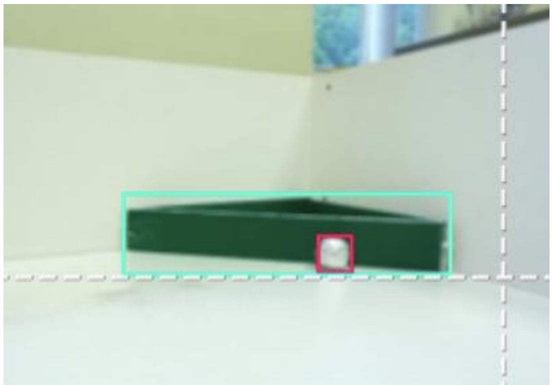
Annotation Interface

 DEMO_YOLO > ANNOTATE image49.jpg 1 / 102 TRAIN Labels Attributes Comments History Raw Data Annotations 0 Group: demo Classes Layers Tags No Tags Applied Type and select tags below to add them to the image	To annotate, simply click and drag to make boxes around the object you wish to detect later on.

 DEMO_YOLO > ANNOTATE image3.jpg 9 / 102 TRAIN Annotations 3 Group: demo Classes Layers dead_victim 1 alive_victim 2 Unused Classes demo Annotation Editor dead_zone Delete Save (Enter) Create class "dead_zone" Options	Make sure each bounding box you make is associated with the correct class label. You can create any class name you want with the ***annotation editor***.
 Annotation Editor dead_zone Delete Save (Enter) 1 alive_victim 2 dead_victim 3 dead_zone 4 demo	Once your classes are created, you can use the keyboard number and the enter key to quickly classify the objects in your images. ***OR*** *you can just click the correct class with your mouse*
 Annotations 1 Group: demo Classes Layers demo 1 Annotation Editor demo Delete Save (Enter) 1 demo Options	The ***drag tool*** on the left side can be used to select and modify any of the bounding rectangles you have made.

	Select the ***bounding box tool*** again to begin creating bounding boxes.
	Use the arrows at the top of the screen to move between images.

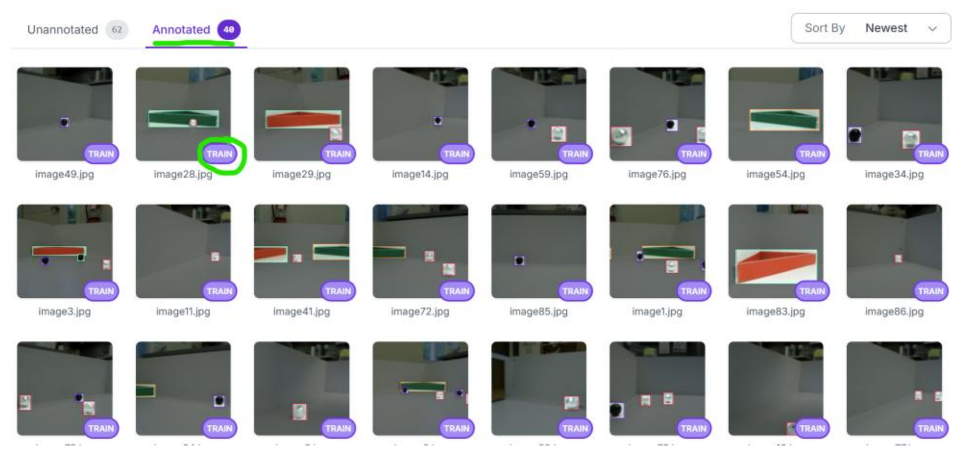
Annotation Tips

	Use *tight* bounding boxes
	Do not cut off any of the object!!
 Classes Layers ● alive_victim 1 ● dead_victim 1 ● dead_zone 2 ● alive_zone 1 Unused Classes *demo*	Annotate objects even if they are only partially visible
	Annotate objects even if they are overlapping

2.2.5 Review Annotations

When you are done annotating the images, you must submit them for "review".

Note: for the demo, I did not annotate the full set of images, but **it is highly recommended that you annotate all of your images**

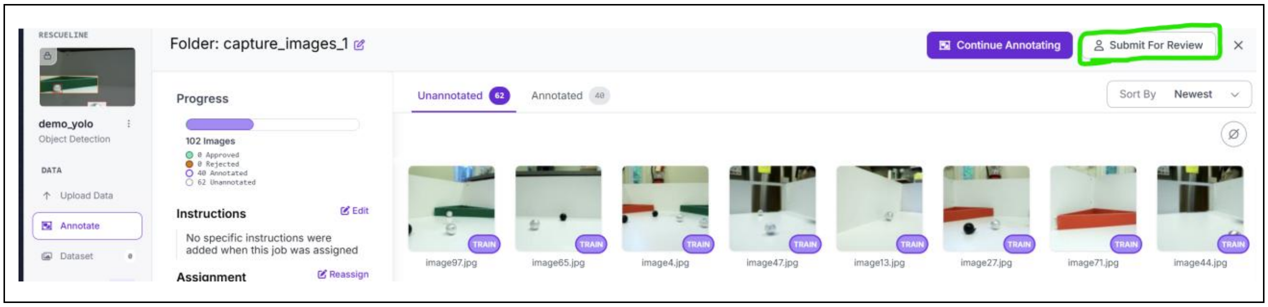


You can easily take a look at your annotated images.

NOTICE: all images say "Train"… this will be modified later automatically when the training set is split

Submit for review

This process is designed for groups of people to work on a project together. You will need to review your own annotations and approve them yourself.



Folder: capture_images_1

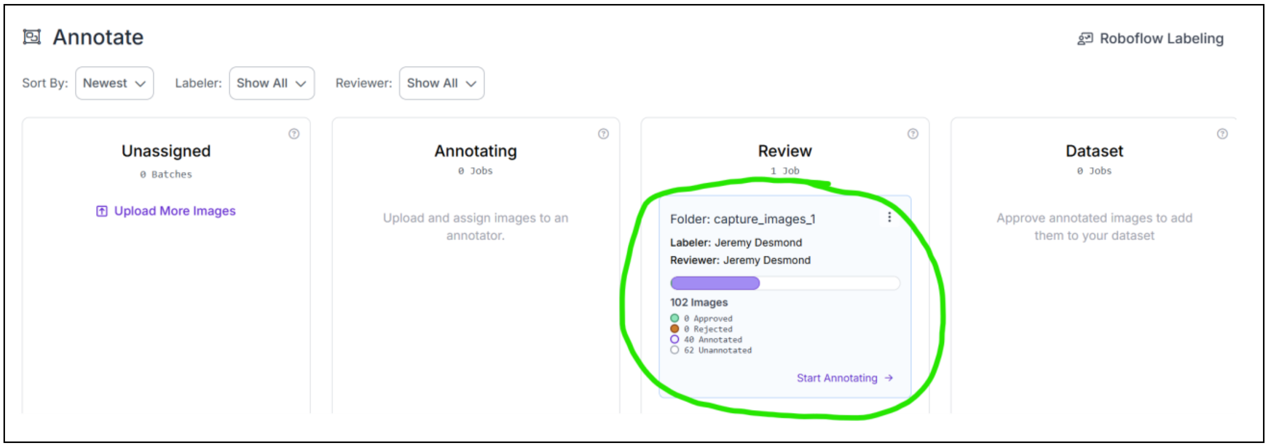
Progress

102 Images
0 Approved
0 Rejected
40 Annotated
62 Unannotated

Instructions

No specific instructions were added when this job was assigned

Assignment



Upload and assign images to an annotator.

Folder: capture_images_1
Labeler: Jeremy Desmond
Reviewer: Jeremy Desmond

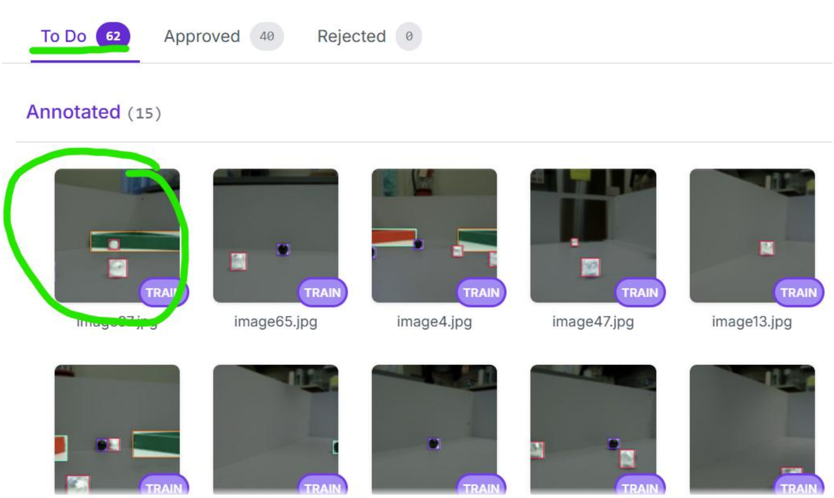
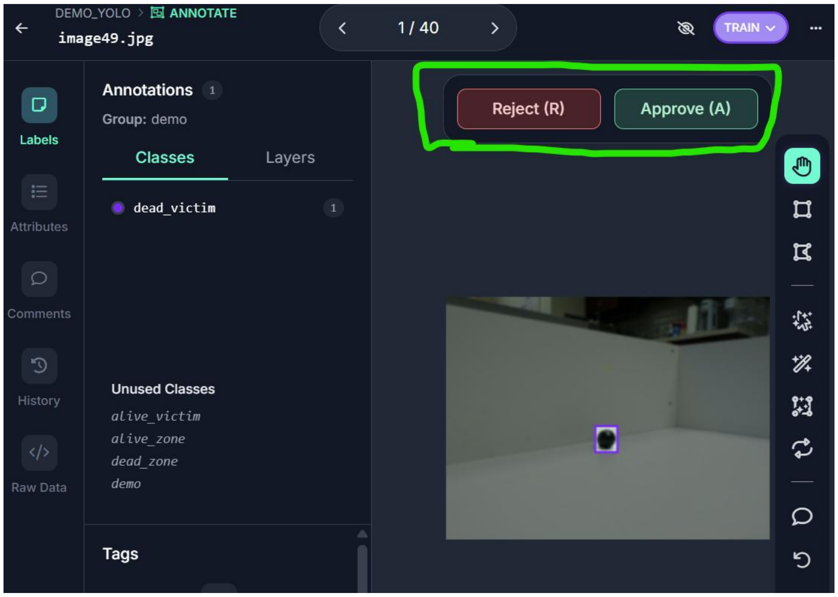
102 Images
0 Approved
0 Rejected
40 Annotated
62 Unannotated

Approve annotated images to add them to your dataset

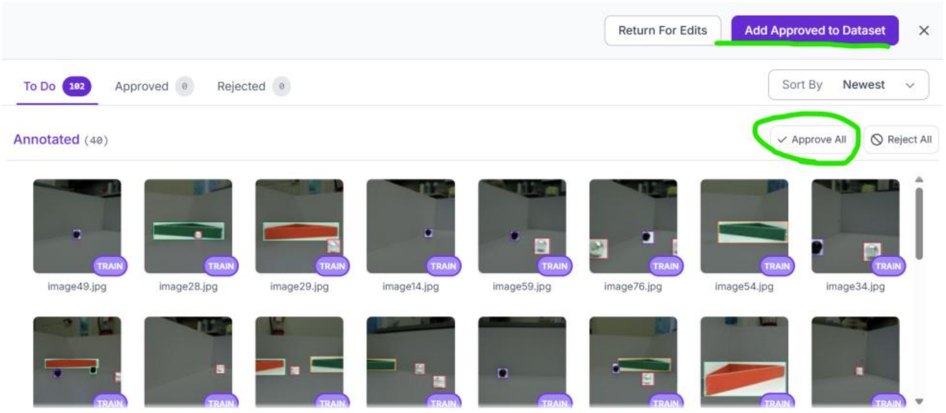
Review and apporve one at a time

Select one of the images in the "**To Do**" tab and you can begin reviewing the images one at a time.

You can make any modifications you want to fix the annotations.

 To Do 62 Approved 40 Rejected 0 Annotated (15) TRAIN image65.jpg image4.jpg image47.jpg image13.jpg	Select any image to start the review
 DEMO_YOLO > ANNOTATE image49.jpg 1 / 40 TRAIN Reject (R) Approve (A) Labels Attributes Comments History Raw Data Annotations 1 Group: demo Classes Layers dead_victim 1 Unused Classes alive_victim alive_zone dead_zone demo Tags	Modify the annotations as needed. Then approve the image when you are satisfied.

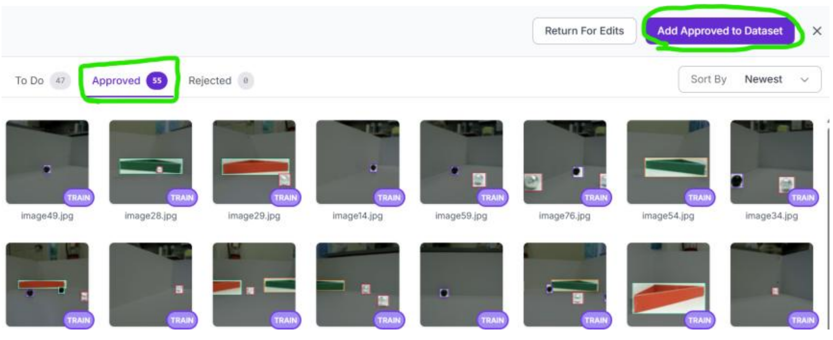
Review and apporve all at once

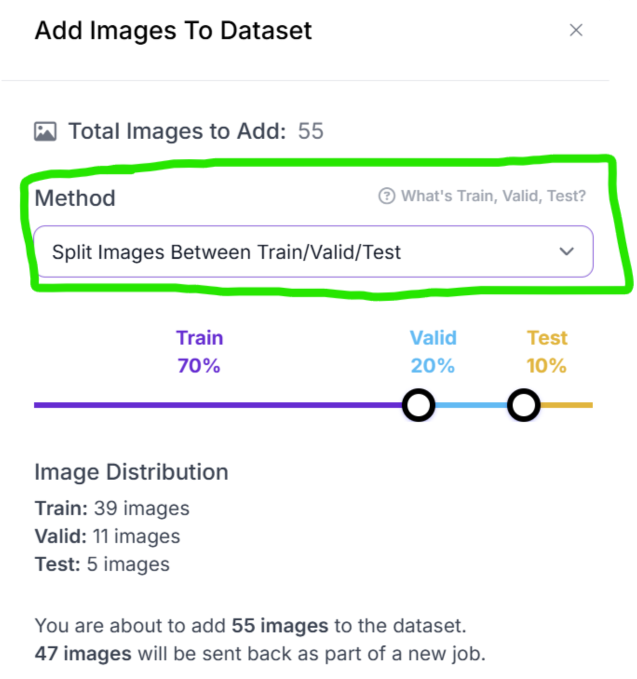


You can approve all annotated images at once and then add them to the dataset.

2.2.6

2.2.7 Adding images to the dataset

 Return For Edits Add Approved to Dataset To Do 47 Approved 55 Rejected 0 Sort By Newest TRAIN image49.jpg image28.jpg image29.jpg image14.jpg image59.jpg image76.jpg image54.jpg image34.jpg	Check your approved images, if you are satisfied you can add the approved images to a dataset.

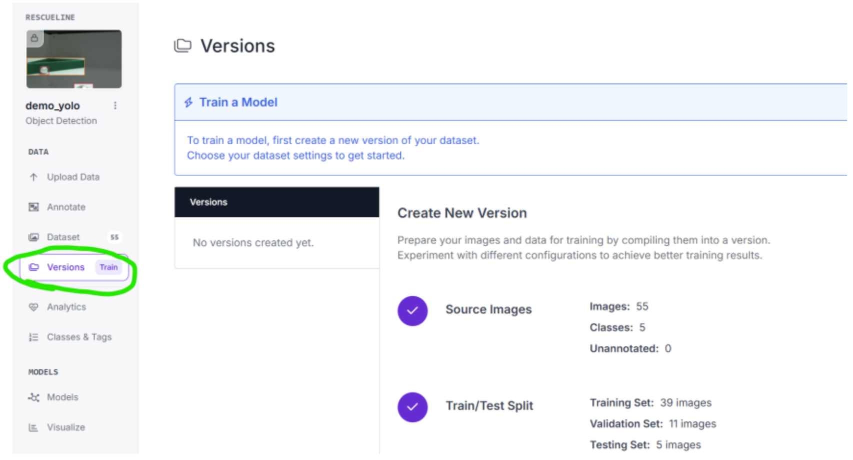
Add Images To Dataset Total Images to Add: 55 Method — What's Train, Valid, Test? Split Images Between Train/Valid/Test Train 70% — Valid 20% — Test 10% Image Distribution **Train:** 39 images **Valid:** 11 images **Test:** 5 images You are about to add **55 images** to the dataset. **47 images** will be sent back as part of a new job.	***IMPORTANT:*** select the method that will automatically split your dataset into the 3 catagories "train" "valid" and "test" Train – training images for the model Valid – images used during training to validate the progress of the model and provide feedback Test- images to test the model's performance after the training is complete

2.2.8 Create the final training set

Pay close attention to the options shown below!!

You can choose to modify these optional settings as you see fit for your training set.

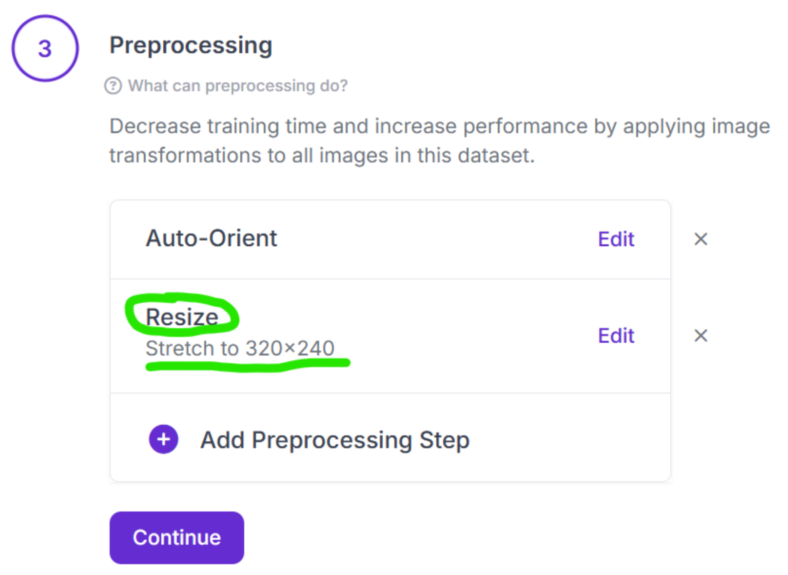
You can redo this at any time with new options selected.

 RESCUELINE demo_yolo Object Detection DATA Upload Data Annotate Dataset 55 Versions Train Analytics Classes & Tags MODELS Models Visualize Versions Train a Model Versions Create New Version Source Images Train/Test SplitTo train a model, first create a new version of your dataset. Choose your dataset settings to get started. No versions created yet. Prepare your images and data for training by compiling them into a version. Experiment with different configurations to achieve better training results. Images: 55 Classes: 5 Unannotated: 0 Training Set: 39 images Validation Set: 11 images Testing Set: 5 images	Navigate to the "**Versions**" tab.

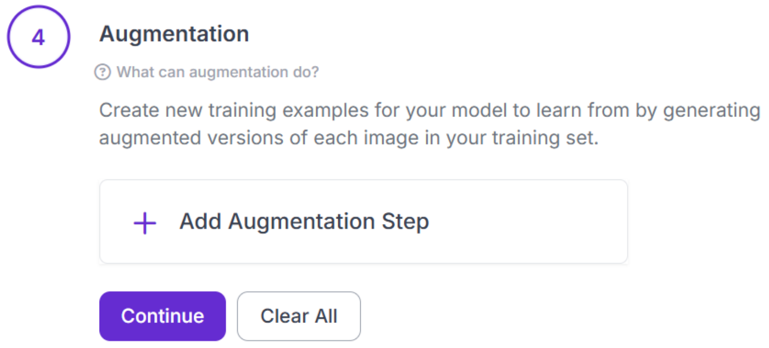
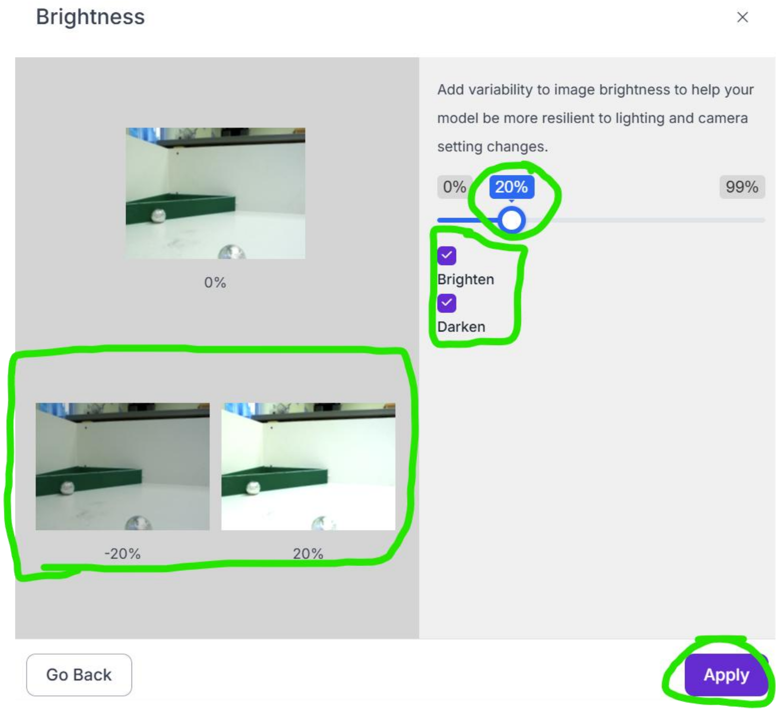
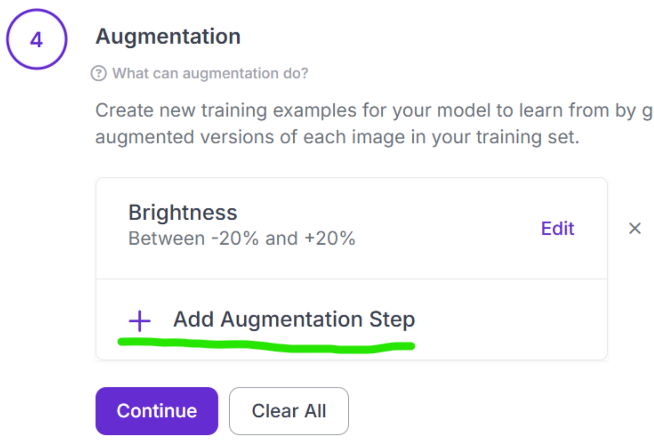
Preprocessing

Modifications to the image before being passed to the mode. By default, roboflow does nothing to the image and simply passes in a 640, 640 image.

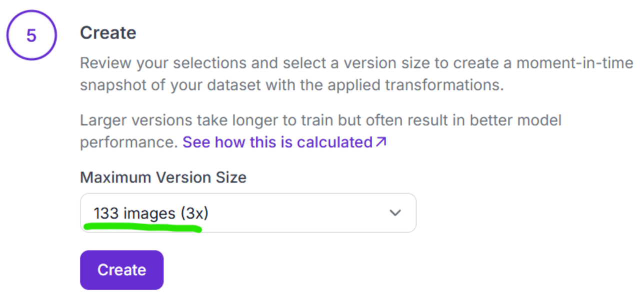
Modify the images in the pre-processing steps and make sure you pass in the same images for inference that the model was trained on.

 3 Preprocessing What can preprocessing do? Decrease training time and increase performance by applying image transformations to all images in this dataset. Auto-Orient Edit × Resize Stretch to 320×240 Edit × Add Preprocessing Step Continue	Resize the images to 320x240 for training. This is to increase model speed. I use 320x240 on my robot as well to increase program speed.

Augmentation

 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. 	The current training images can be augmented and modified to create new training images for different environmental conditions.
 Brightness Add variability to image brightness to help your model be more resilient to lighting and camera setting changes. 0% 20% 99% ☑ Brighten ☑ Darken 	Adds images to the training dataset with augmented lighting conditions. You can select how much the dataset will be augmented.
 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. Brightness Between -20% and +20% Edit × + Add Augmentation Step Continue Clear All	You can add more augmented images if you would like

Finalize the dataset



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: 133 images (3x)

Finalize and create the dataset.

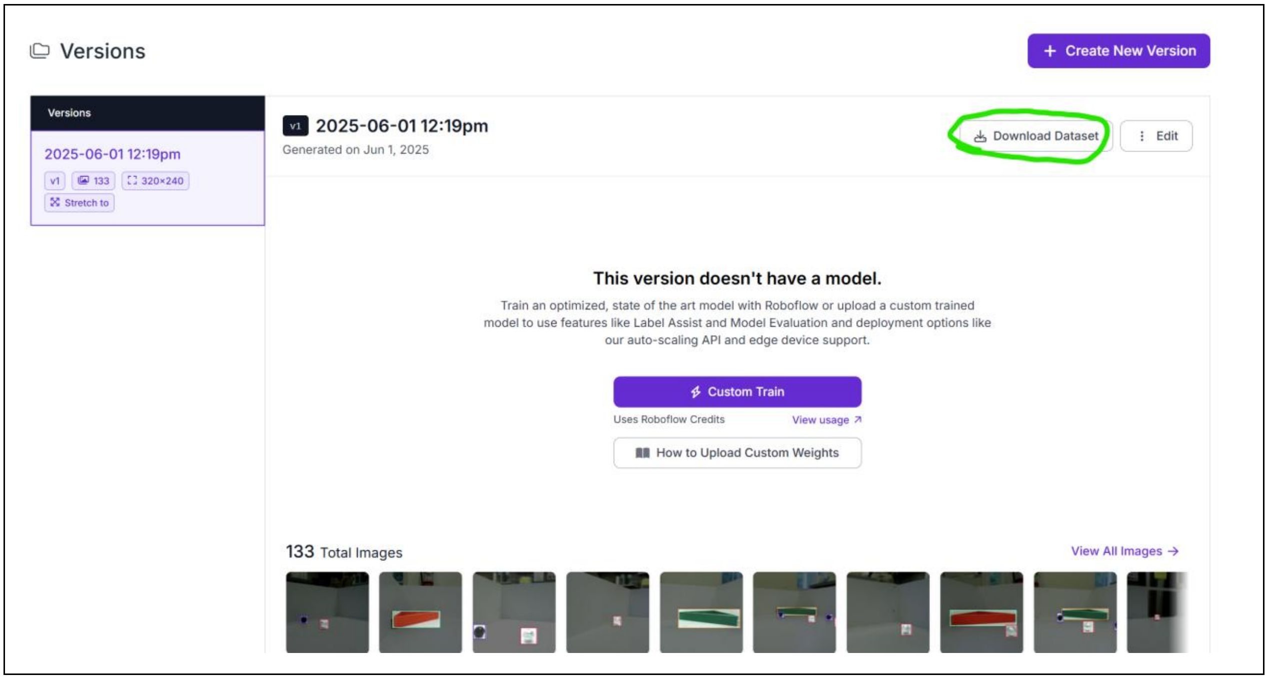
Select the number of variations you would like to create for each of your augmentation steps.

NOTE: if you do not select any augmentations, you will not have this option.

2.2.9

2.2.10 Download the dataset

Once you are done, a dataset will be available for download.



2025-06-01 12:19pm

v1 | 133 | 320×240 | Stretch to

v1 2025-06-01 12:19pm

Generated on Jun 1, 2025

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.

Uses Roboflow Credits

133 Total Images

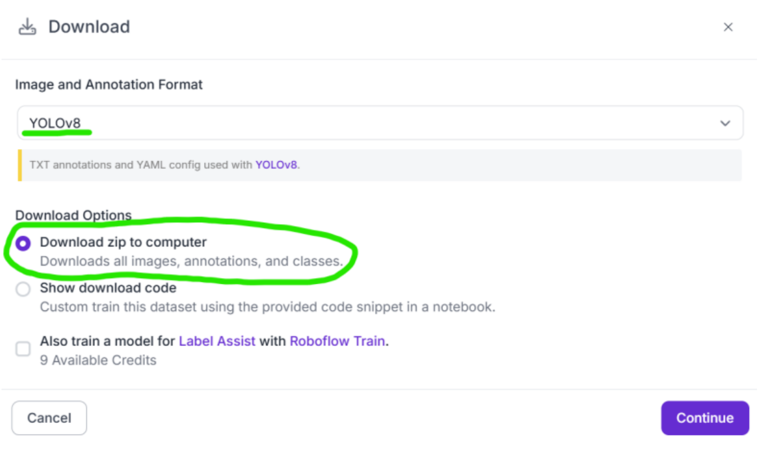


Image and Annotation Format

YOLOv8

TXT annotations and YAML config used with YOLOv8.

Download Options

● Download zip to computer
Downloads all images, annotations, and classes.

○ Show download code
Custom train this dataset using the provided code snippet in a notebook.

□ Also train a model for Label Assist with Roboflow Train.
9 Available Credits

Select the model YOLOv8 (or whichever model you are using).

Download the data locally, do not train a model yet...

Download and ***extract the training data*** to any folder you want

3 Training a Model

To train the model I ran the program on a computer different from my robot.

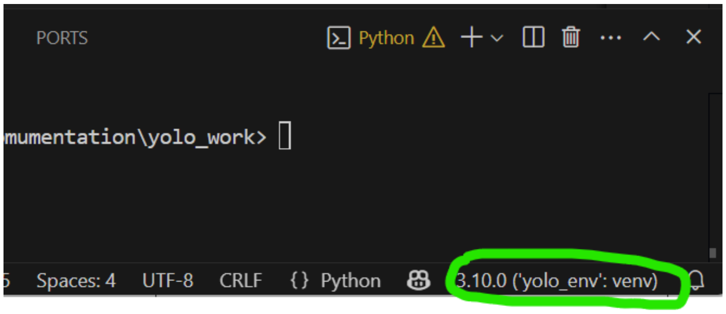
I am using visual studio code on a computer with a Nvidia GPU. *Training times will be significantly slowing if you train using a cpu*

3.1.1 Create a model for python

First, create a model for python use (.pt pytorch model) and then if you want, you can convert it later for use with a c++ file

Virtual Environment

Make sure you have the correct virtual environment selected.



```
mumentation\yolo_work>
```

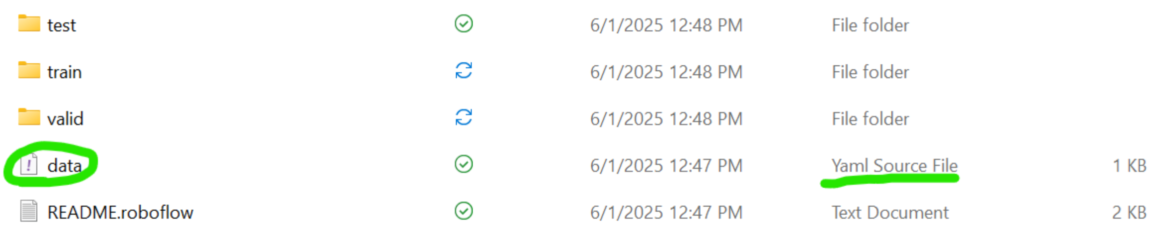
3.10.0 ('yolo_env': venv)

Select the venv in the bottom right corner of VS code

Data path

To train a model with your own custom dataset, you need to specify the correct path to the data.

If you open the folder that you downloaded, you should see a **data.yaml** file. This is the file needed to train a model.



test	6/1/2025 12:48 PM	File folder	
train	6/1/2025 12:48 PM	File folder	
valid	6/1/2025 12:48 PM	File folder	
data	6/1/2025 12:47 PM	Yaml Source File	1 KB
README.roboflow	6/1/2025 12:47 PM	Text Document	2 KB

Code (GPU TRAINING)

```
from ultralytics import YOLO

if __name__ == "__main__":

    data_path = “path to training data as a string”

    # using a model from yolov8
    model = YOLO('yolov8n.yaml')

    # custom training
    results = model.train(data = data_path,
                          imgsz = 320,
                          device = 0,
                          pretrained=False)
    # save the pytorch model
    model.save("model100.pt")
```

Import the necessary libraries

THIS NEEDS TO BE INSIDE a **python main** or it wont work

PATH IS TO THE data.yaml file

Create a model based on Yolov8

Train the model with your own custom data using a path to your data from roboflow

`imgsz = 320` because images from my robot will be 320x240

`device = 0` means GPU training (you can use `device = “cpu”` as well)

`pretrained = False` trains the model from scratch

Save the pytorch mode. You can name it whatever you want.

3.1.2 Convert Model to work with C++

.pt models work with python, not c++

Convert the model to an **onnx** file to work in **c++**

```
from ultralytics import YOLO

if __name__ == "__main__":
    model = YOLO("model100.pt")
    model.export(format = "tflite", imgsz = 320)
```

4 Model Inference

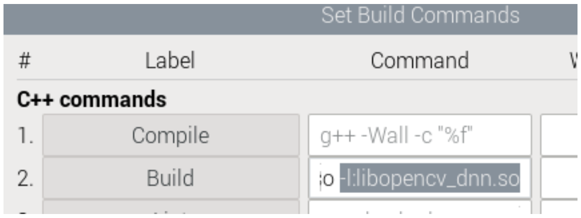
4.1 Python

4.2 C++

Add this to the geany **Build > Set Build Commands**

Add this to the end of **C++ Commands (Build)**

-l:libopencv_dnn.so



Set Build Commands

#	Label	Command
C++ commands		
1.	Compile	g++ -Wall -c "%f"
2.	Build	;o -l:libopencv_dnn.so