

Note: For the sake of simplicity, it is assumed that the detector is being trained on a system that has a NVIDIA GPU, has GPU drivers installed and CUDA (version 11.8 preferably) already installed. Instructions on installing the above mentioned pre-requisites are out of the scope of this guide.

Following steps must have been performed already:

Pre-requisite step 1: Install Python 3.11

Pre-requisite step 2: Create and activate a virtual environment in the project directory and install requirements/packages.

- For Windows (with NVIDIA GPU):

```
python -m venv venv
venv\Scripts\activate
pip install torch torchvision --index-url
https://download.pytorch.org/whl/cu118
pip install -r requirements-gpu.txt
```

- For macOS/Linux (with NVIDIA GPU):

```
python3 -m venv venv
source venv/bin/activate
pip install torch torchvision --index-url
https://download.pytorch.org/whl/cu118
pip install -r requirements-gpu.txt
```

Step 1: Create Dataset

First, we have to extract frames from videos as the detector (Yolo) is trained on frames. Run the following command in the project directory.


```
python utils/extract_frames.py --video your_video.mp4 --output
dataset/images --interval 30
```

Next, we have to annotate the extracted frames, follow the below mentioned steps.

1. Go to labelstud.io and install Label Studio (how to work with Label Studio is out of scope, please refer the Label Studio documentation for the same)

STARTER CLOUD

LAUNCH YOUR LABEL STUDIO PROJECT IN MINUTES

 Label Studio

Use Cases ▾ Learn ▾ Docs ▾ Integrations ▾ Enterprise ▾

Q Search

25,854

Quick Start

RAG Evaluation and Fine-Tuning with Label Studio

Trying to build a RAG pipeline but struggling with reliability? Label Studio can help.

Label Studio is a free, open source data labeling platform that makes labeling data for evaluating and fine-tuning RAG use cases easier and faster, while ensuring that you get the high-quality data that you need to make your project successful.

Get Label Studio - Free and Open Source

Search Results

Furintelligence

Artificial intelligence will definitely create more jobs than it will eliminate. It's the next big wave of innovation.

TurbidMind

Relevant Results

LisemeModel

While the initial impact of AI may seem daunting, technological evolution has historically led to more job creation in the long run.

OpussumLogic

Biased Results

CobesAI

As virtual intelligence will replace human jobs, leading to massive unemployment and societal chaos.

Answer

To have faith is to trust yourself to the water.

How relevant is the answer to the provided context?

☒ [1]

☒ [2]

- Launch Label Studio from the terminal by running the following command and sign in/log in to your account.

```
label-studio start
```

- Follow the Label Studio project creation guide - https://labelstud.io/guide/setup_project
- Once the project is created and labeling rules are understood, upload your extracted frames by clicking on “Import”.

Label Studio

Projects / AntTracker

Settings Light Beta NI

Default

Actions ▾ Columns ▾ Filters ▾ Order by ▾

Label All Tasks ▾

↺ Import Export

☐ ID

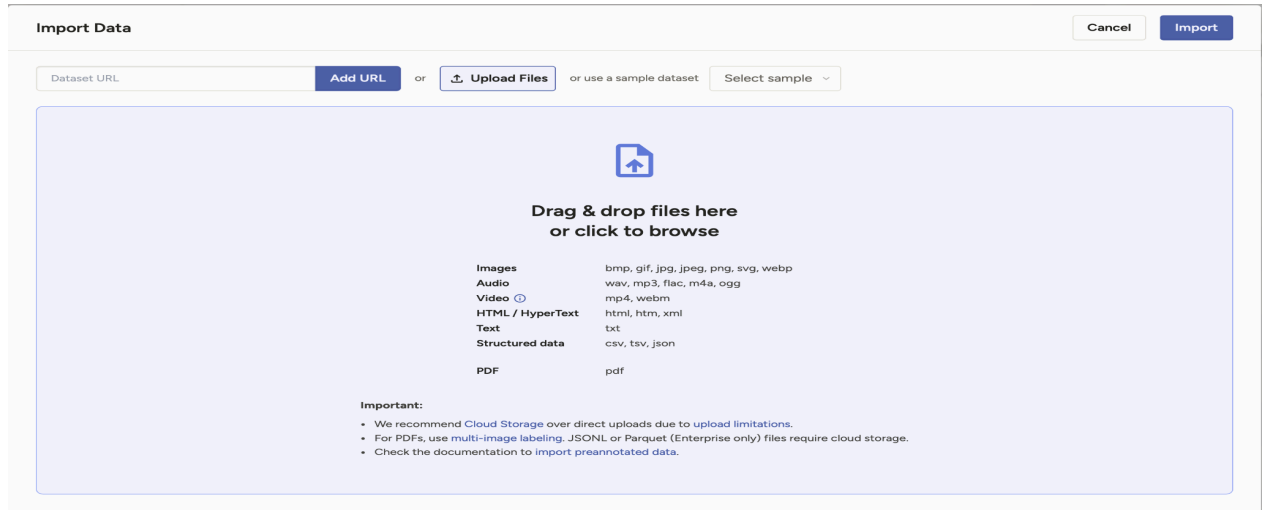
☒ Completed

☐ Annotated by

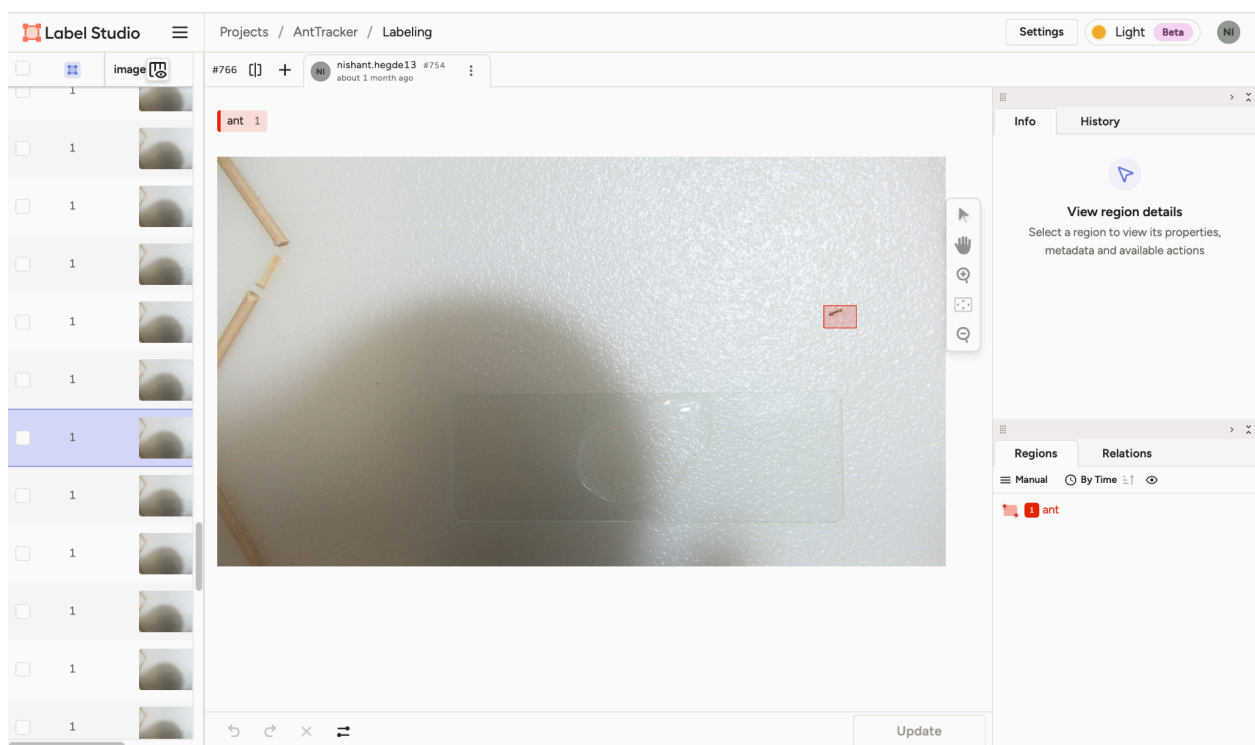
☐ image

☒ img

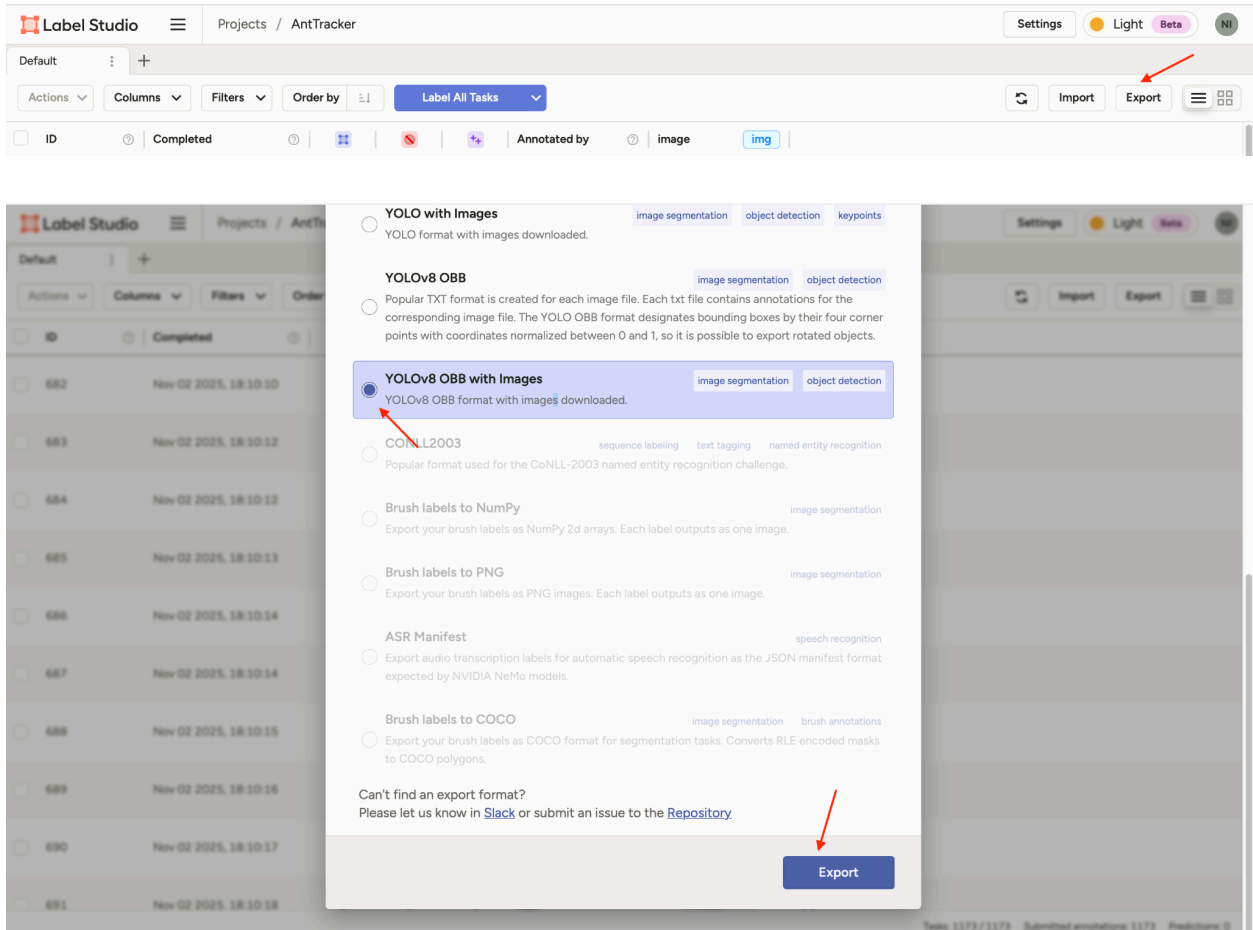
Upload the extracted frames.



5. Draw bounding boxes around each ant on all desired frames. Check documentation for annotation procedures - <https://labelstud.io/guide/labeling>



6. Once all annotations are done, export the annotations and the data by clicking on “Export” and then selecting “YOLOv8 OBB with Images” and then clicking “Export” again.



7. Once the annotations and the images are downloaded, place them in some path and note down the path.
8. Next, we have to split the downloaded data and annotations into train/val/test. Run the below command in the ant-tracker/ directory.

```
python -m src.training.prepare_dataset --source
unprepared_dataset --output dataset
```

Where unprepared_dataset is the path to the dataset we previously downloaded.

9. Train the detector model by running the below command in the project directory

```
python -m src.training.train_yolov8_obb
```

Training time:

GPU: 30 minutes to 4 hours

CPU: Several hours to days (not recommended)

The best model is saved at models/trained/[best.pt](#). Use this model checkpoint file for tracking.