

DETECTING PEDESTRIANS

OUSTER NEAR-IR YOLO TRAINING PIPELINE

FROM PCAP TO LIVE OBJECT DETECTION

WORKSHOP GOALS



Export Near-IR in PNG format



Label using Roboflow



Build dataset



Train YOLO model



Deploy live

WORKFLOW

Full pipeline:

Ouster PCAP -> Near-IR PNGs -> Roboflow Labeling ->
Google Colab Split + Training -> yolo-own-model.pt ->
Live detection on the Ouster

Participants should be able to:

1. read an Ouster PCAP,
2. generate Near-IR PNGs from it,
3. label those images in Roboflow,
4. split the labeled dataset into `train` and `valid` in Google Colab,
5. fine-tune a YOLO model on the Ouster images there,
6. run the resulting `yolo-own-model.pt` / `best.pt` live on the Ouster.

REQUIREMENTS

Python 3.10 or newer

dependencies from
`requirements.txt`

an Ouster PCAP or a live Ouster
sensor

a Roboflow account for labeling

a Google account for Colab

LIDAR DATA – OUSTER SDK

Object Detection and Tracking using Deep Learning and Ouster Python SDK

<https://ouster.com/insights/blog/object-detection-and-tracking-using-deep-learning-and-ouster-python-sdk>



LABELING - ROBOFLOW

<https://roboflow.com/>

- Bounding boxes, person class

<https://universe.roboflow.com/ouster-labeling/osl-64-nearir-labeling>



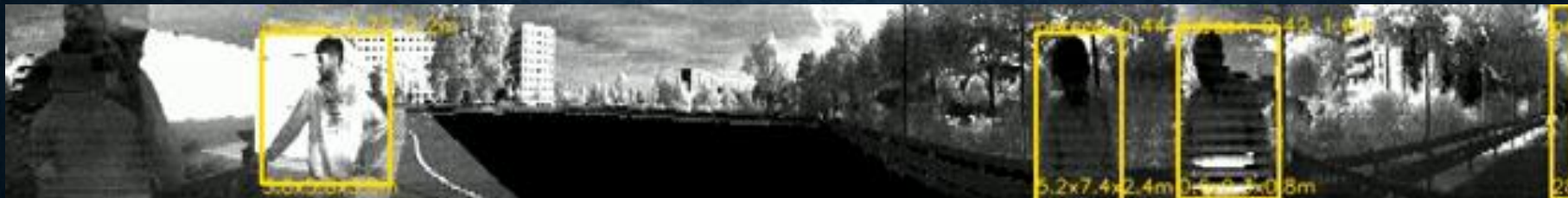
TRAINING YOLO MODEL – GOOGLE COLAB

**YOLO11n, imgsz=1024, epochs=50,
batch=8**



Deployment

Run ouster_yolo.py with best.pt



RECOMMENDED WORKSHOP ORDER



1. Prepare a PCAP



2. Export Near-IR PNGs



3. Label the images in Roboflow



4. Split the dataset into `train` and `valid` in Google Colab



5. Training in Google Colab



6. Download `best.pt`



7. Test live detection on the Ouster

PRACTICAL INFO

[https://github.com
/tewesolowski/BIP_resources](https://github.com/tewesolowski/BIP_resources)

Data for exercising and scripts

<https://my.hidrive.com/share/jea1d9lswa>

PRACTICAL INFO ENVIRONMENT

Python 3.14

ouster_sdk-1.0.0

Roboflow account: Google

```
## ouster_yolo.py
```

1) ERROR line 16

```
from ouster.sdk.core import AutoExposure, ChanField, XYZLut, destagger,  
BeamUniformityCorrector
```

WES added above BeamUniformityCorrector, in new Ouster there is no _utils, removed:

```
#from ouster.sdk.core._utils import BeamUniformityCorrector
```

PRACTICAL INFO

Workshop - Colab

1. Create lidar_workshop folder at Google drive connected to Colab in the root folder

Run first code-box

- Drive asks to connect, permissions,

2. Copy workshop\dataset\train to lidar_workshop\workshop\dataset\train

Run the box 2 & 3

- divides dataset into train/val

Total images: 70

Train: 56

Val: 14

3. Training - takes LONG!

You can create charts to visualize the results using the code from the virtual component (GitHub)