

Object Detection using YOLOv5: [Colab Link](#)

→ Python

→ Packages

- ◆ pytorch/torch
- ◆ Torchvision
- ◆ Numpy
- ◆ pyYAML
- ◆ Opencv
- ◆ Gdown

→ <https://www.curiously.com/posts/object-detection-on-custom-dataset-with-yolo-v5-using-pytorch-and-python/#convert-to-yolo-format>

- ◆ Relevant part of tutorial starts here, converting to darknet format

→ Dictionary with image url and annotations

- ◆ Annotations for bounding box contain 2 points with x and y coordinates, width, height, and label

→ Train-test/val split before converting to darknet format

→ Using OpenCV to read the image and draw bounding box and show label

→ Darknet Format

- ◆ Store labels in txt files (1 per image)
- ◆ Each row contains the following for a single object in the image:
 - Class index -- index of the label in existing categories list
 - X and y coordinates of bounding box center
 - Bounding box height and width
- ◆ Bounding box coordinates must be normalized, so in (0,1]
- ◆ Data in directory containing **labels** and **images** directories

→ YOLO: You Only Look Once

- ◆ One stage object detectors
- ◆ Light and fast
- ◆ Not super accurate, but great for real time
- ◆ Okay, so at this point, YOLO doesn't feel like the right choice for this project -NB
- ◆ Need to clone into <https://github.com/ultralytics/yolov5>
- ◆ Use config.yaml, which specifies paths and gives number and names of classes

Review of different object detection models

→ Need some help understanding what I'm looking at here -NB

Training Custom Data in Detectron2: Colab Link

→ Python

→ Packages/Dependencies

- ◆ detectron2:

 - <https://dl.fbaipublicfiles.com/detectron2/wheels/cu101/torch1.5/index.html>

- ◆ pytorch/torch

- ◆ Torchvision

- ◆ Numpy

- ◆ pyYAML

- ◆ OpenCV

→ Tutorial uses **roboflow** to deploy, convert, preprocess, finetune, etc

→ **COCO JSON format (.coco.json)**

- ◆ Need to register? Instances (train/test/val)

→ Uses COCO validation evaluation?

→ Switching to facebook research's tutorials and codebase, notes in link below

Notes on Detectron2