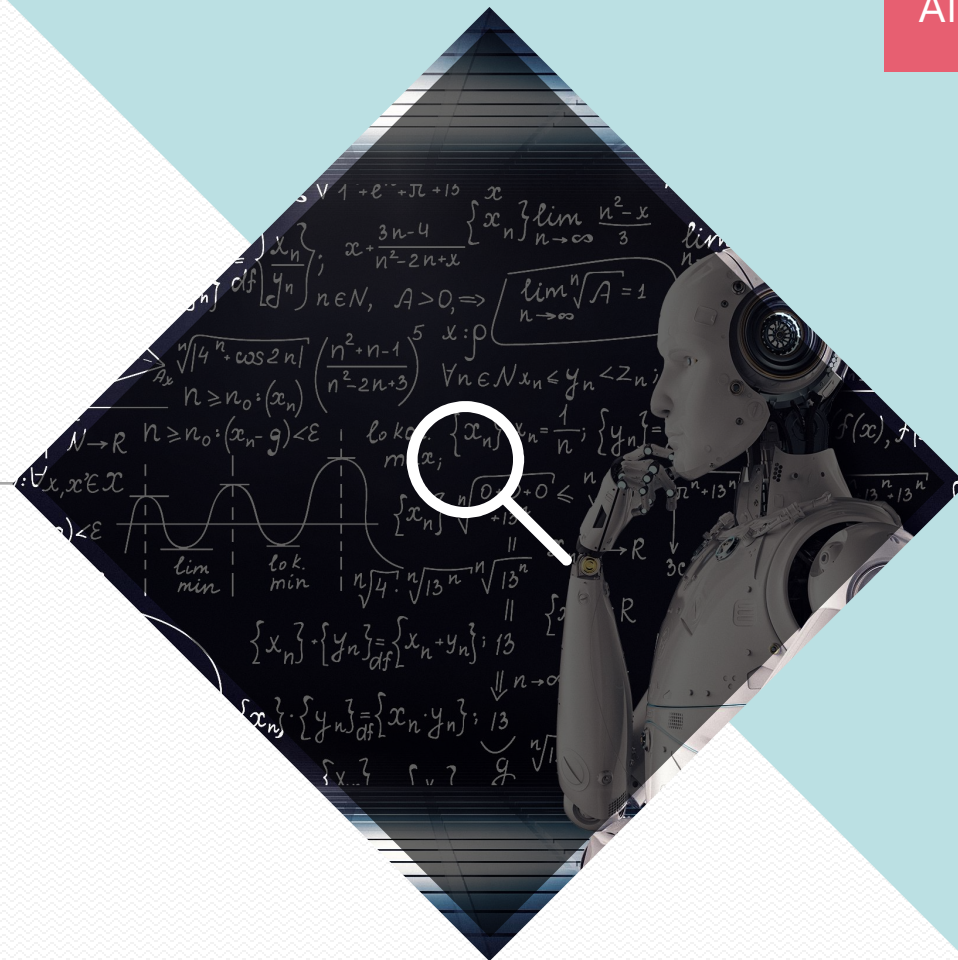


YOLO



Yolo PRACTICE

YOLOv5설치

▼ Install

Clone repo and install [requirements.txt](#) in a [Python>=3.8.0](#) environment, including [PyTorch>=1.8](#).

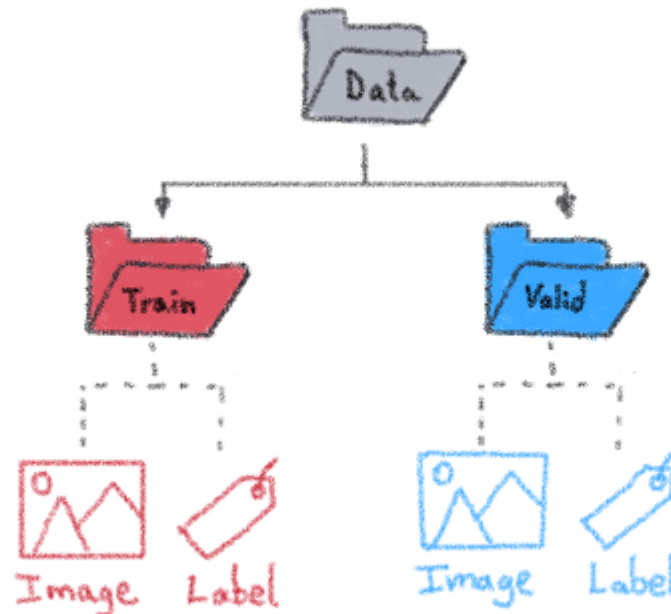
```
git clone https://github.com/ultralytics/yolov5 # clone
cd yolov5
pip install -r requirements.txt # install
```

YOLOv5 training

```
python3 train.py --data dataset.yaml --epochs 300 --weights " --cfg yolov5n.yaml --batch-size 128
```

- dataset.yaml 파일 작성 방법
- Dataset 폴더 구성 방법
- Cat and Dog dataset download (<https://universe.roboflow.com/kct-yh2hv/cat-dog-4paux/dataset/2>)
- Train/Image의 파일 이름과 Train/Label의 파일 이름이 일치 함.

```
data/train/images/img0.jpg # image  
data/train/labels/img0.txt # label
```



- **Set up files and directory structure:** to train the YOLOv5 model, we need to add a **.yaml** file to describe the parameters of our dataset.

YOLOv5 training

```
python3 train.py --data dataset.yaml --epochs 300 --weights " " --cfg yolov5n.yaml --batch-size 128
```

- dataset.yaml 파일 작성 방법

```
names: ['aeroplane', 'apple', 'backpack', 'banana', 'baseball bat', 'baseball glove', 'bear', 'bed', 'bench', 'bicycle',  
'bird', 'boat', 'book', 'bottle', 'bowl', 'broccoli', 'bus', 'cake', 'car', 'carrot', 'cat', 'cell phone', 'chair', 'clock',  
'cow', 'cup', 'diningtable', 'dog', 'donut', 'elephant', 'fire hydrant', 'fork', 'frisbee', 'giraffe', 'hair drier', 'handbag',  
'horse', 'hot dog', 'keyboard', 'kite', 'knife', 'laptop', 'microwave', 'motorbike', 'mouse', 'orange', 'oven', 'parking  
meter', 'person', 'pizza', 'pottedplant', 'refrigerator', 'remote', 'sandwich', 'scissors', 'sheep', 'sink', 'skateboard',  
'skis', 'snowboard', 'sofa', 'spoon', 'sports ball', 'stop sign', 'suitcase', 'surfboard', 'teddy bear', 'tennis racket',  
'tie', 'toaster', 'toilet', 'toothbrush', 'traffic light', 'train', 'truck', 'tvmonitor', 'umbrella', 'vase', 'wine glass',  
'zebra']
```

YOLOv5 training

```
python3 train.py --data dataset.yaml --epochs 300 --weights "" --cfg yolov5n.yaml --batch-size 128
```

- dataset.yaml 예]

```
# Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt, or 3)
# imgs1, path/to/imgs2, ..]
path: /home/team996/workspace/D04_CNN/issue2_catanddogs/ # dataset root dir
train: train/images # train images (relative to 'path') 80% images
val: valid/images # val images (relative to 'path') 20% images
test: # test images (optional)

# Classes (2 classes)
names: ['Cat', 'Dog']
```





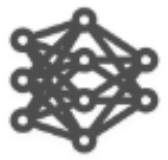
YOLOv5 training

```
python3 train.py --data dataset.yaml --epochs 300 --weights " " --cfg yolov5n.yaml --batch-size 128
```

yolov5s	64
yolov5m	40
yolov5l	24
yolov5x	16

2. Select a Model

Select a pretrained model to start training from. Here we select [YOLOv5s](#), the second-smallest and fastest model available. See our README [table](#) for a full comparison of all models.

				
Nano YOLOv5n	Small YOLOv5s	Medium YOLOv5m	Large YOLOv5l	XLarge YOLOv5x
4 MB _{FP16} 6.3 ms _{V100} 28.4 mAP _{COCO}	14 MB _{FP16} 6.4 ms _{V100} 37.2 mAP _{COCO}	41 MB _{FP16} 8.2 ms _{V100} 45.2 mAP _{COCO}	89 MB _{FP16} 10.1 ms _{V100} 48.8 mAP _{COCO}	166 MB _{FP16} 12.1 ms _{V100} 50.7 mAP _{COCO}

YOLOv5 Inference

```
python3 detect.py --weights yolov5s.pt --source 0
```

webcam

img.jpg

image

vid.mp4

video



```
from ultralytics import YOLO
import cv2
```

```
model = YOLO("yolov5s.pt")
```

```
# from ndarray
```

```
im2 = cv2.imread("cat1.jpg")
```

```
results = model.predict(source=im2, save=True, save_txt=False)
```

```
# save predictions as labels
```

```
if results[0].boxes.conf > 0.5:
    print("A confidence is .....")
    print(results[0].boxes.conf)
```



