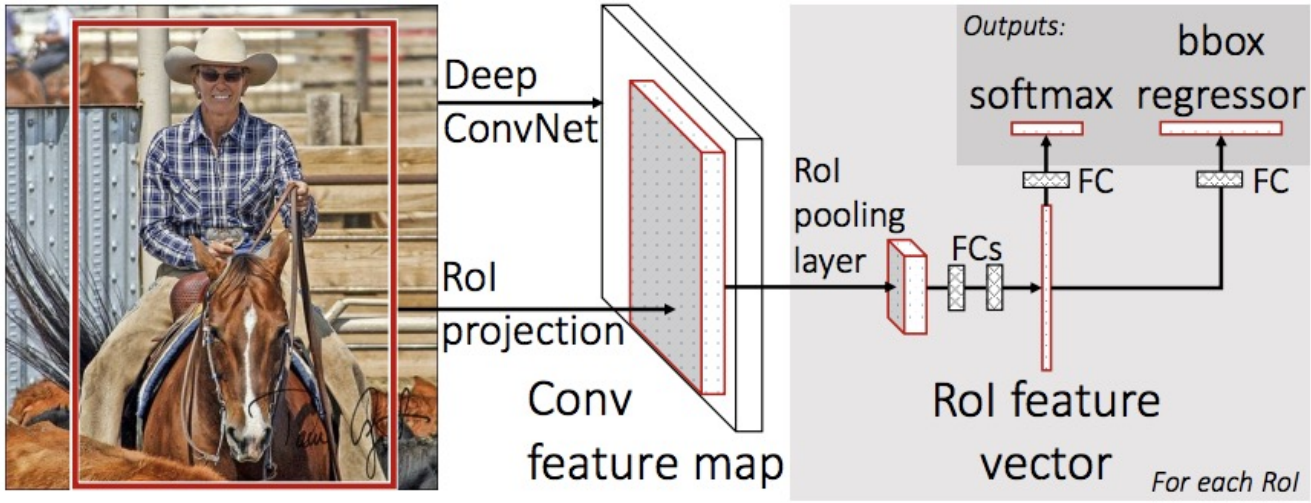
YOLOv6 – train, test and deploy

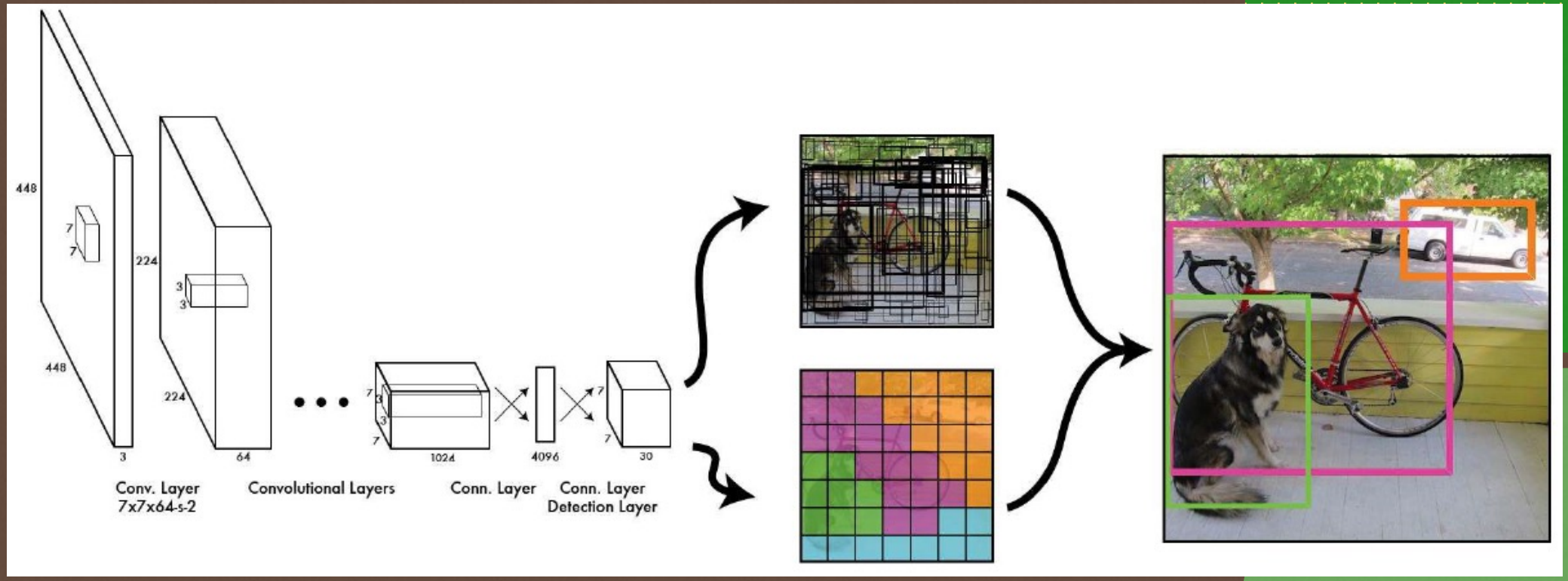
Mì AI

2 stages algorithm



R-CNN, Faster R-CNN

YOLO



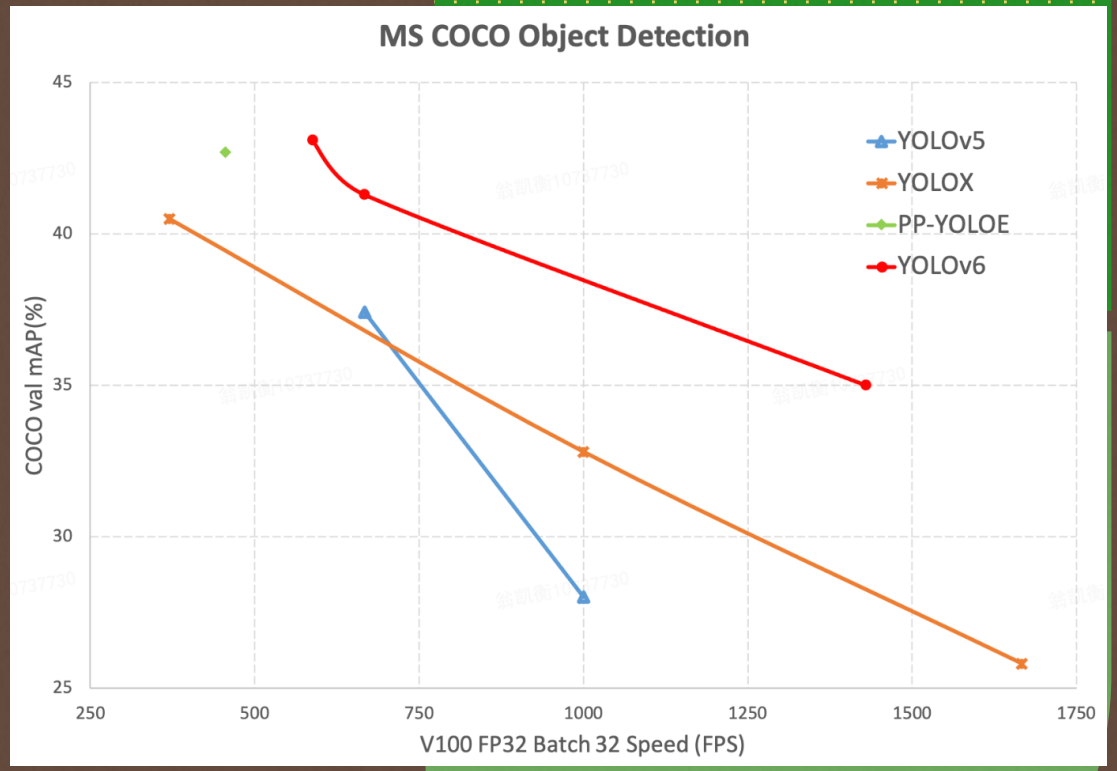
YOLO suggested a different methodology where both stages are conducted in the same neural network. First, the image is divided into cells, each having an equal dimensional region of SxS. Then, each cell detects and locates the objects it contains with bounding box coordinates (relative to its coordinates) with the object label and probability of the thing being present in the cell

What versions does YOLO have?

- YOLOv1 (Jun, 2015): You Only Look Once: Unified, Real-Time Object Detection
- YOLOv2 (Dec, 2016): YOLO9000:Better, Faster, Stronger
- YOLOv3 (Apr, 2018): YOLOv3: An Incremental Improvement
- YOLOv4 (Apr, 2020): YOLOv4: Optimal Speed and Accuracy of Object Detection
- YOLOv5 (May, 2020): Github repo (No paper was released yet)

YOLOv6


Why YOLOv6
COCO val mAP(%)
V100 FP16 Batch 32 Speed (ms/img)
D0
D1
D2
EfficientDet
YOLOv5n
YOLOv5s
YOLOX-s
YOLOv6n
YOLOv6s


MS COCO Object Detection
COCO val mAP(%)
V100 FP32 Batch 32 Speed (FPS)
YOLOv5
YOLOX
PP-YOLOE
YOLOv6

YOLOv6 vs YOLOv5

YOLOv6 Vs. YOLOv5

Model	Size (pixels)	mAP^{val} 0.5:0.95	Params (M)	Flops (G)
YOLOv6-n	640	35.0 ✅	4.3	11.1
YOLOv5-n	640	28.0	1.9	4.5
YOLOv6-s	640	43.1 ✅	17.2	44.2
YOLOv5-s	640	37.4	7.2	16.5

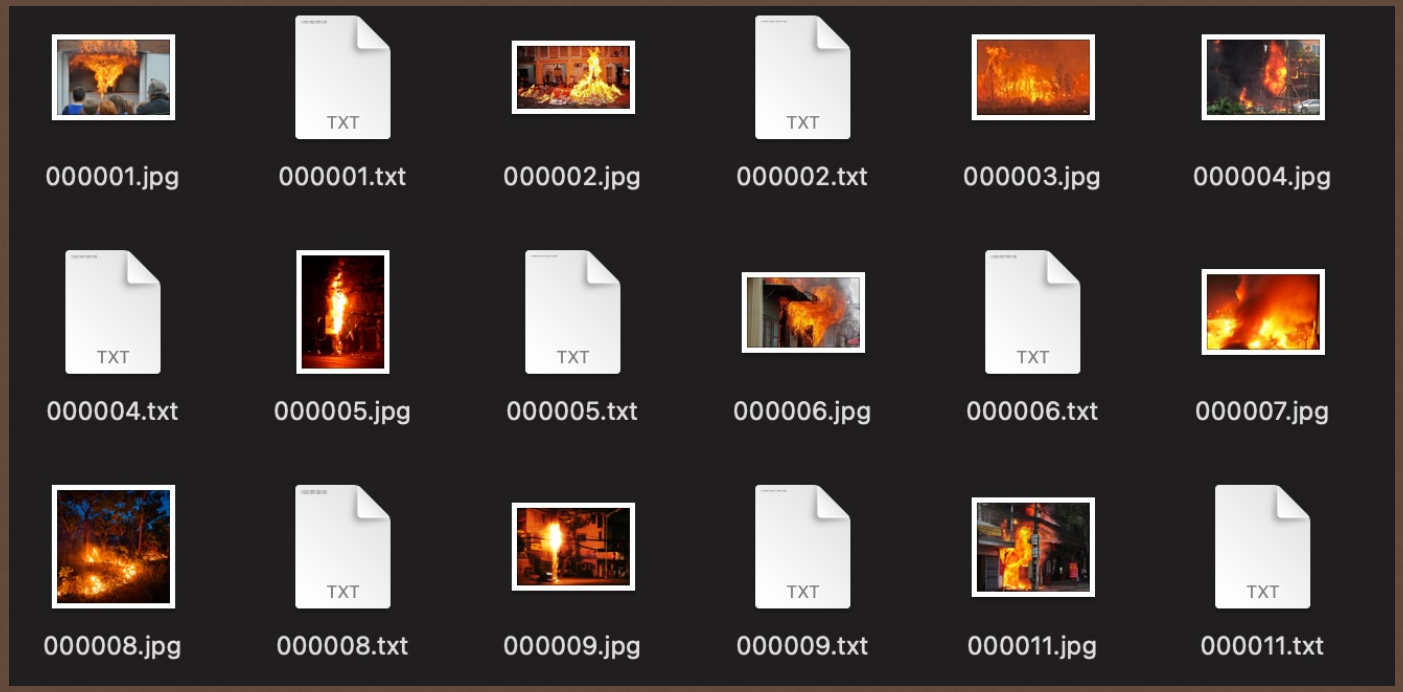
YOLOv6 vs YOLOv5

YOLOv6

YOLOv5

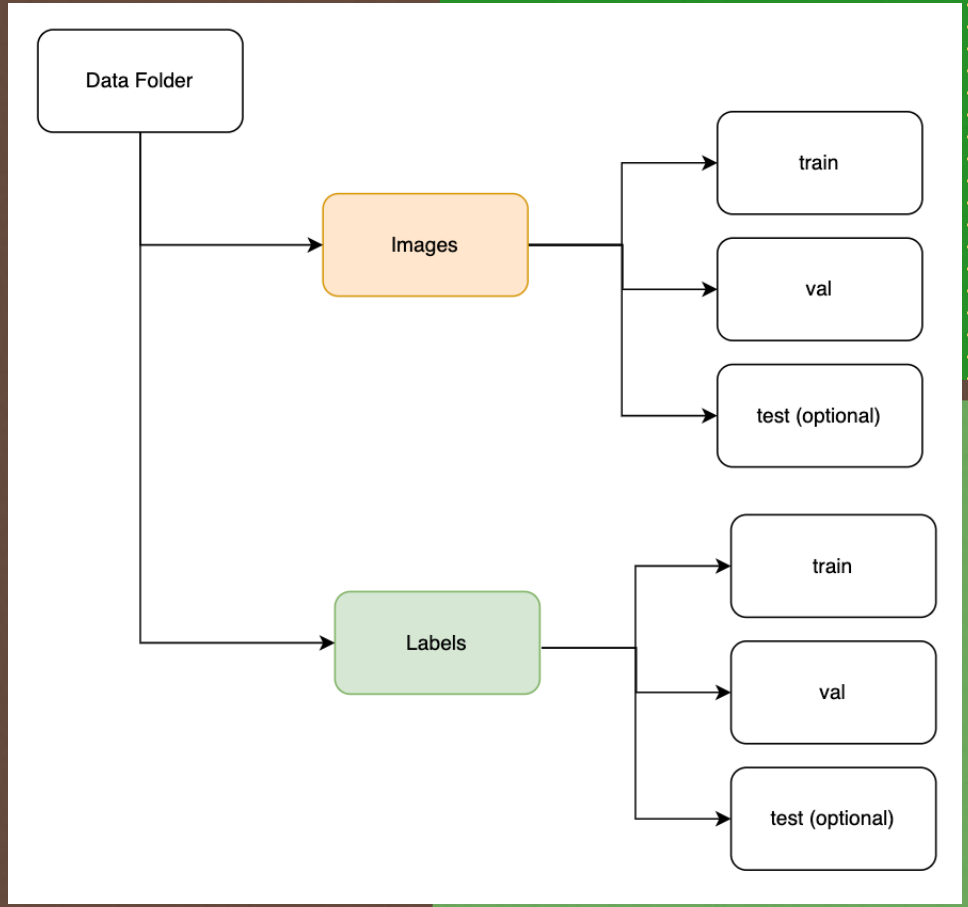
Train YOLOv6 with custom data

- Use Fire data:
  - Can download from ”Thư viện Miai” (miai.vn)
  - Label file is YOLO TXT format.



Train YOLOv6 with custom data

- Use Fire data:
  - Data structure:



Train YOLOv6 parameters

- --data-path, default='./data/coco.yaml', type=str, help='path of dataset')
- --conf-file, default='./configs/yolov6s.py', type=str, help='experiments description file')
- --img-size, type=int, default=640, help='train, val image size (pixels)')
- --batch-size, default=32, type=int, help='total batch size for all GPUs')
- --epochs, default=400, type=int, help='number of total epochs to run')
- --eval-interval, type=int, default=20, help='evaluate at every interval epochs')
- --eval-final-only, action='store_true', help='only evaluate at the final epoch')
- --heavy-eval-range, default=50,help='evaluating every epoch for last such epochs (can be jointly used with --eval-interval)')
- --output-dir, default='./runs/train', type=str, help='path to save outputs')
- --name, default='exp', type=str, help='experiment name, saved to output_dir/name')
- --resume, type=str, default=None, help='resume the corresponding ckpt')

Hands on