基于 YOLOv8 的 FPS 游戏图像识别

CONTENTS

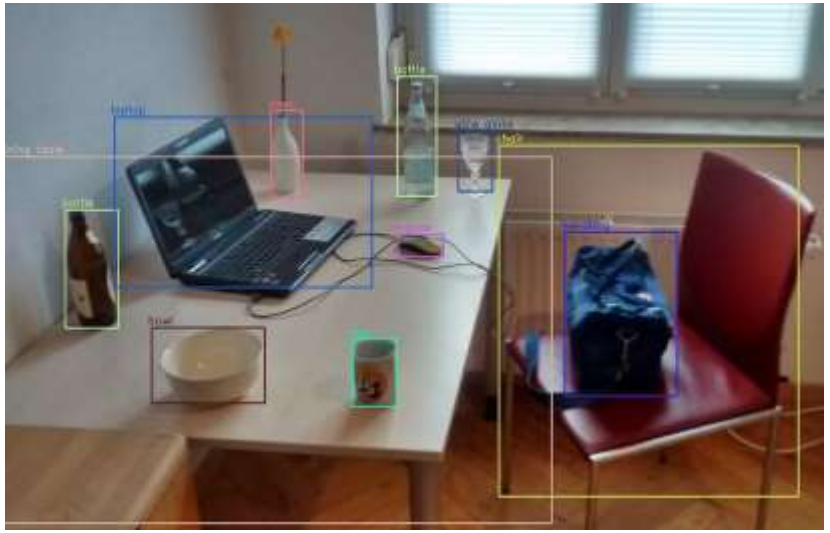
01

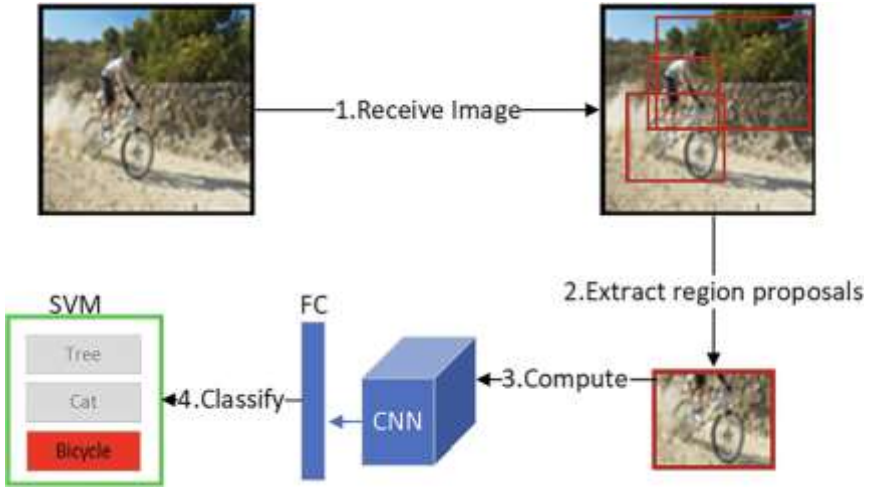
原理

图像识别

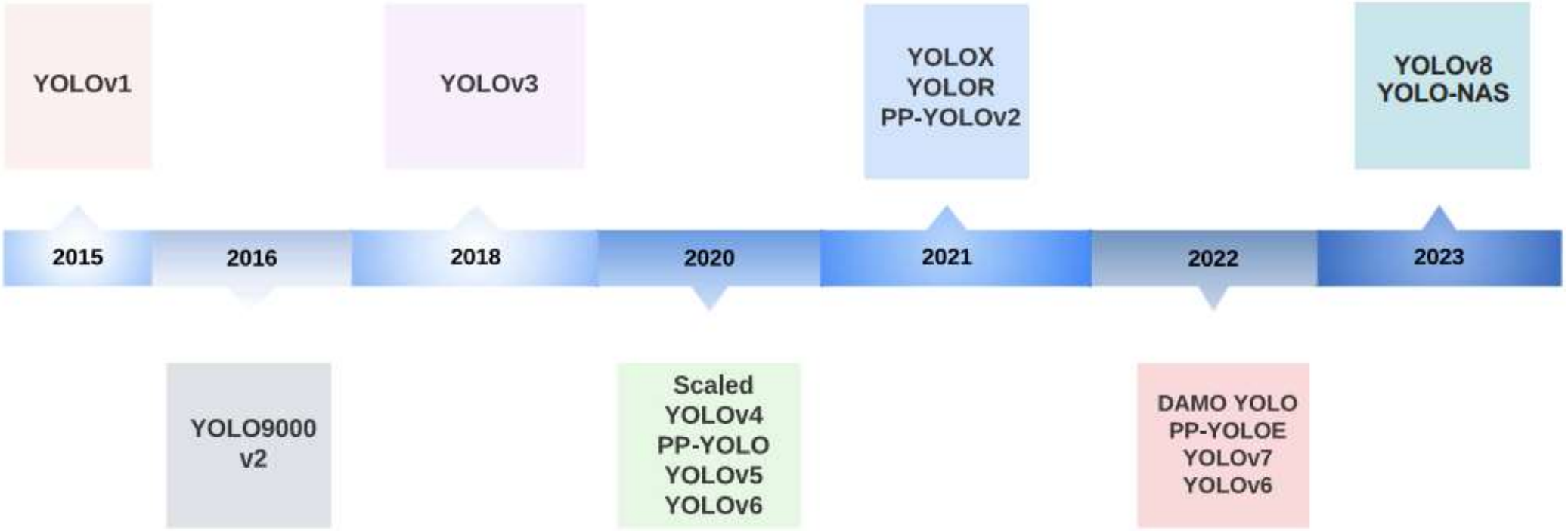
图像识别（image recognition）是指寻找并鉴别图像中的物体的过程，是计算机视觉领域的经典任务，应用广泛。

卷积神经网络（CNN）是实现图像识别的主流方法。

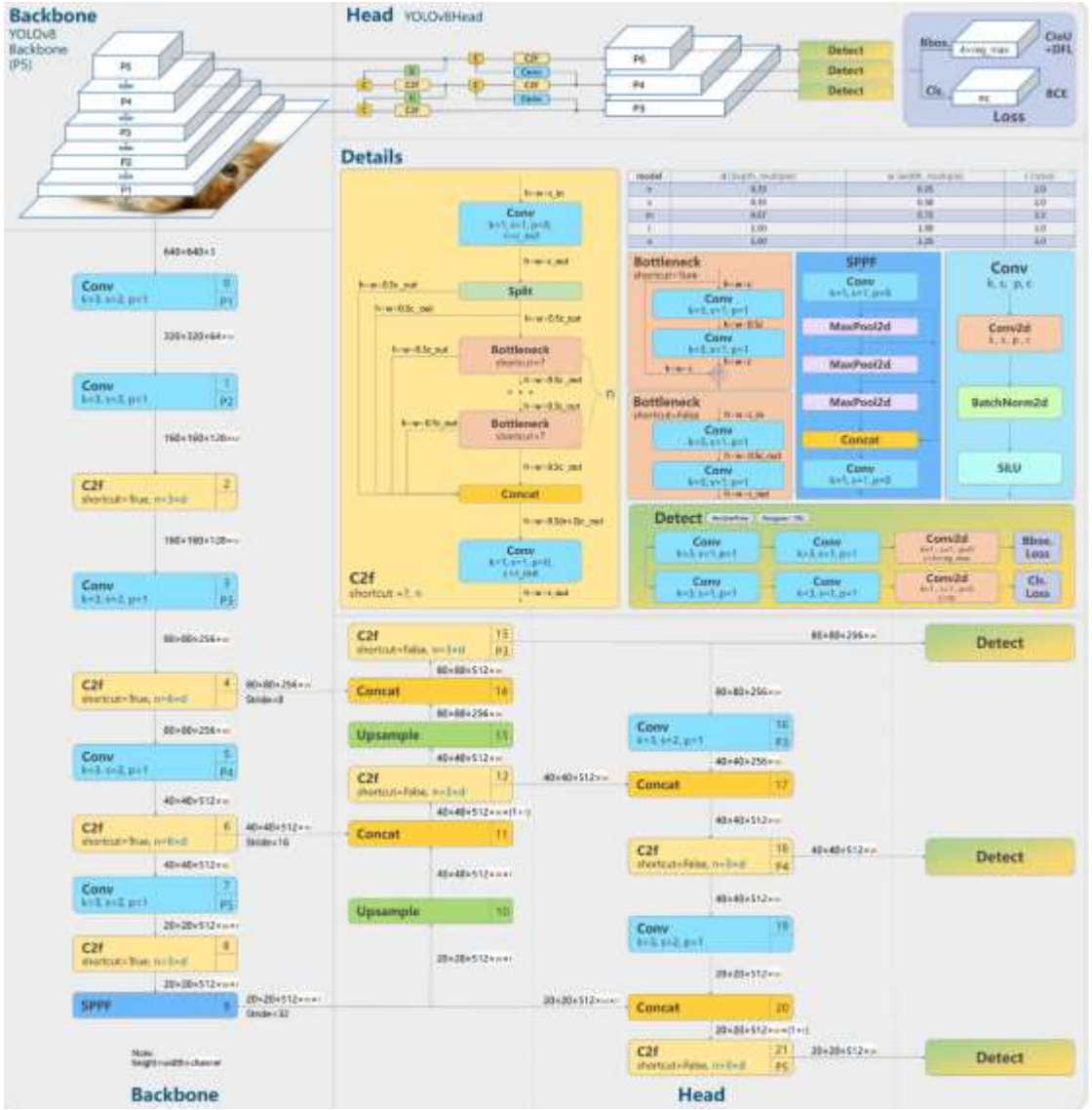




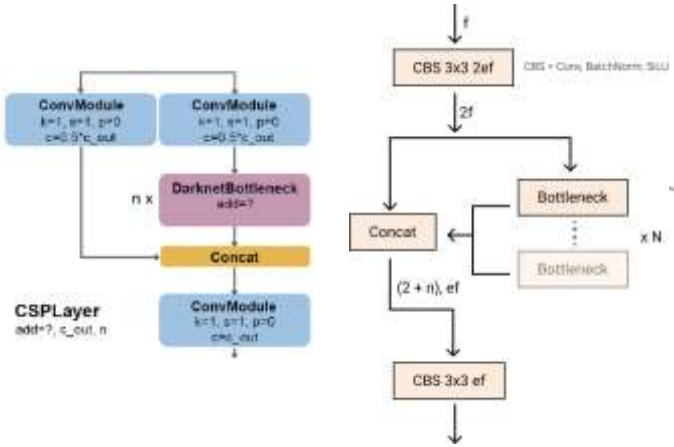
You Only Look Once



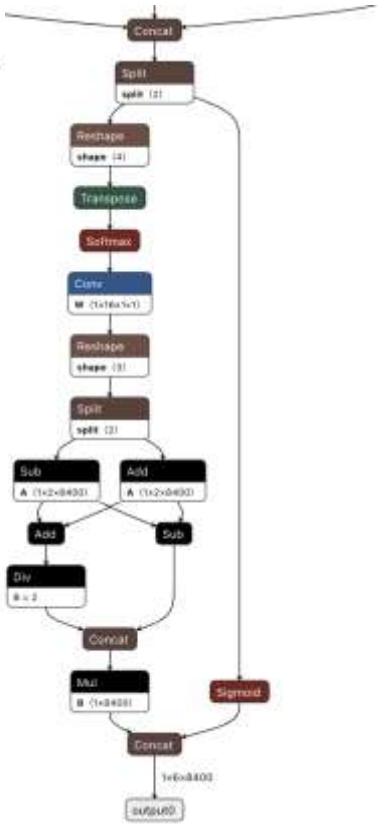
YOLOv8



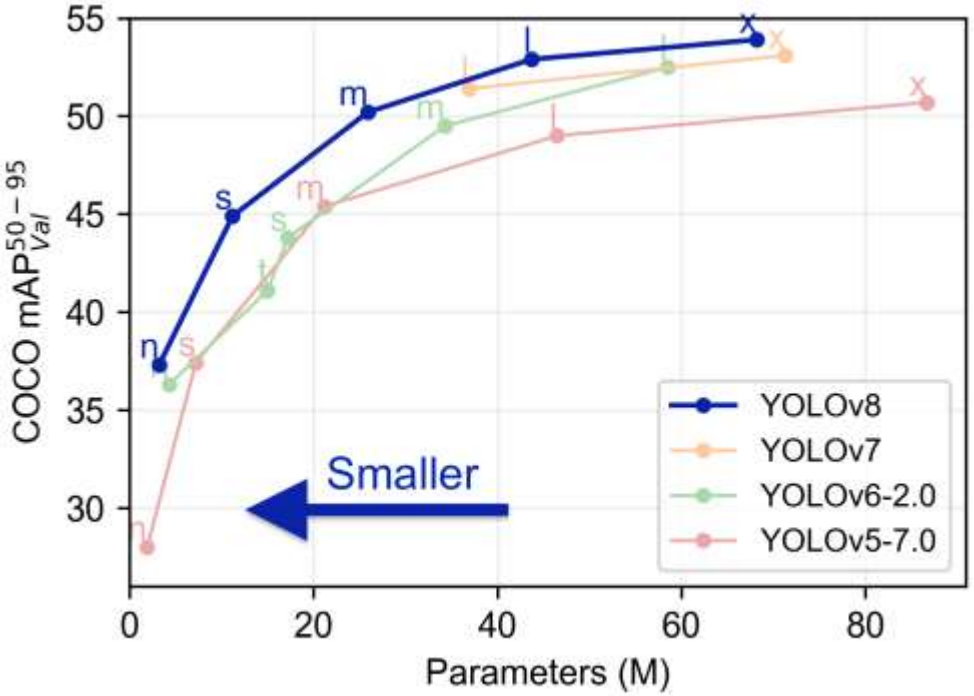
- C2f module

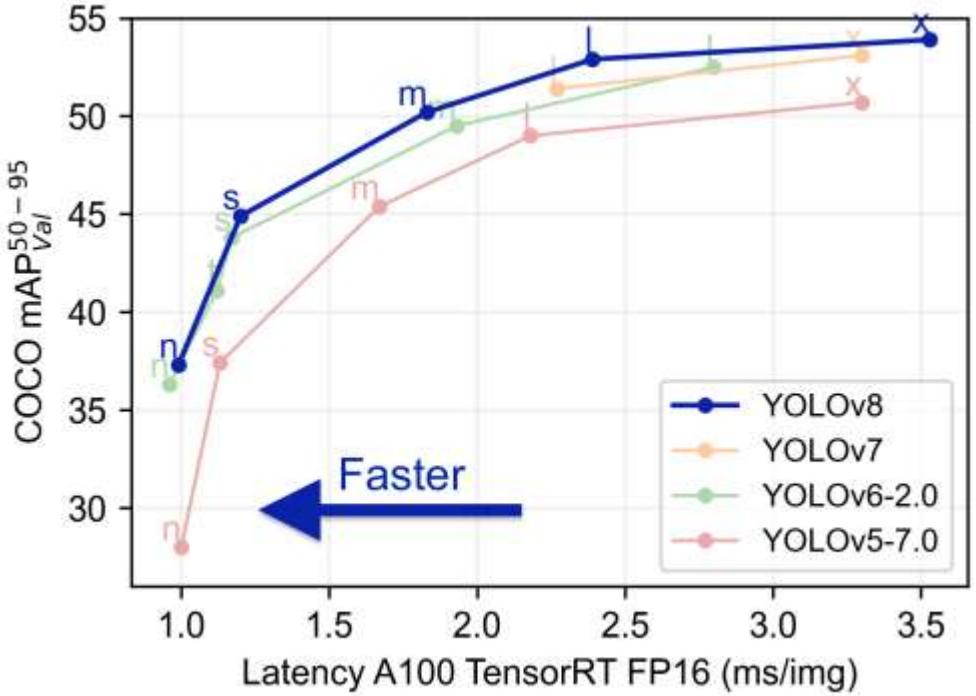


- Anchor-free
- Decoupled head
  - 图像识别
  - 语义分割
  - 图像分类
  - 姿态估计
- CIoU and DFL loss functions



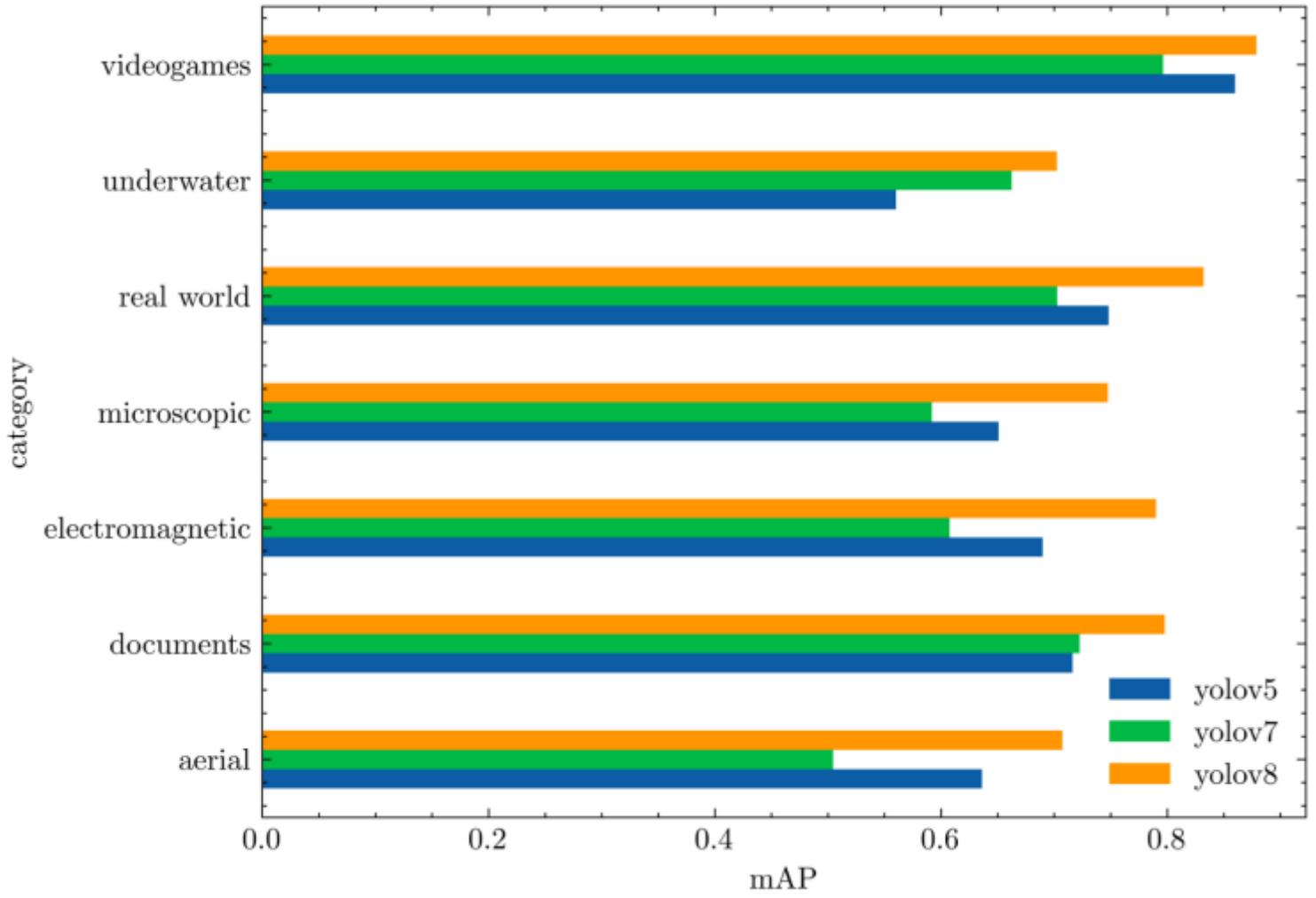
YOLOv8





Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

YOLOv8


videogames
underwater
real world
microscopic
electromagnetic
documents
aerial
category
0.0
0.2
0.4
0.6
0.8
mAP
yolov5
yolov7
yolov8

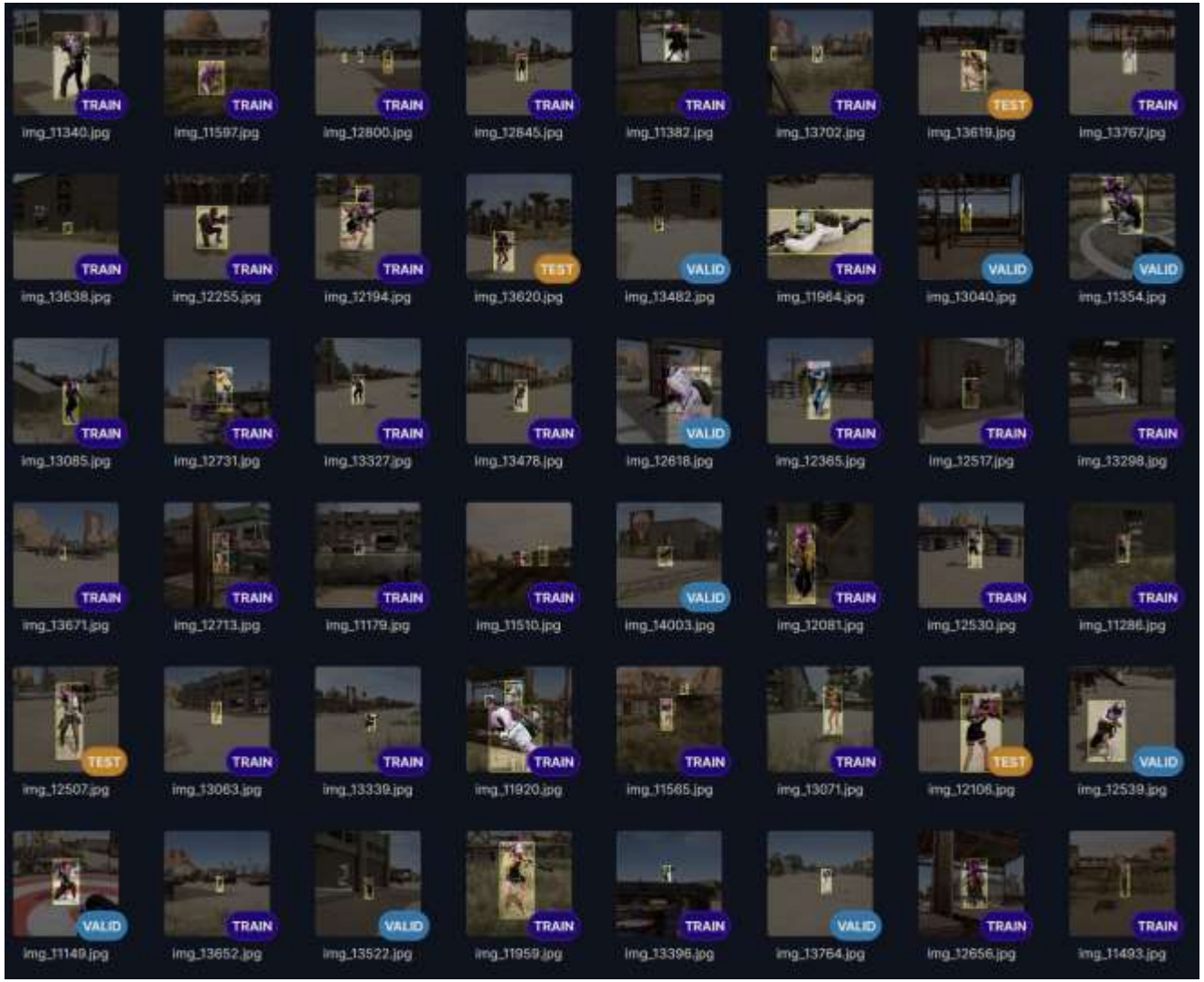
02

实验

框架

PyTorch 2.0 + Ultralytics + roboflow

数据集



2744张

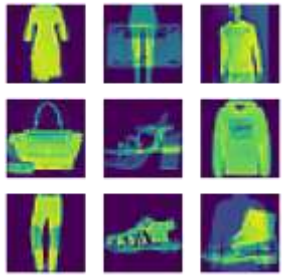
预处理:

- 640×640

数据增强:

- Mosaic
- Copy-Paste

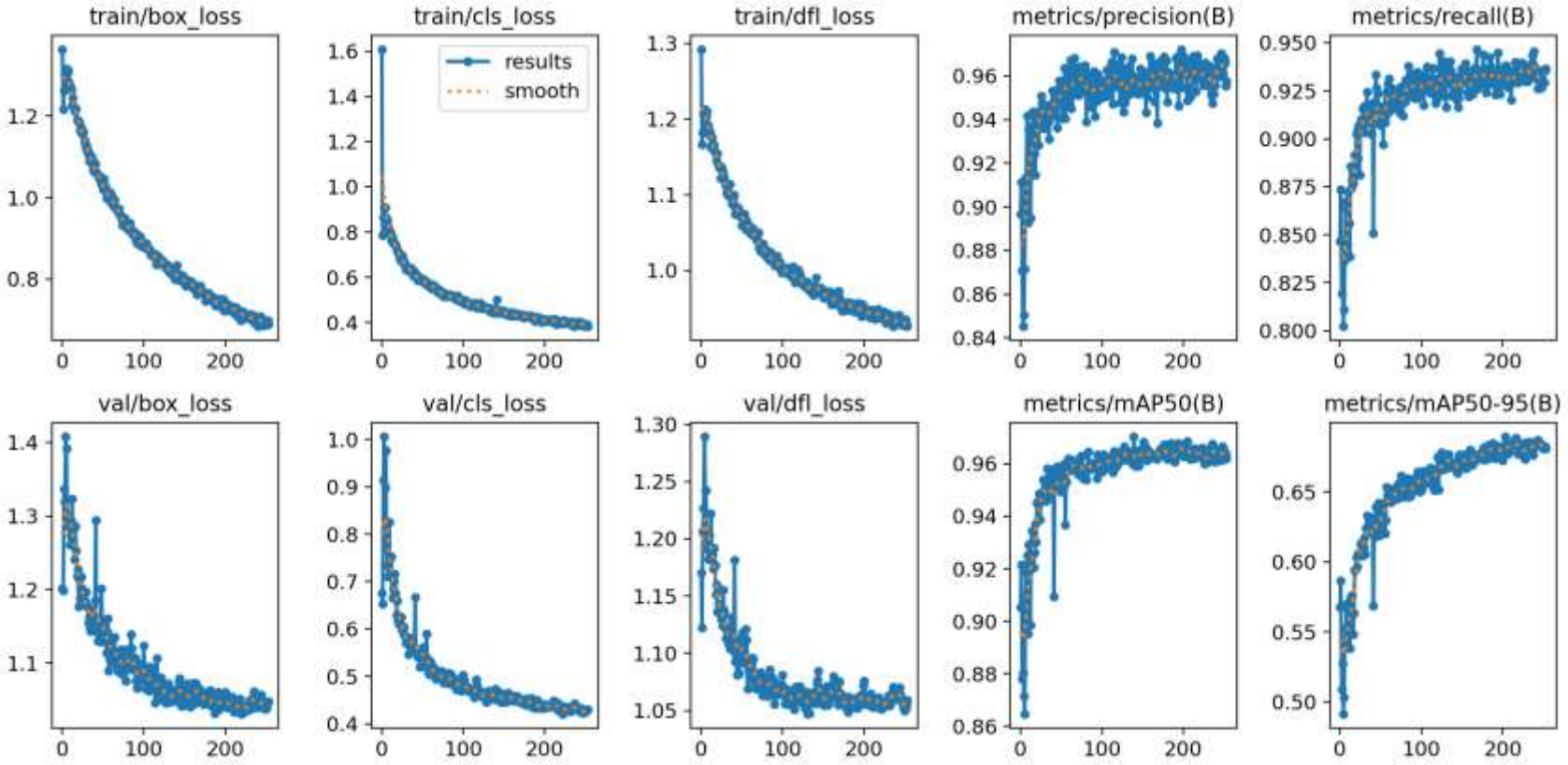
- Random affine
- mixup



- HSV augmentation
- Random horizontal/vertical flip

训练

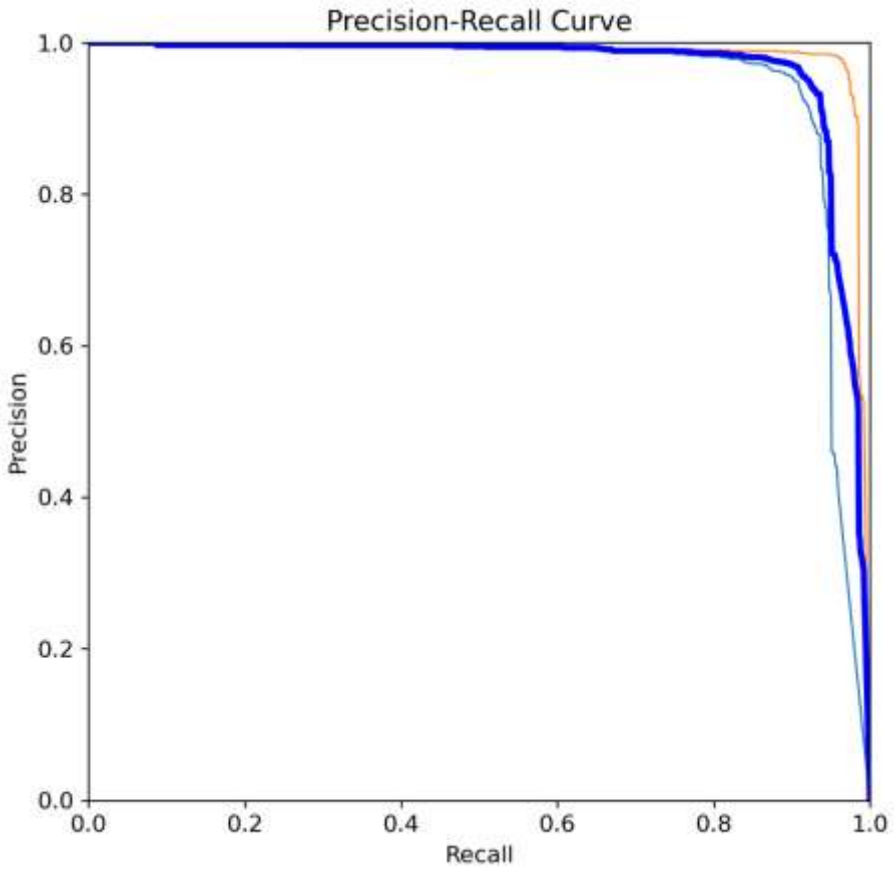
- Optimizer：SGD
- Batch size：16
- Learning rate：0.01
- Early stopping
- Epochs：254（203）
- Automatic Mixed Precision

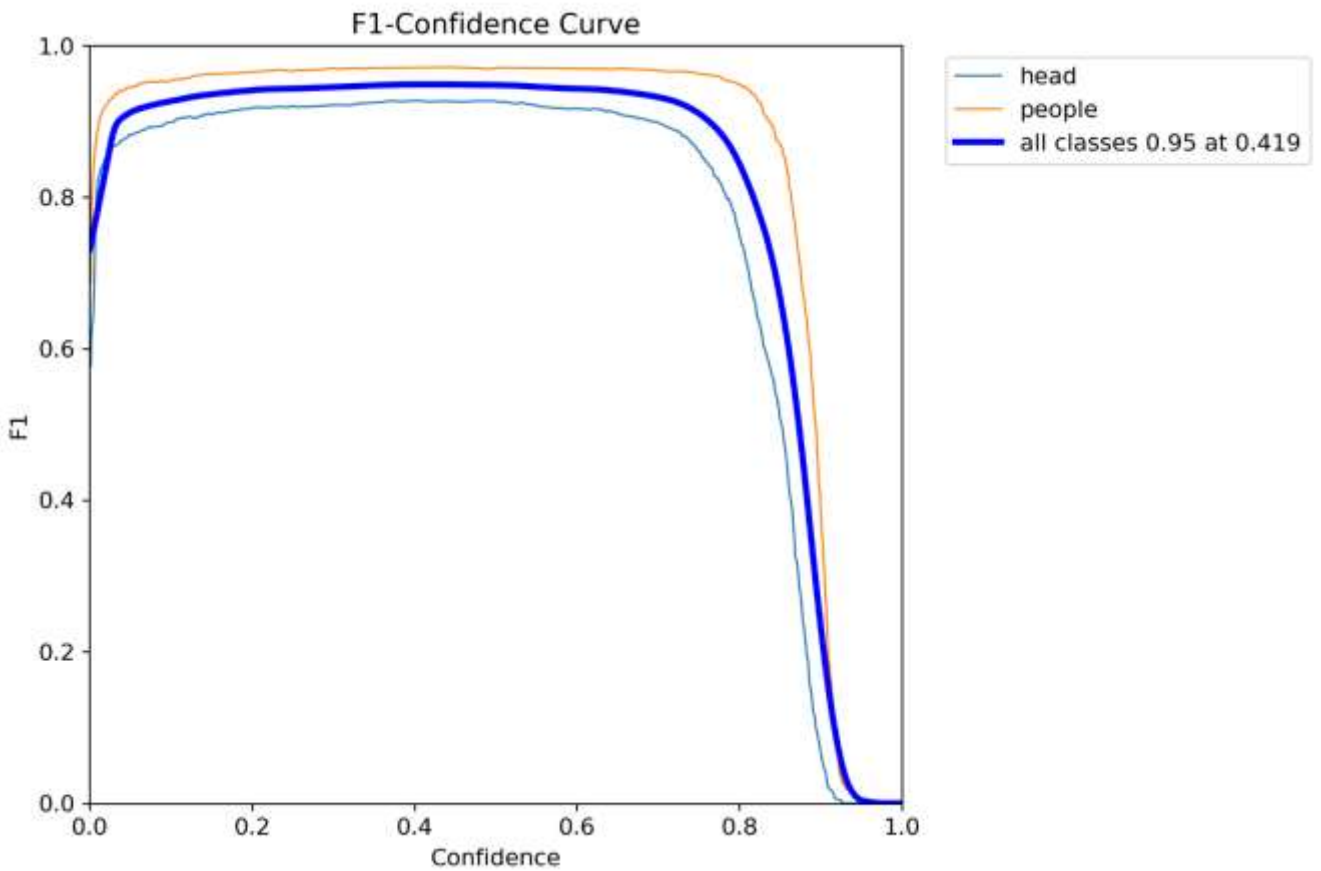


YOLOv8s：$mAP_{50} = 97.0\%, mAP_{50-95} = 68.9\%$

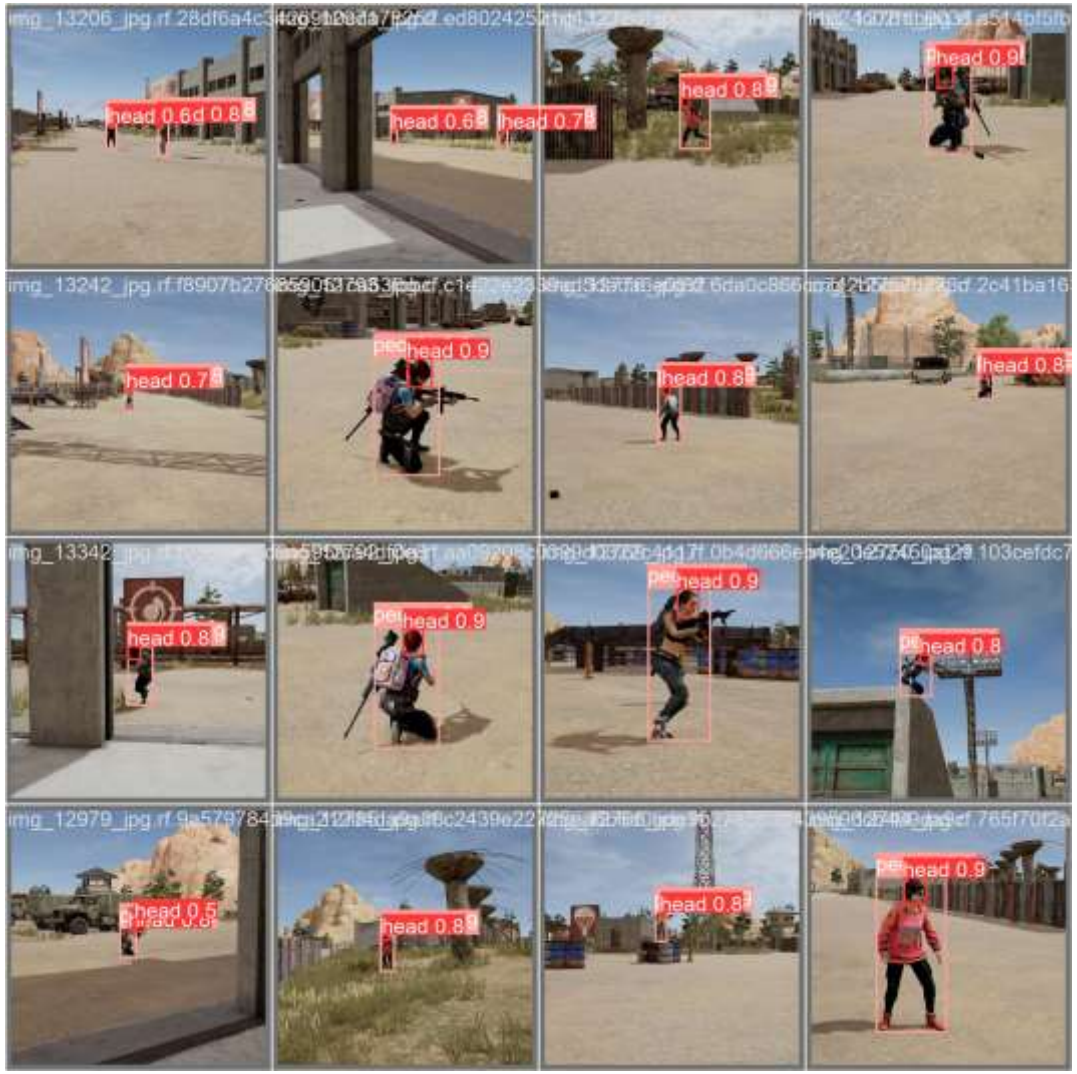
YOLOv8n：$mAP_{50} = 96.5\%, mAP_{50-95} = 67.2\%$

精度曲线

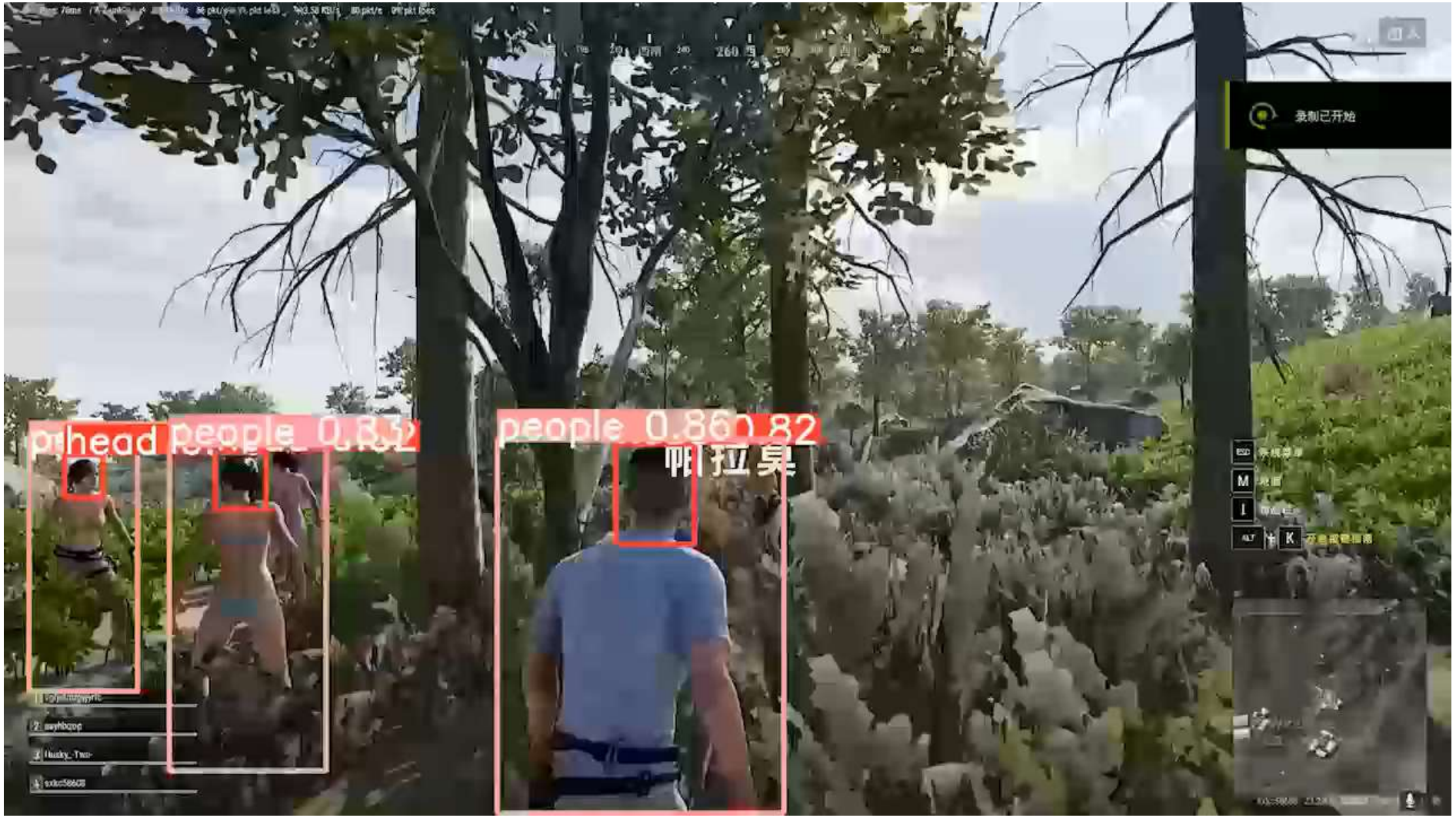

Precision-Recall Curve
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


F1-Confidence Curve
F1
Confidence
head
people
all classes 0.95 at 0.419
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

测试数据集

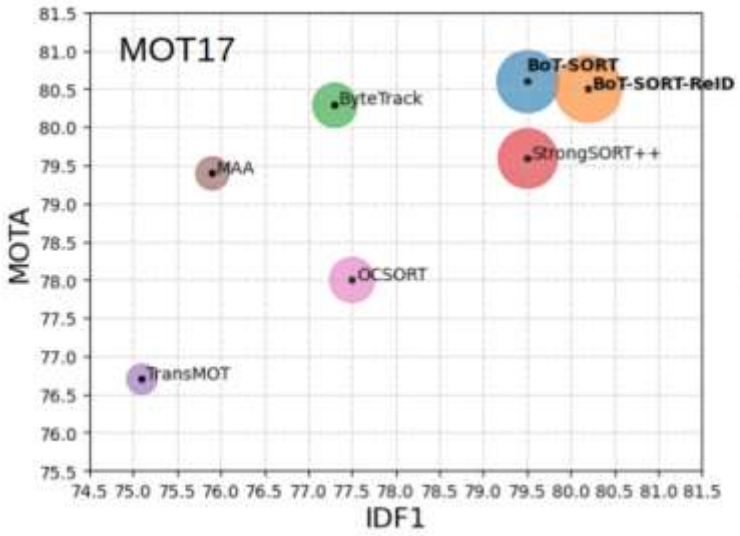


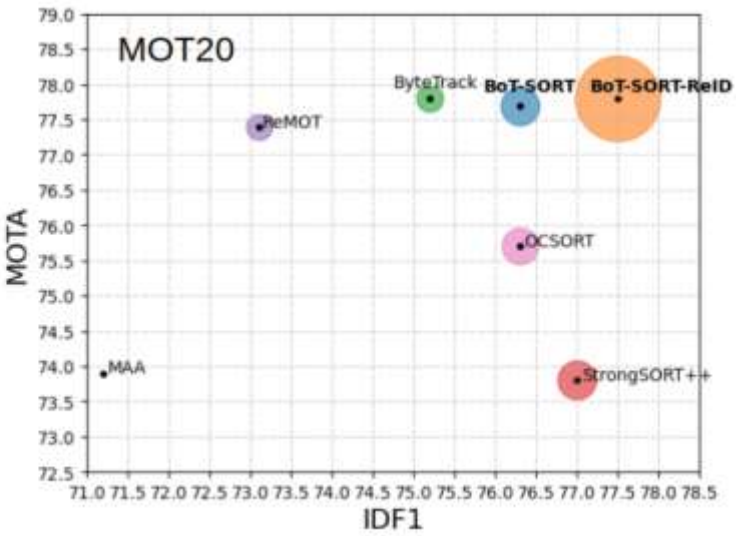
视频测试

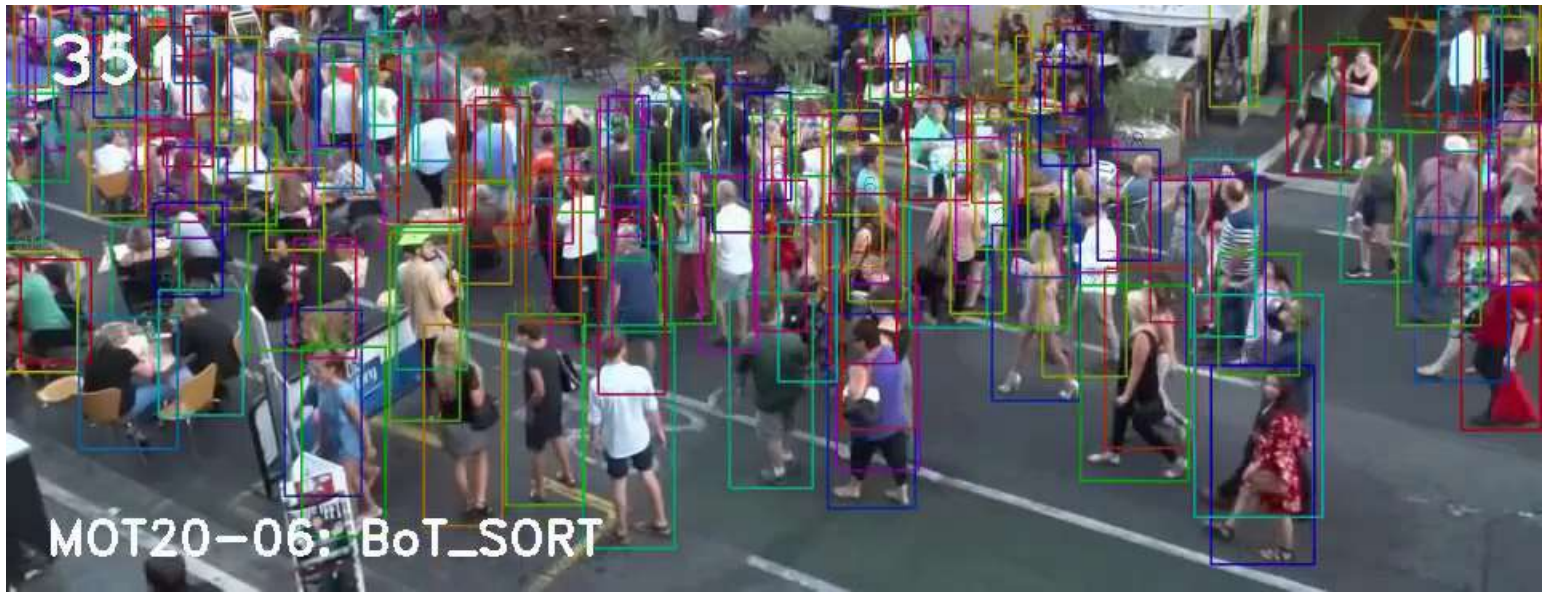


物体追踪

BoT-SORT







视频测试：物体追踪

03

总结

总结

- 本次实验使用 YOLOv8 卷积神经网络，结合多种数据增强方法，实现了对 FPS 游戏图像的识别。实验所得模型在测试集上的 mAP_{50} 达到了 97.0%，mAP_{50-95} 达到了 68.9%，取得了较好的识别效果。

- 为了改善 YOLOv8 进行实时识别时出现的不连续识别问题，实验在 YOLOv8 模型的基础上集成了用于物体追踪的 BoT-SORT 算法，提高了物体识别的连续性。

改进方向

- 数据集质量

- 多模态：声音
- 基于 Transformer 的图像识别

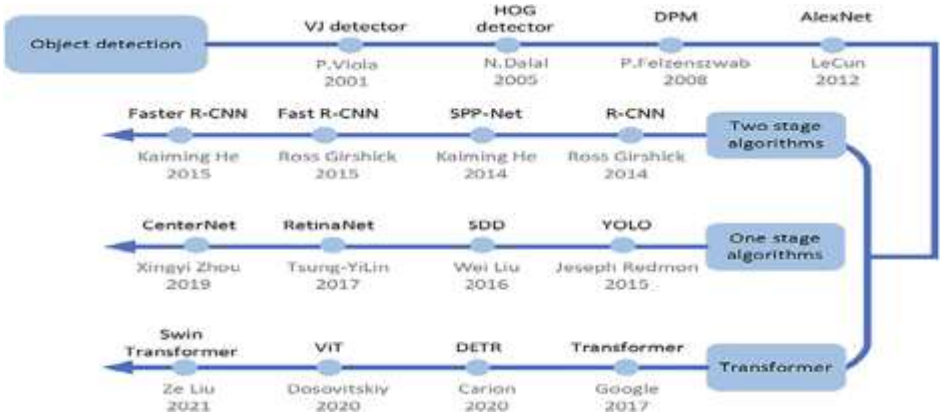


Table 5 Model complexity of mainstream detection methods trained on COCO datasets

Methods	Flops (G)	Param (M)	Inference time (fps)
Faster R-CNN+ResNet-101	283.14	60.52	15.6
RetinaNet+ResNet-101	315.39	56.74	15.0
Yolov4+DarkNet-53	195.55	61.95	66.0
Mask R-CNN+Swin-T	263.78	47.79	15.3
DETR+ResNet-50	91.64	41.3	–

Table 3 Performance comparison of one stage algorithms

Framework	Backbone	Dataset	$AP_{0.5}$	$AP_{[0.5,0.95]}$
SSD512	VGG16	VOC07	71.6	
SSD512	VGG16	MS COCO	46.5	26.8
YOLO	Modified GoogLeNet	VOC07+VOC12	63.4	
YOLOv2	DarkNet-19	MS COCO	44.0	21.6
YOLOv3	DarkNet-53	MS COCO	51.5	28.2
YOLOv4	CSPDarknet53	MS COCO	64.9	43.0
RetinaNet	ResNet-101	MS COCO	53.1	34.4
Center Net	Hourglass-104[58]	MS COCO	62.4	44.9

Table 4 Transformer based detection methods performance comparison on the COCO val2017

Framework	Backbone	$AP_{0.5}$	$AP_{[0.5,0.95]}$
RetinaNet	ResNet-101	53.1	34.4
RetinaNet	PVT-Medium	63.1	41.9
RetinaNet	Twins-PCPVT-B	65.6	44.3
RetinaNet	Twins-SVT-B	66.7	45.3
RetinaNet	Swin-S	65.7	44.5
DETR	ResNet-101	64.9	44.9

谢谢

2023.6.1

参考：

- Arkin, Ershat, Nurbiya Yadikar, Xuebin Xu, Alimjan Aysa, and Kurban Ubul. “A Survey: Object Detection Methods from CNN to Transformer.” Multimedia Tools and Applications 82, no. 14 (October 21, 2022): 21353–83. https://doi.org/10.1007/s11042-022-13801-3.
- Terven, Juan, and Diana Cordova-Esparza. “A Comprehensive Review of YOLO: From YOLOv1 and Beyond.” arXiv, May 19, 2023. https://doi.org/10.48550/arXiv.2304.00501.
- Jocher, Glenn, Ayush Chaurasia, and Jing Qiu. “YOLO by Ultralytics.” Python, January 2023. https://github.com/ultralytics/ultralytics.
- Aharon, Nir, Roy Orfaig, and Ben-Zion Bobrovsky. “BoT-SORT: Robust Associations Multi-Pedestrian Tracking.” arXiv, July 7, 2022. https://doi.org/10.48550/arXiv.2206.14651.