

Simple Demo with BCCD Dataset

Object Detection with Yolo7

Xu Han

1654759534@qq.com

School of CST

Beijing Institute of Technology

December 14, 2022





1. Background

- 1.1. Object Detection
- 1.2. Application
- 1.3. Models

2. Our Work

- 2.1. Dataset Analyze
- 2.2. Train Preparation
- 2.3. Train Model
- 2.4. Evaluate Model
- 2.5. Fine-tuning Model



1. Background

1.1. Object Detection

1.2. Application

1.3. Models

2. Our Work

2.1. Dataset Analyze

2.2. Train Preparation

2.3. Train Model

2.4. Evaluate Model

2.5. Fine-tuning Model

A Simple Introduction to Object Detection:

- 目标检测（Object Detection）的任务是找出图像中所有感兴趣的目标（Object），并确定他们的类别和位置，是计算机视觉领域的核心问题之一。
- 由于各类物体有不同的外观、形状和姿态，加上成像时的光照、遮挡等因素的干扰，目标检测一直是计算机视觉领域最具有挑战性的问题
- 核心问题：
 - 1、分类问题：即图片（或某个区域）中的图像属于哪个类别
 - 2、定位问题：目标可能出现在图像的任何位置
 - 3、大小问题：目标有各种不同的大小
 - 4、形状问题：目标可能有各种不同的形状

A Simple Introduction to Object Detection:

- 目标检测（Object Detection）的任务是找出图像中所有感兴趣的目标（Object），并确定他们的类别和位置，是计算机视觉领域的核心问题之一。
- 由于各类物体有不同的外观、形状和姿态，加上成像时的光照、遮挡等因素的干扰，目标检测一直是计算机视觉领域最具有挑战性的问题
- 核心问题：
 - 1、分类问题：即图片（或某个区域）中的图像属于哪个类别
 - 2、定位问题：目标可能出现在图像的任何位置
 - 3、大小问题：目标有各种不同的大小
 - 4、形状问题：目标可能有各种不同的形状

A Simple Introduction to Object Detection:

- 目标检测（Object Detection）的任务是找出图像中所有感兴趣的目标（Object），并确定他们的类别和位置，是计算机视觉领域的核心问题之一。
- 由于各类物体有不同的外观、形状和姿态，加上成像时的光照、遮挡等因素的干扰，目标检测一直是计算机视觉领域最具有挑战性的问题
- 核心问题：
 - 1、分类问题：即图片（或某个区域）中的图像属于哪个类别
 - 2、定位问题：目标可能出现在图像的任何位置
 - 3、大小问题：目标有各种不同的大小
 - 4、形状问题：目标可能有各种不同的形状



1. Background

1.1. Object Detection

1.2. Application

1.3. Models

2. Our Work

2.1. Dataset Analyze

2.2. Train Preparation

2.3. Train Model

2.4. Evaluate Model

2.5. Fine-tuning Model

Application of Object Detection:

- 1)、人脸检测：智能门控、员工考勤签到、智慧超市、人脸支付
- 2)、行人检测：智能辅助驾驶、智能监控
- 3)、车辆检测：自动驾驶、违章查询、关键通道检测
- 4)、遥感检测：大地遥感、农作物监控



1. Background

- 1.1. Object Detection
- 1.2. Application
- 1.3. Models**

2. Our Work

- 2.1. Dataset Analyze
- 2.2. Train Preparation
- 2.3. Train Model
- 2.4. Evaluate Model
- 2.5. Fine-tuning Model

基于神经网路的目标检测主要分为两大系列—RCNN 系列和 YOLO 系列

RCNN 系列是基于区域检测的代表性算法，YOLO 是基于区域提取的代表性算法

RCNN

RCNN

Fast-RCNN

Faster-RCNN

YOLO

YOLOv1、YOLOv2、YOLOv3

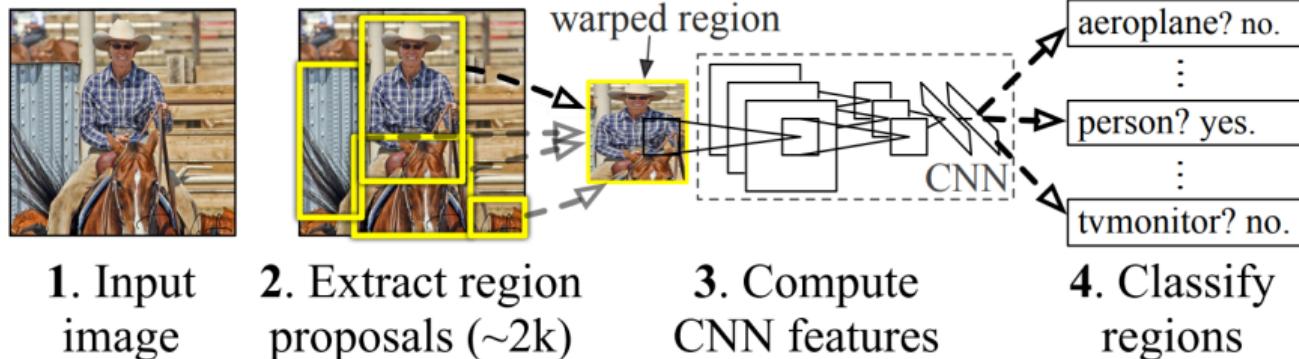
YOLOv4、YOLOv7

YOLOv5、YOLOv6、...

1)、R-CNN

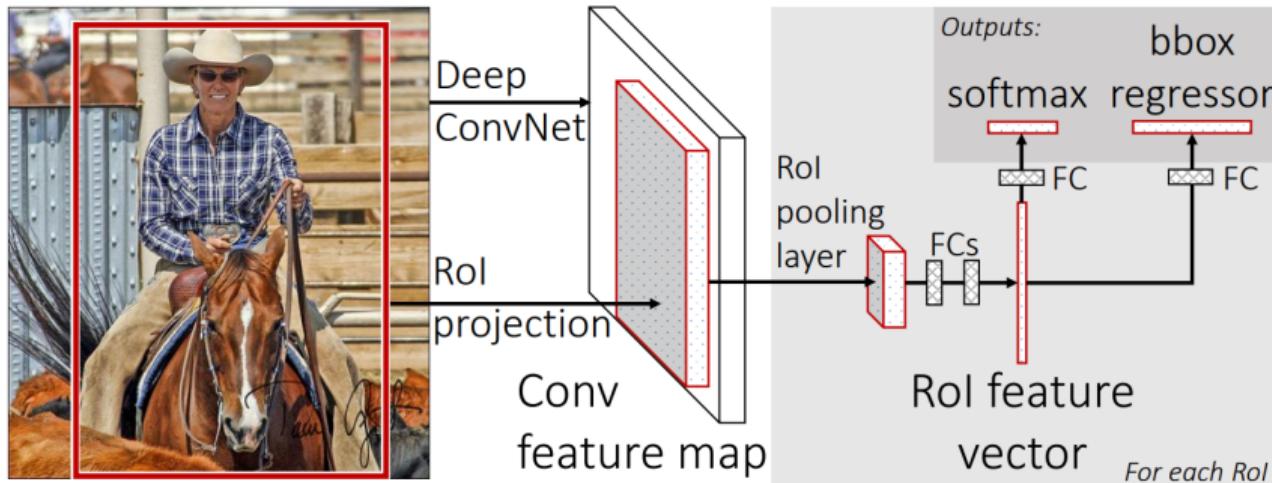
R-CNN (Regions with CNN features), 是 R-CNN 系列的第一代算法，其实没有过多的使用"深度学习"的思想，而是将"深度学习"和传统的"计算机视觉"的知识相结合。比如 R-CNN pipeline 中的第二步和第四步其实就属于传统的"计算机视觉"技术。使用 selective search 提取 region proposals, 使用 SVM 实现分类。

R-CNN: *Regions with CNN features*



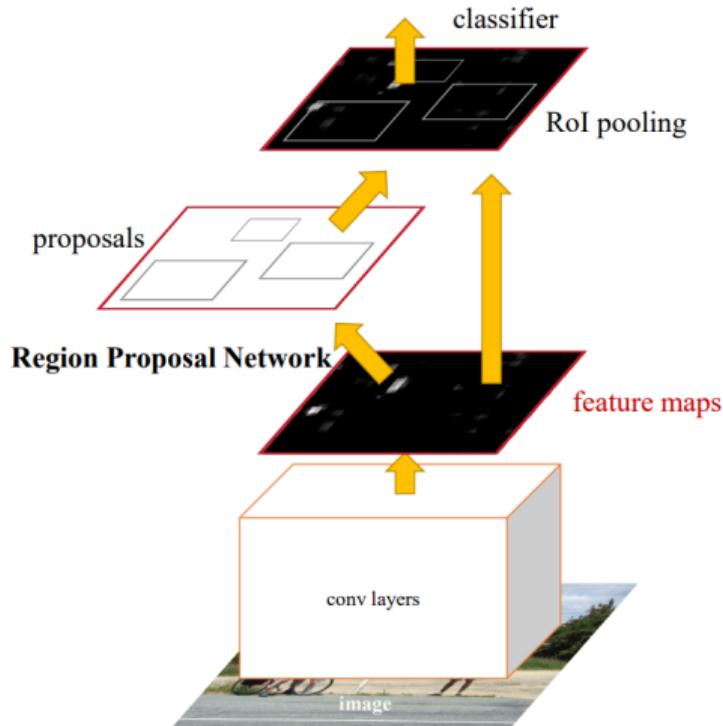
2)、Fast R-CNN

Fast R-CNN 是基于 R-CNN 和 SPPnets 进行的改进。SPPnets 的创新点在于只进行一次图像特征提取（而不是每个候选区域计算一次），然后根据算法，将候选区域特征图映射到整张图片特征图中。



3)、Faster R-CNN

经过 R-CNN 和 Fast R-CNN 的积淀，ROSS B.Girshick 在 2016 年提出了新的 Faster R-CNN 在结构上将特征提取、region proposal 提取、bbox regression、分类都整合到了一个网络之中，使得综合性能有较大提高，在检测速度方面尤其明显。



4)、YOLOv1(2016)

YOLOv1(You Only Look Once) 是继 R-CNN, Fast R-CNN 和 faster R-CNN 之后, Ross Girshick 针对 DL 目标检测速度问题提出的另一种框架, 其核心思想是生成 RoI+ 目标检测两阶段 (two stage) 算法用一套网络的一阶段 (one stage) 算法替代, 直接在输出层回归 bounding box 的位置和所属类别。

5)、YOLOv2(2016)

Ross Girshick 吸收 fast R-CNN 和 SSD 算法, 设计了 YOLOv2 (论文原名《YOLO9000: Better, Faster, Stronger》), 在精度上利用一些训练技巧, 在速度上应用了新的网络模型 DarkNet19, 在分类任务上采用联合训练方法, 结合 wordtree 等方法, 用 YOLOv2 的检测种类扩充到了上千种。

6)、YOLOv7(2022)

YOLOv7 是目前 YOLO 系列最先进的算法，YOLOv7 相同体量下比 YOLOv5 精度更高，速度更快 (120%FPS)。

其大部分继承自 YOLOv5，包括整体网络架构、配置文件的设置和训练、推理、验证过程等等。

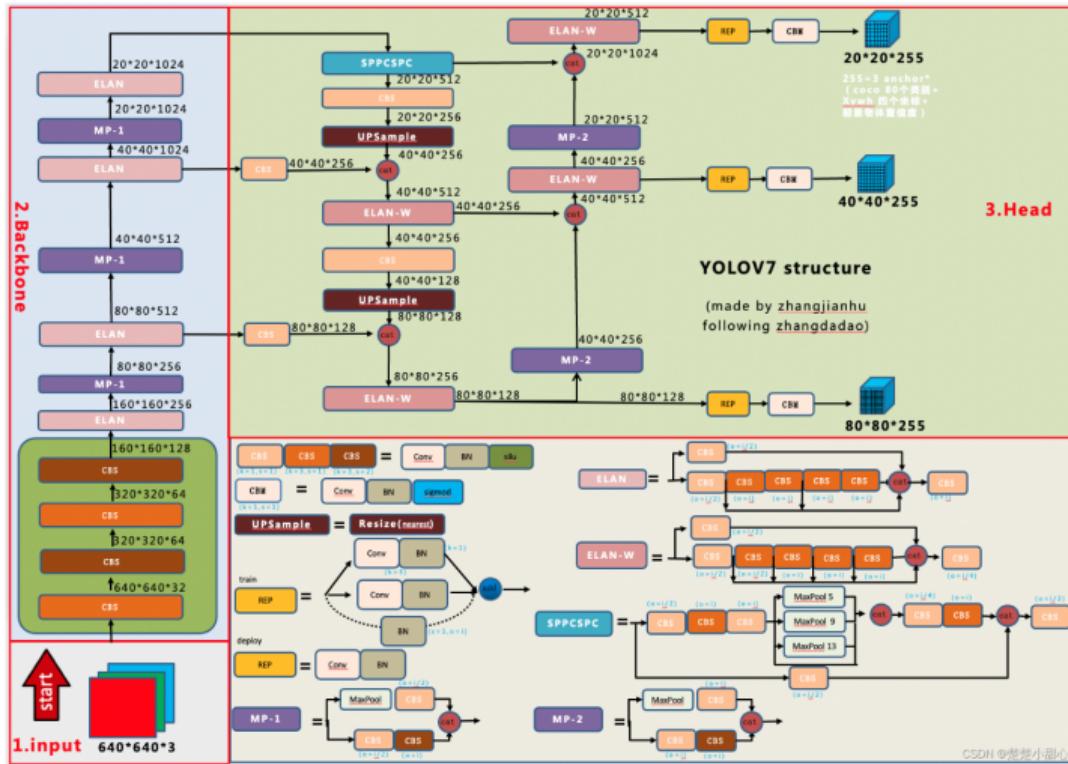
此外，v7 也有不少继承自 YOLOR，毕竟是同一个作者前后年的工作，包括不同网络的设计、超参数设置以及隐性只是学习的加入。

还有就是在帧样本匹配时仿照了 YOLOX 的 SimOTA 策略。YOLOv7 包括了近几年最新的 trick：高效聚合网络、重参数化卷积、辅助头检测、缩放模型等。

Models



北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY



1. Background

- 1.1. Object Detection
- 1.2. Application
- 1.3. Models

2. Our Work

- 2.1. Dataset Analyze
- 2.2. Train Preparation
- 2.3. Train Model
- 2.4. Evaluate Model
- 2.5. Fine-tuning Model



1. Background

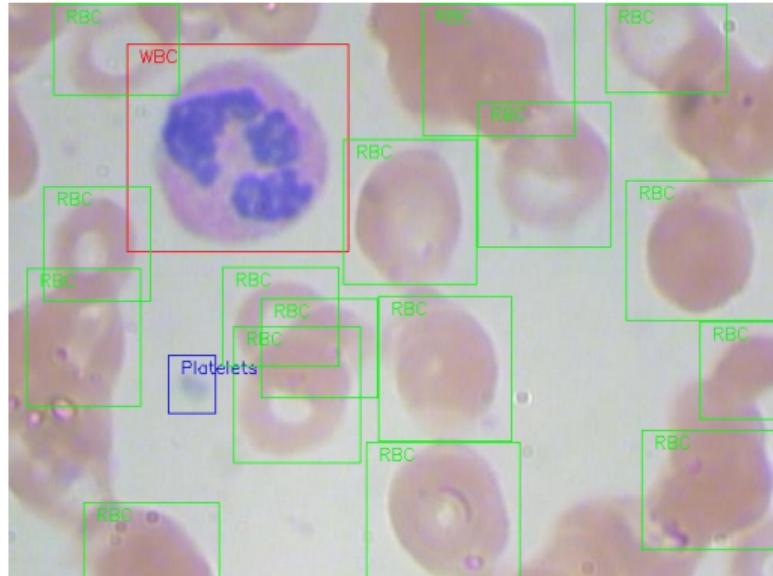
- 1.1. Object Detection
- 1.2. Application
- 1.3. Models

2. Our Work

- 2.1. Dataset Analyze
- 2.2. Train Preparation
- 2.3. Train Model
- 2.4. Evaluate Model
- 2.5. Fine-tuning Model

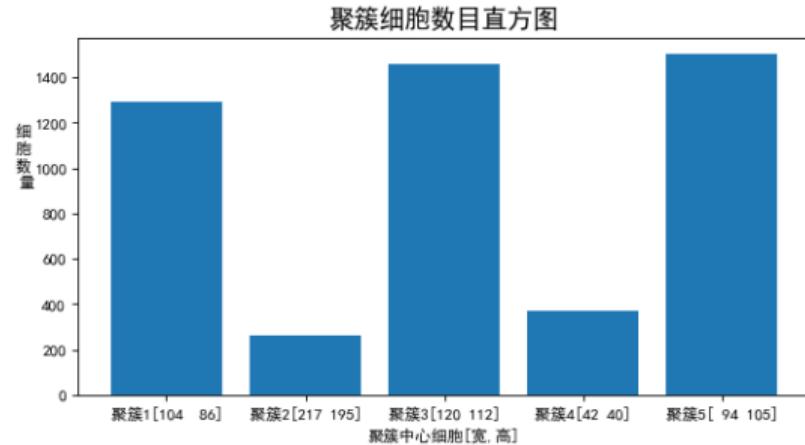
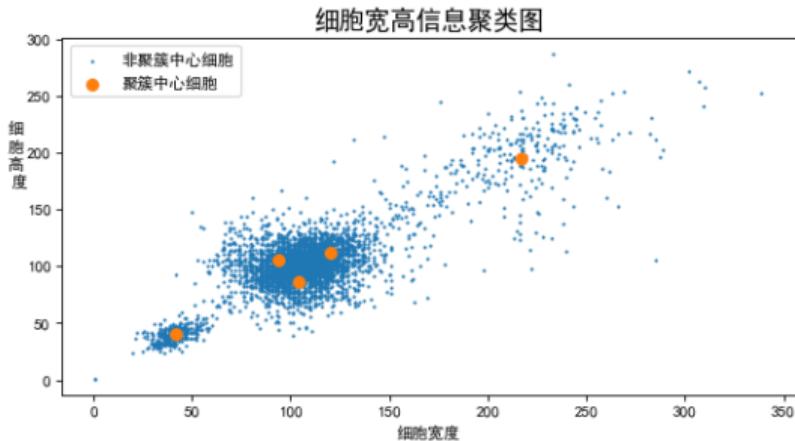
Dataset Analyze

我们所用的数据集是 BCCD Dataset，BCCD 数据集是一个小型的血细胞检测数据集，它由 364 张显微镜下的图片以及 xml 文件格式的标注构成，我们通过 git 获取数据集，并对其中的内容做了简单的统计与展示

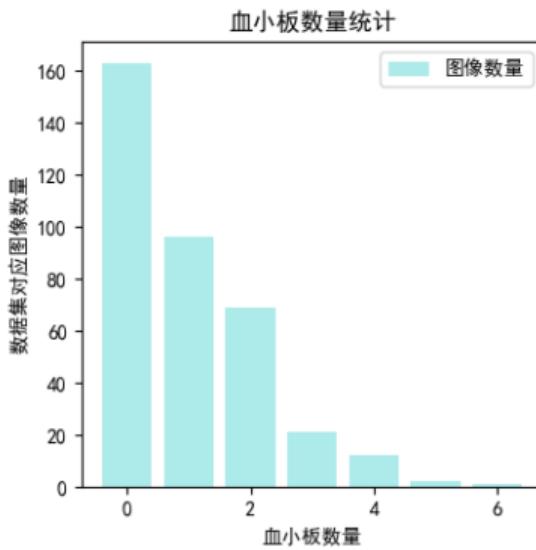
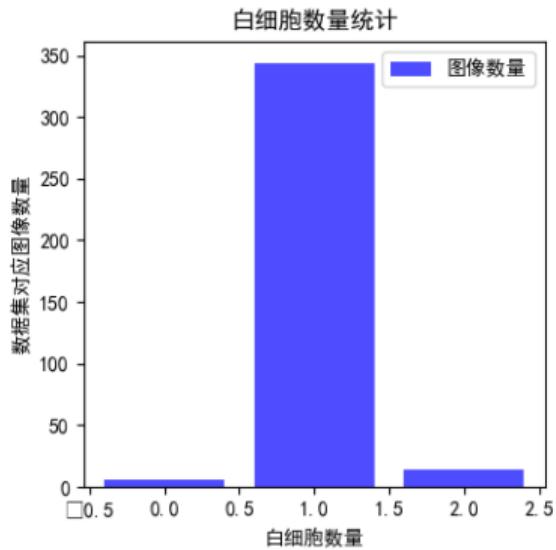
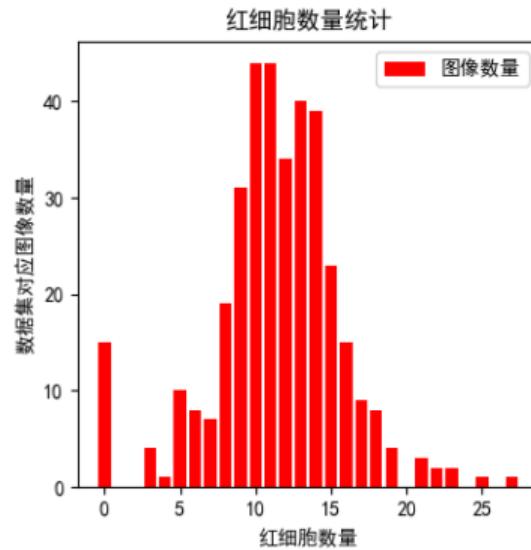


图：已绘制标记框

我们使用 KMeans 聚类算法对细胞大小进行了聚类，用宽高描述细胞大小，聚簇设为 5，



同时我们也对细胞数量进行了统计



可以看到，红细胞数量最多，白细胞大多数都是每张图只有一个，血小板数量最少，有160张图片没有血小板，因此可能存在训练数据的缺失，血小板数量远远小于红细胞以及白细胞，所以考虑采取措施在训练时来进行训练数据扩充



1. Background

- 1.1. Object Detection
- 1.2. Application
- 1.3. Models

2. Our Work

- 2.1. Dataset Analyze
- 2.2. Train Preparation**
- 2.3. Train Model
- 2.4. Evaluate Model
- 2.5. Fine-tuning Model

yolo7 训练开源工具包需要的数据结构如下：

dataset

...Annotations//放标注好的 XML 文件

...images//放图片

...ImageSets//存放生成的.txt 文件

...Labels//存放从 XML 文件提取的.txt 文件

```
print(len(os.listdir('Dataset/images/train')))  
print(len(os.listdir('Dataset/images/val')))  
print(len(os.listdir('Dataset/images/test')))  
[18]  
.. 205  
87  
72
```



1. Background

- 1.1. Object Detection
- 1.2. Application
- 1.3. Models

2. Our Work

- 2.1. Dataset Analyze
- 2.2. Train Preparation
- 2.3. Train Model**
- 2.4. Evaluate Model
- 2.5. Fine-tuning Model

我们使用预训练的 yolov7-tiny.pt，这个权重对于数据量比较小的模型，训练效果要更好，同时我们修改相应的 yaml 文件，调整模型的相关参数，使其适应我们使用的 BCCD 数据集，在训练过程中，我们发现 BCCD 数据标注导出时出现问题，对 yolo7 的 dataset 文件进行了修改。

我们使用的 GPU 为 3060Ti，设置 epoch=50，batch-size=8，设置 batch-size 时发现若设置为 32 或更高时，会出现显卡显存不够用的情况，因此将其设置为 8，同时我们发现，训练迭代次数超过 50 以后，损失函数已经收敛，因此认为迭代次数设置较高对模型性能影响不大，所以综合考虑将其设置为 100



1. Background

- 1.1. Object Detection
- 1.2. Application
- 1.3. Models

2. Our Work

- 2.1. Dataset Analyze
- 2.2. Train Preparation
- 2.3. Train Model
- 2.4. Evaluate Model**
- 2.5. Fine-tuning Model

在目标识别领域，一般使用以下特征对模型进行评估：

1、PR 曲线：准确性（precision）以及召回率（recall），理想情况下，准确率与召回率应都很高，但是但是一般情况下准确率高、召回率就低，召回率低、准确率高

2、AP 与 mAP：AP (Average Precision)，平均精准度，简单来说就是对 PR 曲线上的 Precision 值求均值。而 mAP 就是 AP 的平均值，对于 PR 曲线来说，使用积分来进行计算：

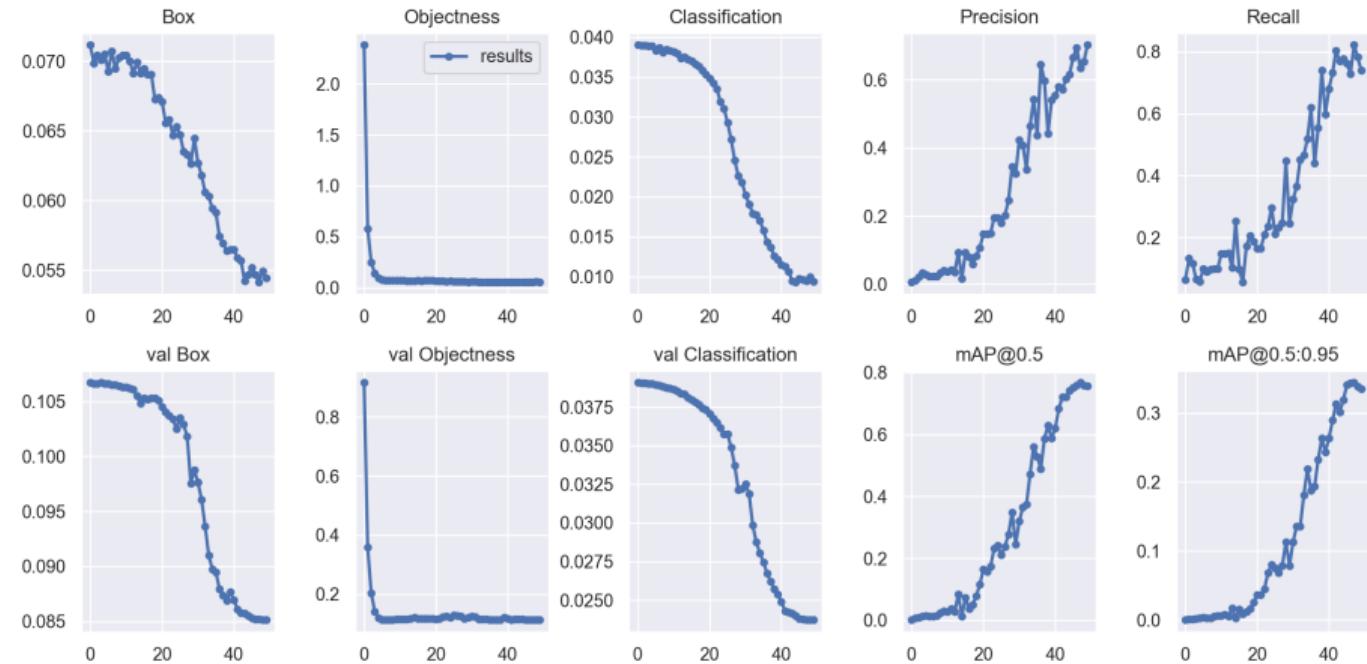
$$AP = \int_0^1 p(r) dr$$

3、F1 分数：是一个权衡精准率与召回率指标的值，可以画出曲线

Evaluate Model



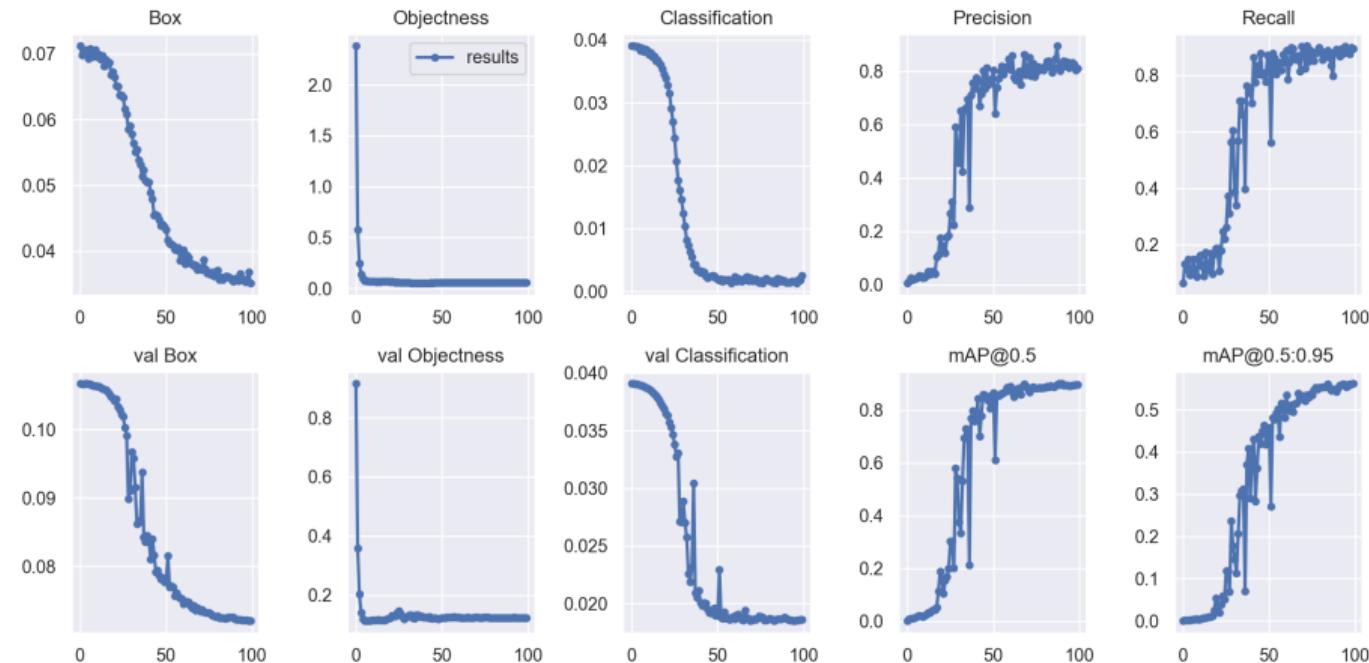
下面对模型训练结果展示：epoch=50



Evaluate Model



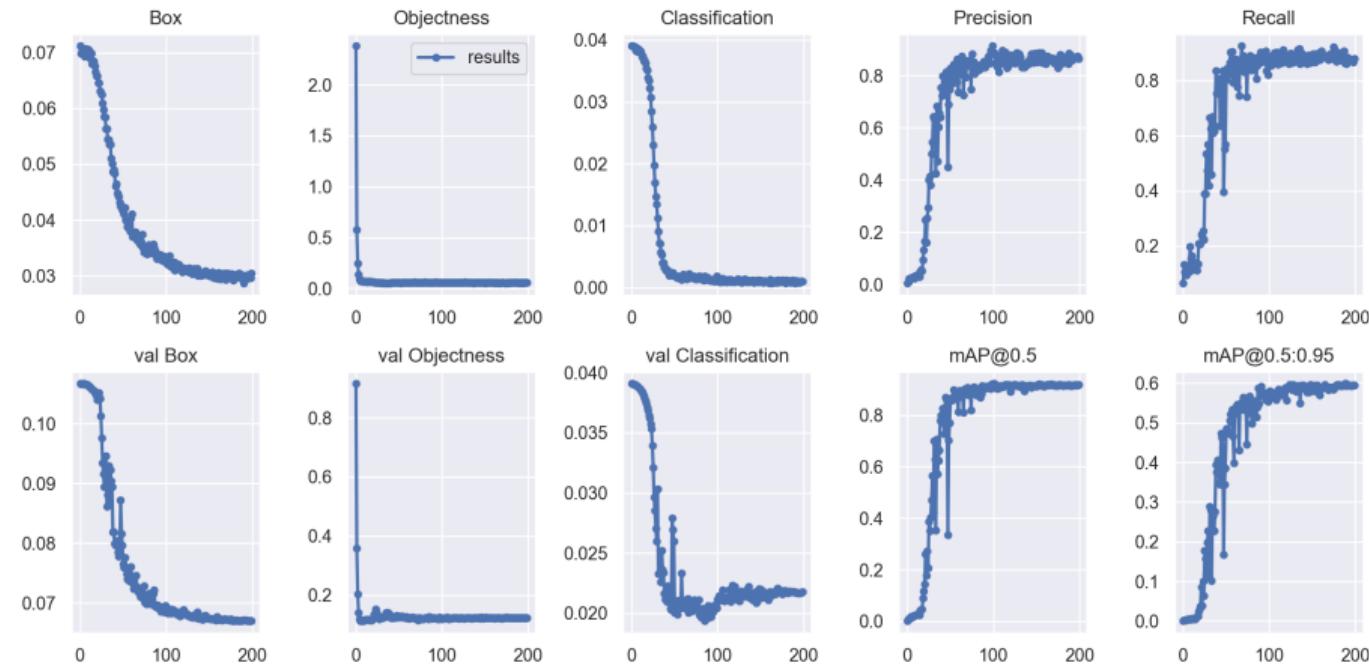
epoch=100



Evaluate Model



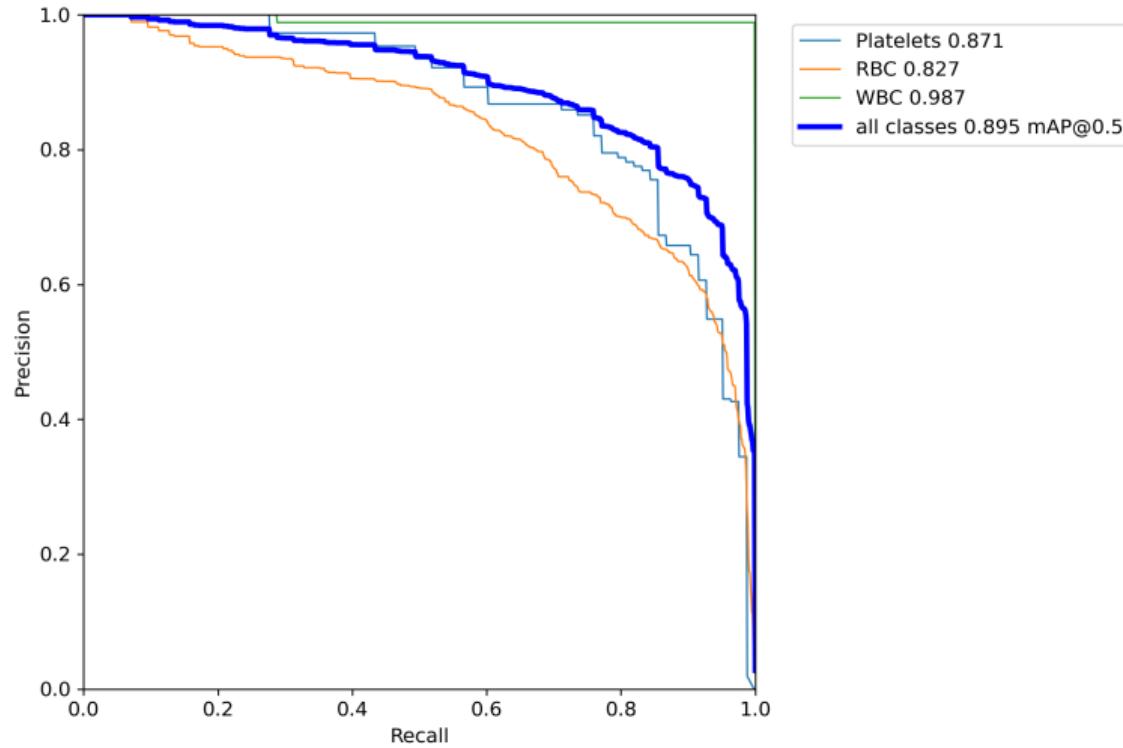
epoch=200



Evaluate Model



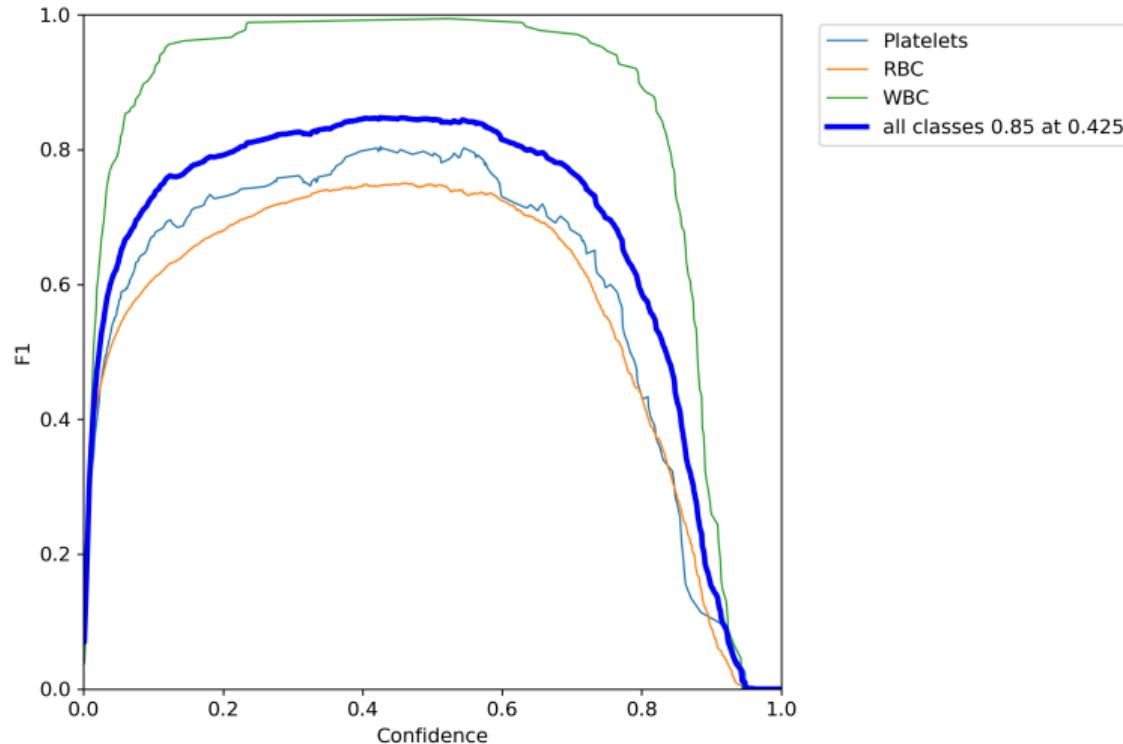
epoch=100



Evaluate Model



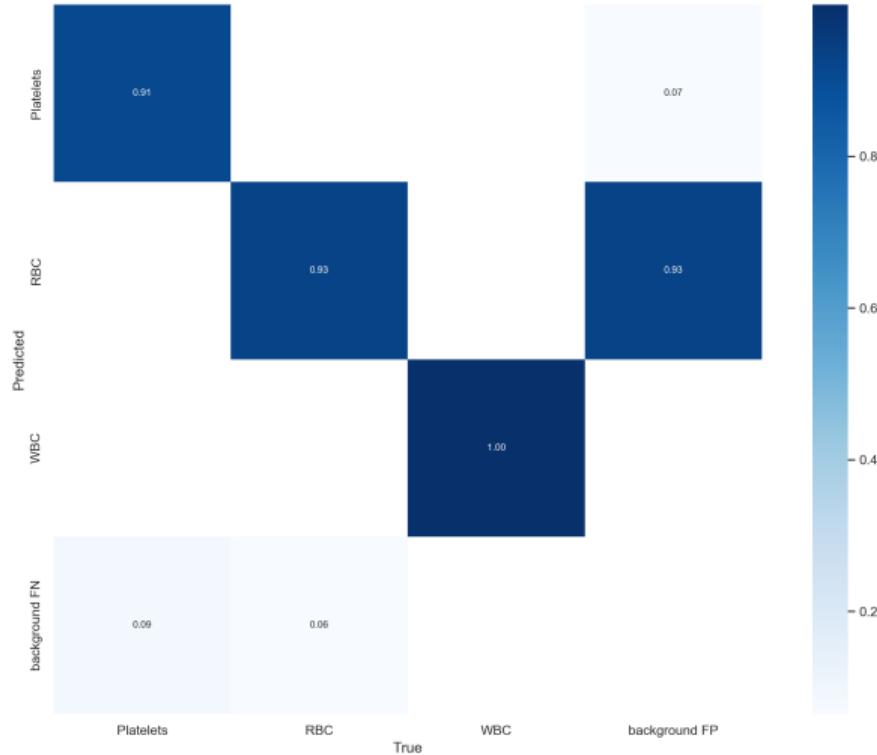
epoch=100



Evaluate Model



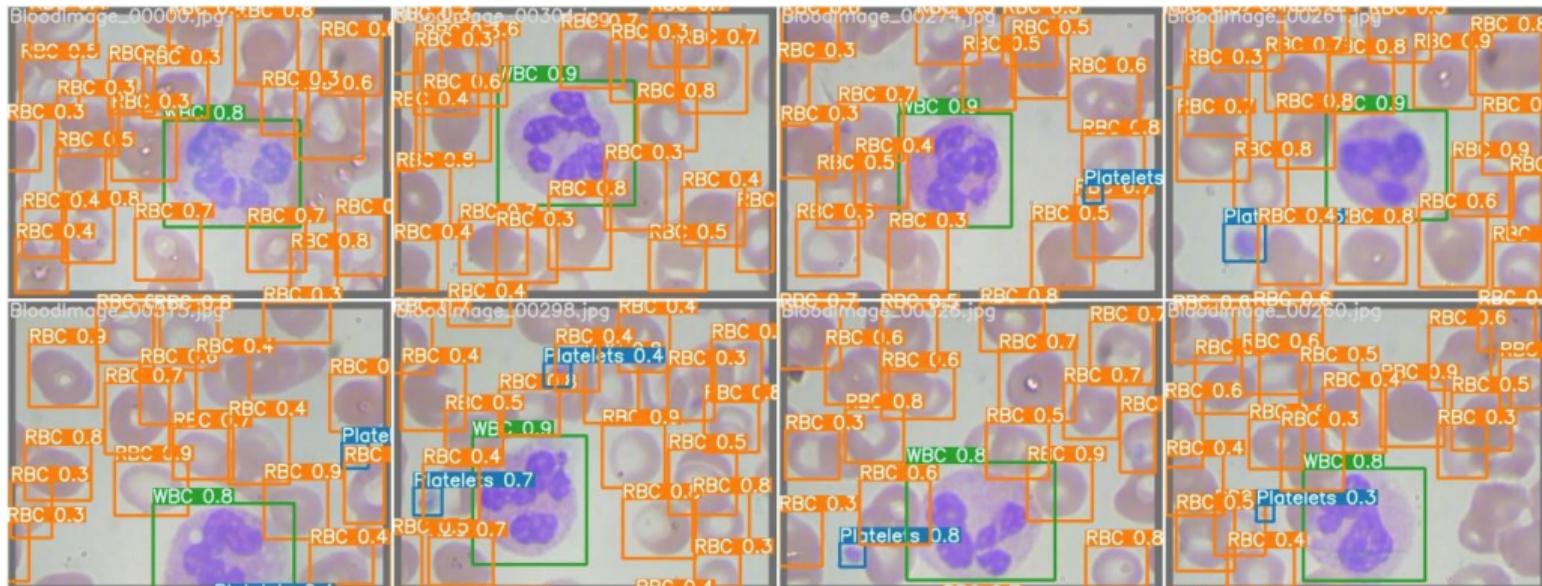
epoch=100



Evaluate Model



epoch=100





1. Background

- 1.1. Object Detection
- 1.2. Application
- 1.3. Models

2. Our Work

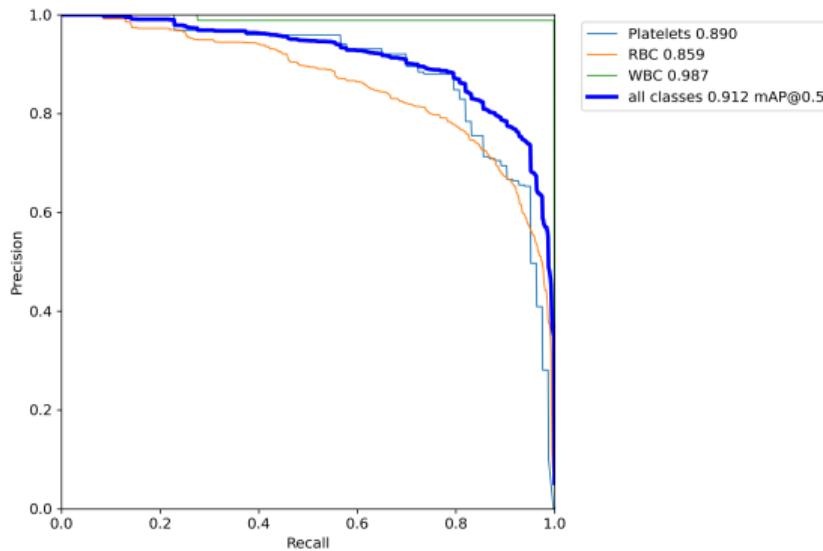
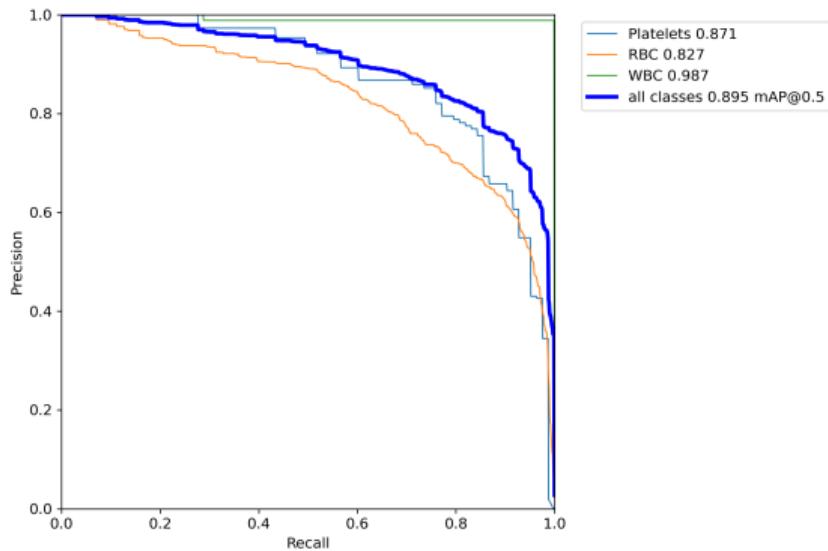
- 2.1. Dataset Analyze
- 2.2. Train Preparation
- 2.3. Train Model
- 2.4. Evaluate Model
- 2.5. Fine-tuning Model**

除了采用默认的 yolov7 工具包训练配置文件，我们尝试对 yaml 文件进行调整

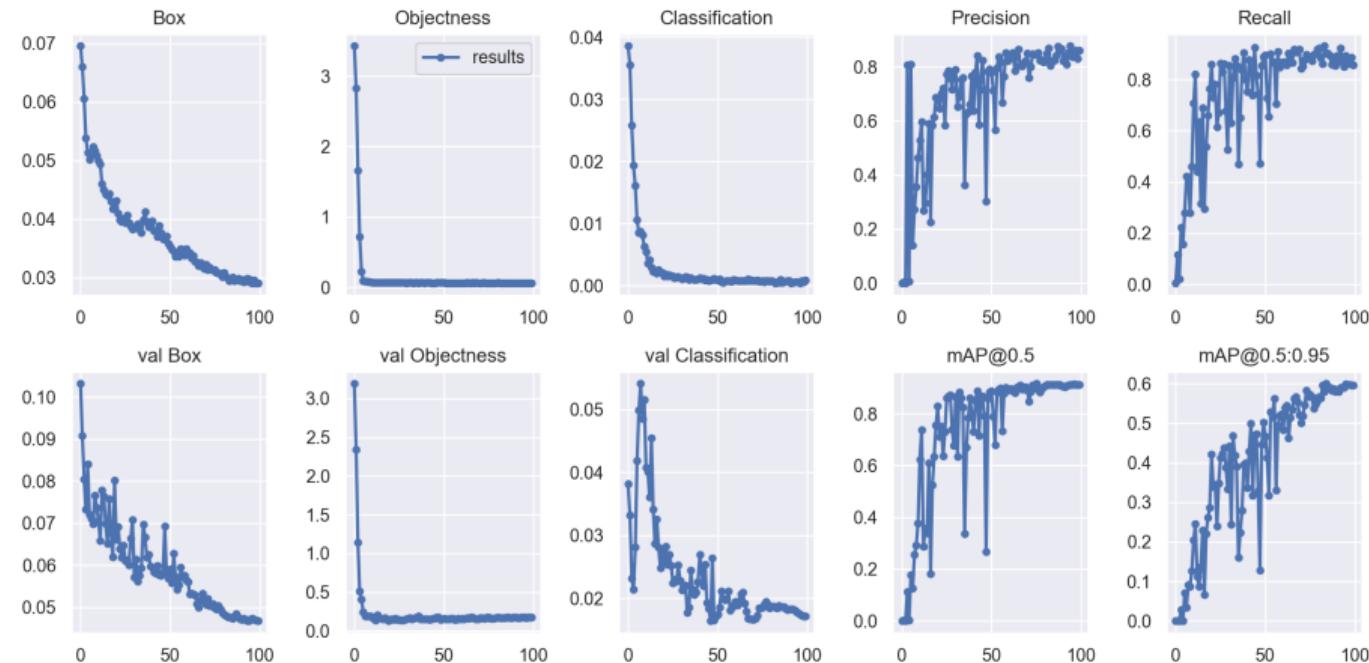
-hyp：超参数配置文件，我使用的是默认的 data/hyp.scratch.tiny.yaml 文件，在网上查阅相关资料后，我得知由于自制的数据集本来复杂度就不够，开启在线数据增强的话，很容易训练效果不好，造成过拟合，于是我尝试调整 data/hyp.scratch.tiny.yaml，中的相关参数，将其设置为 0）

```
fl_gamma: 0.0 # focal loss gamma (efficientDet default gamma=1.5)
hsv_h: 0.00 # image HSV-Hue augmentation (fraction)
hsv_s: 0.0 # image HSV-Saturation augmentation (fraction)
hsv_v: 0.0 # image HSV-Value augmentation (fraction)
degrees: 0.0 # image rotation (+/- deg)
```

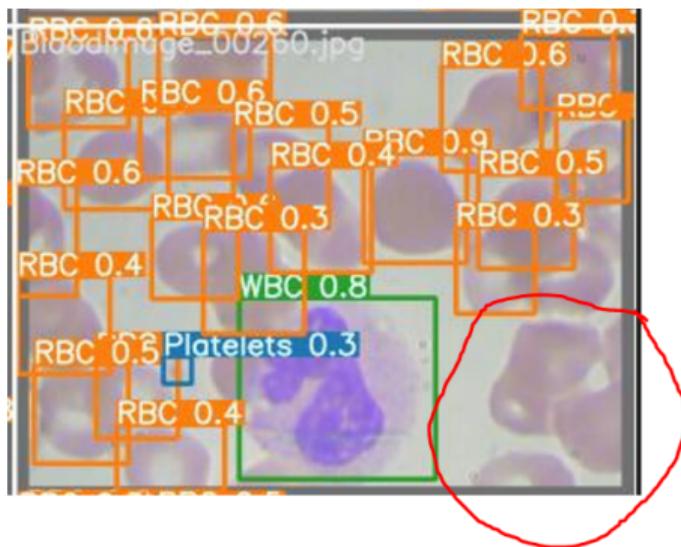
下图为前后对比 (PR 曲线), 左边为未修改, 右边为修改后



下图为经过微调后模型的训练效果，可以看到曲线波动比较大，但是最后的效果要好一点



下图为同一张测试图片在两个模型 test 时确定的框数



右图是修改过后的

Thank you for your attention!
Questions?