

# 基于 YOLO 的血细胞检测

学院：信息工程学院

班级：XG21B2DZ

学号：1211002019 1211002017 1211002002

组员：孟凡千 罗欧栗 鲍捷

日期：2024/06/17

指导老师：刘铁

## 目录

1. 引言与现状.....	3
2. 数据集介绍.....	3
2.1 BCCD 数据集.....	3
2.2 BCCD - v4 416x416_aug 数据集.....	4
3. 算法模块.....	5
3.1 传统算法.....	5
3.2 YOLOv5.....	5
3.2.1 YOLOv5 网络架构.....	5
3.2.2 输入端: Mosaic 数据增强、自适应锚框计算、自适应图片缩放 .....	6
3.2.3 Backbone: Focus 结构, CSP 结构 .....	8
3.2.4 Neck: FPN+PAN 结构 .....	9
3.2.5 Prediction: GIOU_Loss.....	9
3.3 YOLOv5+Transformer Block .....	10
3.4 YOLOv10.....	10
3.4.1 去除 NMS 的双重分配策略 .....	11
3.4.2 整体效率-准确性驱动的设计 .....	11
3.5 YOLOv10+Transformer Block .....	12
4. 实验结果.....	13
4.1 YOLOv5s 训练 50 轮 .....	13
4.2 YOLOv5s 训练 100 轮 .....	15
4.3 YOLOv5s+Transformer Block 训练 50 轮.....	16
4.4 YOLOv5s+Transformer Block 训练 100 轮.....	17
4.5 YOLOv10s 训练 100 轮 .....	18
4.6 YOLOv10b 训练 100 轮.....	19
4.7 YOLOv10x(YOLOv10 中最大的模型)训练 100 轮 .....	20
5. 基于 YOLOv10 与 Gradio 的实时对象检测 Web 应用 .....	21
5.1 系统架构.....	21
5.2 功能介绍.....	21
5.3 实验结果与分析.....	22
6. 总结与展望.....	23

小组成员	贡献概述	分工占比
孟凡千	搜寻数据集、读论文、搭建模型框架、寻找创新点、搭建 web 应用、训练模型、测验模型	47.5%
罗欧栗	搜寻数据集、读论文、搭建模型框架、寻找创新点、搭建 web 应用、训练模型、测验模型	47.5%
鲍捷	搜寻数据集	5%

# 1. 引言与现状

计算机视觉是人工智能的重要分支，通过计算机对图像或视频进行处理、理解和分析，以实现环境的识别和理解。近年来，随着深度学习技术的迅猛发展，计算机视觉在目标检测、图像分类、语义分割等方面取得了显著进展。

目标检测作为计算机视觉的核心任务之一，旨在从图像中检测并识别目标物体的类别和位置。YOLO（You Only Look Once）系列模型因其高效的检测速度和精确的检测性能，在众多目标检测方法中脱颖而出，广泛应用于自动驾驶、安防监控、医疗影像等领域。YOLOv5 作为 YOLO 系列的最新版本，进一步提升了模型的性能和适应性。

在医疗领域，血细胞检测是诊断和治疗多种疾病的重要手段。传统的血细胞检测方法依赖于显微镜观察和手动计数，存在效率低、易受主观因素影响等问题。现有的基于深度学习的血细胞检测方法虽然在一定程度上提高了检测效率和准确性，但仍然面临背景复杂、细胞形态多样、重叠严重等挑战。

目前，许多研究采用卷积神经网络（CNN）进行血细胞检测，取得了一定的成果。例如，U-Net 和 Mask R-CNN 等模型被广泛应用于医学图像分割和目标检测任务。然而，这些模型在处理具有复杂背景和多样形态的血细胞图像时，效果仍有待提升。同时，卷积神经网络在捕捉全局特征和处理长距离依赖关系方面存在局限，这限制了其在血细胞检测中的表现。

为应对上述挑战，本文提出了一种基于 YOLOv5 和 Transformer Block 的血细胞检测方法。通过在 YOLOv5 中引入 Transformer Block 结构，模型能够更好地捕捉图像中的全局特征和长距离依赖关系，从而提高对复杂场景中血细胞的检测能力。我们还测试了 YOLOv10 模型，并将其应用于血细胞检测任务中。

本文的主要贡献如下：

提出了基于 YOLOv5 和 Transformer Block 的血细胞检测方法，并验证了其在复杂场景下的优越性能。

测试了 YOLOv10 模型，通过优化网络结构和引入先进的注意力机制，进一步提升了检测精度和效率。

在 BCCD 数据集上进行了大量实验，验证了所提方法的有效性和鲁棒性。

通过本文的研究，我们希望为血细胞检测提供一种高效、准确的解决方案，并推动计算机视觉技术在医疗领域的应用和发展。

## 2. 数据集介绍

### 2.1 BCCD 数据集

BCCD 数据集是一组血细胞照片，最初由 cosmicad 和 akshaylambda 开源。该数据集由 364 张图像组成，涵盖三类细胞：白细胞（WBC）、红细胞（RBC）和血小板（Platelets）。整个数据集中共有 4888 个标签，分布在这三类细胞中（不包括空标签）。

根据 Roboflow 的“数据集健康检查”提供的统计数据，可以看到每一类细胞的数量分布

情况。此外，数据集中还提供了示例图像，以便更直观地了解图像的内容。这个数据集是一个小规模的目标检测数据集，通常用于评估模型性能。它是医学影像能力的一个初步示例。数据集以公共领域的方式发布，任何人都可以自由使用它，无需提供归属，但如果能提及数据来源将会更好。

数据集的一张原图上面有许多细胞，有的细胞会重叠在一起，如图 1 右所示。该图片对应的标签是以数据的形式呈现，第一列有 0,1,2 这三种标签，分别表示白细胞（WBC）、红细胞（RBC）和血小板（Platelets）这三类。后面紧跟的四列数据分别表示中心点坐标，长度以及宽度。



图 1 原图（左）和标签（右）

## 2.2 BCCD - v4 416x416\_aug 数据集

该数据集通过 roboflow.ai 于 2021 年 2 月 23 日导出，共包含 874 张图像。所有细胞都以 YOLO v5 PyTorch 格式进行了标注。每张图像都经过了以下预处理：

1. 调整大小至 416x416 像素（拉伸处理）
2. 为了增加数据集的多样性，每张源图像都应用了以下增强操作，生成了三个版本：
  - (1) 50%的概率进行水平翻转
  - (2) 50%的概率进行垂直翻转
  - (3) 以相等的概率进行以下 90 度旋转之一：不旋转、顺时针旋转、逆时针旋转、倒置
3. 随机裁剪图像的 0 到 15%
4. 随机调整亮度，范围为-15%到+15%
5. 随机调整曝光，范围为-20%到+20%

这些预处理和数据增强操作有助于提高模型的泛化能力和鲁棒性，使其在实际应用中表现更好。原图和标签图与 BCCD 形式一致，如图 2 所示。

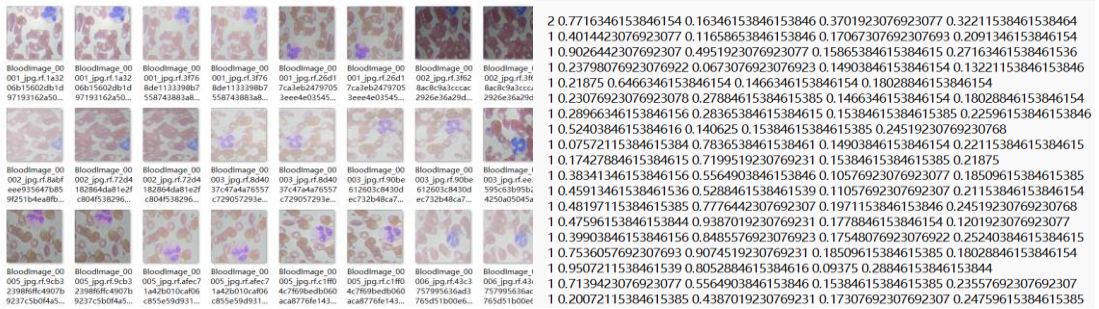


图 2 原图（左）和标签（右）

## 3.算法模块

### 3.1 传统算法

首先，我们使用 Python 编写了一个用于计数图像中的血细胞的程序。它采用了传统的计数算法，结合了计算机视觉和图像处理技术。

- 1) **颜色阈值分割：**定义血细胞的颜色范围，并创建一个掩码来分割出这些颜色区域。
- 2) **自适应阈值化：**使用自适应阈值化进一步改善掩码，使其更适合后续处理。
- 3) **形态学操作：**使用闭运算来连接断裂的细胞，形成完整的细胞区域。
- 4) **连通组件标记：**标记图像中的连通区域，每个连通区域代表一个细胞。
- 5) **区域过滤：**根据细胞的大小过滤掉不符合预期大小的区域。
- 6) **计数：**计算过滤后的区域数量，即血细胞的数量。
- 7) **结果可视化：**在原始图像上绘制标记的细胞边界，以验证计数结果。

传统算法分割以及计数的结果图如图 3 所示。

传统图像处理方法相对简单，易于理解和实现。二是实时性强：处理速度较快，适用于实时应用。三是低资源需求：不需要大量计算资源，适合在嵌入式系统上运行。

但是传统算法鲁棒性差：对于复杂背景和噪声敏感，需要精心调整参数（如阈值和滤波器大小）。对颜色依赖强：依赖于颜色分割，对于颜色变化或非红色细胞效果较差。难以处理重叠细胞：形态学操作虽然可以改善，但对重叠或形态复杂的细胞效果有限。参数调整繁琐：需要根据不同图像不断调整参数，难以通用。这种算法适用于血细胞计数等生物医学图像分析任务，但需要根据具体情况调整参数以获得最佳效果。

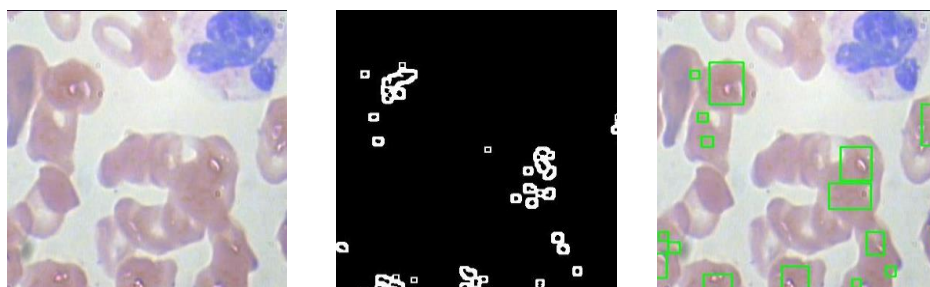


图 3 原图（左）、分割结果图（中）和计数结果（右）

### 3.2 YOLOv5

#### 3.2.1 YOLOv5 网络架构

YOLOv5（You Only Look Once version 5）是一种用于实时目标检测的深度学习模型。它的网络结构可以分为三个主要部分：Backbone、Neck 和 Head，如图 4 所示。

Backbone 的功能是提取图片的基本特征。其结构通常采用 CSPDarknet53 作为主干网络。CSP（Cross Stage Partial）网络通过跨阶段分离的方式来减少计算量，同时保持模型的准确性。Backbone 部分主要包含卷积层、残差模块和 CSP 模块。Neck 的功能是融合不同尺度的



特征。

其结构通常采用 PANet（Path Aggregation Network）结构。PANet 通过路径聚合网络来增强特征融合能力，特别是对不同尺度的特征进行融合，以便更好地检测不同大小的目标。Head 的功能是输出检测结果，包括目标的类别和位置。其结构由一系列卷积层组成，用于预测每个网格单元的目标类别和边界框。YOLOv5 的 Head 部分会输出三个尺度的预测，以便检测不同大小的目标。

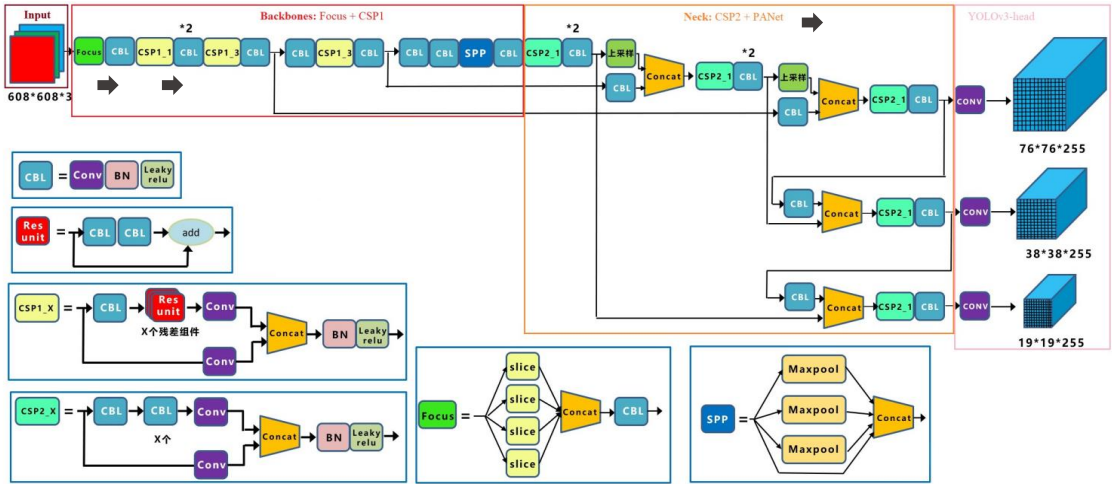


图 4 YOLOv5 网络结构

### 3.2.2 输入端：Mosaic 数据增强、自适应锚框计算、自适应图片缩放

#### (1)Mosaic 数据增强

在 YOLOv4 版本中，引入了一种名为"Mosaic"的数据增强技术，以提高模型的泛化能力和性能。Mosaic 数据增强的工作原理如下，效果如图 5 所示：

1. **选取图像：**从训练集中随机选取 4 张图像。
2. **缩放：**将这 4 张图像缩放到相同的分辨率。
3. **拼接：**将这 4 张图像在水平或垂直方向上拼接成一个大图像。具体来说，如果选择水平拼接，则将第一张和第二张图像拼接在一起，第三张和第四张图像拼接在一起，然后将这两个拼接后的图像再拼接在一起，形成一个 4 行 1 列的图像布局。
4. **数据增强：**对拼接后的大图像进行随机的数据增强操作，如随机水平翻转、随机颜色抖动等。
5. **输入模型：**将经过增强的大图像输入到 YOLO 模型中进行训练。

总的来说，Mosaic 数据增强是一种有效的技术，可以在不显著增加计算成本的情况下，提高目标检测模型的泛化能力和性能。

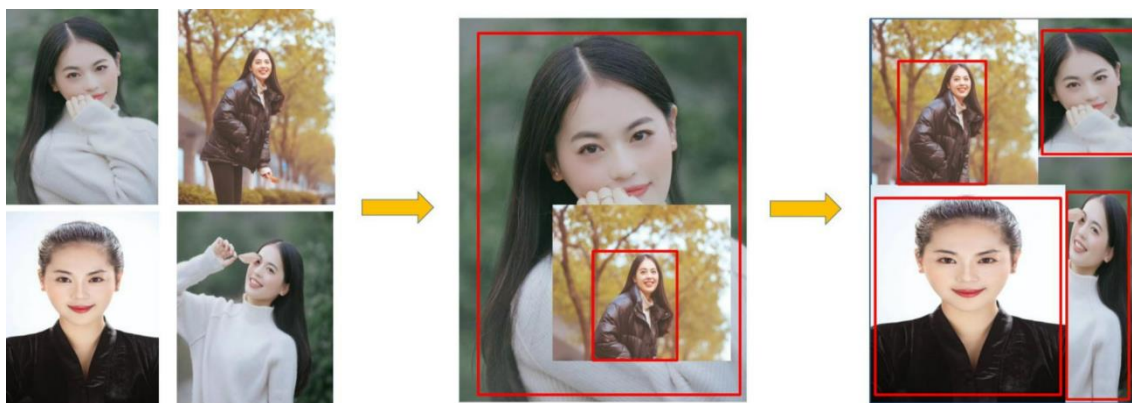


图 5 Mosaic 数据增强过程效果图

## (2)自适应锚框计算

在 Yolo 算法中，针对不同的数据集，都会有初始设定长宽的锚框。

在网络训练中，网络在初始锚框的基础上输出预测框，进而和真实框 `groundtruth` 进行比对，计算两者差距，再反向更新，迭代网络参数。

因此初始锚框也是比较重要的一部分，比如 YOLOv5 在 Coco 数据集上初始设定的锚框：

```
anchors:
- [10,13, 16,30, 33,23] # P3/8
- [30,61, 62,45, 59,119] # P4/16
- [116,90, 156,198, 373,326] # P5/32
```

图 6 YOLOv5 在 Coco 数据集上初始设定的锚框

在 YOLOv3、YOLOv4 中，训练不同的数据集时，计算初始锚框的值是通过单独的程序运行的。但 YOLOv5 中将此功能嵌入到代码中，每次训练时，自适应的计算不同训练集中的最佳锚框值。当然，如果觉得计算的锚框效果不是很好，也可以在代码中将自动计算锚框功能关闭。`parser.add_argument(*name_or_flags:'--noautoanchor', action='store_true', help='disable autoanchor check')`控制的代码即 `train.py` 中上面一行代码，设置成 `False`，每次训练时，不会自动计算。

## (3)自适应图片缩放

在常用的目标检测算法中，不同的图片长宽都不相同，因此常用的方式是将原始图片统一缩放到一个标准尺寸，再送入检测网络中。

比如 YOLO 算法中常用 416\*416, 608\*608 等尺寸，比如对下面图 7 中 800\*600 的图像进行缩放。

因此在 YOLOv5 的代码中 `datasets.py` 的 `letterbox` 函数中进行了修改，对原始图像自适应的添加最少的黑边。图像高度上两端的黑边变少了，在推理时，计算量也会减少，即目标检测速度会得到提升。通过这种简单的改进，推理速度得到了 37% 的提升，可以说效果很明显。



图 7 自适应缩放示意图

### 3.2.3 Backbone: Focus 结构, CSP 结构

#### (1) Focus 结构

Focus 结构, 在 YOLOv3&YOLOv4 中并没有这个结构, 其中比较关键的是切片操作。

比如图 8 右侧的切片示意图,  $4 \times 4 \times 3$  的图像切片后变成  $2 \times 2 \times 12$  的特征图。

以 YOLOv5s 的结构为例, 原始  $608 \times 608 \times 3$  的图像输入 Focus 结构, 采用切片操作, 先变成  $304 \times 304 \times 12$  的特征图, 再经过一次 32 个卷积核的卷积操作, 最终变成  $304 \times 304 \times 32$  的特征图。

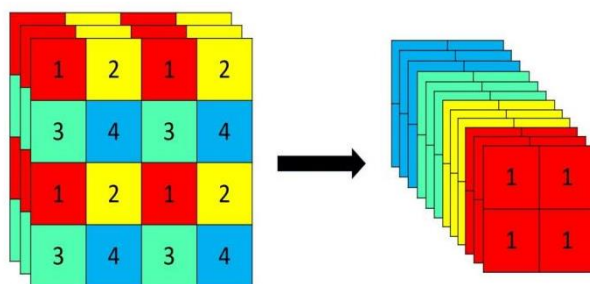


图 8 Focus 切片示意图

#### (2) CSP 结构

YOLOv5 中的 CSP 结构指的是 CSPNet (Cross Stage Partial Network) 的设计, 它是一种网络结构的改良, 旨在减少计算成本并提高网络的学习能力, 其结构示意图如图 9 所示。CSPNet 的核心思想是将特征图分成两部分, 然后在网络的不同阶段交叉合并这些部分。

将输入分为两个分支, 一个分支先通过 CBL, 再经过多个残差结构 (Bottleneck \* N), 再进行一次卷积; 另一个分支直接进行卷积; 然后两个分支进行 concat, 再经过 BN (正态分布), 再来一次激活, 最后进行一个 CBL。

CSP1\_X 应用于 backbone 主干网络部分, backbone 是较深的网络, 增加残差结构可以增加层与层之间反向传播的梯度值, 避免因为加深而带来的梯度消失, 从而可以提取到更细粒度的特征并且不用担心网络退化。



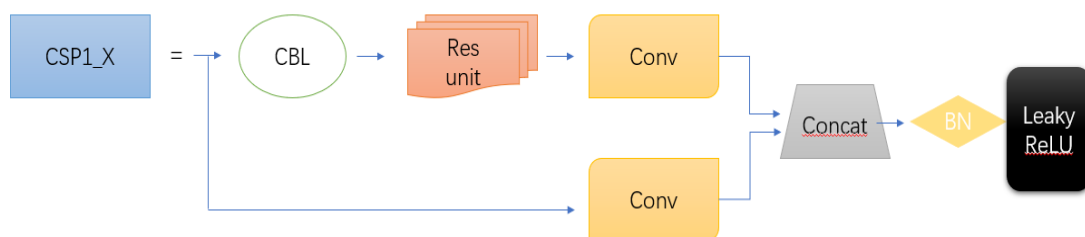


图 9 CSP 结构示意图

### 3.2.4 Neck: FPN+PAN 结构

FPN 是一种流行的特征金字塔结构，用于构建多尺度的特征图，以便于检测不同尺寸的物体，其结构图如图 10 所示。在 FPN 中，低层次的特征图（高分辨率、低语义信息）会与高层次的特征图（低分辨率、高语义信息）融合。这种自顶向下的结构使得每个层级的特征图都包含了丰富的语义信息，有助于提高小物体的检测性能。

PAN 则是在 FPN 的基础上进行改进，增加了自底向上的路径，以增强低层次特征图的信息。这种结构可以进一步增强特征图之间的信息流动，特别是将高层次的语义信息更有效地传递到低层次的特征图中。

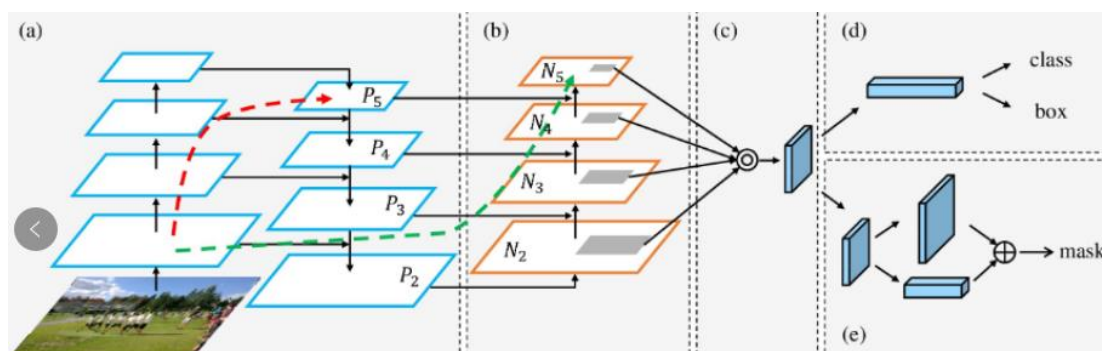


图 10 Neck 结构示意图

### 3.2.5 Prediction: GIOU\_Loss

GIOU Loss 在 IOU 的基础上进行了改进，引入了一个概念，即最小闭包区域（即包含两个边界框的最小矩形区域），并考虑了边界框与最小闭包区域的关系。

$$GIoU = IoU - \frac{|C - B|}{|C|}$$

其优点有：可以处理不相交的边界框，具有更稳定的梯度，能提高定位精度并且适用于各种尺寸的目标。其运用于图片的效果如图 11 所示。

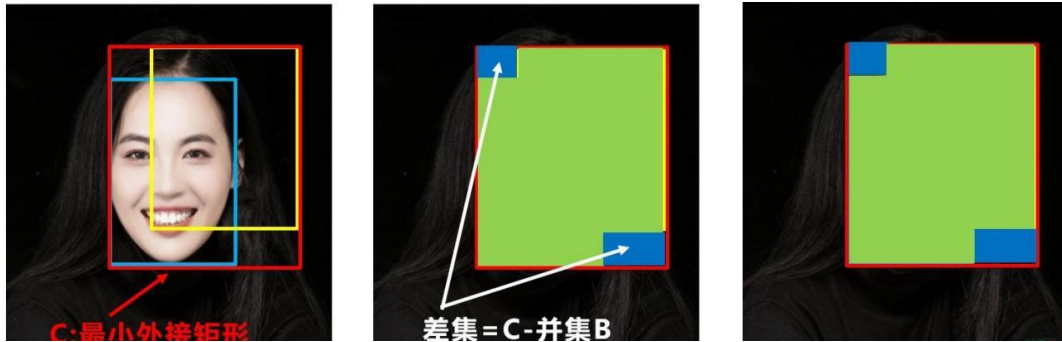


图 11 GIoU\_Loss 在图片上的演示示意图

### 3.3 YOLOv5+Transformer Block

在本次研究中，我们在 YOLOv5 的基础上引入了 Transformer Block 结构，以期提升模型在血细胞检测任务中的性能。YOLOv5 是由 Ultralytics 开发的目标检测模型，具有较高的检测速度和精度。通过引入 Transformer Block，我们希望利用其在捕捉长距离依赖关系和全局信息方面的优势，进一步提高模型的表现。

在 YOLOv5 的骨干网络中，我们在 SPPF（Spatial Pyramid Pooling-Fast）层之后引入了一个 Transformer Block。具体的网络结构如下：

```
# YOLOv5 v6.0 backbone with TransformerBlock
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [64, 6, 2, 2]], # 0-P1/2
   [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
   [-1, 3, C3, [128]],
   [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
   [-1, 6, C3, [256]],
   [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
   [-1, 9, C3, [512]],
   [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
   [-1, 3, C3, [1024]],
   [-1, 1, SPPF, [1024, 5]], # 9
   [-1, 1, TransformerBlock, [1024, 1024, 8, 1]], # 10 Add TransformerBlock
  ]
```

图 12 YOLOv5 中引入 Transformer Block 的代码设计

在引入 Transformer Block 后，模型能够更好地捕捉图像中的全局特征，提高对复杂场景中目标的检测能力。

### 3.4 YOLOv10

近年来，YOLO 系列模型在计算机视觉领域的实时目标检测中占据了主导地位。它们凭借在计算成本和检测性能之间的有效平衡，获得了广泛的应用。然而，传统的 YOLO 模型依赖于非极大值抑制（NMS）进行后处理，这不仅影响了模型的端到端部署，还增加了推理延迟。此外，YOLO 模型各组件的设计缺乏全面检查，导致计算冗余明显，模型能力有限。

为了解决这些问题，YOLOv10 应运而生。YOLOv10 在模型架构和后处理方法上进行了全面优化，旨在提高性能和效率。

### 3.4.1 去除 NMS 的双重分配策略

传统 YOLO 模型在训练过程中采用一对多标签分配策略，这虽然能提供丰富的监督信号，但需要在推理过程中使用 NMS 进行后处理，从而增加了推理时间。为解决这一问题，YOLOv10 提出了双重标签分配策略。具体来说，在训练时，模型同时进行一对多和一对一的标签分配，其中一对一分配用于最终的预测，从而在推理时无需 NMS。这一策略既保留了丰富的监督信号，又大幅提升了推理效率。YOLOv10 与其他模型的结果如图 13 所示。

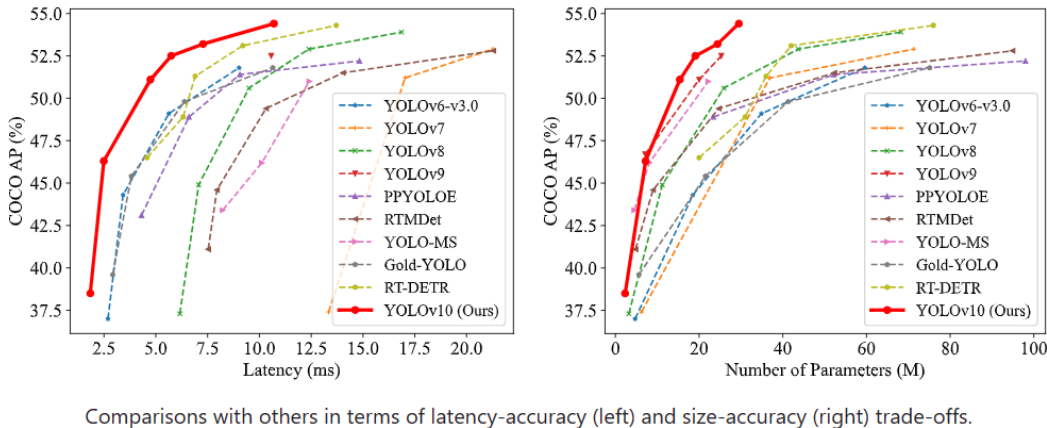


图 13 YOLOv10 与其他模型的对比结果

### 3.4.2 整体效率-准确性驱动下的模型设计

为了进一步优化模型的效率和准确性，YOLOv10 对各个组件进行了全面检查和改进。

#### (1) 轻量级分类头

- YOLOv10 采用了轻量级分类头设计，减少了计算冗余，提高了模型的计算效率。
- 1.减少计算冗余：**传统的分类头通常包含大量的全连接层和复杂的操作，这些操作虽然能够提高模型的准确性，但也会带来大量的计算冗余。YOLOv10 通过采用轻量级设计，减少了不必要的计算步骤，从而降低了计算复杂度。
  - 2.提高计算效率：**轻量级分类头通过简化网络结构，使得模型在推理阶段的计算速度显著提升。这对于实时应用尤为重要，例如在自动驾驶、视频监控等需要快速响应的场景中，YOLOv10 的轻量级设计能够提供更快的检测速度。
  - 3.减少参数量：**轻量级分类头通常包含较少的参数，这不仅减少了模型的存储需求，还降低了训练过程中对硬件资源的依赖。这意味着在相同的硬件环境下，YOLOv10 可以处理更大的数据集或进行更多次的迭代训练，从而提高模型的泛化能力。
  - 4.保持高准确率：**尽管采用了轻量级设计，YOLOv10 在分类精度上并没有显著下降。通过精心设计的网络结构和优化算法，YOLOv10 在保持高计算效率的同时，依然能够提供较高的检测准确率。这使得 YOLOv10 在实际应用中既高效又可靠。

**5.适应多种设备：**由于轻量级分类头减少了计算和存储需求，YOLOv10 可以更容易地部署在各种设备上，包括资源受限的嵌入式设备和移动设备。这种灵活性使得 YOLOv10 能够在更广泛的应用场景中发挥作用。

## (2) 空间-通道解耦下采样

通过空间-通道解耦下采样策略，YOLOv10 在保持精度的同时，降低了计算复杂度。

### 1.空间和通道的独立处理：

传统的下采样方法通常同时处理空间和通道信息，这会导致大量的计算冗余。空间-通道解耦下采样策略将空间信息和通道信息独立处理，分别进行降维和压缩，从而减少了不必要的计算步骤。

### 2.降低计算复杂度：

空间-通道解耦下采样通过将空间维度和通道维度的操作解耦，减少了每一步计算所需的操作数。这种方法不仅降低了模型的计算复杂度，还减少了内存占用，使得模型在推理阶段更加高效。

## (3) 大核卷积和部分自注意模块

为了增强模型的特征提取能力，YOLOv10 引入了大核卷积和部分自注意模块。

**1.提升感受野：**大核卷积通过使用较大的卷积核，增加了模型的感受野，使得模型能够捕捉更大范围的特征。这对于检测大尺度目标和复杂背景下的目标非常有帮助。

**2.减少层数：**相比于使用多个小卷积核层堆叠来增加感受野，大核卷积可以在较少的层数下实现相同的效果，从而减少了计算量和参数数量。

**3.提高特征表达能力：**大核卷积能够更好地整合全局信息，提高特征的表达能力，使得模型在处理复杂场景时更加鲁棒。

**4.增强特征选择性：**部分自注意模块通过自注意机制，能够动态地调整特征图中各个位置的权重，使得模型能够更有选择性地关注重要的特征区域。这种机制能够有效提升模型的特征提取能力。

**5.降低计算成本：**相比于全局自注意机制，部分自注意模块仅在局部范围内计算注意力权重，从而显著降低了计算成本。这使得模型在保持高性能的同时，依然能够高效运行。

**6.提高模型的鲁棒性：**自注意模块通过捕捉特征之间的长距离依赖关系，使得模型在面对复杂背景和遮挡时，依然能够准确识别目标，增强了模型的鲁棒性。

## 3.5 YOLOv10+Transformer Block

在本次研究中，我们在 YOLOv10 的基础上引入了 Transformer Block 结构，以期提升模型在血细胞检测任务中的性能。YOLOv10 是 YOLO 系列的最新版本，具有更高的检测速度和精度。通过引入 Transformer Block，我们希望利用其在捕捉长距离依赖关系和全局信息方面的优势，进一步提高模型的表现。

在 YOLOv10 的骨干网络中，我们在 PSA（Pyramid Spatial Attention）层之后引入了一个 Transformer Block。具体的网络结构如下：

```

backbone:
  # [from, repeats, module, args]
  - [-1, 1, Conv, [64, 3, 2]] # 0-P1/2
  - [-1, 1, Conv, [128, 3, 2]] # 1-P2/4
  - [-1, 3, C2f, [128, True]]
  - [-1, 1, Conv, [256, 3, 2]] # 3-P3/8
  - [-1, 6, C2f, [256, True]]
  - [-1, 1, SCDowN, [512, 3, 2]] # 5-P4/16
  - [-1, 6, C2fCIB, [512, True]]
  - [-1, 1, SCDowN, [1024, 3, 2]] # 7-P5/32
  - [-1, 3, C2fCIB, [1024, True]]
  - [-1, 1, SPPF, [1024, 5]] # 9
  - [-1, 1, PSA, [1024]] # 10
  - [-1, 1, TransformerBlock, [1024, 1024, 8, 1]] # 插入 Transformer Block

```

图 14 YOLOv10 中加入 Transformer Block 代码设计

每个 Transformer Block 包含以下部分：

**1.多头自注意力机制（Multi-Head Self-Attention）：**输入特征通过多个不同的注意力头进行处理，以捕捉不同的特征模式。

**2.前馈神经网络（Feed-Forward Neural Network）：**用于对注意力机制的输出进行进一步处理和非线性变换。

引入 Transformer Block 后的 YOLOv10 模型在血细胞检测任务中表现出色。具体表现为：

**1.检测精度的提升：**加入 Transformer Block 后，模型在 BCCD 数据集上的检测精度提高，特别是在复杂背景和重叠细胞的检测中表现尤为突出。

**2.泛化能力的增强：**Transformer Block 增强了模型的全局特征提取能力，使其在不同场景下的检测效果更加稳定和鲁棒。

通过上述改进，我们成功地提升了 YOLOv10 在血细胞检测任务中的性能。

## 4.实验结果

### 4.1 YOLOv5s 训练 50 轮

我们可视化了训练损失曲线 `train`、验证损失曲线 `val` 以及评价指标曲线 `metrics`，如图 15 所示。

其中 `train/box_loss` 表示训练过程中边界框回归的损失。随着训练迭代次数的增加，`box_loss` 稳步下降，表明模型在定位细胞方面逐渐变得更好。`train/obj_loss` 表示训练过程中目标置信度的损失。该损失也在逐渐下降，表明模型在区分细胞和背景方面表现得越来越好。`train/cls_loss` 表示训练过程中分类损失。该损失快速下降，表明模型在识别细胞类别方面的性能快速提升。

`val/box_loss` 表示验证过程中边界框回归的损失。验证损失总体呈下降趋势，但初期有较大的波动。随着训练进行，损失趋于稳定，表明模型在验证数据上的定位性能也在提升。`val/obj_loss` 表示验证过程中目标置信度的损失。该损失波动较大，但整体趋势显示略有上升，这可能表明模型在某些验证样本上的置信度估计不够稳定。`val/cls_loss` 表示验证过程中



分类损失。该损失也在下降，表明模型在验证数据上识别细胞类别的性能逐渐提高。

`metrics/precision` 表示精确度（Precision）。精确度在初期快速上升并趋于稳定，保持在较高水平（接近 0.8 到 0.9），表明模型在减少误检方面表现良好。`metrics/recall` 表示召回率（Recall）。召回率也保持在较高水平（约 0.9），表明模型在检测到大多数细胞方面表现出色。`metrics/mAP_0.5` 表示在 IoU 阈值为 0.5 时的平均精度（Mean Average Precision）。`mAP_0.5` 快速上升并趋于稳定，保持在较高水平（接近 0.9），表明模型整体检测性能良好。`metrics/mAP_0.5:0.95` 表示在多个 IoU 阈值（从 0.5 到 0.95）下的平均精度。`mAP_0.5:0.95` 也快速上升并趋于稳定，保持在较高水平（接近 0.75 到 0.8），表明模型在不同精度要求下表现稳定且出色。

最后使用训练的最终模型预测得到的样例图如图 16 所示。

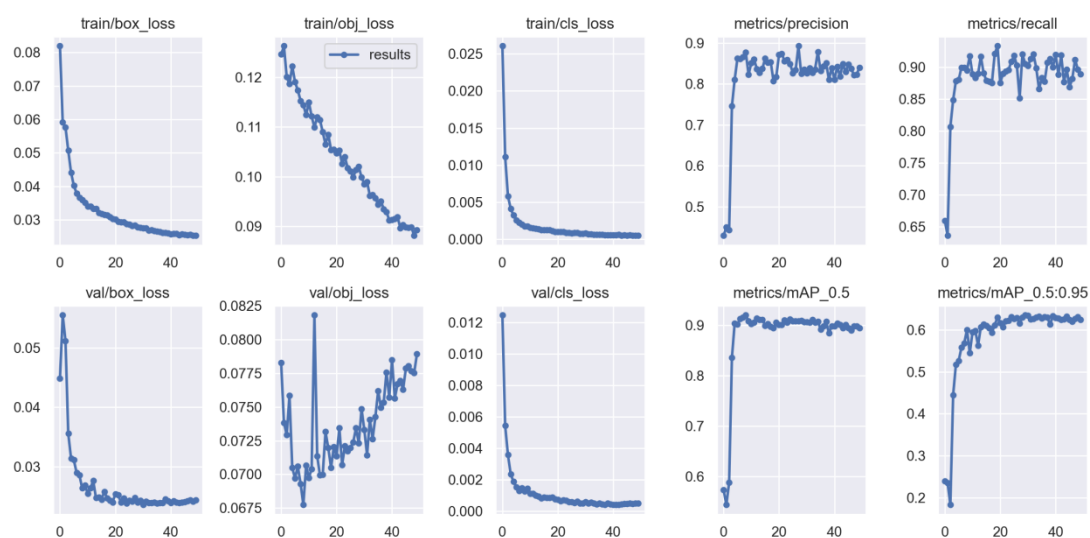


图 15 YOLOv5s 训练 50 轮过程可视化曲线

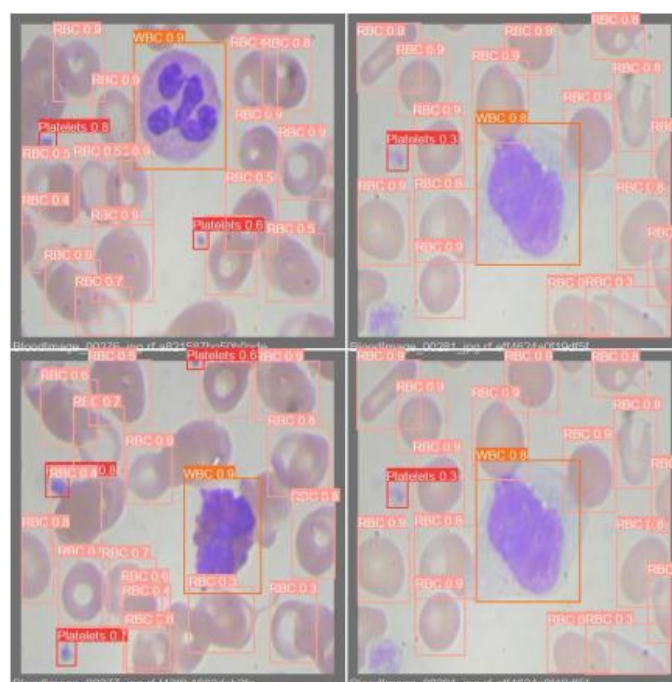


图 16 YOLOv5s 训练 50 轮预测检测计数结果

## 4.2 YOLOv5s 训练 100 轮

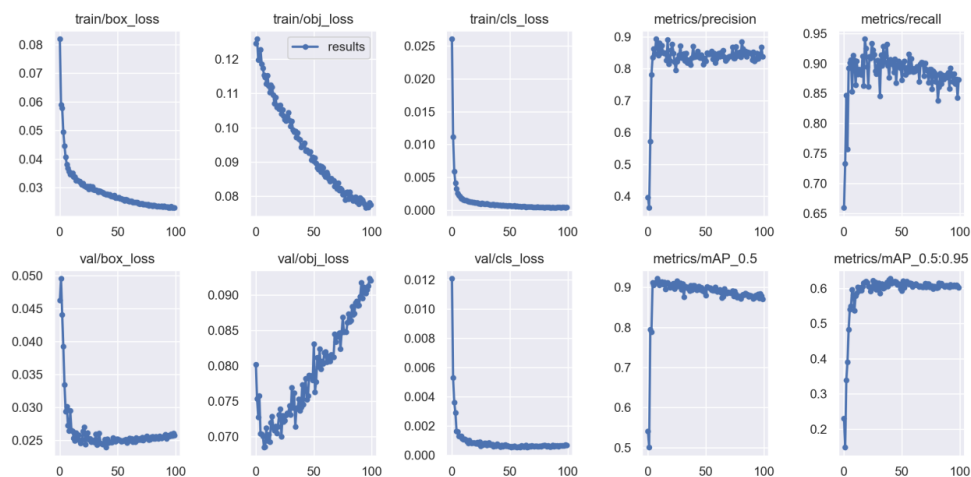


图 17 YOLOv5s 训练 100 轮过程可视化曲线

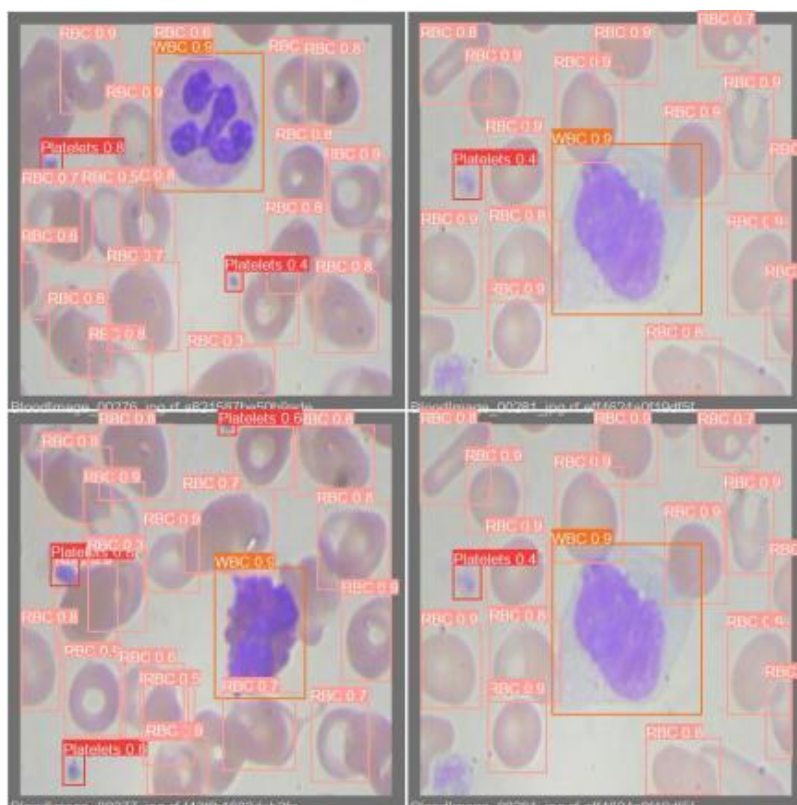


图 18 YOLOv5s 训练 100 轮预测检测计数结果

### 4.3 YOLOv5s+Transformer Block 训练 50 轮

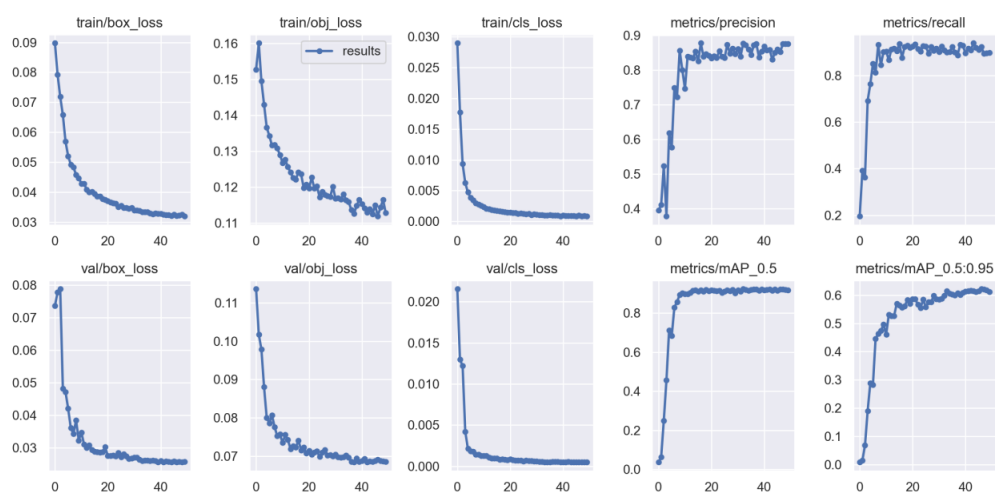


图 19 YOLOv5s+Transformer Block 训练 50 轮过程可视化曲线

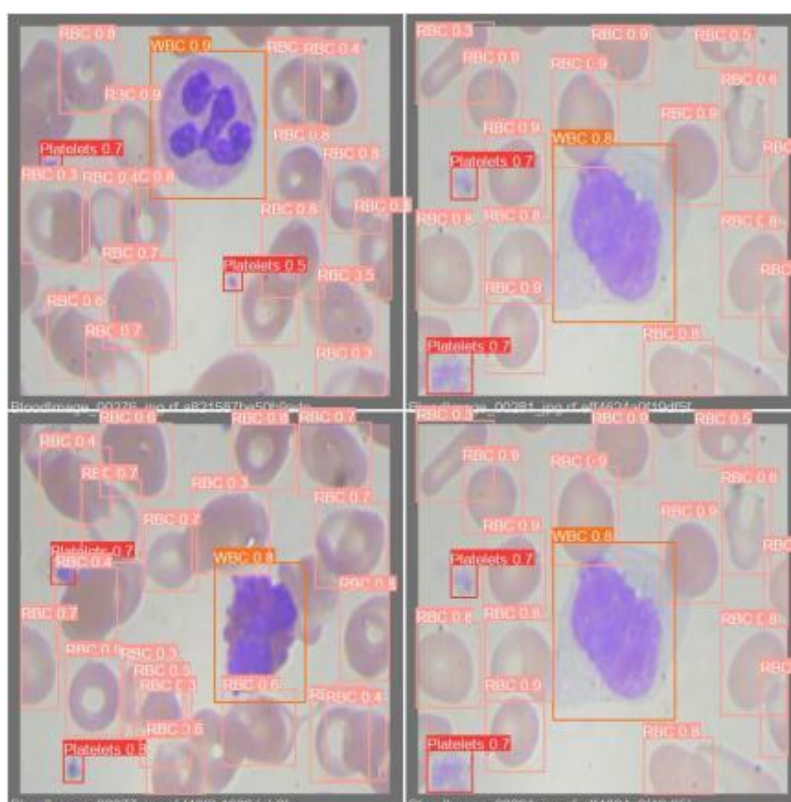


图 20 YOLOv5s+Transformer Block 训练 50 轮预测检测计数结果

## 4.4 YOLOv5s+Transformer Block 训练 100 轮

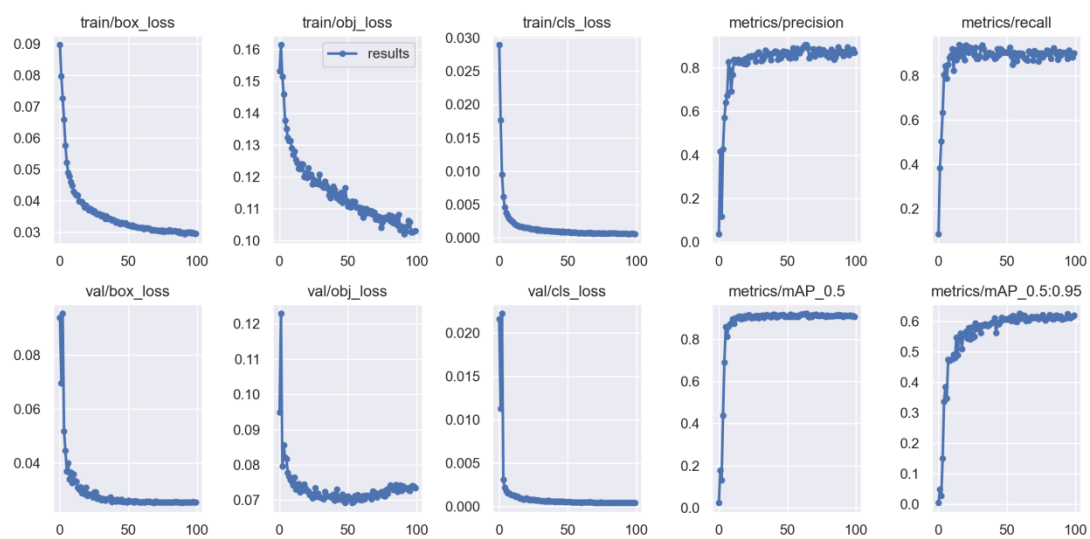


图 21 YOLOv5s+Transformer Block 训练 100 轮过程可视化曲线

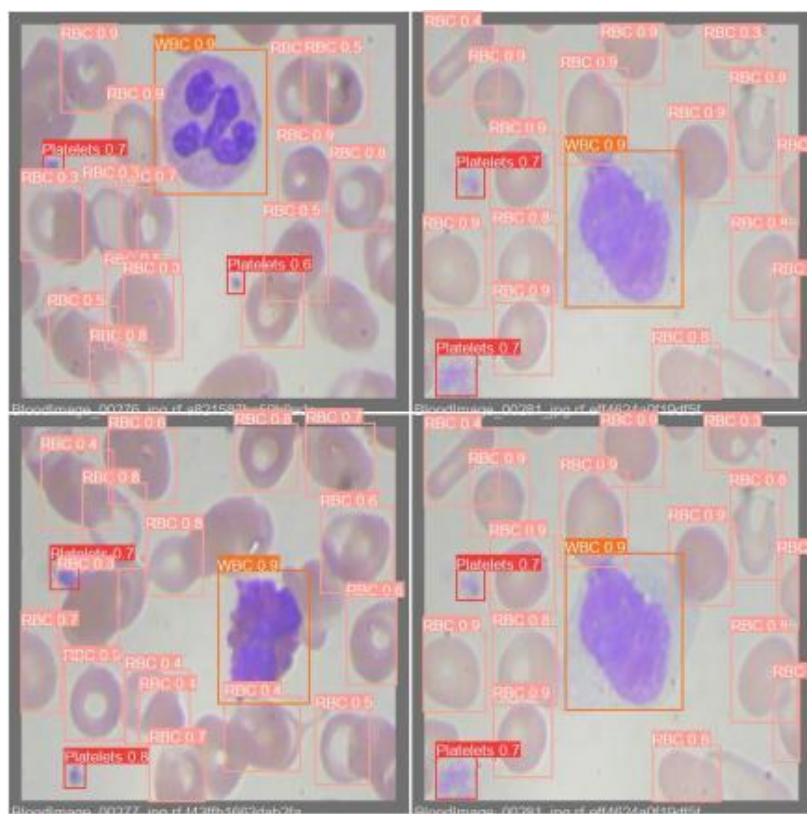


图 22 YOLOv5s+Transformer Block 训练 100 轮预测检测计数结果



## 4.5 YOLOv10s 训练 100 轮

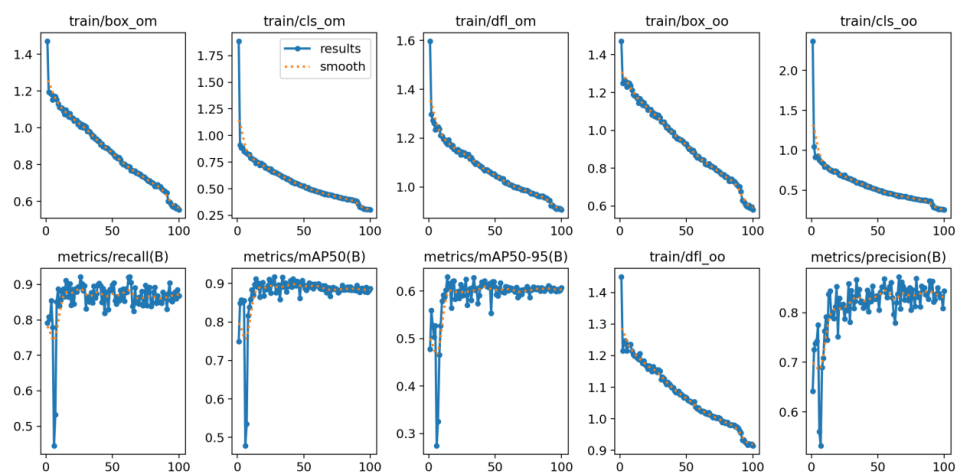


图 23 YOLOv10s 训练 100 轮过程可视化曲线

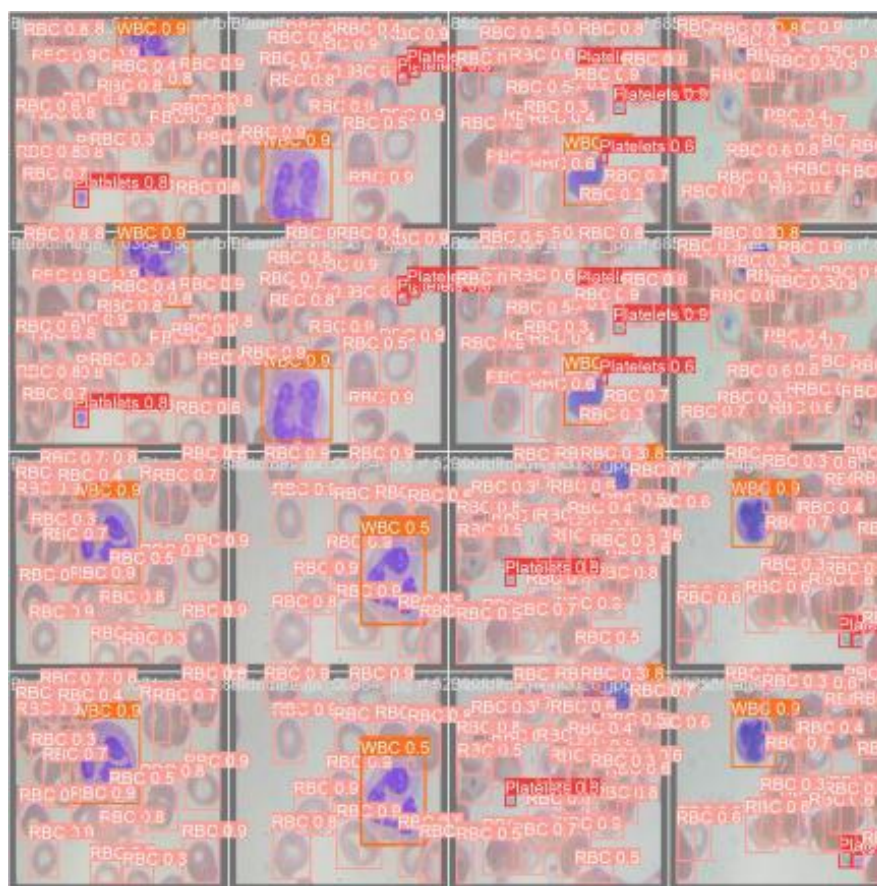


图 24 YOLOv10s 训练 100 轮预测检测计数结果



## 4.6 YOLOv10b 训练 100 轮

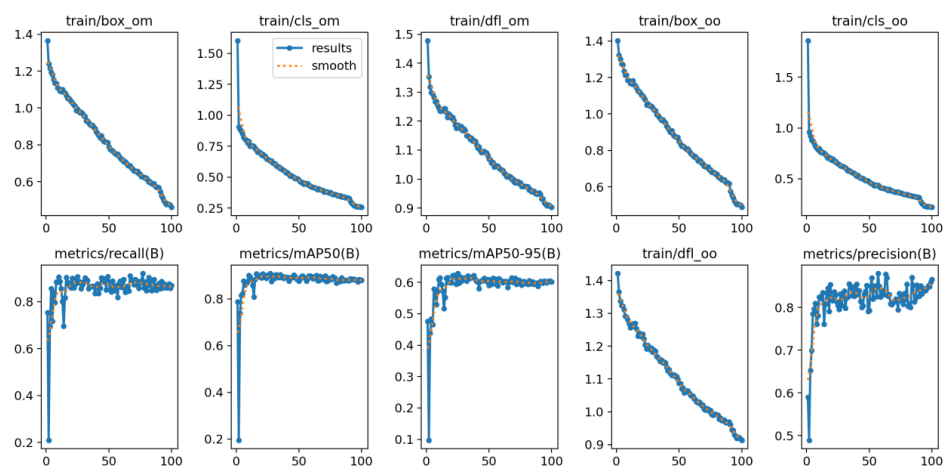


图 25 YOLOv10b 训练 100 轮过程可视化曲线

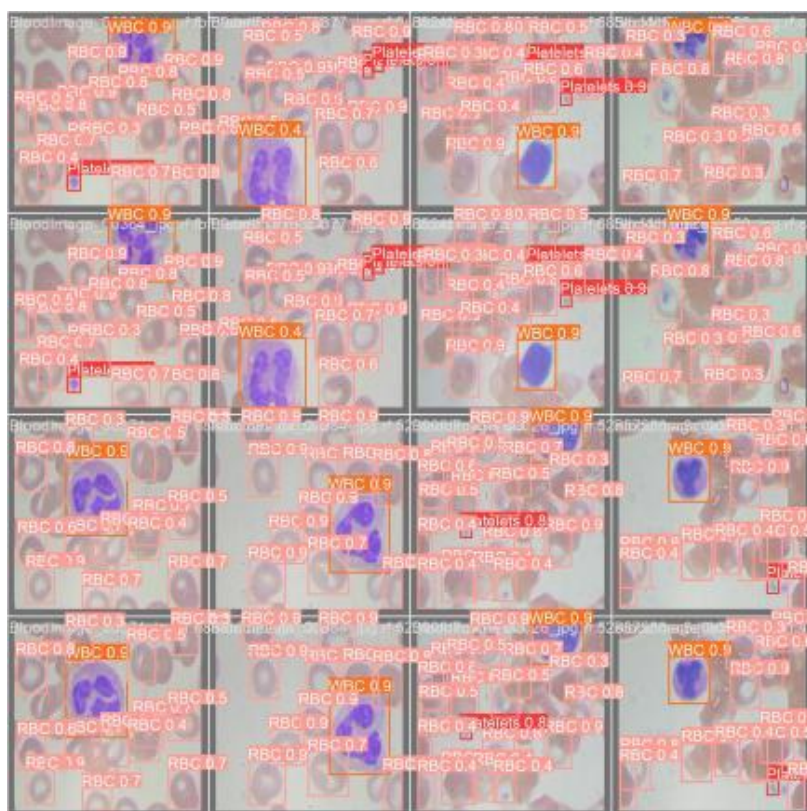


图 26 YOLOv10b 训练 100 轮预测检测计数结果

## 4.7 YOLOv10x(YOLOv10 中最大的模型)训练 100 轮

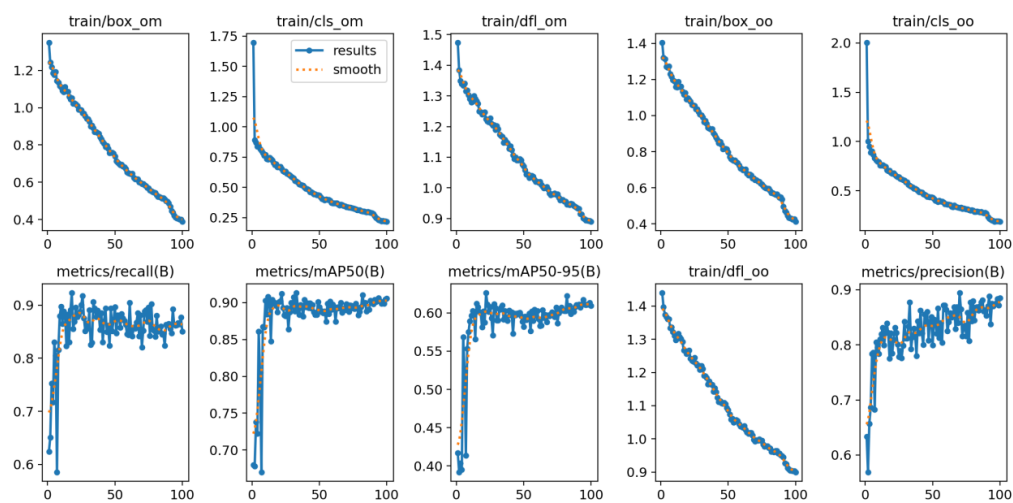


图 27 YOLOv10x 训练 100 轮过程可视化曲线



图 28 YOLOv10x 训练 100 轮预测检测计数结果

# 5.基于 YOLOv10 与 Gradio 的实时对象检测 Web 应用

为了能够对 YOLOv10 模型进行可视化并实时检测对象，我们开发了一款基于 Gradio 的 Web 应用。Gradio 是一个开源的 Python 库，可以快速创建用户友好的 Web 界面，用于模型的演示和测试。在本部分中，我们详细介绍了如何利用 Gradio 构建一个实时对象检测的 Web 应用，并展示 YOLOv10 模型的检测结果。

## 5.1 系统架构

- 我们的 Web 应用主要由以下几个部分组成：
  - 模型加载：**在应用启动时加载 YOLOv10 模型，包括多个预训练模型，如 yolov10n.pt、yolov10s.pt、yolov10b.pt、yolov10x.pt 和 best.pt。
  - 前端界面：**使用 Gradio 构建用户交互界面，包括图像上传、模型选择、图像尺寸调整和置信度阈值设置等功能。
  - 推理函数：**定义 YOLOv10 模型的推理函数，处理用户上传的图像，并返回检测结果。

## 5.2 功能介绍

**图像上传：**用户可以通过界面上上传图像，支持多种图像格式。

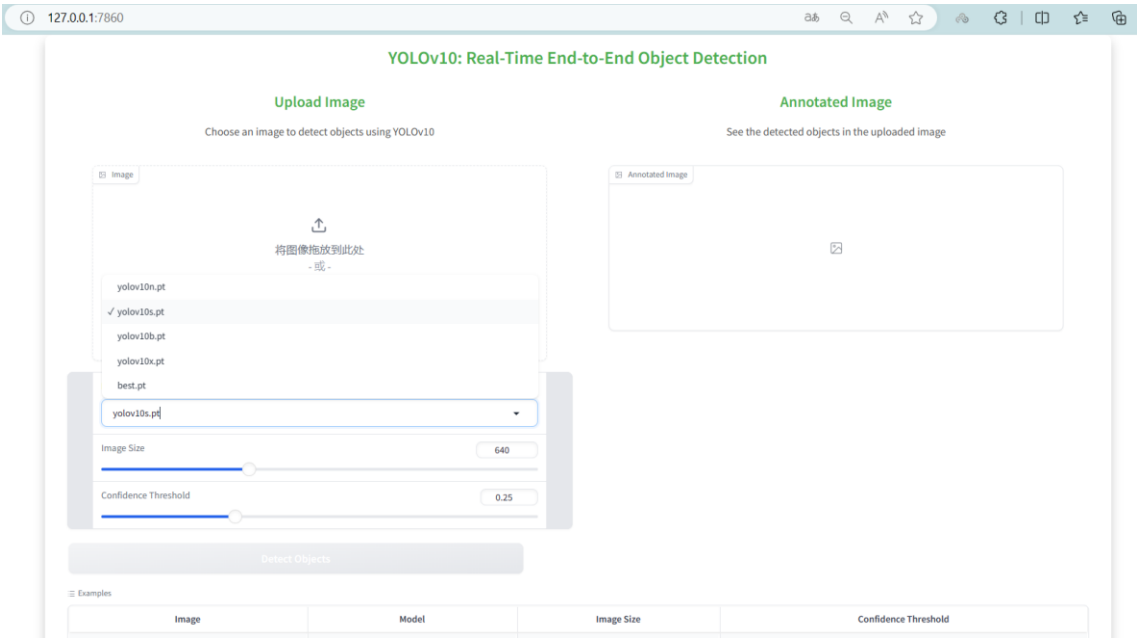


图 29 交互界面

**模型选择：**用户可以从多个预训练模型中选择一个进行对象检测。

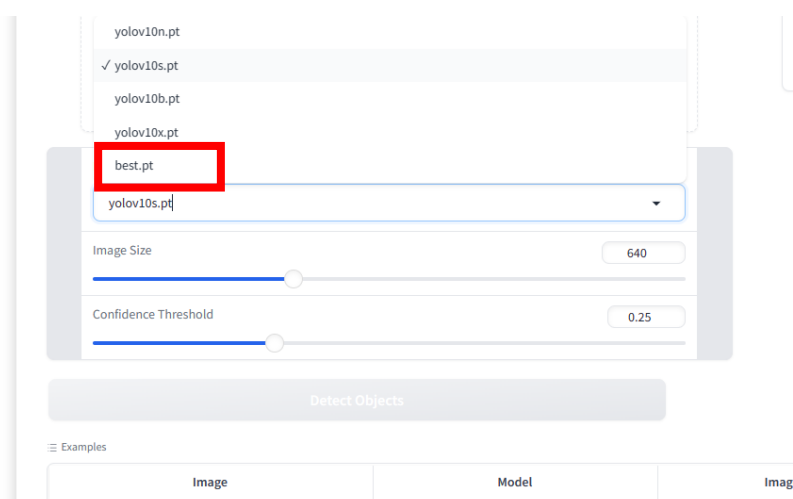


图 30 上传模型

**图像尺寸调整：**用户可以通过滑动条调整输入图像的尺寸，以适应不同的检测需求。

**置信度阈值设置：**用户可以设置置信度阈值，以控制检测结果的精度。

**检测结果显示：**检测完成后，应用会在界面上显示带有标注的图像，展示检测到的对象。

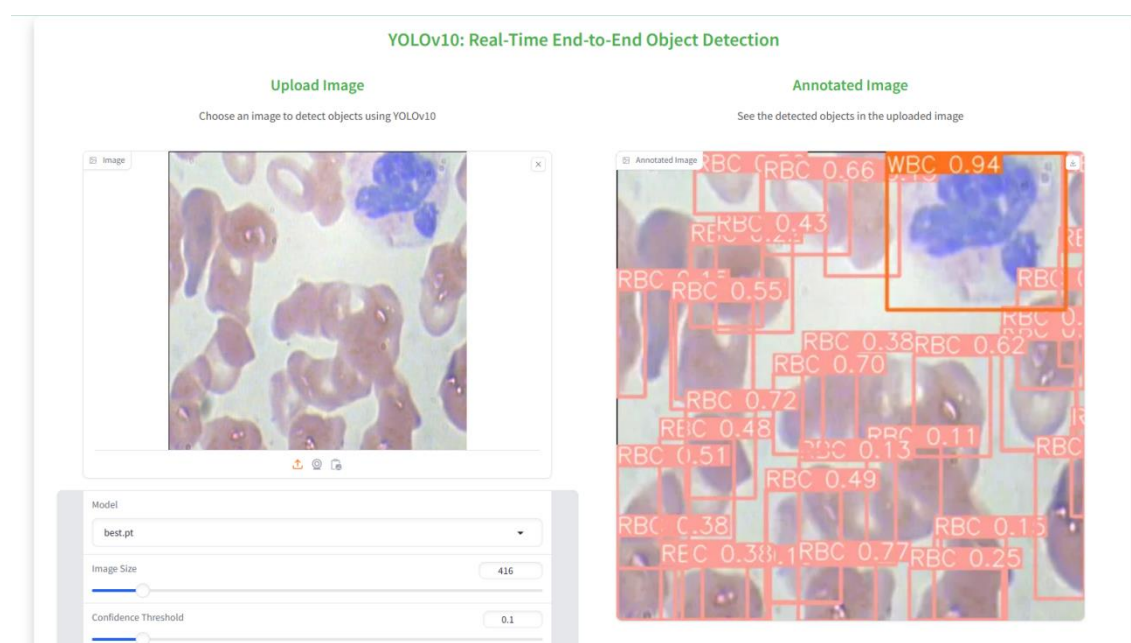


图 31 预测结果

## 5.3 实验结果与分析

通过本 Web 应用，我们可以方便地对 YOLOv10 模型进行测试和验证。用户可以上传任意图像，选择不同的模型和参数进行检测，并实时查看检测结果。该应用不仅提高了模型测试的效率，还为用户提供了直观的检测结果展示，有助于进一步优化和改进模型。

通过上述改进，我们成功地提升了 YOLOv10 在血细胞检测任务中的性能，并通过 Gradio 提供了一个用户友好的 Web 界面，使得模型的使用更加便捷和高效。

## 6.总结与展望

在本文中，我们探讨了基于 YOLO 架构的血细胞检测方法，并引入了 YOLOv5 和 YOLOv10 模型来提升检测性能。通过对比传统的 YOLOv5 和改进后的 YOLOv5+Transformer Block，以及新版本的 YOLOv10 和其变体，我们发现改进后的模型在血细胞检测任务中表现出更高的精度和效率。特别是，YOLOv10 结合了去除 NMS 的双重分配策略和整体效率-准确性驱动的设计，在不同的数据集上均展示了优异的性能。

YOLOv10 的多项改进，包括轻量级分类头设计、增强的特征提取能力以及更高的推理速度，使其在实际应用中具有较大的优势。我们通过实验验证了这些改进带来的性能提升，结果表明，YOLOv10 不仅在检测精度上优于传统模型，同时在推理速度和计算资源消耗方面也有显著优势。这些特点使得 YOLOv10 特别适用于实时对象检测的应用场景，如自动驾驶和视频监控。

展望未来，我们认为以下几个方向值得进一步研究：

**(1) 模型优化：**进一步优化 YOLOv10 的结构，以减少计算冗余和模型参数量，提高模型在移动设备上的部署性能。

**(2) 数据增强：**探索更多的数据增强技术，以提升模型的鲁棒性和泛化能力，尤其是在处理复杂背景和多样化细胞形态时。

**(3) 多任务学习：**结合多任务学习方法，使模型不仅能够进行细胞检测，还能同时进行分类和计数，从而提高临床应用的实用性。

**(4) 跨领域应用：**将改进后的 YOLO 模型应用于其他医学影像分析任务，如肿瘤检测、器官分割等，以验证其广泛适用性和优越性能。

总的来说，本研究展示了 YOLO 在血细胞检测中的巨大潜力，并为未来的改进和应用指明了方向。我们相信，通过持续的优化和创新，基于 YOLO 的检测模型将在医学影像分析中发挥越来越重要的作用。