



基于 YOLO 的血细胞检测

学院：	信息工程学院
班级：	XG21B2DZ
学号：	1211002019
组员：	孟凡千
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
3. 算法模块	4
3.1 YOLOv5	4
3.1.1 YOLOv5 网络架构	4
3.1.2 输入端: Mosaic 数据增强、自适应锚框计算、自适应图片缩放	5
3.1.3 Backbone: Focus 结构, CSP 结构	6
3.1.4 Neck: FPN+PAN 结构	7
3.1.5 Prediction: GIoU_Loss	8
3.2 YOLOv5+Transformer Block	8
4. 实验结果	9
4.1 YOLOv5s 训练 50 轮	9
4.2 YOLOv5s 训练 100 轮	11
4.3 YOLOv5s+Transformer Block 训练 50 轮	12
4.4 YOLOv5s+Transformer Block 训练 100 轮	13
5. 基于 YOLOv10 与 Gradio 的实时对象检测 Web 应用	14
5.1 系统架构	14
5.2 功能介绍	14
5.3 实验结果与分析	15
6. 总结与展望	16

1.引言与现状

计算机视觉是人工智能的一个重要领域，通过对图像或视频的处理、理解和分析，实现环境的识别和理解。近年来，随着深度学习技术的快速发展，计算机视觉在目标检测、图像分类和语义分割等方面取得了显著进展。

目标检测是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、安防监控和医疗影像等领域。作为 YOLO 系列的最新版本，YOLOv5 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。尽管基于深度学习的血细胞检测方法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、细胞形态多样和重叠严重等挑战。

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。此外，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。

为应对这些挑战，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：

- 提出了一种基于 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，数据集中还提供了一个示例图像，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。后面紧跟着的四列数据分别表示中心点坐标，长度以及宽度。

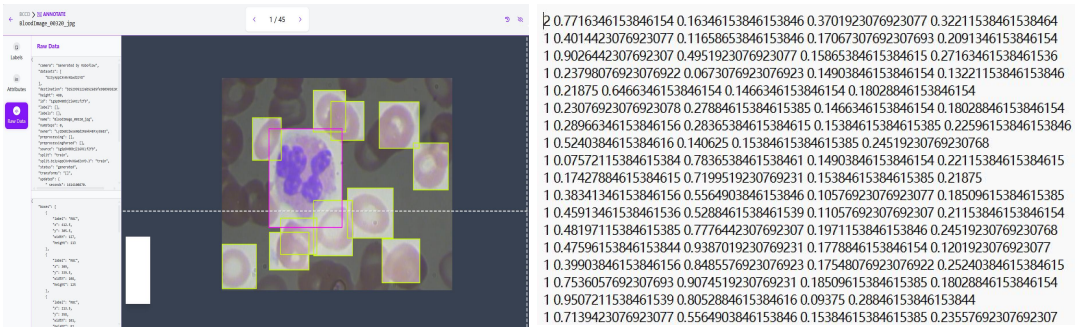


图 1 原图（左）和标签（右）

3.算法模块

3.1 YOLOv5

3.1.1 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 2 所示。

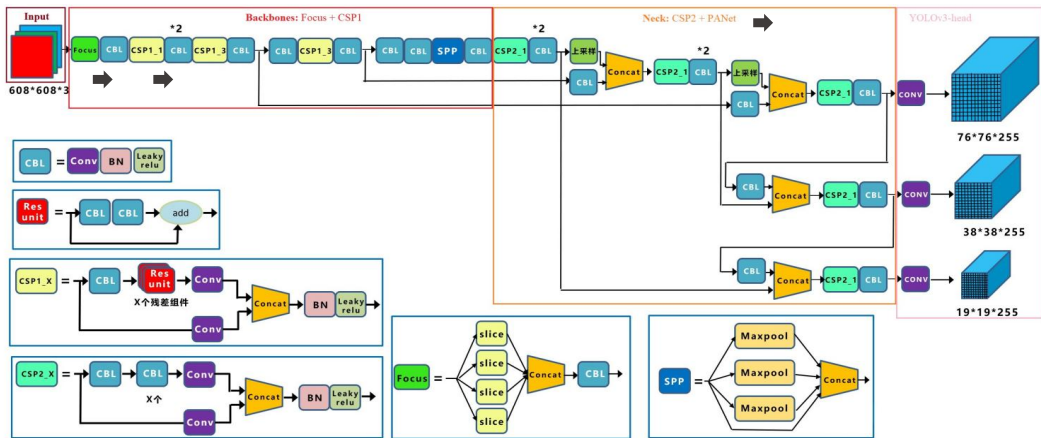


图 2 YOLOv5 网络结构

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。

3.1.2 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 3 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。

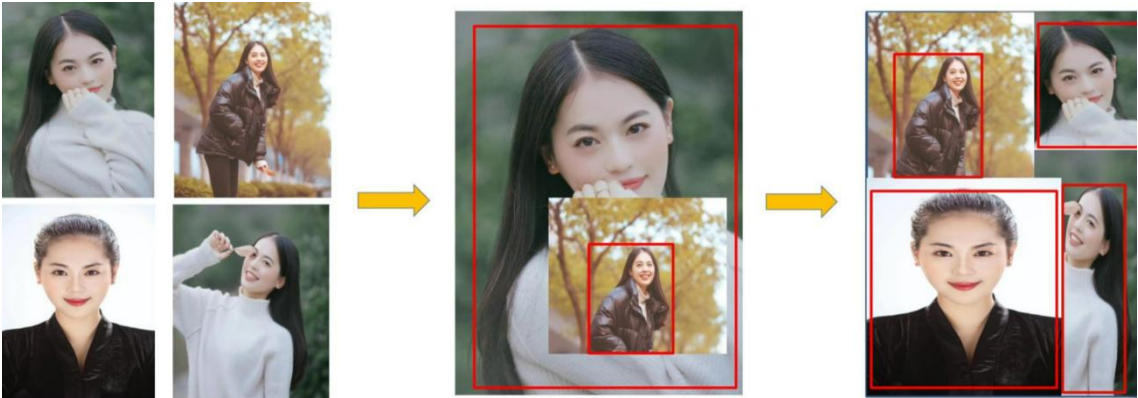


图 3 Mosaic 数据增强过程效果图

(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 进行对比，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 4 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 5 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 5 自适应缩放示意图

3.1.3 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 6 右侧的切片示意图，4*4*3 的图像切片后变成 2*2*12 的特征图。

以 YOLOv5s 的结构为例，原始 608*608*3 的图像输入 Focus 结构，采用切片操作，先变成 304*304*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*304*32 的特征图。

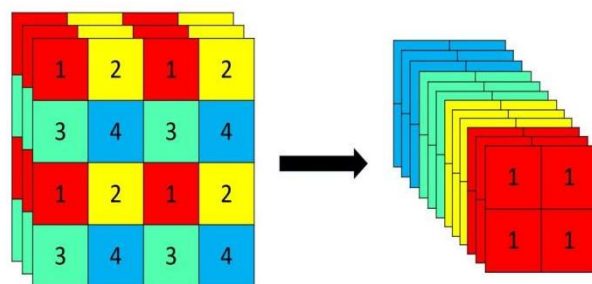


图 6 Focus 切片示意图

(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 7 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。

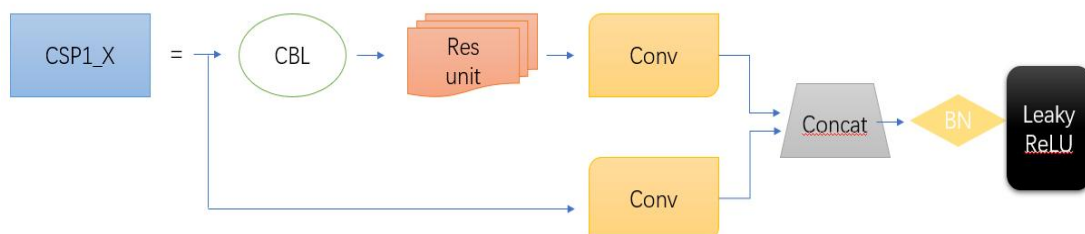


图 7 CSP 结构示意图

3.1.4 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 8 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。

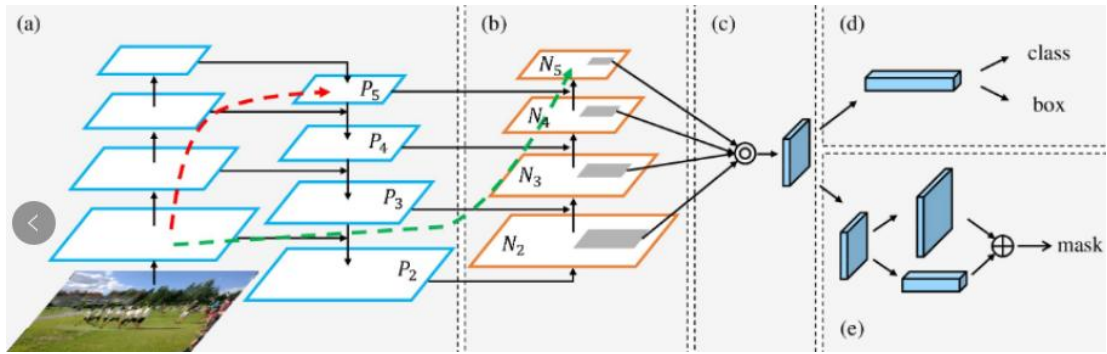


图 8 Neck 结构示意图

3.1.5 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 9 所示。

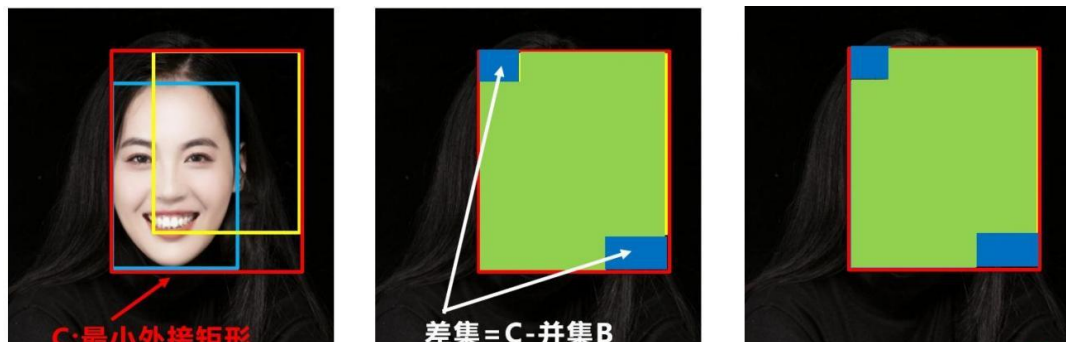


图 9 GIoU_Loss 在图片上的演示示意图

3.2 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 10 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。

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 11 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 12 所示。

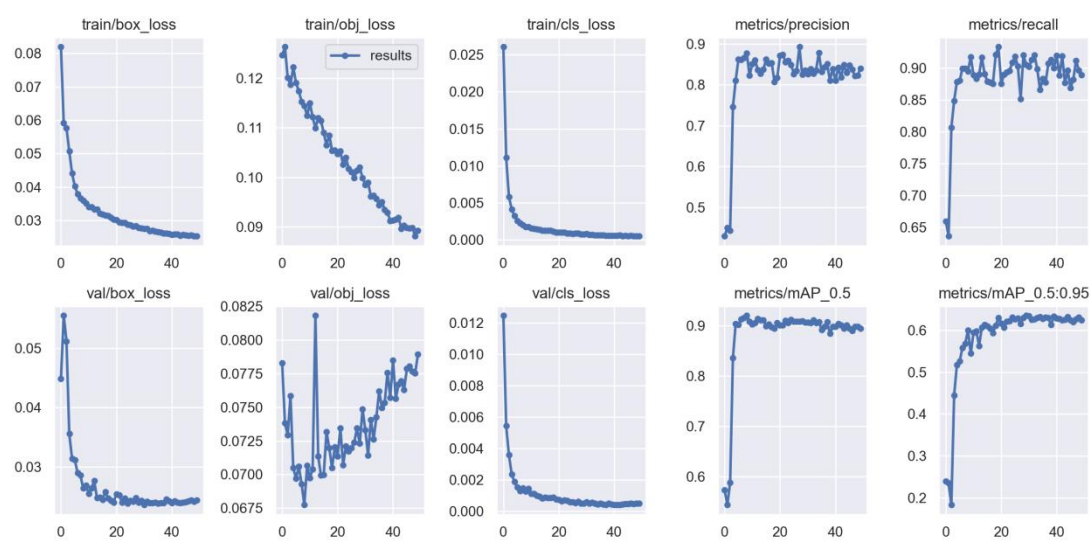


图 11 YOLOv5s 训练 50 轮过程可视化曲线

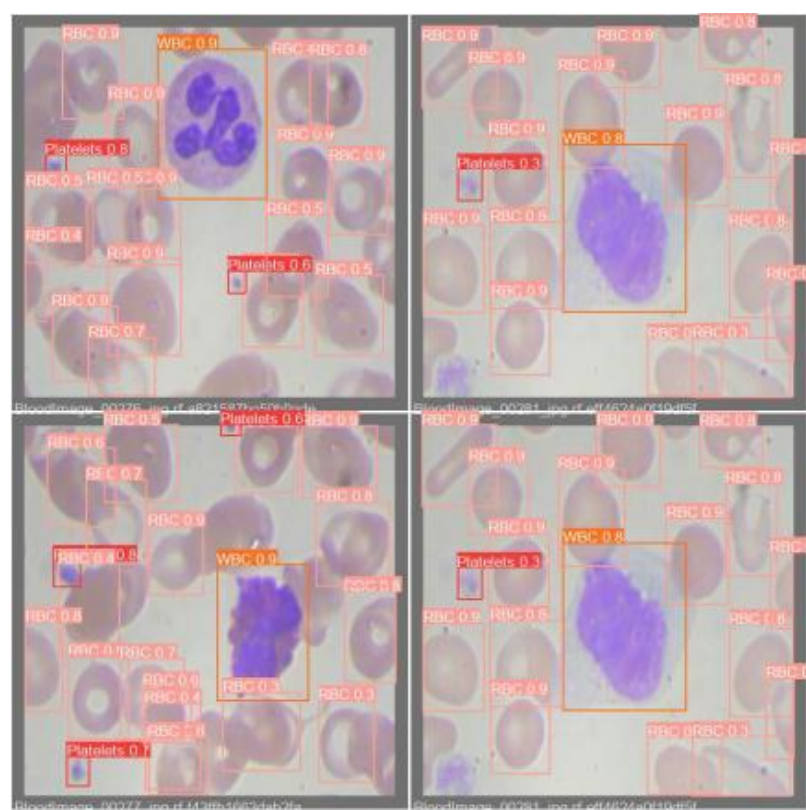


图 12 YOLOv5s 训练 50 轮预测检测计数结果

4.2 YOLOv5s 训练 100 轮

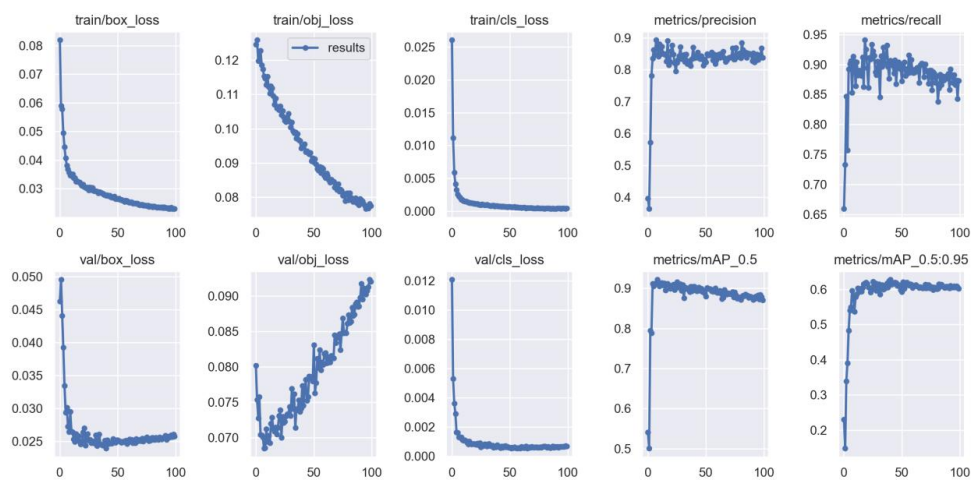


图 13 YOLOv5s 训练 100 轮过程可视化曲线

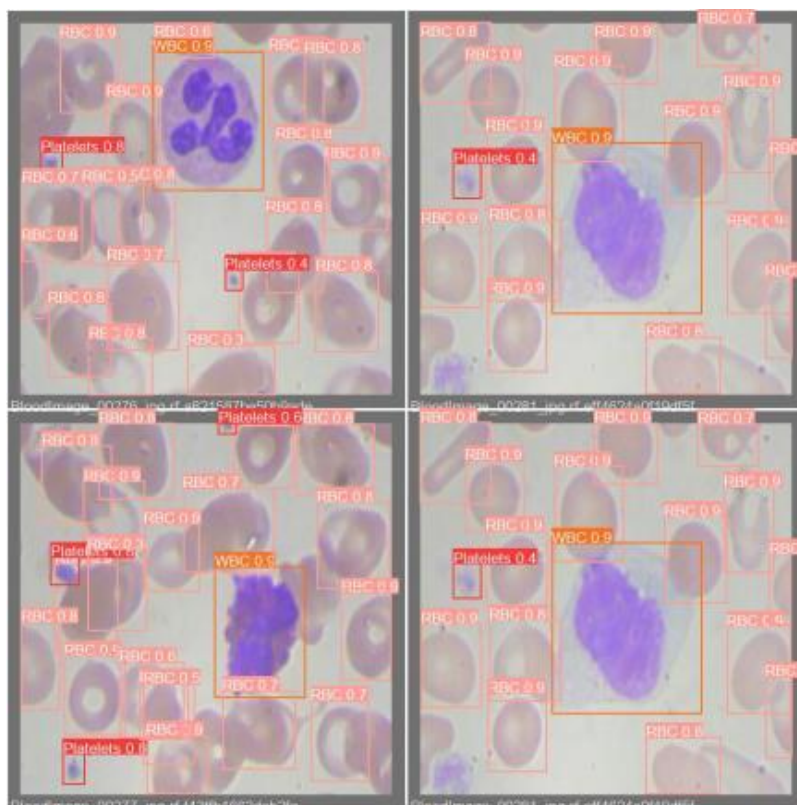


图 14 YOLOv5s 训练 100 轮预测检测计数结果

4.3 YOLOv5s+Transformer Block 训练 50 轮

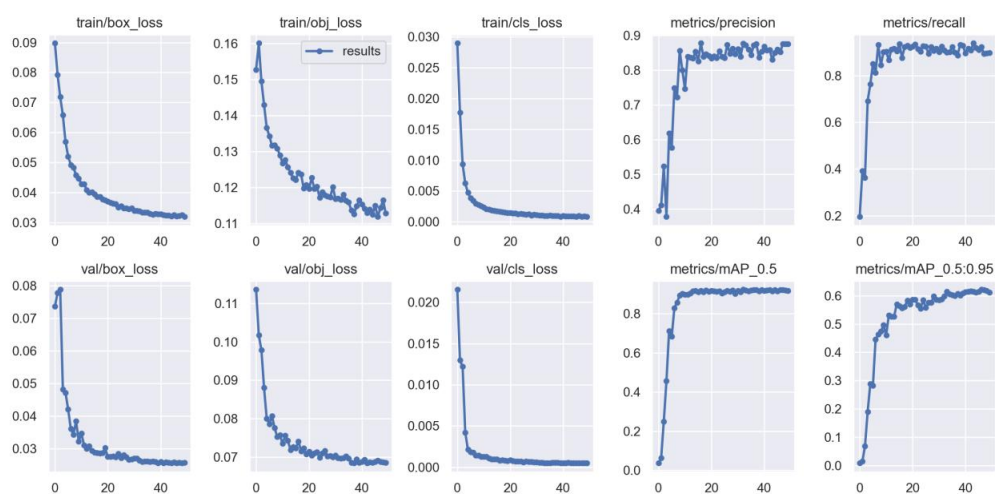


图 15 YOLOv5s+Transformer Block 训练 50 轮过程可视化曲线

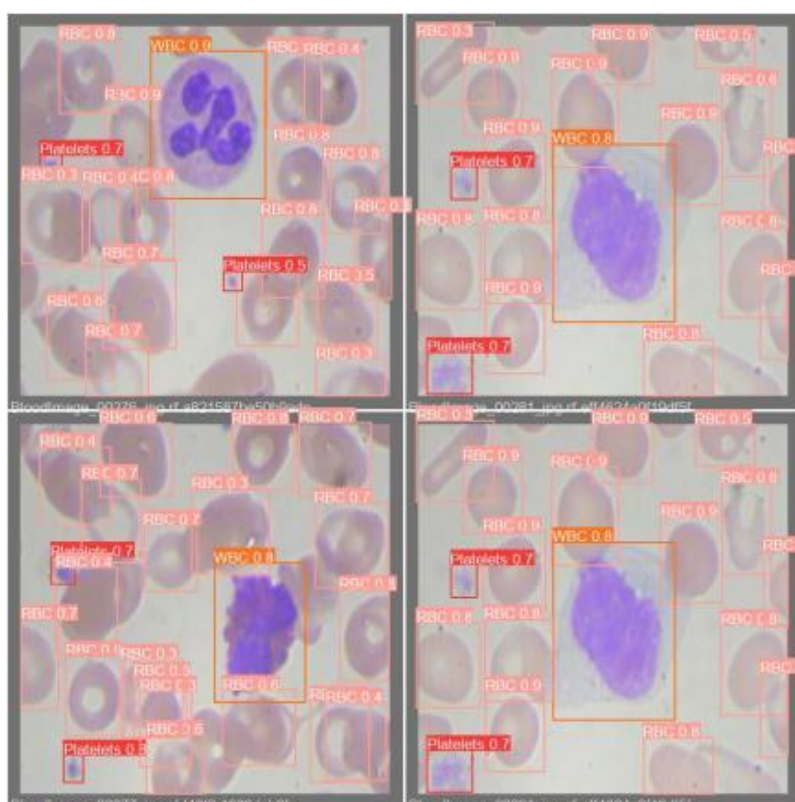


图 16 YOLOv5s+Transformer Block 训练 50 轮预测检测计数结果

4.4 YOLOv5s+Transformer Block 训练 100 轮

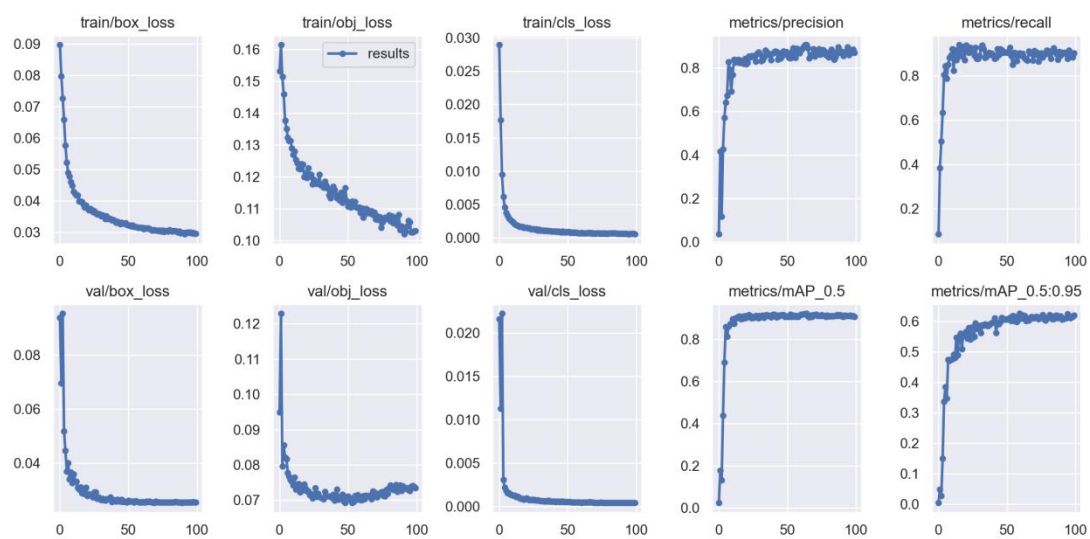


图 17 YOLOv5s+Transformer Block 训练 100 轮过程可视化曲线

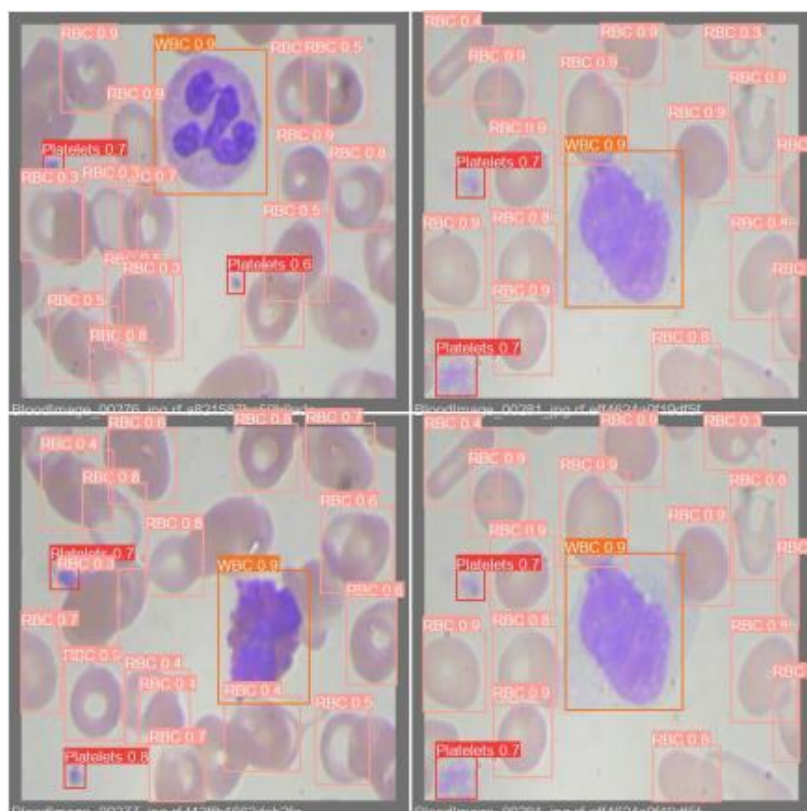


图 18 YOLOv5s+Transformer Block 训练 100 轮预测检测计数结果

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。

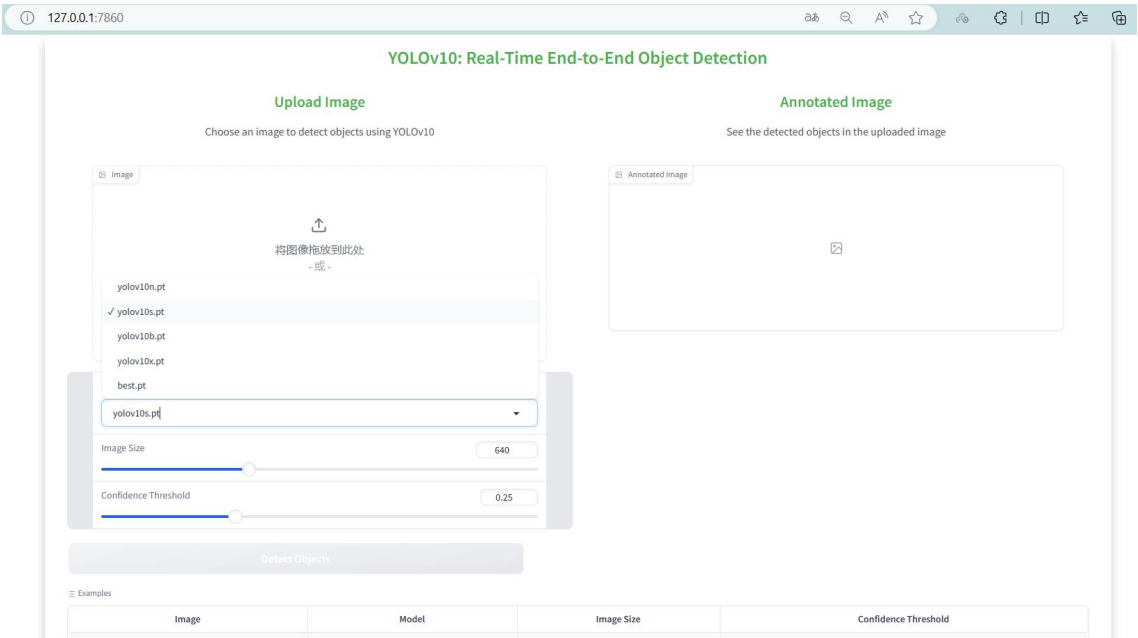


图 19 交互界面

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。

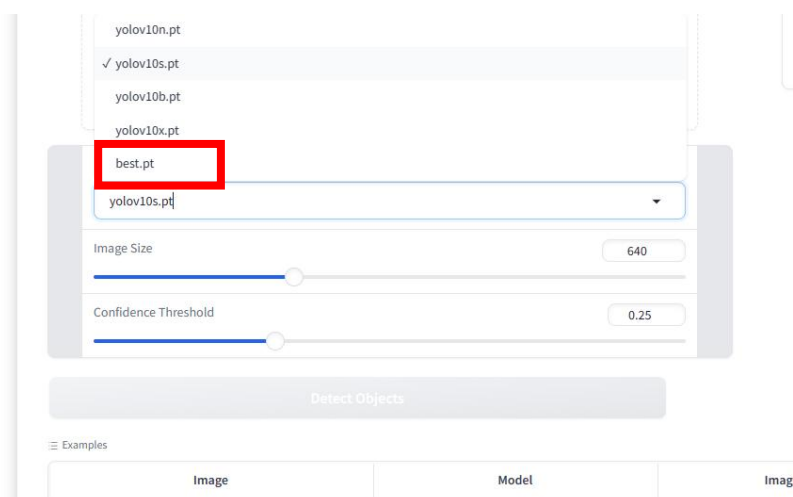


图 20 上传模型

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。

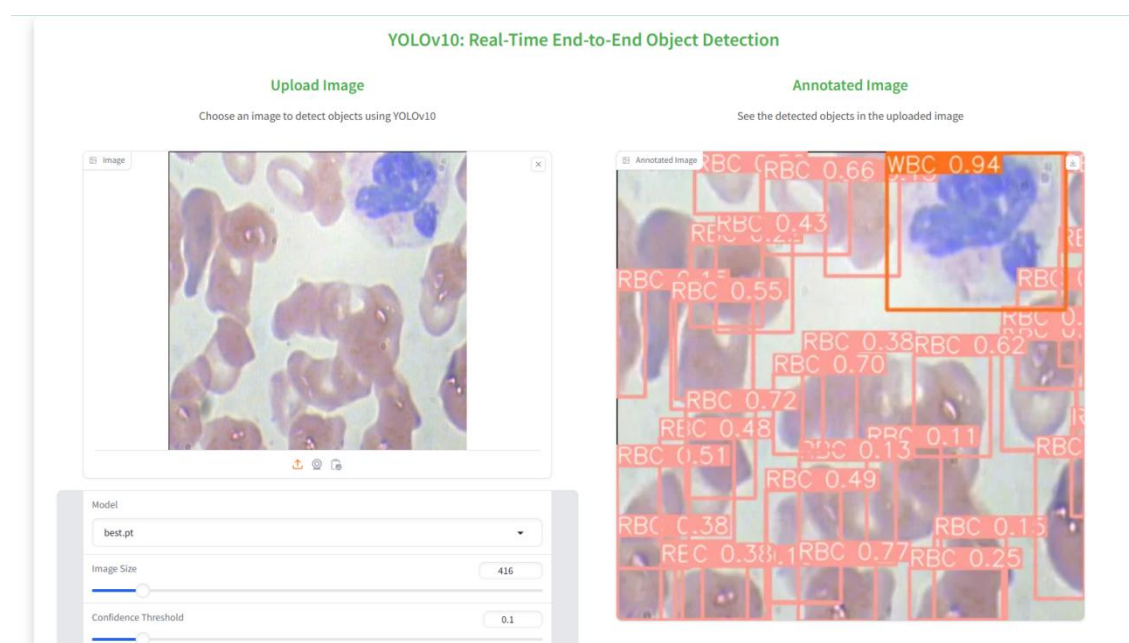


图 21 预测结果

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，我们研究了基于 YOLO 架构的血细胞检测方法，并引入了 YOLOv5 和 YOLOv10 模型以提升检测性能。通过比较传统的 YOLOv5、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。尤其是，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卓越的性能。

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显著优势。实验结果表明，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：

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。

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。

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。

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。