

@ AIGO (九点打卡, 六点打卡)

YOLO模型评估报告

YOLOv8-m 性能评估

报告日期 模型版本
2025-07-16 YOLOv8-m

目录

1	摘要	1
2	评估结果	2
3	分析	7
4	结论	9

1

摘要

本报告对YOLOv8-m模型在人体检测任务上的性能进行评估。评估基于公共数据集，采用精确率-召回率曲线(PR曲线)、接收者操作特征曲线(ROC曲线)等指标，从多个维度分析模型性能。

关键发现

平均精度

YOLOv8-m 模型在测试集上达到了 0.4955 的AP值，显示了良好的检测精度。

区分能力

ROC曲线下面积(AUC)为 0.8548，表明模型具有出色的真假样本区分能力。

适用场景

该模型在复杂场景和需要高精度的应用中表现最佳。

阈值建议

根据不同应用场景需求，建议选择不同的检测阈值：高精度场景选择高精确率阈值点，低漏检场景选择高召回率阈值点，平衡场景选择F1最佳点。

模型性能摘要

YOLOv8-m

0.4955

0.8548

平均精度 (AP)

ROC 曲线下面积 (AUC)

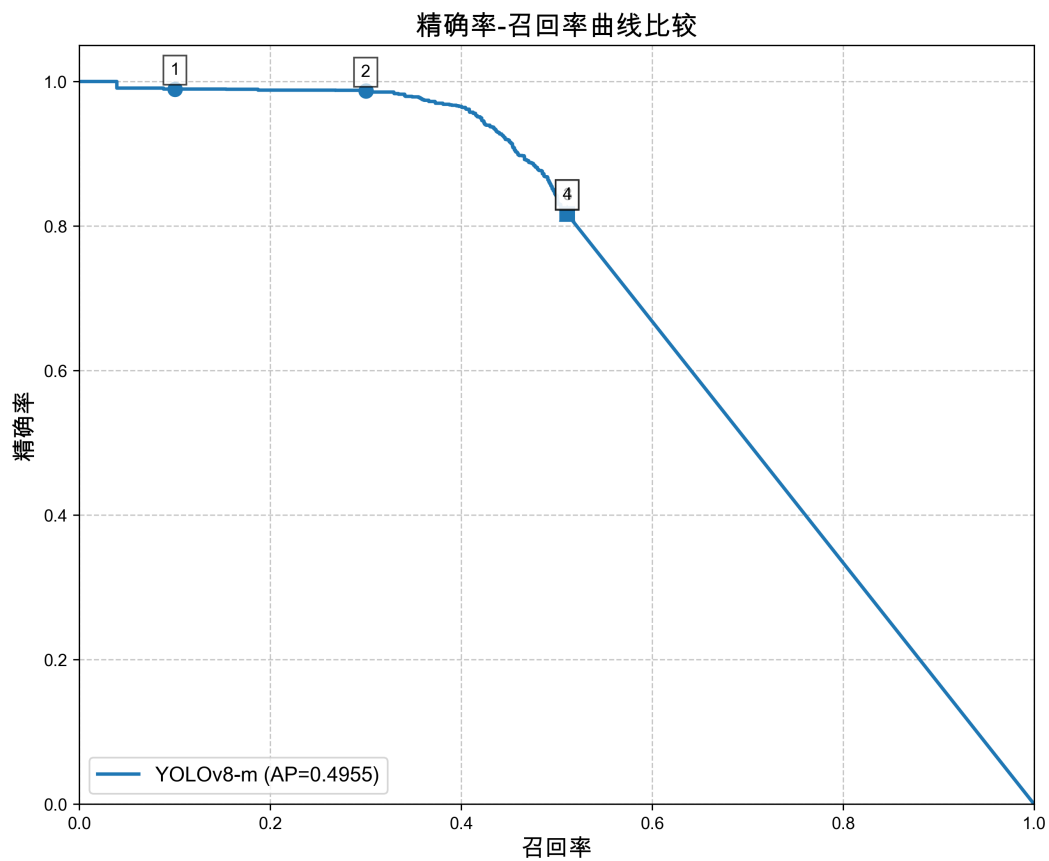
模型性能指标

模型	AP	AUC	评估日期
YOLOv8-m	0.4955	0.8548	2025-07-16

2 评估结果

精确率-召回率曲线

展示不同置信度阈值下模型精确率和召回率之间的权衡。



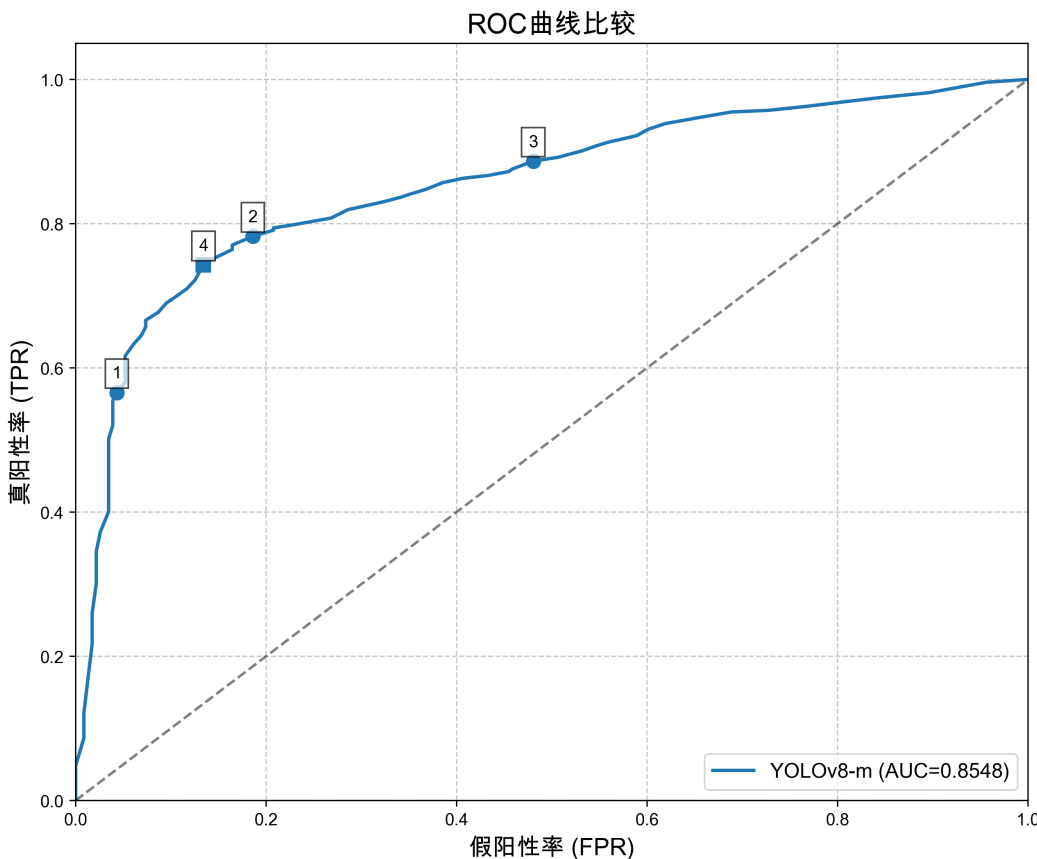
分析要点

曲线越接近右上角，表示模型性能越好。AP值是曲线下面积，代表模型整体精确度。

PR曲线关键点数据

ROC曲线

展示不同检测阈值下真阳性率与假阳性率之间的关系。



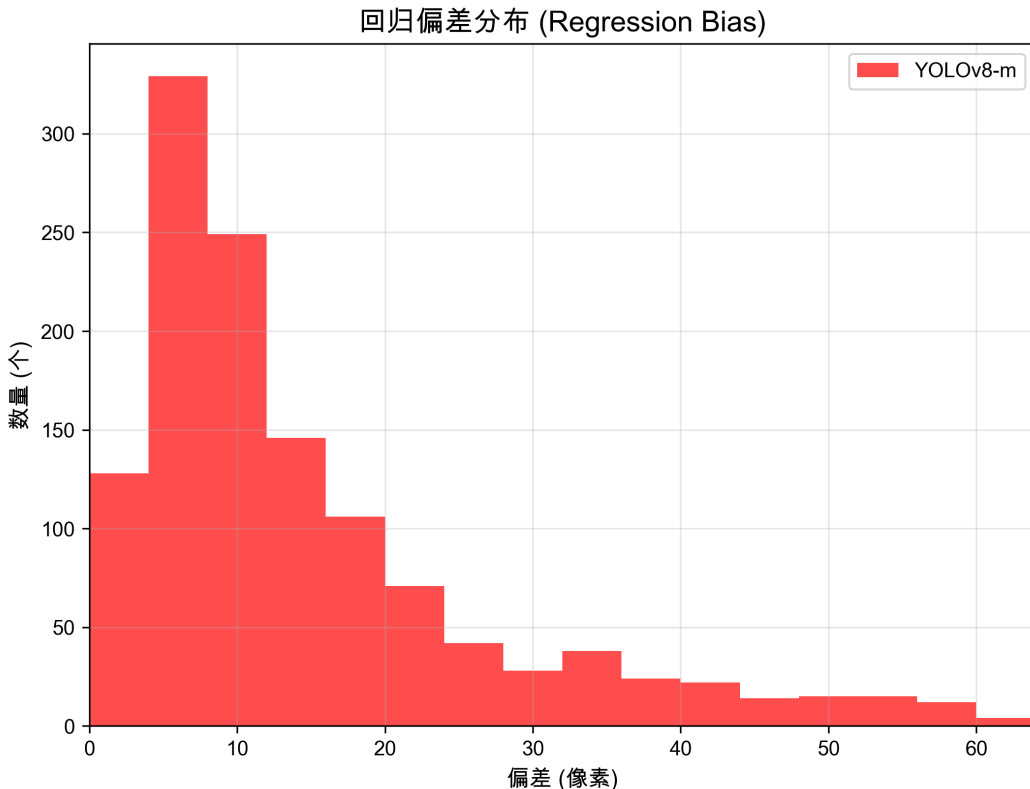
分析要点

曲线越接近左上角，表示模型区分能力越强。AUC值越接近1，表示模型区分正负样本的能力越好。

ROC曲线关键点数据

回归偏差分布

展示预测边界框与真实边界框之间的定位误差分布。

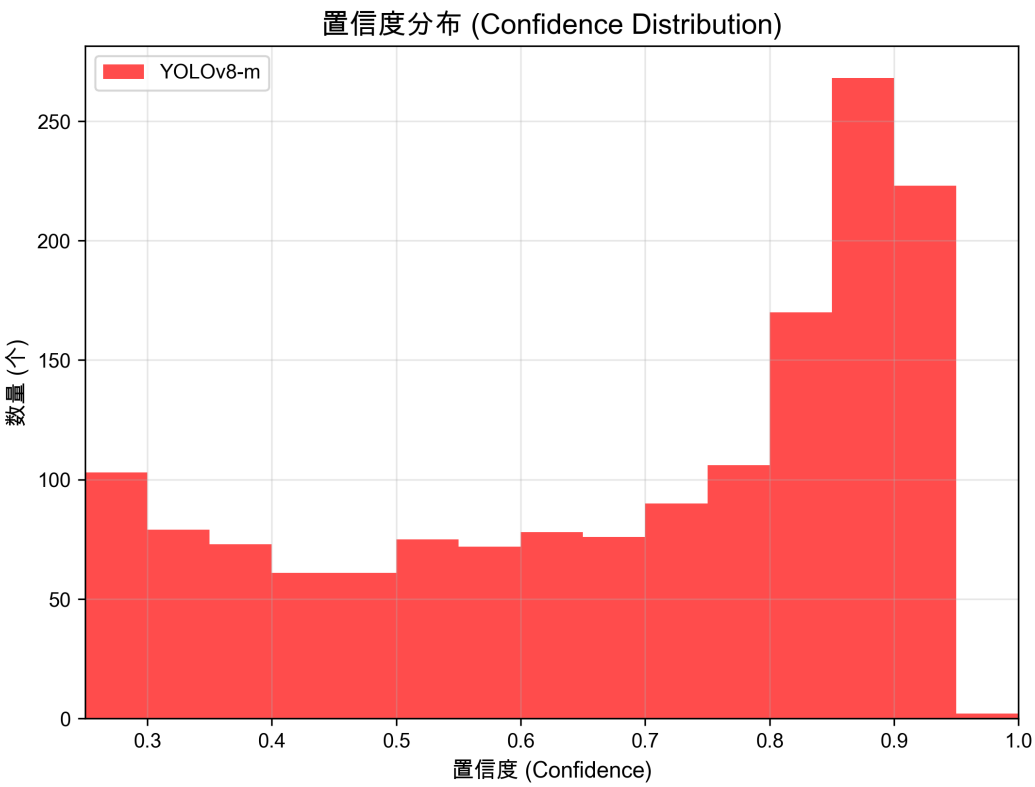


分析要点

分布越集中在左侧，表示模型定位精度越高。较大的偏差可能由遮挡、小目标或极端姿态引起。

置信度分布

展示模型对检测结果的置信度分数分布。



分析要点

理想情况下，分布应与实际准确率一致。高置信度峰值表示模型对其预测结果有较高的确定性。

阈值选择指南

根据不同应用场景的需求，可以选择不同的阈值：

应用场景	优先指标	建议选择	典型阈值特征
安全关键应用	高精确率，低误报	$P > 0.9$ 的点	较高阈值，精确率 > 0.9
医疗筛查应用	高召回率，低漏检	高召回率区间点	较低阈值，召回率 > 0.8
平衡型应用	精确率与召回率平衡	Best F1 或 $P=R$ 点	中等阈值，F1 分数最高
低资源应用	高特异性，极低误报	High Spec 点	较高阈值， $FPR < 0.05$

3 分析

模型分析

YOLOv8-m

优势

- 高检测精度 (AP: 0.4955)
- 对于需要精确检测的应用场景表现出色
- 适合大规模复杂场景的人体检测任务

局限性

- 计算资源需求较高
- 在资源受限设备上推理速度较慢
- 模型体积较大，不适合轻量级部署

应用场景建议

高精度场景

适合要求非常准确检测、误检率低的应用（如安防系统）。选择高精确率阈值点。

平衡性能场景

适合需要在检测率和误报之间取得良好平衡的通用应用。选择F1最佳点。

实时应用场景

适合需要快速处理的实时应用。调整阈值以平衡速度和准确性需求。

4

结论

本评估展示了 YOLOv8-m 模型在人体检测任务上的性能特点。模型在测试集上达到了 0.4955 的AP值和 0.8548 的AUC值，表现良好。

为获得最佳效果，建议根据具体应用需求选择合适的阈值点，考虑因素包括所需精度、计算资源限制，以及最小化误检与最大化检出之间的相对重要性。

AIGO LAB

技术评估报告 · 2025-07-16

模型性能基准测试

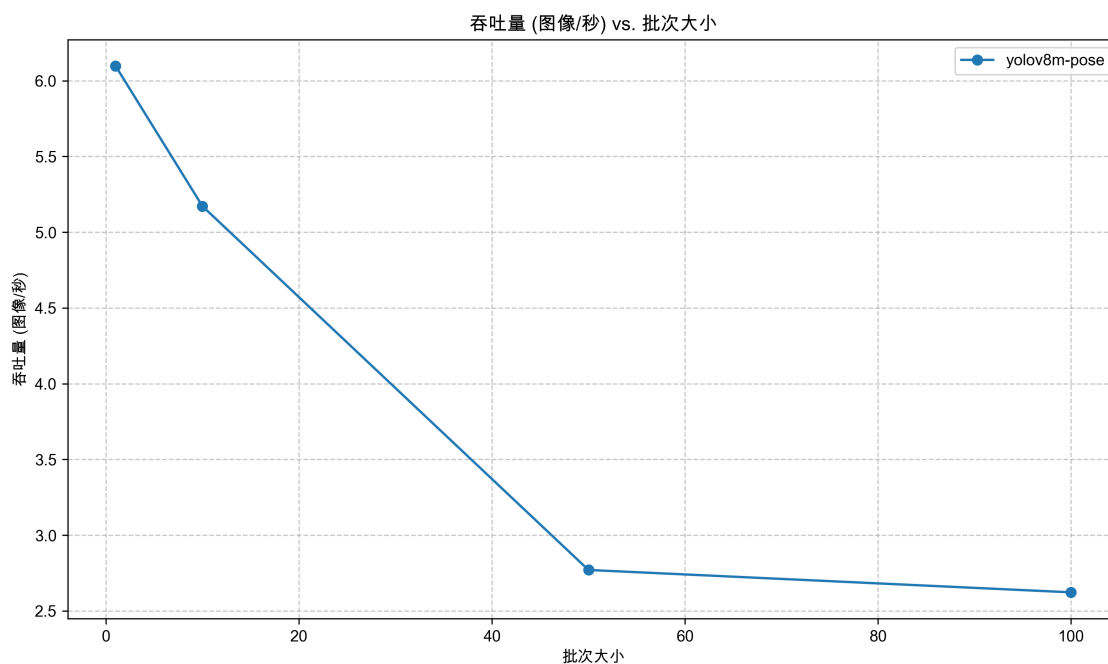
系统信息

操作系统	Darwin 24.5.0 (LeiondeMacBook-Air.local)
CPU	Apple M4 (10 物理核心, 10 逻辑核心)
内存	总计: 16.00 GB, 可用: 6.67 GB (使用率: 58.3%)
计算设备	MPS (Apple Silicon) - Apple Silicon (显存: 共享系统内存)
软件环境	Python 3.13.2, PyTorch 2.7.1, CUDA None

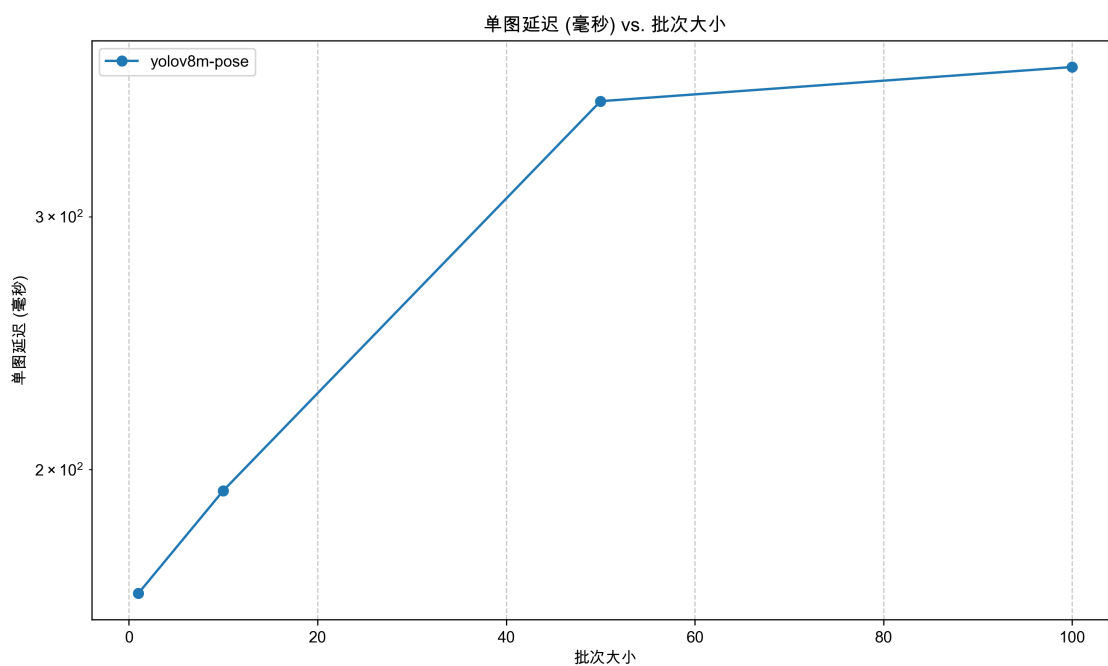
测试结果概览

本测试评估了不同YOLO模型在各批次大小下的性能表现，包括推理延迟、吞吐量和内存使用情况。

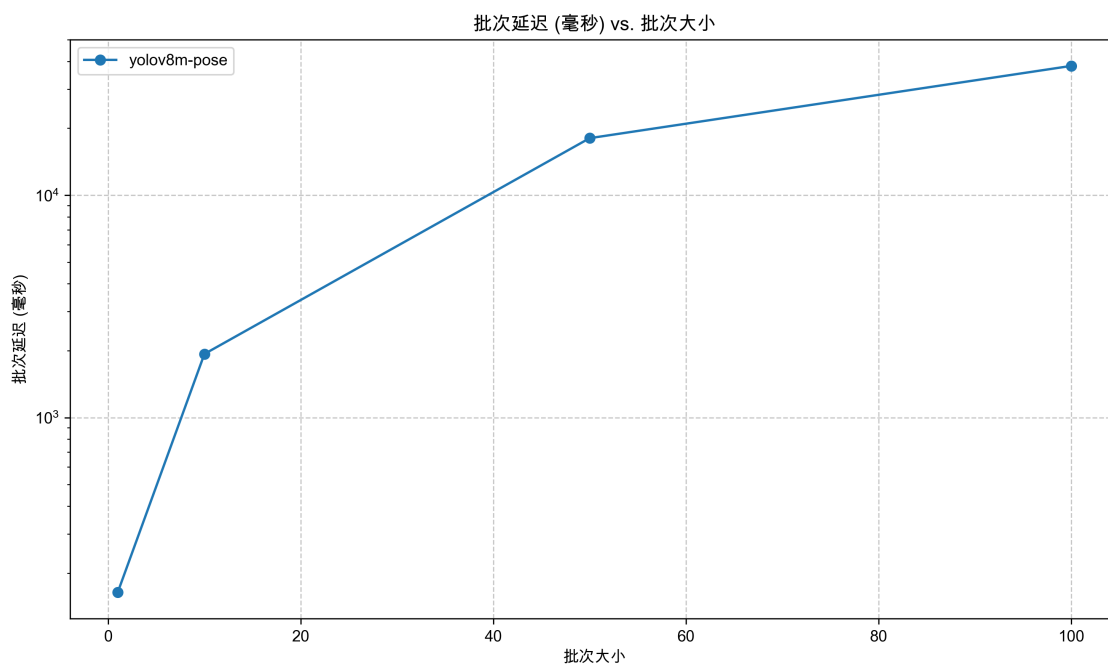
吞吐量比较



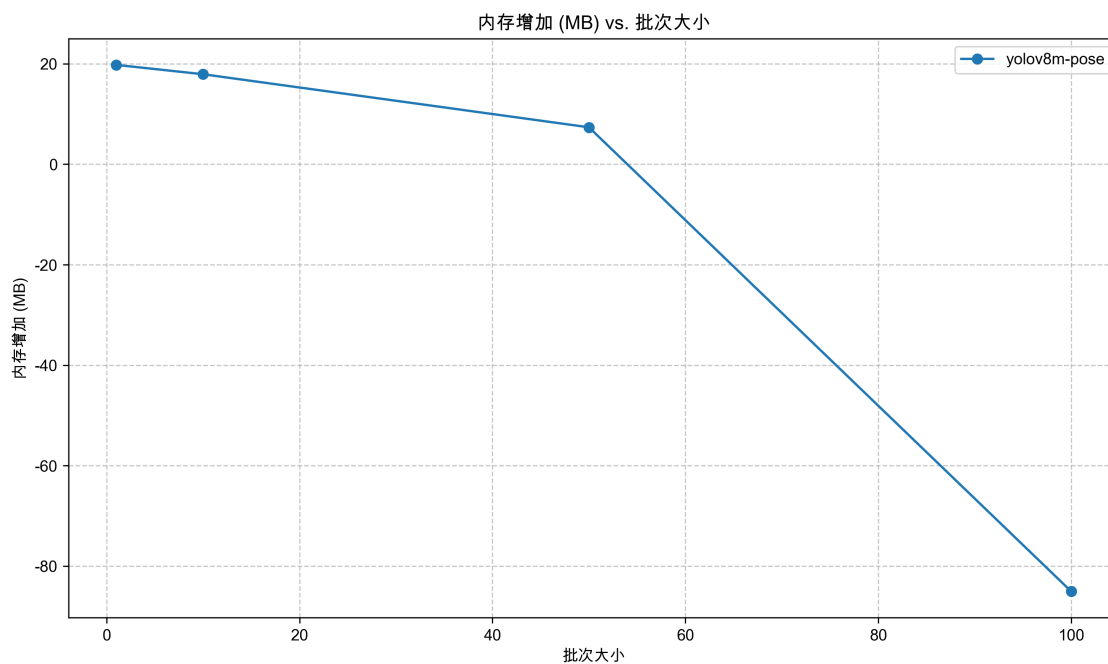
单图延迟比较



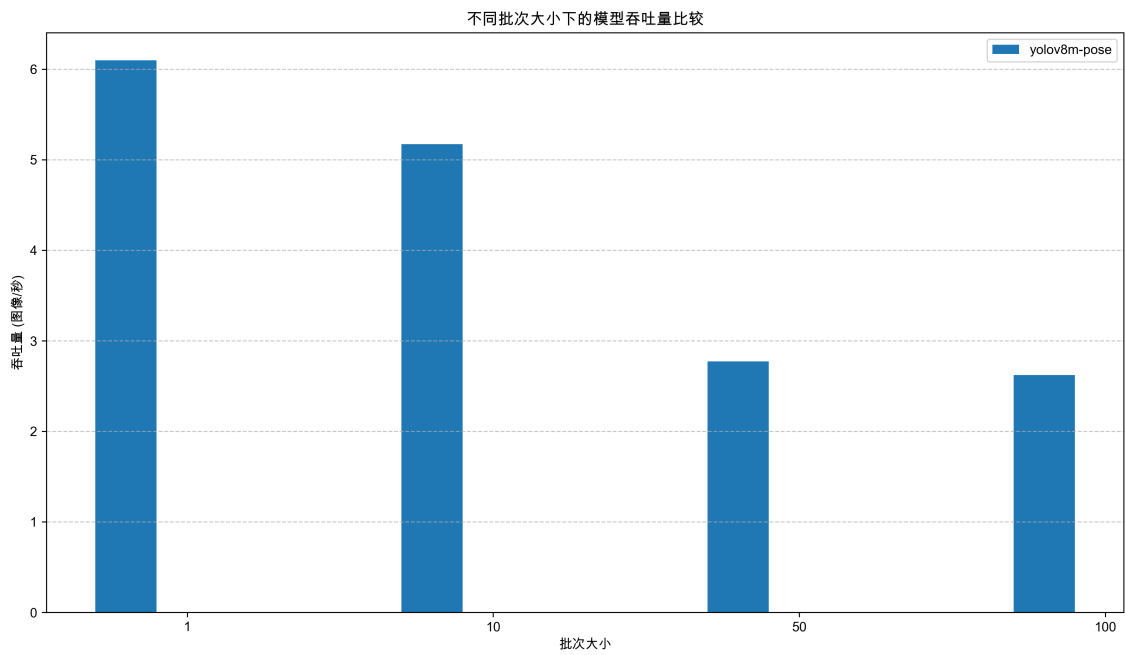
批次延迟比较



内存使用比较



吞吐量柱状图比较



yolo8m-pose 模型性能

6.1

最大吞吐量 (图像/秒)
批次大小: 1

163.99

最低单图延迟 (毫秒)
批次大小: 1

163.99

单批次延迟 (毫秒)
批次大小: 1

-85.0

最大内存增加 (MB)
批次大小: 100

详细性能数据

批次大小	批次延迟 (毫秒)	单图延迟 (毫秒)	吞吐量 (图像/秒)	内存增加 (MB)	CPU使用率 (%)	内存使用率 (%)
1	163.99	163.99	6.10	19.78	29.0	60.5
10	1933.41	193.34	5.17	17.92	43.3	67.8
50	18048.49	360.97	2.77	7.36	67.4	65.9
100	38129.34	381.29	2.62	-85.03	76.1	58.4

结论与建议

根据基准测试结果，可以得出以下结论和建议：

- **延迟敏感场景：**对于需要低延迟的实时应用，应使用较小的批次大小（通常为1）。YOLOv8n-pose模型通常提供最低的单图延迟。
- **吞吐量优先场景：**对于需要高吞吐量的离线处理，应使用较大的批次大小。随着批次大小增加，吞吐量通常会提高，但会达到硬件极限。
- **内存受限场景：**对于内存受限的设备，应考虑使用较小的模型（如YOLOv8n-pose）并限制批次大小。
- **资源均衡：**YOLOv8s-pose模型在性能和资源消耗之间提供良好的平衡，适合中等配置设备。
- **高精度需求：**对于需要高精度的应用，YOLOv8m-pose模型提供最佳性能，但需要更多计算资源。