

基于 YOLOv8 的头盔检测模型设计与实现

陈邵杰, 郭昊, 蔡明珠, 刘政, 黄俊毅, 王晶昊, 杨双菁, 田睿朴, 苗馨月

Abstract—随着人工智能的发展和交通安全需求的增加, 基于深度学习的目标检测技术在行人头盔佩戴检测中发挥着重要作用。在本研究中, 针对行人检测场景采用 YOLOv8l 模型进行头盔检测, 并通过稀疏化和剪枝技术对模型进行优化, 在高精度的同时提升推理效率。利用稀疏化方法对 YOLOv8l 模型的权重进行优化, 增强模型的稀疏性, 减少计算冗余。通过剪枝技术有效去除冗余的网络参数和连接, 大幅减少模型的计算量和参数规模, 优化后的模型在嵌入式和边缘计算设备上运行效率显著提升。实验结果表明, 经过稀疏化和剪枝后的 YOLOv8l 模型参数量减少了约 30%, 推理速度提高了近 40%, 并且在行人头盔检测任务中维持了高达 98% 的准确率。我们还将优化后的模型与 YOLOv3、YOLOv5、YOLOv7 等多个版本进行了对比, 验证了 YOLOv8l 在行人检测精度和速度方面的优势。

Index Terms—交通管理, YOLOv8, 头盔检测, 深度学习, 目标检测

I. 引言

A. 选题背景与意义

Subsection text here.

1) 课题背景: subsection text here.

2) 研究意义: subsection text here.

B. 研究问题分析

Subsection text here.

II. 行人数据图像处理

The conclusion goes here.

A. 实验数据集

Subsection text here.A'

B. 图像预处理

Subsection text here.

C. yolo 标准数据集

Subsection text here.

感谢电子科技大学机器学习课程闫老师与张老师的教学。
2024 年 10 月

D. 数据集标注

Subsection text here.

III. 基于 YOLOv8 的检测系统实现

YOLOv8 是一种精准高效的目标检测解决方案, 系统采用 YOLOv8 作为核心检测算法, 识别电动车及其驾驶员, 并判断是否存在佩戴头盔行为。训练与测试使用的均为 540*355 的 YOLO 标准数据集, 图 3-1 为整个实验流程图:

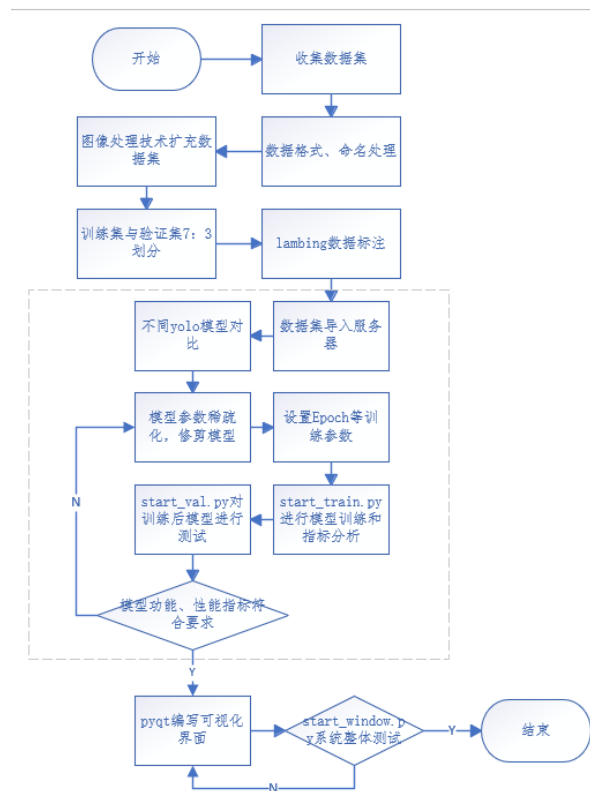


Fig. 3-1 实验流程图

A. 训练环境搭建

对于大量图片数据集与复杂模型, 通过 GPU 进行模型训练大加速, 通过 AutoDL 平台租借云服务器实现

YOLOv8 模型的搭建，训练，测试与部署工作。

本次实验使用的硬件环境为：CPU 16 vCPU Intel(R) Xeon(R) Platinum 8481C, GPU RTX 4090D, 显存大小 24G, 硬盘为 80G。

软件环境：操作系统 ubuntu22.04, 编程语言 python 版本 3.8.5, pytorch 版本 1.10.0, Cuda 版本 11.8。

B. 模型训练

本次实验采用 YOLOv8 模型进行目标检测训练，整个训练过程依托于高性能服务器硬件及成熟的软件环境，确保了模型的高效收敛和稳定性能。

训练流程从数据准备开始，输入图像维度为像素 540*355，首先对数据集进行了精细的标注、清洗和划分，形成训练集、验证集和测试集，为模型的训练和评估提供了坚实的数据基础。并且通过高性能硬件配置使得数据处理与模型计算可以高效并行执行，显著提高了训练效率。

整个训练过程共进行了 100 个 epoch，训练阶段首先加载数据集，并进行数据增强操作，提高模型的泛化能力。接着初始化 YOLOv8 模型，设置包括学习率和批次大小等超参数，以确保训练初期能够有效学习特征。每个 epoch 包含前向传播、损失计算、反向传播和权重更新等步骤。

在服务器上，GPU 的强大算力极大加速了这些运算，在卷积层的特征提取和参数更新过程中尤为显著。每个 epoch 结束后，模型会在验证集上进行评估，以监控性能并避免过拟合的发生。训练日志详细记录了每个 epoch 的损失、精度和召回率等关键指标，并且实时生成了损失曲线和精度曲线，以便于监控训练过程中的动态表现和分析模型的改进方向。

在进行模型参数配置时，为了确保对比实验的一致性和准确性，我们需要核心参数固定不变，以排除对训练效果产生的潜在影响。训练 epoch 为 100 轮，使用常见的随机梯度下降优化器，初始学习率和周期学习率均设置为 0.01，设置 batch 数量为 64。经训练，YOLOv8 模型最终达到了预期的检测效果，能够精准地识别目标物体，表现出良好的鲁棒性和检测性能。通过数据预处理、模型初始化、迭代训练和模型评估的完整闭环与高性能服务器硬件的支持，使得模型在合理时间内完成了高效训练。

C. 模型部署

在本项目中，我们将 YOLOv8 目标检测模型部署到基于 PySide6 开发的图形化界面应用中。模型部署的核心步骤包括开发 UI 界面，将经过训练的 YOLOv8 模型进行格式转换等。

在应用程序中，用户能够通过直观的界面上传图像或视频流，并实时查看检测结果。这一过程中，我们充分利

用了 PySide6 的强大功能，创建了用户友好的操作面板，使用户体验更加流畅。此外，部署过程中还进行了全面的测试，以确保在不同输入场景下的稳定性和可靠性。通过此次部署，YOLOv8 模型得以在实际应用中发挥重要作用，推动了目标检测技术的实用化进程。

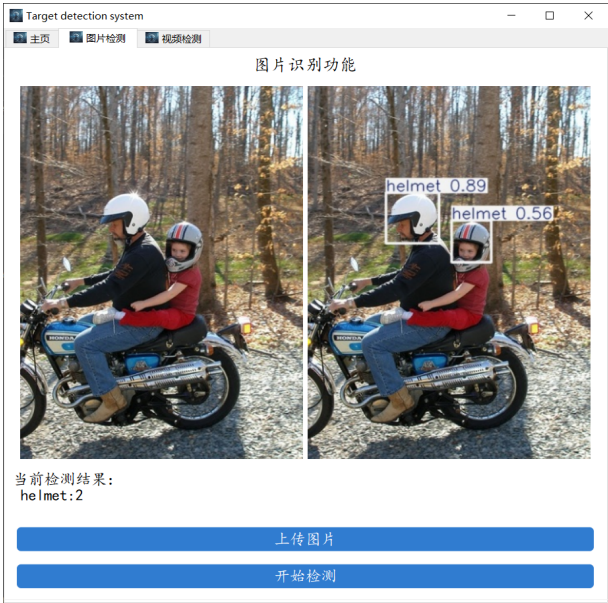


Fig. 3-2 部署界面

IV. 调整与改进

V. 实验结果

The conclusion goes here.

VI. 总结与展望

The conclusion goes here.

致谢

The authors would like to thank...

REFERENCES

[1] Woo S, Park J, Lee J Y, et al. CBAM: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[2] Vaswani A. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017.