

计算机视觉技术作业报告

|  |  |
| --- | --- |
| 题目： | 探究Dysample算子对YOLOv8在道路病害检测中的改进效果 |

院 系： 计算机科学与工程学院

学 号： 2023201237

学生姓名： 　 宋星晴

学 期： 2023-2024年2学期

2024年6月20日

# 代码GITHUB网址：

参考原文github链接：https://github.com/tiny-smart/dysample

我的github链接：https://github.com/sxqxhwyb/cv-task

# 动机：

在看了Dysample[1]这篇文章后，根据它能够显著降低计算工作量、延迟以及本身满足即插即用的特性，应用在YOLOv8模型中可以起到优化的作用。

# 创新点：

将Dysample算子应用到YOLOv8中提升道路病害检测效果。

# 方法：

4.1 YOLOv8算法

YOLOv8由Ultralytics于2023年1月发布。YOLOv8建立在之前版本的基础上，引入了无锚分割头结构，并优化了主干和中间架构，以提高灵活性、效率、准确性和性能。YOLOv8 包含一整套视觉人工智能 (AI) 任务，涵盖检测、分割、姿态估计、跟踪和分类。这种灵活性使用户能够跨各种应用程序和领域利用 YOLOv8 的功能。

YOLOv8的网络结构如图1所示，由三个主要部分组成：backbone、neck、head。其中backbone主要负责从输入图像中提取特征，neck负责融合从backbone网络中提取的多尺度特征，head根据检测目标获取位置、类别、置信度等信息。对backbone和neck处理后的特征进行分析并给出检测结果。YOLOv8提供五个版本：YOLOv8n、YOLOv8s、YOLOv8m、YOLOv8l 和 YOLOv8x，每个版本都有不同的网络深度和特征图宽度。模型的参数和计算量随着模型深度和宽度的增加而增加，以满足不同场景的要求。

|  |
| --- |
|  |
| 图1 YOLOv8网络架构 |

4.2 Dysample算子改进YOLOv8

本文使用Dysample算子[1]对YOLOv8进行了改进，如图2所示。在neck中将原来的上采样方法替换为Dysample算子，使得在保留模型优越识别性能的同时，减少了模型计算复杂度和延迟。

|  |
| --- |
|  |
| 图2 改进后的YOLOv8网络架构 |

# 实验及分析：

5.1 数据集

本文使用来自开源基于人群感知的道路损坏检测挑战赛（CRDDC’2022）[3]的RDD2022数据集[2]进行模型训练。RDD2022数据集包含从日本、印度、捷克共和国、挪威、美国和中国六个国家收集的47,420张道路图像。该数据集按照PASCAL视觉对象类 (VOC)数据集[4]的格式准备，其中XML文件为超过55,000个道路损坏实例提供注释。 如图3所示，数据集包含四种损坏类型：纵向裂纹 (D00)、横向裂纹 (D10)、鳄鱼裂纹 (D20) 和坑洞 (D40)，分别有26,016、11,830、10,617和6544个实例。这些损坏类型的分布表明，纵向裂纹最为普遍，而坑洼则相对较少。

|  |
| --- |
|  |
| 图3 道路损坏类别的示例图像 |

由于RDD2022数据集比较大，因此在实际实验中使用其中部分数据集。如图4所示，该数据集被分为三组：训练集、验证集和测试集。按照7:2:1的比例划分数据集，训练集包含11149张图片，验证集包含3185张图片，测试集包含481张图片。

|  |
| --- |
|  |
| 图4 数据集划分结果 |

5.2 实验设置

实验使用的GPU为NVIDIA RTX A4000，使用了8核CPU，pytorch 框架版本为 2.0.1+cu118，python版本为3.10，cuda版本为11.8。在本文中实验参数设置如下：输入图片的大小设置为640×640，初始化学习率为0.01，使用SGD优化器。在相同实验环境下使用同样的训练参数训练150轮，当模型50轮训练中都没有得到性能提升时，将触发早停机制提前结束训练。具体的实验环境参数设置如表1所示。

表1 训练参数设置

|  |  |
| --- | --- |
| 参数 | 设置 |
| epochs | 150 |
| patience | 50 |
| batch | 32 |
| imgsz | 640 |
| workers | 8 |
| optimizer | SGD |

5.3 实验结果及分析

分析使用Dysample改进后的YOLOv8算法在RDD2022数据集上的性能。如表2所示，改进后的模型相比原模型GFLOPs不变，说明两者的计算复杂度相当。虽然在准确率上，改进后的模型相比原模型降低了0.006。但是，召回率提升了0.018。同样，mAP50和mAP50-90分别提升了0.006和0.002。更有利的一点是，改进后的模型花费更短的训练时间。

表2 模型在RDD验证集上的结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| model | GFLOPs | P | R | mAP50 | mAP50-95 | hours |
| yolov8n | 8.1 | 0.61 | 0.545 | 0.574 | 0.283 | 3.952 |
| yolov8n\_Dysample | 8.1 | 0.604 | **0.563** | **0.58** | **0.285** | 3.493 |

图5和图6是两模型训练结果的截图，从图中可以看出，改进后的模型显著提高了鳄裂和坑洞类别的召回率，分别提升了0.024和0.0025。

|  |
| --- |
|  |
| 图5 yolov8n训练RDD数据集结果 |
|  |
| 图6 yolov8n\_dysample训练RDD数据集结果 |

# 参考文献

1. Liu W, Lu H, Fu H, et al. Learning to Upsample by Learning to Sample[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 6027-6037.
2. Arya D, Maeda H, Ghosh S K, et al. Rdd2022: A multi-national image dataset for automatic road damage detection[J]. arXiv preprint arXiv:2209.08538, 2022.
3. Arya D, Maeda H, Ghosh S K, et al. Crowdsensing-based road damage detection challenge (CRDDC’2022)[C]//2022 IEEE International Conference on Big Data (Big Data). IEEE, 2022: 6378-6386.
4. Everingham M, Eslami S M A, Van Gool L, et al. The pascal visual object classes challenge: A retrospective[J]. International journal of computer vision, 2015, 111: 98-136.