

基于 YOLOv8 的南极动物语义分割实验报告

姓名：翟笑晨

学号：231880394

2025 年 12 月 14 日

1 实验方法

1.1 数据获取与标注

本次实验的数据集来源于 NJU 大神提供的南极动物图像。为了构建高质量的训练集，我们首先在 Roboflow 平台上对部分图像进行了精细的人工标注，采用全监督学习结合迁移学习的策略进行实验。

- 标注过程：使用多边形工具（Polygon Tool）精确勾勒出图像中的动物轮廓（如企鹅、海鸥、海豹等）。
- 数据集划分：在 Roboflow 上将标注好的数据按照一定比例划分为训练集（Train）、验证集（Valid）和测试集（Test）。
- 数据导出：将处理好的数据集以 COCO JSON 格式导出，文件夹命名为 Polar Animal.v9。

1.2 数据预处理

为了适配 YOLOv8 模型的输入格式，我们需要将 COCO 格式转换为 YOLO 格式。

- 格式转换：编写了 `scripts/convert.py` 脚本，解析 COCO JSON 文件，提取其中的类别信息（Categories）、图像信息（Images）和标注信息（Annotations）。
- 坐标归一化：将多边形分割点的像素坐标转换为相对于图像宽高的归一化坐标 (x, y) ，并保存为 YOLO 格式的 `.txt` 文件。
- 配置文件生成：脚本自动生成了 `antarctic.yaml` 配置文件，定义了数据集路径和类别映射关系（如 0: objects, 1: Penguin, 2: Seagull, 3: Seal）。

1.3 模型选择与架构

本次实验选用了 **YOLOv8n-seg** 模型。YOLOv8 (You Only Look Once version 8) 是目前最先进的实时目标检测和分割模型之一。

1.3.1 YOLOv8 模型原理

YOLOv8 是一种基于 Anchor-free 的单阶段目标检测与分割模型。其核心思想是将图像划分为 $S \times S$ 的网格，每个网格负责预测中心点落在该网格内的目标。

- **骨干网络**：使用 CSPDarknet53 提取图像特征。该结构引入了 C2f 模块，通过丰富的梯度流信息提升了特征提取能力。
- **颈部**：使用 PANet 结构进行多尺度特征融合，增强了模型对不同大小目标的检测能力。
- **头部**：采用解耦头，分别预测类别、边界框和分割掩码 (Mask)。
- **分割分支**：YOLOv8-seg 在检测头的基础上增加了一个 Proto 分支用于生成原型掩码，以及一个 Mask Coefficients 分支用于预测每个实例的掩码系数。最终的实例掩码由原型掩码与系数线性组合并通过 Sigmoid 激活函数得到：

$$M = \sigma\left(\sum_{i=1}^k c_i \cdot P_i\right) \quad (1)$$

其中 M 是最终掩码， P_i 是原型掩码， c_i 是对应的系数。

1.3.2 损失函数

YOLOv8 的损失函数由三部分组成：

$$L_{total} = \lambda_{box} L_{box} + \lambda_{cls} L_{cls} + \lambda_{seg} L_{seg} \quad (2)$$

- **分类损失 (L_{cls})**：使用二元交叉熵损失 (BCE Loss) 计算类别概率的误差。
- **边界框损失 (L_{box})**：结合了 CIoU Loss 和 DFL (Distribution Focal Loss)，用于回归边界框的位置。
- **分割损失 (L_{seg})**：计算预测掩码与真实掩码之间的像素级交叉熵损失。

1.4 训练配置

实验在本地环境进行，配置如下：

- **硬件环境**：NVIDIA GeForce RTX 4060 Laptop GPU (8GB 显存)

- 深度学习框架: PyTorch, Ultralytics YOLOv8
- 超参数设置:
 - Epochs: 100 (设置了 Early Stopping, 实际训练 57 轮)
 - Batch Size: 8
 - Image Size: 640
 - Optimizer: AdamW (lr=0.001)
 - Model Weights: yolov8n-seg.pt (预训练权重)

2 实验结果

2.1 训练过程分析

模型训练在第 57 轮时触发了早停, 图 1 详细展示了训练过程中各项指标的变化曲线。我们可以从损失函数和评估指标两个维度来分析模型的训练效果:

1. 损失函数 (Losses): **Box Loss** 持续下降, 表明模型在预测目标位置方面越来越精准; **Seg Loss** 显著降低, 说明模型对动物轮廓的分割能力不断增强; **Cls Loss** 的收敛则表明模型能够准确识别目标类别。

2. 评估指标 (Metrics): **Precision & Recall** 在训练初期波动上升后趋于稳定, 说明模型在减少误检的同时检出了更多真实目标; **mAP50** 在前 30 轮增长迅速并稳定在 0.24 左右, 表明具备了基础检测能力; **mAP50-95** 的稳步提升则反映了模型在高质量分割 (如边缘贴合) 方面的进步。

总体而言, 训练集和验证集的损失均呈现收敛趋势, 且未出现明显的过拟合现象 (验证集损失未反弹), 证明模型训练是有效的。

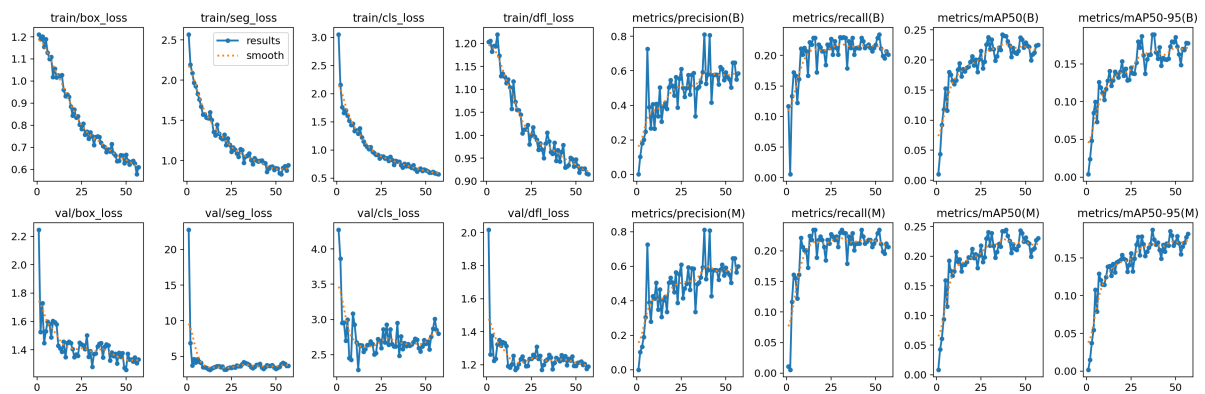


图 1: 训练过程中的损失函数与评估指标曲线

2.2 分割效果展示

为了直观评估模型的性能，我们使用训练好的最佳模型对测试集进行了推理，并将结果拼接展示。



图 2: 测试集上的语义分割预测结果拼接图

如图 2 所示，模型能够较好地定位并分割出图像中的企鹅，海鸥等目标。

- 个体识别：在第一行第四列的图片中，模型准确识别出了岩石前景中的一只企鹅，并给出了较高的置信度（0.94）。
- 群体分割：在第二行第二列和第四行第四列的图片中，面对冰块上密集的企鹅群，模型虽然能够检测出多个个体，但在重叠区域的分割边界上略显模糊。

- 复杂目标检测：在第三行第一列的图片中，远处水中的海豹由于目标过小且背景复杂，模型未能成功检出，这反映了模型在小目标检测上的局限性。

3 评价分析

本次实验使用了多个指标对模型性能进行全面评估，包括精确率（Precision）、召回率（Recall）以及平均精度均值（mAP）。

3.1 评估指标统计

表 1 列出了模型在验证集上的最佳评估结果（Epoch 37）。

表 1: 模型评估指标 (Mask & Box)

Metric	Box (检测)	Mask (分割)
Precision (P)	0.595	0.595
Recall (R)	0.228	0.228
mAP@50	0.243	0.243
mAP@50-95	0.188	0.187

- **精确率 (Precision, P):** 表示预测为正样本的目标中，真正为正样本的比例。0.595 的值意味着模型预测出的目标中约有 60% 是正确的，误检率相对较低。
- **召回率 (Recall, R):** 表示所有真实正样本中，被模型正确预测出来的比例。0.228 的值较低，说明模型漏检了较多的目标，这可能与数据集中存在大量模糊、遮挡或小目标有关。
- **mAP@50:** 当 IoU (Intersection over Union) 阈值设为 0.5 时的平均精度均值。0.243 的得分表明模型在较为宽松的匹配标准下具有一定的检测能力。
- **mAP@50-95:** 在 IoU 阈值从 0.5 到 0.95 步长为 0.05 的范围内计算的平均精度均值。这是一个更严格的指标，0.187 的得分反映了模型在高精度定位和分割边缘贴合度上仍有提升空间。

3.2 混淆矩阵与曲线分析

图 3 展示了混淆矩阵与 Mask PR 曲线，反映了模型在不同类别上的分类准确度与 Mask 的 P/R 曲线。

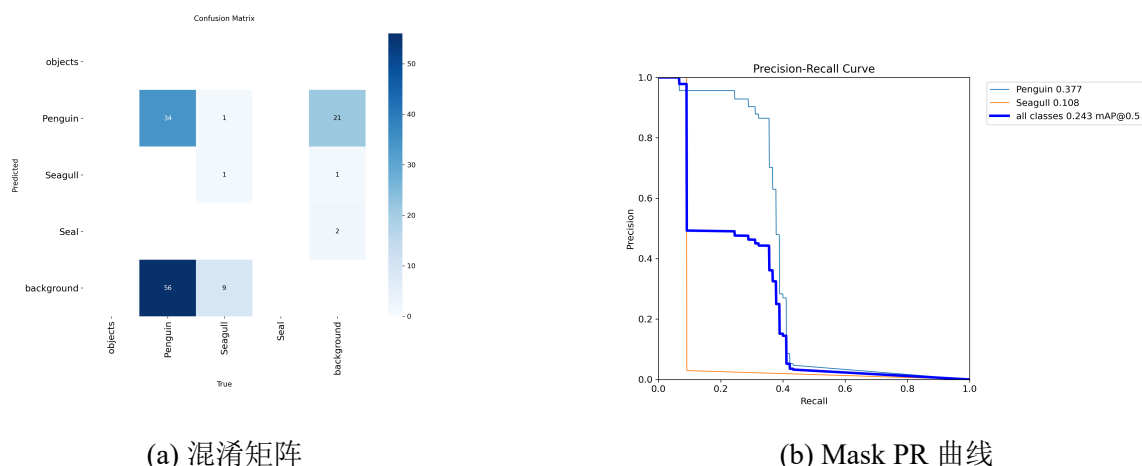


图 3: 模型性能详细分析图

- **混淆矩阵分析:** 图 3a 中, 对角线上的数值代表正确分类的比例。模型在 **Penguin** 类别上的表现最好, 而在 **Seagull** 和 **Seal** 上存在较多的误判 (被预测为背景或混淆)。这主要是因为这两类样本数量极少, 模型未能充分学习其特征。
- **PR 曲线分析:** 图 3b 展示了不同类别的 Precision-Recall 曲线。曲线下面积越大, 模型性能越好。可以看出 **Penguin** 的曲线包围面积最大 (mAP@50=0.377), 而 **Seagull** 和 **Seal** 的曲线则迅速下降, 进一步印证了类别不平衡带来的影响。

4 讨论与总结

4.1 模型表现讨论

1. 优点:

- (a) **实时性强:** YOLOv8n-seg 作为轻量级模型, 参数量小, 推理速度极快 (在 RTX 4060 上仅需约 5ms/张), 完全满足实时监测的需求。
- (b) **主要目标识别准确:** 对于图像中特征明显、数量最多的企鹅目标, 模型展现出了较好的分割效果, 能够准确勾勒出其轮廓。

2. 不足与挑战:

- (a) **相似背景干扰:** 企鹅的背部颜色与南极岩石极为相近, 且训练集中包含岩石背景的负样本较少。这导致模型在复杂背景下容易将岩石误检为企鹅, 或者无法将企鹅从岩石背景中清晰分离。
- (b) **密集目标检测:** 在企鹅数量较多且密集分布的场景中 (如繁殖地), 个体之间存在严重遮挡。模型难以逐个区分个体, 容易出现漏检或将多个个体分割为一个连通区域 (掩码粘连)。

- (c) 小样本与低质量数据：海豹和海鸥的训练样本数量极少，且部分海豹图片质量较差（如只露出头部、大部分身体在水中、光照条件差）。这导致模型对这两类的特征学习不充分，泛化能力差。
- (d) **RLE** 格式解析问题：在数据预处理阶段，部分复杂的 **RLE**（Run-Length Encoding）标注无法被脚本正确解析，导致部分训练数据的标签丢失。这直接减少了有效训练样本的数量，限制了模型的性能上限。

4.2 总结与改进方向

本次作业成功实现了基于 YOLOv8 的南极动物语义分割。我们完成了从数据标注（Roboflow）、格式转换、模型微调到推理测试的全流程，并对结果进行了详细分析。尽管模型在主要类别上表现尚可，但在小样本类别和复杂场景下仍有不足。未来的改进方向包括：

- 优化数据转换脚本：改进 `convert.py`，引入更强大的 **RLE** 解码库（如 `pycoco-tools`），确保所有标注数据都能被正确转换为 YOLO 格式，最大化利用现有数据。
- 数据增强策略：针对小样本类别（海鸥、海豹），在训练时引入 **Mosaic**、**Mixup**、**Copy-Paste** 等数据增强策略，增加其在训练 **batch** 中的出现频率和多样性，缓解类别不平衡问题。
- 引入注意力机制：可以在 YOLOv8 的骨干网络或颈部引入注意力机制（如 **CBAM**、**SE**），增强模型对关键特征（如企鹅与岩石的纹理差异）的关注度，提升在复杂背景下的分割精度。
- 模型微调与集成：尝试使用参数量更大的模型（如最新的 YOLO11），或者采用测试时增强（TTA）和模型集成（Ensemble）技术，以进一步提升模型的泛化能力和鲁棒性。
- 探索弱监督与无监督学习：考虑到人工标注成本高昂，未来可尝试基于图像级标签的弱监督学习，或利用自监督学习挖掘大量未标注数据的特征，降低对精细标注的依赖。

附录：测试集分割结果

根据作业要求，我们对指定的测试集数据进行了预测。下图展示了测试集中每张图片的原始图像、Ground Truth 掩码以及模型的预测结果对比。我们将每张图片的原图、掩码、预测结果拼接在一起，以便于直观地评估模型的分割性能。为了方便直观展示，图片都已经经过后期去除识别框和置信度的标注。

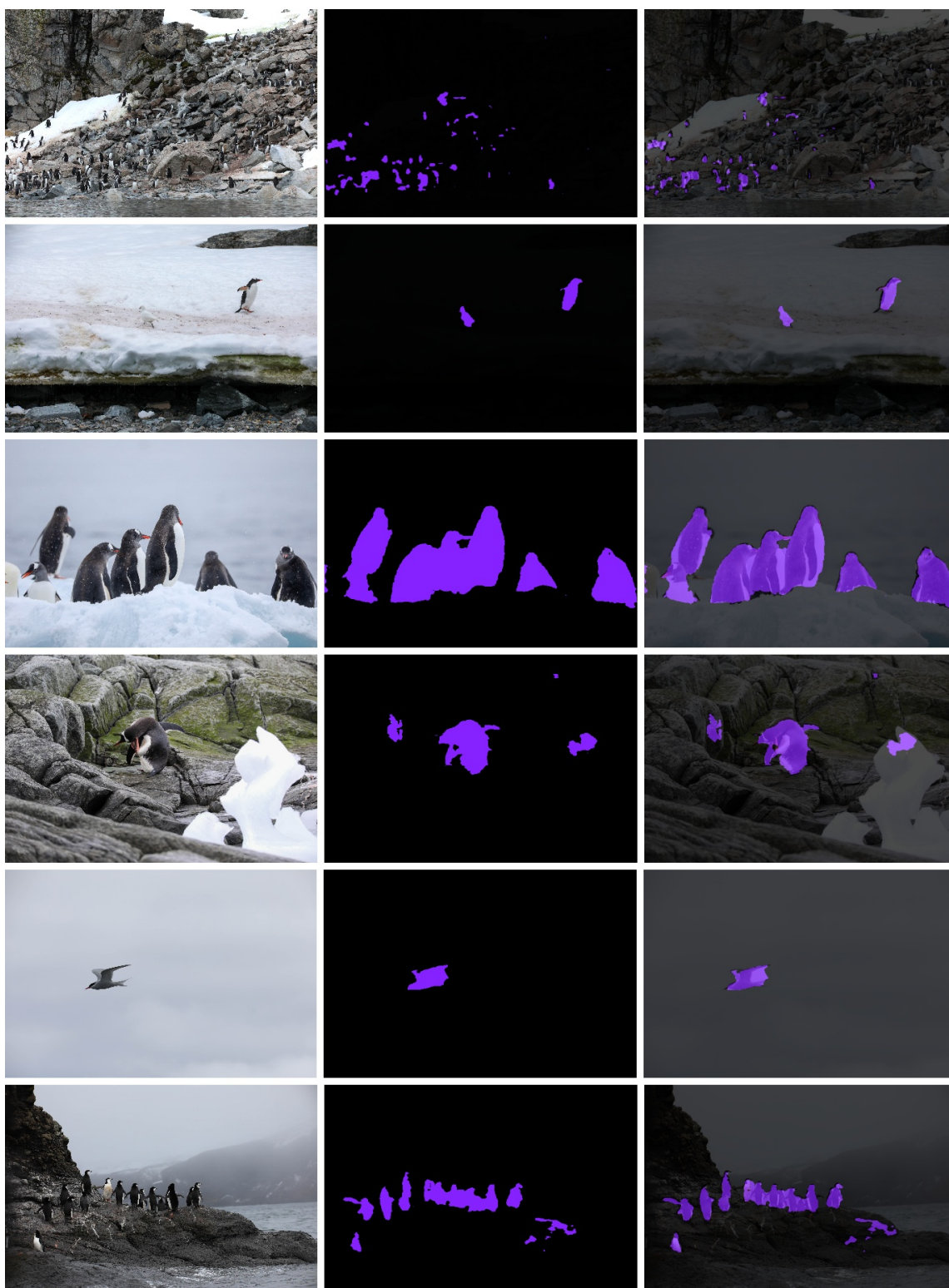


图 4: 测试集预测结果对比 (左: 原图, 中: 掩码, 右: 预测结果)

参考文献

1. Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLO by Ultralytics (Version 8.0.0) [Computer software]. <https://github.com/ultralytics/ultralytics>
2. Lin, T. Y., et al. (2014). Microsoft COCO: Common Objects in Context. In European Conference on Computer Vision.
3. Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

LLMs 协助声明

部分相关代码（包括 `scripts/convert.py`, `scripts/stitch.py` 等）在编写过程中使用了 GitHub Copilot 进行辅助生成、代码调试。同时该报告也使用了 LLM 进行文档润色。所有生成内容均经过人工审核与验证。