OCINet模型优化与国产化适配

上海大学大学生本科生创新项目结项汇报

汇报人：王春阳、戴鑫、范舒舰

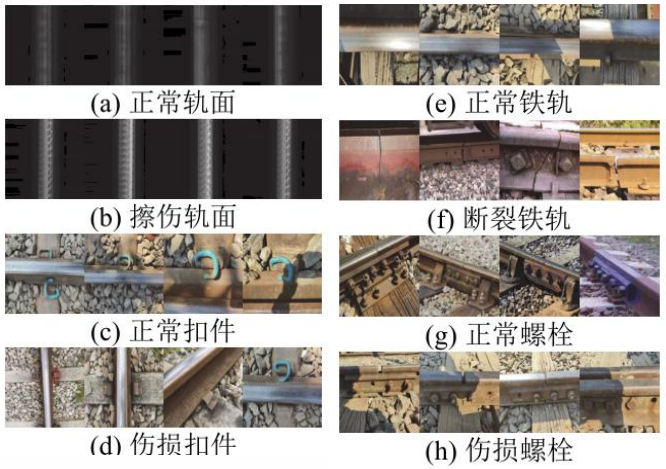
指导教师：李恭杨

2026年4月23日

01 项目概述与研究背景

项目核心主题

基于混合精度训练与昇腾NPU适配的OCINet模型优化研究



背景与研究意义

行业刚需：自动化运维升级
传统人工巡检效率低、主观性强，亟需高精度、高可靠的自动化检测技术替代。

核心痛点：计算与部署成本
高精度深度学习模型伴随巨大计算量与显存占用，导致训练资源紧张，且部署阶段存在高昂的硬件生态依赖。

项目目标：性能优化
通过混合精度训练与NPU深度适配，解决资源瓶颈，实现模型轻量化部署。

遇到的问题：高精度模型与计算资源的矛盾

训练阶段：资源瓶颈

模型结构极其复杂
如OCINet等先进语义分割模型，网络层级深、参数量巨大，基础架构负载重。

GPU 显存严重受限
实验室本地服务器显存不足，直接训练极易触发OOM (显存溢出)错误，导致任务中断。

研究迭代效率低下
受显存制约无法使用大 Batch Size，单轮训练耗时过长，严重拖慢算法研究进度。

部署阶段：成本与生态依赖

特定硬件架构强依赖
主流高精度模型 深度绑定 NVIDIA CUDA 架构，难以在通用计算平台上高效运行。

硬件部署与维护成本高
高性能 GPU 采购成本昂贵，且对运行环境要求高，长期的能耗与维护成本不可忽视。

供应链与生态“卡脖子”风险
过度依赖国外软硬件生态，存在技术断供风险，不符合国产化自主可控的战略发展要求。

02

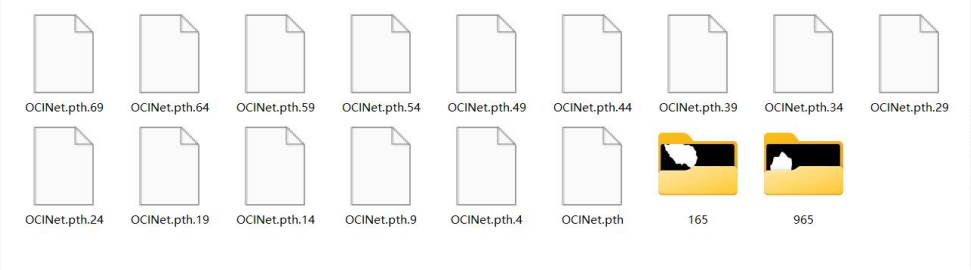
模型概况

OCINet 原理简析

原模型简介

OCINet 全称与简介

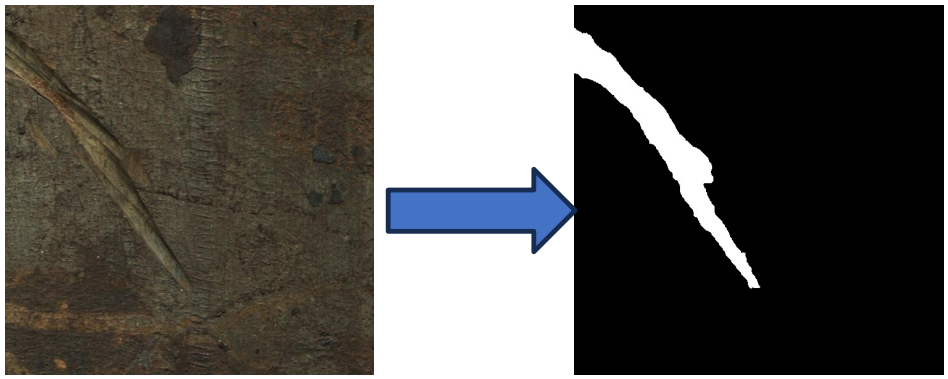
Orientation and Context Information Network，是面向显著目标检测（Saliency Detection的深度学习模型，通过多尺度协作注意力机制 强化不同分辨率特征的交互，提升目标区域的特征表征能力，最终输出高精度的特征图。



模型文件及输出特征图

```
root@autodl-container-8f5c4683c1-a6b85bd0:~/autodl-tmp/OCINet-main# python evaluation_Dice.py
Dataset 965: PA:0.869, mIoU:0.713, mDice:0.819; Dataset 165: PA:0.743, mIoU:0.650, mDice:0.741
```

训练模型d评价结果



应用示例：铁轨裂缝缺陷智能检测

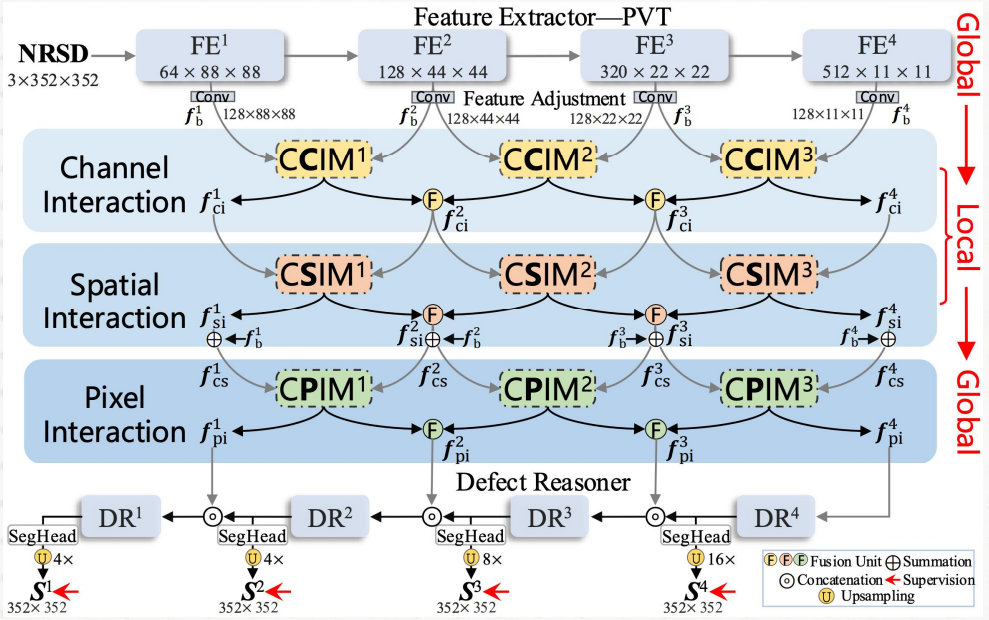
核心网络架构

U 型编码 - 解码网络

以U-Net为基础

核心优势

以 PVTv2 为编码器、以 SalReasoner 为解码器的 U 型网络，在编码器和解码器之间的跳跃连接位置，插入了CCIM（通道）→ CSIM（空间）→ CPIM（自注意力） 三重双向跨尺度协作注意力模块，解决了标准 U-Net 跳跃连接中高低层特征语义鸿沟的问题，大幅提升了目标检测的精度和鲁棒性。



核心网络架构

03 / 核心交互模块 (Interaction Modules)

OCINet设计的精髓，通过三个协同工作的模块进行深度的信息交互与特征增强。

CCIM (协作通道注意力)

将两个尺度的特征图拆分为多个通道分组，每组独立做跨尺度通道注意力，实现细粒度协作。

CSIM (协作空间注意力)

将两个尺度的特征图拆分为多个空间分组，每组独立做跨尺度空间注意力。

CPIM (空间自注意力)

借鉴 Transformer 的 注意力特征，实现跨尺度的长距离空间依赖捕捉。

04 / 解码器 / 缺陷推理器 (Decoder)

接收增强后的特征，通过逐步上采样与融合，生成最终的缺陷分割预测结果。

层级结构设计

采用 U-Net 的对称解码器结构，逐步恢复特征图分辨率，同时融合深层语义与浅层细节。

多尺度预测输出

生成不同分辨率的预测结果，能够同时精准捕捉微小缺陷的细节特征与大面积缺陷的整体轮廓。


03
INNOVATION POINT 01
基于AMP的混合精度训练

问题提出：本地训练的显存瓶颈

核心现象：显存快速耗尽

在8GB显存的本地GPU环境中，使用 FP32 (32位浮点) 精度训练 OCINet 这类复杂模型时，模型参数与中间激活值会迅速填满显存空间，引发资源不足问题。

训练频繁中断

运行中频繁触发 CUDA OOM (Out of Memory) 错误，导致训练流程被迫终止。

训练效率极低

被迫使用极小的 Batch Size (如1或2)，严重降低了梯度下降的效率与模型收敛速度。

算法迭代受阻

漫长的单次训练耗时与不稳定的运行环境，严重拖慢了研究验证与模型调优的节奏。

典型场景：尝试使用更合理的 Batch Size (如 16) 进行训练时，往往在 epoch 初期就因显存不足而失败，无法进行完整的实验验证。

解决方案：引入PyTorch自动混合精度 (AMP)

代码改造方案

放大梯度

```
问题  输出  调试控制台  终端  端口

d:\OCINet-main\OCINet-main\test_OCINet.py:48: UserWarning: `nn.functional.upsample` is deprecated. Use `nn.functional.interpolate` i
nstead.
  res = F.upsample(res, size=gt.shape, mode='bilinear', align_corners=False)
Running time 0.02570
FPS 38.90648
165
Running time 0.02817
FPS 35.49506
PS D:\OCINet-main\OCINet-main> & D:/python3.13/python.exe d:/OCINet-main/OCINet-main/evaluation_Dice.py
Dataset 965: PA:0.854, mIoU:0.718, mDice:0.824; Dataset 165: PA:0.729, mIoU:0.655, mDice:0.742
PS D:\OCINet-main\OCINet-main>
```

AMP 混合精度训练可将模型显存占用降低 40%-50%、训练速度提升 50%，同时保证性能损失极小且收敛性与 FP32 完全一致，既能突破中低端硬件显存限制，又能大幅缩短模型迭代与研发周期。

04

创新点二：面向昇腾NPU的国产化适配

核心适配目标与价值

战略自主可控响应

响应国家关键核心技术自主创新号召，实现算力底座的国产化替代，摆脱外部依赖。

NPU 算力深度挖掘

深度适配昇腾NPU的达芬奇架构特性，充分发挥国产芯片的矩阵运算与并行计算优势。

模型部署基础环境原理验证

Linux 文件链接机制验证

基于 Linux 开发环境完成硬链接与软链接全流程机制验证，明确二者 inode 归属、生命周期、资源复用的底层差异，为适配部署阶段模型权重文件、驱动动态库的多目录复用、路径挂载方案设计提供底层依据，规避部署中模型路径失效、资源调用异常问题。

```
(base) homea@homea-virtual-machine:~/OSsystem/link$ ./link

create a1:
stat  a1  : inode=1835126, nlink=1
1835126 -rw-r--r-- 1 homea homea 6 1月   5 00:26 a1

hard link a2 -> a1:
stat  a1  : inode=1835126, nlink=2
stat  a2  : inode=1835126, nlink=2
1835126 -rw-r--r-- 2 homea homea 6 1月   5 00:26 a1
1835126 -rw-r--r-- 2 homea homea 6 1月   5 00:26 a2

symlink a3 -> a1:
stat  a1  : inode=1835126, nlink=2
lstat a3  : inode=1835131, nlink=1
stat  a3  : inode=1835126, nlink=2
1835126 -rw-r--r-- 2 homea homea 6 1月   5 00:26 a1
1835131 lrwxrwxrwx 1 homea homea 2 1月   5 00:26 a3 -> a1

delete a1:
stat  a2  : inode=1835126, nlink=1
lstat a3  : inode=1835131, nlink=1
a3: No such file or directory
1835126 -rw-r--r-- 1 homea homea 6 1月   5 00:26 a2
1835131 lrwxrwxrwx 1 homea homea 2 1月   5 00:26 a3 -> a1
```

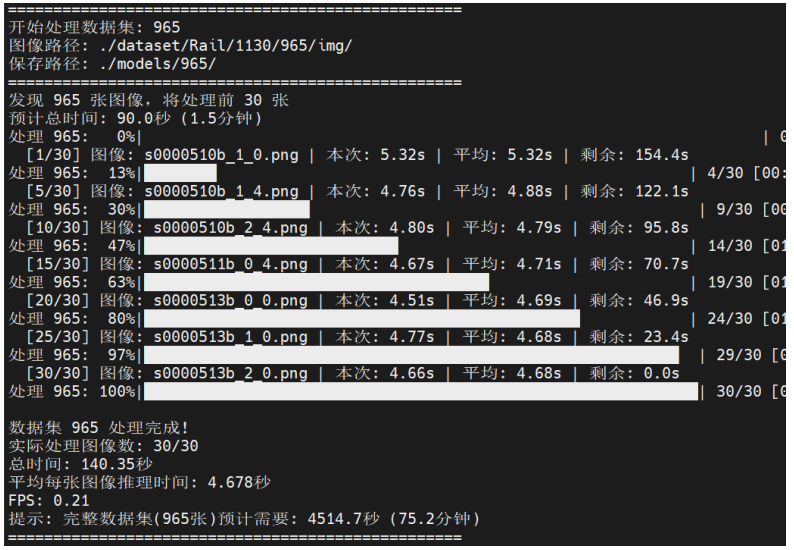
AscendOMRunner 推理封装类

核心设计理念

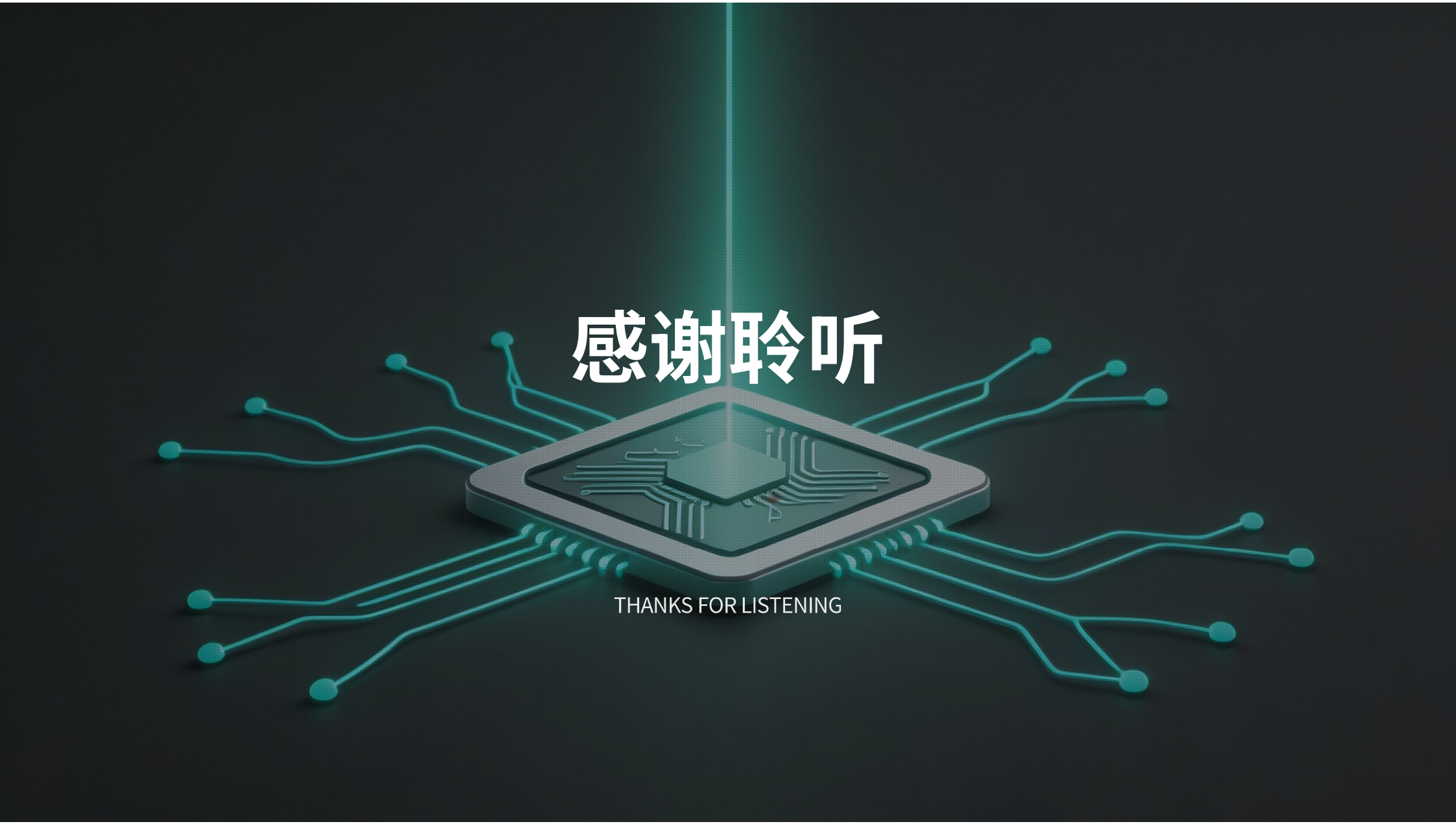
设计 Python 封装类 AscendOMRunner，对昇腾 CANN 的 ACL 底层接口进行深度抽象与封装，实现与原有 PyTorch 推理逻辑的完全解耦。

关键性能优化

- 全流程自动化：自动管理设备初始化、模型加载及资源释放。
- 内存预分配：一次性申请设备内存，避免动态申请的额外开销。
- 类型自动映射：自动处理异构数据类型转换，简化用户操作。
- 推理预热机制：消除首次推理耗时，确保性能统计真实准确。



在 Linux 服务器环境下完成 OCINet 轨道缺陷检测模型批量推理实验，分别对 965 张、165 张 Rail 轨道数据集图像进行全量推理。实验运行稳定，965 张数据集单图平均推理耗时 4.678s；165 张数据集单图平均推理耗时稳定在 4.63s。


感谢聆听
THANKS FOR LISTENING