第一通道技术规格书：全谱系图像篡改检测

模块名称：Channel 1: Omni-Forgery Detection
文件路径：channel_1_forgery/
定位：系统物理层防线，负责基于像素级微观特征的异常检测。

1. 核心技术目标

本模块旨在构建一个统一的“数字取证”检测器，需通过单一模型架构，同时兼容检测以下两类伪造手段。

1.1 覆盖范围

1. **传统数字篡改 (Traditional Manipulation)**
   - **拼接 (Splicing)**：将异源图像区域复制并粘贴到目标图像中。
   - **复制移动 (Copy-Move)**：在同一图像内复制区域以覆盖或增加物体。
2. **AIGC 局部生成 (AI Inpainting)**
   - **AI 消除**：使用智能消除笔 (Magic Eraser) 等 AI 工具擦除物体。
   - **创成式填充**：使用 Generative Fill 生成局部内容。

1.2 检测原理

本模块基于**数字取证 (Digital Forensics)** 原理，捕捉图像成像一致性的破坏：

- **噪声不一致**：AI 生成区域的噪声分布与相机传感器自然噪声不匹配。
- **边缘伪影**：拼接区域存在非自然的边缘突变。

2. 技术选型与模型架构

鉴于开发周期与算力限制，本项目严禁从零训练，必须采用**预训练模型 (Pre-trained Models)** 进行推理。

2.1 首选方案：MVSS-Net++

- **全称**：Multi-View Multi-Scale Supervised Networks
- **核心优势**：该模型包含专门的**噪声提取分支 (Noise Branch)** 和 **边缘提取分支 (Edge Branch)**。噪声分支对 AI Inpainting 留下的平滑痕迹具有极高的敏感度，边缘分支能有效捕捉传统 PS 的拼接痕迹。
- **部署要求**：需在 GitHub 检索 PyTorch 实现版本，并下载官方提供的预训练权重 (pth文件)。

2.2 备选方案：MantraNet

1. **全称**：Minimal Annotation Training
2. **核心优势**：作为通用的异常检测网络，其鲁棒性较强，对各种未知的篡改类型有较好的泛化

能力。若 MVSS-Net++ 部署遇到不可解决的环境问题，可回退至此模型。

首选方案：MVSS-Net++ (Multi-View Multi-Scale Supervised Networks)

- **优势**：包含专门的噪声提取分支 (Noise Branch)，对 AI Inpainting 留下的平滑痕迹具有极高敏感度。
- **部署**：需在 channel_1_forgery/ 目录下部署推理脚本，并加载预训练权重。

备选方案：MantraNet

- **优势**：通用性强，鲁棒性高。

3. 接口定义与目录结构 (Interface & Directory)

本模块代码必须严格放置于 channel_1_forgery/ 目录下，并对外暴露统一接口。

3.1 实际目录结构要求

```
MultiChannel-Reasoning-System/
├── data/
│   ├── images/            # [数据仓] 图片必须存放在这里 (e.g., 001.jpg)
│   └── YuanJing_AI_Data_Standard.xlsx  # [标准表]
├── channel_1_forgery/     # [本模块工作区]
│   ├── weights/           # [模型仓] 存放 .pth 权重文件 (记得git ignore)
│   ├── inference.py       # [主代码] 必须包含 Detect 类
│   └── utils.py           # [工具类] 图像预处理等
└── main_demo.py           # [调用方] 根目录主程序
```

3.2 输入输出规范

维度	参数名	类型	描述
Input	image_path	str	待检测图片的**相对路径**。例如 ./data/images/001_p_graph.jpg。代码需具备处理相对路径的能力。
Output 1	score	float	**篡改置信度** (0.0 - 1.0)。

			> 0.5 判定为 Fake。 该分数直接影响最终的真假判决。
Output 2	message	str	**诊断信息**。例如 "Detected Splicing" 或 "Detected Inpainting"。
Visual	mask	Image	**黑白掩码图** (Mask) 白色 = 篡改区域， 黑色 = 真实区域。

4. Excel 数据字段对代码逻辑的影响

请务必理解 data/YuanJing_AI_Data_Standard.xlsx 中各字段如何影响本模块的开发与测试。

4.1 关键字段解析

字段名称	对 Channel 1 代码逻辑的影响
Image_Path	**直接输入**。 代码读取图片时，必须严格按照此字段填写的路径加载。如果 Excel 里写的是 ./data/images/...，代码就必须从根目录解析，或者自行拼接绝对路径。
Ch1_Tamper_Label	**Ground Truth (标准答案)**。 用于验证模型准确率。代码跑出的 score 如果 > 0.5，则此字段必须为 1；反之应为 0。开发阶段需编写脚本对比这两者的差异。

Tamper_Type	**分类统计 (新增字段)**。 取值：PS / AIGC / None。 代码逻辑本身不读取此字段，但在测试报告中，需分别统计模型对 PS 和 AIGC 样本的检出率，以证明"兼容性"。

4.2 构造数据时的注意事项

- **PS样本**：填入 Excel 时，Ch1_Tamper_Label 设为 1，Tamper_Type 设为 PS。
- **AIGC样本**：填入 Excel 时，Ch1_Tamper_Label 设为 1，Tamper_Type 设为 AIGC。
- **真图**：Ch1_Tamper_Label 设为 0，Tamper_Type 设为 None。

5. 数据构造与验收标准 (SOP)

为验证模型对 **AIGC** 与 **传统篡改** 的兼容性，需构造以下两类数据放入 data/images/ 进行测试。

5.1 数据构造方法 (SOP)

Type A: 传统篡改样本 (10张)

- **工具**：Photoshop。
- **操作**：将一张图中的物体（如车、人）抠出，粘贴到另一张背景图中；或复制图中的云朵/草地覆盖另一区域。
- **要求**：尽量保持边缘肉眼难以察觉，测试模型对微观边缘的敏感度。

Type B: AI 消除/重绘样本 (10张)

- **工具**：手机相册"消除笔"、美图秀秀 AI 消除、或网页版 Magic Eraser。
- **操作**：移除图像中的显著物体（如路边的垃圾桶、电线杆）。
- **要求**：**不要**使用全图生成的图片（如 Midjourney 生成的画），只针对真实照片进行局部修改。

5.2 验收标准

在包含 20 张混合样本（10张 PS + 10张 AI 消除）的测试集中：

1. 模型输出 Score > 0.8 的样本数量应不少于 15 张。
2. 对于 AI 消除的样本，生成的 **Mask (掩码图)** 必须高亮显示被涂抹的区域，证明模型成功捕捉到了噪声不一致性。

6. 开发里程碑

3. **环境搭建**：在 channel_1_forgery 下创建 inference.py，跑通 Mock 逻辑。
4. **模型集成**：下载 MVSS-Net 权重，替换 Mock 逻辑。
5. **兼容性测试**：使用 Excel 表中的混合数据进行测试，确保 PS 和 AIGC 样本均能生成高亮 Mask。