> 说实话我觉得可以前后端一个人做，压两个人以最快速度出训练结果，再来支援。
>
> and我觉得前后端可以并行开展。一方面先mock数据（主要是JS文件），提前协商好参数。另外一方面我们把训练的结果部署在云端，提供API调用就可以避免用户主机差异。
>
> 这个我找了解决方案，避免大家混淆，就先不贴出来了。

- **每个人的任务内容（责任模块）**
- **制作周期（需要多少周）**
- **任务之间的依赖关系（是否可并行）**
- **任务难度排序（从最难到最容易）**

☆ 总体任务拆分逻辑

本项目包含四个主要部分：

1. **数据预处理管线（Data Pipeline）**
2. **分类模型训练（AI Model）**
3. **系统前端 UI（Frontend）**
4. **系统后端服务（Backend API）**

4 个人刚好对应 4 个模块，职责边界清晰、互相依赖少，可并行率高。

🧩 四个人任务分配（最优划分方案）

下面是最合理的划分（含周期 + 并行度 + 难度）。

👨‍🔬 成员 A：数据处理 & 预处理管线负责人（Data Engineer）

✓ 负责内容

- 数据读取（.hdr/.dat/.tif）
- 噪声波段剔除、平滑
- PCA / 手动 / 自动波段选择
- 光谱标准化
- Patch 提取（给 3D CNN 用）
- 数据可视化辅助（光谱曲线、RGB 组合）

✓ 产出物

- `processing/` 下所有 `.py` 文件
- 光谱曲线预览接口
- 波段筛选与标准化函数库

✓ 制作周期

- **2 周**

✓ 是否可并行?

- **完全可并行**（与 B/C/D 同时进行）
- 但 **模型训练前**需要 pipeline 先稳定（第 2 周完成即可）

✓ 难度（4 个人中）

☆☆☆☆（第二难）

理由：
需要理解 HSI 数据结构 + 信号处理方法 + 数学变换（PCA 等）

👨‍🎨 成员 B：AI 模型开发负责人（Model Developer）

✓ 负责内容

- SVM / RF 等传统 ML 模型
- 3D CNN 或 HybridSN 模型搭建
- 训练脚本设计
- 性能评估（OA、Kappa、混淆矩阵）
- 模型 A / 模型 B 对比分析

✓ 产出物

- `models/svm_model.py`
- `models/cnn3d_model.py`
- `models/train_cnn.py`
- 评估指标模块

✓ 制作周期

- **3 周**

理由：深度学习部分需要：
数据 reshape、patch 构造、GPU 调试、训练曲线等。

✓ 是否能并行?

- **前 1 周** 可以与 A 完全并行（用 mock 数据训练）
- **第 2 周起** 需要 A 的预处理管线输出稳定数据
- 后端集成（D）需要 B 的输出模型（第 3 周）

✓ 难度（4 人中）

☆☆☆☆（最难）

👨‍💻 成员 C：前端 UI / 可视化负责人（Frontend Developer）

✓ 负责内容

- 整体页面 UI 结构（导航栏、子页面）
- 数据导入界面
- 假彩色显示（RGB 组合）
- 分类结果图显示（模型 A / B）
- 显示真实标注 overlay
- 性能评估界面（热力图、图表展示）
- 与后端 API 对接

✓ 产出物

- Vue/React 页面：`DataImport.vue`, `Preprocess.vue`, etc.
- 图像显示组件
- 可交互光谱曲线组件

✓ 制作周期

- **3 周**

✓ 是否能并行?

- **完全可以与 A 和 B 并行**
- **第 2 周后** 才需要后端接口（D）支持

✓ 难度（4 人中）

☆☆☆☆（第二难 ≈ A）

前端要处理图像与曲线，有一定复杂度。

👨‍💻 成员 D：后端 API & 系统整合负责人（Backend Engineer）

✓ 负责内容

- Flask/FastAPI 后端搭建
- REST API 设计
  - `/upload_hsi`
  - `/preprocess`
  - `/classify_svm`
  - `/classify_cnn`

    - `/evaluate`
- 前后端数据传输格式
- 模型部署（加载 SVM / CNN 模型）
- 性能调优
- 后端日志、任务队列（可选）

✓ 产出物

- `app.py`
- API 路由文件
- 数据存储/缓存逻辑

✓ 制作周期

- **2 周**

✓ 是否能并行?

- 第 1 周可与所有成员并行 → 做 API 框架
- 第 2 周开始需要 A/B 的功能来完善 API
- 第 3 周主要是 C 整合测试（轻松）

✓ 难度（4 人中）

☆☆☆（第三难）

后端难度中等，主要是打通数据流。

📦 项目前后顺序（关键路径）

下面我给你画一个 **时间线甘特图风格的任务依赖**：

```
第 1 周：
A：数据预处理 —— 开始
B：模型初版（随机数据）—— 开始
C：前端框架 + UI 布局 —— 开始
D：后端框架（API 结构）—— 开始

第 2 周：
A：预处理模块完成（所有波段处理/PCA/标准化）
B：使用 A 的数据训练模型
C：所有界面完成基本原型（可点击）
D：API 初版实现（接收数据＋返回假数据）

第 3 周：
A：提供稳定数据给 B/D
B：CNN 训练完成、输出模型
C：前端与 API 接口联调
D：整合 A+B，API 输出真实分类图

第 4 周：
C：前端优化 & 全流程跑通
D：整体系统测试
```

```
B：性能评估图表生成
A：补充可视化工具（光谱曲线）

第 5 周：
- 系统联调
- bug 修复
- 答辩 PPT 准备
- 视频演示录制
```

📊 四个人难度排序（从难到易）

排名	角色	难度	难在哪里
1	**B：AI 模型开发**	☆☆☆☆	深度学习、Patch 处理、GPU
2	**A：数据预处理**	☆☆☆☆	光谱特征、PCA、数据结构
3	**C：前端 & 可视化**	☆☆☆	图像显示、图谱绘制、对接 API
4	**D：后端整合**	☆☆☆	与模型/前端都要连通

🧩 四人分工最终总结（非常适合直接复制到报告）

```
成员 A（数据工程）：
负责数据读取、预处理、光谱分析模块。
周期：2 周。可并行度高。难度第二。

成员 B（模型工程）：
负责 SVM 与 3D CNN 建模、训练与评估。
周期：3 周。部分依赖 A 的预处理。难度最高。

成员 C（前端开发）：
负责 UI 布局、图像显示、光谱曲线和分类图可视化。
周期：3 周。可与 A/B 并行。难度第二。

成员 D（后端开发）：
负责 Flask/FastAPI API、模型部署与系统整合。
周期：2 周。后半段需 A/B 输出。难度第三。
```