《深度学习与图像识别》实验二 · 作业布置

本机部署，跑通第一棒

2026 年 11 月 2 日（第 7 周周一）布置 · 计 10 分 · 在本组实验一工作目录原地继续 · 提交截止 2026-11-27（周五）16:30，登录课程网站「实验与考核」页在线提交

一、全班统一动作模板（15 组同构，部署对象各组不同）

① 进入实验一已建的本组工作目录（**不换目录、不重建**）→ ② 按本组任务单装环境或编译 → ③ 跑通基准与探针（或本组对应流程）→ ④ **亲手记下第一组基线数字** → ⑤ 终端完整输出当场存档。

二、工具与环境

各组工具链见各组页第 2 栏：VS2022（第 1–5 组编译）、Python 独立虚拟环境（第 6–8、13–15 组）、ffmpeg + PySceneDetect + OpenCV（第 9 组）、conda + 4080（第 10–12 组，时段按第 10 组值守制度预约）、Ollama（第 14 组）。**除需 4080 的组外，全程本机即可完成。**

三、证据三件套（缺一件 = 无证据 = 未完成）

命令全文（你敲的是哪条命令，原样复制） + **完整日志**（终端从开始到结束的完整输出，不许截选） + **产物路径**（跑出来的文件在哪、叫什么）。实验记录里只许出现你亲手跑出来的内容，禁止把 AI 的回答直接粘进去。

四、四条纪律（违反任何一条，视同作业未完成）

- **不许只抄不看**：交付物会被随机抽问，讲不清楚就是没做；
- **基线数字不许抄 AI 给的**：耗时、EPE、指标必须是自己代码在自己机器上跑出来的；
- **降级与失败如实记录**：降级链走到哪级、卡在哪、换了什么，一条一条写——降级不是失败，无记录才是失败；
- **4080 只读、预约、严禁回写**（第 10–12 组）：代码主线只读，时段提前预约，产出与证据只存本组任务夹，严禁回写服务器目录。

五、线上提交（实验二起统一走网站）

登录课程网站 → 实验与考核 → 「提交实验二报告」：**报告 PDF/Word 必传**，附件 zip 选传（可打包日志与产物）；**每组子文件夹 ≤200MB**；以最后一次上传为准（旧版自动删除）；**截止 2026-11-27（周五）16:30，逾期通道自动关闭。**15 组操作指南与完整任务书见课程网站实验二卡片链接。

六、15 个小组任务索引（每人只精读自己组那一页）

组号	组名	本组实验二部署对象 / 交付物	页码
1	六目拼接 · 匹配决策组	在自己电脑上把 360Stitcher 编译出来	2
2	六目拼接 · 特征质量组	部署流程与第 1 组一致；多做一步「占位模型」对照——验证“换模型”这件事本身干净无副作用。	3
3	六目拼接 · 缝合融合组	部署流程与第 1 组一致；把日志里增益补偿（luminance 计时行）与融合（blend 计时行）的计时基线记下来——这是实验三、实验四 A/B 对比的 before。	4
4	六目拼接 · 数据与标签组	用 deploy/simulation 五张基准图把「遥测→三段标签」全流程跑通	5
5	六目拼接 · 测试集与评测组	两条路线都在本机部署并通过探针——路线一 360stitcher.exe（九头探针 PASS 12/12）、路线二 360stitcher_e2e.exe（引擎探针 PASS 5/5）。裁判先当一次被评者	6
6	六目拼接 · 端到端数据组	真数据到位前	7
7	六目拼接 · 端到端训练组	不上 GPU	8
8	六目拼接 · 端到端优化组	跑通引擎模式 benchmark	9
9	水下双目 · 数据组	装通 ffmpeg / PySceneDetect / OpenCV 工具链	10
10	水下双目 · 部署组	按清单在 4080 上独立复现部署（资产包里已有可跑环境——你的价值是独立复现与压测）	11
11	水下双目 · 训练组	让一个 batch 走完全程——数据能进、loss 能算、梯度能传	12
12	水下双目 · 评测组	跑通「推理→视差 float 存盘→可视化→EPE/bad-N 指标」链路	13
13	数据基座 · 数据结构化组	本地部署 Python 解析环境（PyMuPDF / pdfplumber / camelot 三选二装通）	14
14	数据基座 · 智能问答与图检索组	搭通“一句话问数”小 demo：本地小模型 + 演示库 + 三个白名单工具 + SQL 安全检查——军规：模型只提出 SQL	15
15	数据基座 · 虚假内容识别与对抗训练组	把实验一的三条规则扩成可配置的规则引擎	16

第 1 组 · 项目一 · 六目拼接（360Stitcher）

匹配决策组

实验二

2026-11-02 布置

1 实验二的任务（一句话）

在自己电脑上把 360Stitcher 编译出来，看到九个头以 ONNX 形式挂上系统，并用探针确认部署无误。

2 本组工作目录（文件夹）

> 本机 · `stitcher_cpp/`（实验一已从 `4080` 拿全的工程根：`training/ ≈65MB`、`src/`、`tools/`）——不换目录、不重新下载

> 编译入口：`stitcher_cpp/build/360stitcher.sln`（VS2022 打开，配置切 `Release` 后整体生成）

> 运行目录：`build/Release/`（`benchmark` 与探针都必须在此目录下运行，否则找不到 `deploy/simulation` 基准图）

> 命令：`360stitcher.exe --benchmark-stitch=deploy/simulation --provider=onnx` ＋ `python tools/nine_head_probe.py`

实验一拿全的 training/ 整目录这次直接派上用场。三个头目录（group01_policy / group04_mc / group05_geometry）的 train.py 实验三才动，本实验只编译与验证。杀毒软件可能静默隔离 onnx 文件——探针 FAIL 先查 models/ 下文件名是否与头名完全一致。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。我在 VS2022 用 Release 编译 stitcher_cpp/build/360stitcher.sln，成功后在 build/Release 运行 360stitcher.exe --benchmark-stitch=deploy/simulation --provider=onnx，再跑 python tools/nine_head_probe.py。请解读三件事：(1) --provider=onnx 让程序以什么方式加载 models/ 下的九个头文件；(2) 日志里九行 [Provider]（形如 wired: orb=1500）各自确认了什么；(3) nine_head_probe.py 的 12 项检查分别在验什么。每条引用 tools/ 与 src/ 相关代码行号；拿不准就说“存疑”。之后我会贴编译或运行报错原文，请逐条解释原因与最小修复步骤，不许猜。

4 验收标准：满足全部条件才可结束迭代

- ☐ VS2022 Release 编译成功，末段日志已存档（Debug 版会一路报错找不到 dll）
- ☐ benchmark 日志出现九行 [Provider] 加载记录（形如 wired: orb=1500）
- ☐ nine_head_probe.py 输出 PASS (12/12)
- ☐ 三段终端完整输出（编译末段 / benchmark / 探针）缺一不可

5 实验二报告需要写的内容

- 部署过程记录：环境（VS 版本）、编译步骤、遇到的报错与解决
- 三段证据的说明：每段日志对应哪一步、各自说明了什么

6 产生 / 更新文件后，需要更新与上传什么

- 实验二证据（命令全文 + 完整日志 + 产物路径三件套）归入「证据包-实验二」，学期末随总清单提交
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）
- 本实验不训练模型、不改 models/ 下任何文件——探针通过即是交付

第 2 组 · 项目一 · 六目拼接（360Stitcher）

特征质量组

实验二

2026-11-02 布置

1 实验二的任务（一句话）

部署流程与第 1 组一致；多做一步「占位模型」对照——验证“换模型”这件事本身干净无副作用。

2 本组工作目录（文件夹）

本机 · `stitcher_cpp/`（实验一已拿全，`training/` 含 `common/head_specs.py`）

编译入口：`build/360stitcher.sln`（VS2022 · Release）；运行目录：`build/Release/`

加餐工具：`tools/make_placeholder_models.py`（生成九个输出恒定的占位模型）

占位模型是输出恒定的空壳，装上后 wired 值应当与装前完全一致——这是给实验三“真换模型”立的干净对照组。两头都要留证据：只截装完的、没截装前的等于没做对照。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。请解读 stitcher_cpp/tools/make_placeholder_models.py：(1) 它生成的九个占位 onnx 内部结构是什么、输出为什么恒定；(2) 装上占位模型后 benchmark 的 wired 值为什么不该变化；(3) 如果 wired 值变了，说明部署链路哪里可能有问题。每条引用代码行号。另外请解释这个对照实验对实验三交付真模型的意义：怎样用它排除“换模型本身带入的副作用”。

4 验收标准：满足全部条件才可结束迭代

- ☐ 部署三段日志齐备（编译末段 / 九行 [Provider] / 探针 PASS 12/12）
- ☐ make_placeholder_models.py 生成九个占位模型后重跑同一基准
- ☐ 装前、装后两份 wired 值对照记录——完全一致才算通过

5 实验二报告需要写的内容

- 部署过程记录（同第 1 组口径）
- 占位模型对照实验记录：装前/装后 wired 值对照表与结论

6 产生 / 更新文件后，需要更新与上传什么

- 实验二证据三件套归入「证据包-实验二」
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）
- 对照记录存本组工作目录——实验三交付真模型时重做同一对照

第 3 组 · 项目一 · 六目拼接（360Stitcher）

缝合融合组

实验二

2026-11-02 布置

1 实验二的任务（一句话）

部署流程与第 1 组一致；把日志里增益补偿（luminance 计时行）与融合（blend 计时行）的计时基线记下来——这是实验三、实验四 A/B 对比的 before。

2 本组工作目录（文件夹）

> 本机 · `stitcher_cpp/`（实验一已拿全，含 `src/stitch/stitch_initialize.cpp`——实验一解读的增益补偿段就是这次计时对象）

> 编译入口：`build/360stitcher.sln`（VS2022 · Release）；运行目录：`build/Release/`

同一基准至少跑三遍取稳定读数。日志级别没开的话计时行根本不打印——先确认日志配置。后台程序会让计时抖动，关干净多跑几轮。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。360stitcher 的 benchmark 日志里有 luminance 与 blend 两类计时行。请在 src/stitch/ 相关代码中找到它们的打点位置，解读：(1) 每类计时行覆盖的计算范围（从哪一步到哪一步）；(2) 为什么基线要以“多轮稳态”为准而不是首轮；(3) 首轮读数里混进了什么一次性开销。每条引用代码行号，我会在文件里逐条对账。

4 验收标准：满足全部条件才可结束迭代

- ☐ 部署三段日志齐备（编译末段 / 九行 [Provider] / 探针 PASS 12/12）
- ☐ 同一基准至少三遍的计时日志，luminance 与 blend 稳定读数记录成表
- ☐ 基线表注明测试图目录与运行遍数（下学期回看才知道测的是什么）

5 实验二报告需要写的内容

- 部署过程记录（同第 1 组口径）
- 增益补偿与融合的基线计时表——注明此表将作为实验三/四 A/B 的 before

6 产生 / 更新文件后，需要更新与上传什么

- 实验二证据三件套归入「证据包-实验二」
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）
- 基线计时表存本组工作目录（gain_field 头设计文档的素材）

第 4 组 · 项目一 · 六目拼接（360Stitcher）

数据与标签组

实验二
2026-11-02 布置

1 实验二的任务（一句话）

用 deploy/simulation 五张基准图把「遥测→三段标签」全流程跑通，再批量喂 UDIS-D 公开拼接对（至少 50 对），产出公开仿真标签交给第 1–3 组——他们的开工口粮就在这场实验里。

2 本组工作目录（文件夹）

本机 · `stitcher_cpp/tools/`（`check_telemetry.py`、`label_builder.py`）与 `training_data/`（遥测输出目录，如 `simulation_run1`）

命令①：`python scripts/check_telemetry.py training_data/simulation_run1` → 期望 `RESULT: PASS`

命令②：`360stitcher.exe --benchmark-training-telemetry`（部署步骤同第 1 组实验二）

UDIS-D 图对：从 `4080 · campus06 · data/raw_pairs/` 按 `≥50` 对选取子集下载到本机（只下所选，不搬 `≈12GB` 整库）

遥测目录路径写错，校验器查了个空目录也“不报错”——先看它统计了多少帧再下结论。标签生成后必须跑校验再往下传：校验 5 秒钟，省下游一晚上。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。请解读 stitcher_cpp/tools/ 下的遥测与标签工具：(1) check_telemetry.py 校验遥测目录的哪些方面，“RESULT: PASS”到底保证了什么；(2) label_builder.py 从遥测到 match/geometry/policy 三段标签的生成顺序与字段来源；(3) 标签校验查什么、查不过怎么办。每条引用代码行号。之后我贴批量跑 UDIS-D 的命令与统计输出，请帮我核对行数与字段是否齐全。

4 验收标准：满足全部条件才可结束迭代

- □ 遥测体检 RESULT: PASS 日志（含统计帧数）
- □ 五张基准图跑遥测的完整终端日志
- □ 三段标签文件与校验输出（字段齐全、行数可查）
- □ UDIS-D ≥50 对批量产出：match / geometry 各 ≥200 行并过校验——第一份交付物

5 实验二报告需要写的内容

- 全流程记录：体检 → 遥测 → 标签 → 校验各步命令与输出
- 批量标签统计（各段行数）与交给第 1–3 组的交接说明

6 产生 / 更新文件后，需要更新与上传什么

- 公开仿真标签放入本机 training/data/labels/，完成后通知第 1–3 组取用
- 实验二证据三件套归入「证据包-实验二」
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）

第 5 组 · 项目一 · 六目拼接（360Stitcher）

测试集与评测组

实验二
2026-11-02 布置

1 实验二的任务（一句话）

两条路线都在本机部署并通过探针——路线一 360stitcher.exe（九头探针 PASS 12/12）、路线二 360stitcher_e2e.exe（引擎探针 PASS 5/5）。裁判先当一次被评者，以后各组交付，你验收的手感就来自今天。

2 本组工作目录（文件夹）

> 路线一：本机 · `stitcher_cpp/`（`build/360stitcher.sln` · `Release`；运行于 `build/Release/`）

> 路线二：本机 · `e2e_gpu/`（实验一已拿全）`--bin/360stitcher_e2e.exe --benchmark-stitch=bin/simulation --engine=ropstitch`

> 两个探针：`tools/nine_head_probe.py`（`12/12`）与 `training/engine_probe_test.py`（`5/5`）

跑路线二引擎探针时别开着 360stitcher_e2e.exe——文件占用会让“交付还原”失败。两个探针一个 12/12 一个 5/5，别背混——裁判先把自己手里的验收线背熟。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。请对比解读两个探针：stitcher_cpp/tools/nine_head_probe.py 的 12 项检查与 e2e_gpu/training/engine_probe_test.py 的 5 项检查——(1) 各自验什么；(2) 引擎探针为什么会“真实替换模型做实验”、它最后怎么把交付还原；(3) 两条路线的验收线各由哪个探针把守。每条引用代码行号。

4 验收标准：满足全部条件才可结束迭代

- □ 路线一部署日志 + 九行 [Provider] + 探针 PASS (12/12)
- □ 路线二部署日志（[Engine] loaded 与 engine active） + 探针 PASS (5/5)
- □ 四样证据齐全（两段部署日志 + 两个探针完整输出）

5 实验二报告需要写的内容

- 两路线部署记录与差异笔记——给全队当验收手感基准

6 产生 / 更新文件后，需要更新与上传什么

- 实验二证据三件套归入「证据包-实验二」
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）

第 6 组 · 项目一 · 六目拼接（360Stitcher）

端到端数据组

实验二

2026-11-02 布置

1 实验二的任务（一句话）

真数据到位前，用合成对把「生成→训练口径→推理自测」整条流程跑顺——流程里的坑今天踩完，实验三就只剩数据问题。

2 本组工作目录（文件夹）

本机 · `e2e_gpu/training/`（实验一已拿全：`make_demo_data.py`、`infer_test.py`、`requirements-cpu.txt`）

数据装载位：`e2e_gpu/training/datasets/`（产物 `train/demo 32 对`、`val/demo 8 对`）

命令：`pip install -r requirements-cpu.txt → python make_demo_data.py → python infer_test.py`

到 outputs/ 查看 infer_overlay.jpg 叠加可视化，确认对齐合理才算自测通过。加餐：python make_real_dataset.py 生成 8 对真实测试批次（只放 val，评测用、不进训练）。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。请解读 e2e_gpu/training/ 下三个脚本：(1) make_demo_data.py——合成对怎么生成、已知平移真值怎么写进标签、为什么说“端到端也需要标签”；(2) datasets/ 的组织约定（left/right 同名成对 512×512、train/val 划分）；(3) infer_test.py——叠加可视化怎么判断“对齐合理”。每条引用代码行号。

4 验收标准：满足全部条件才可结束迭代

- ☐ requirements-cpu.txt 安装完整日志
- ☐ make_demo_data.py 完整输出（train/demo 32 对 + val/demo 8 对）
- ☐ infer_test.py 输出与 infer_overlay.jpg 叠加可视化截图
- ☐ 加餐（选做）：make_real_dataset.py 8 对真实批次的输出

5 实验二报告需要写的内容

- 流程记录：安装→生成→自测各步命令与输出
- 产物清单（目录树与文件数）与叠加可视化说明

6 产生 / 更新文件后，需要更新与上传什么

- 实验二证据三件套归入「证据包-实验二」
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）
- 数据集目录留档——实验三第 7 组直接吃你的清单开训

第 7 组 · 项目一 · 六目拼接（360Stitcher）

端到端训练组

实验二

2026-11-02 布置

1 实验二的任务（一句话）

不上 GPU，本机 CPU 把「训练→导出→交付→探针」四个环节全走通——冒烟 5 分钟，救你半天 GPU 时段。

2 本组工作目录（文件夹）

> 本机 · `e2e_gpu/training/`（`train.py`、`export_onnx.py`、`deliver_model.bat` 均在 `training/` 根目录；任务夹 `group11_e2e_train`）

> 命令链：`python train.py --epochs 2 --batch 4` → `python export_onnx.py` → 双击 `deliver_model.bat` → `python engine_probe_test.py`

跳过 CPU 冒烟直接上 GPU 是大坑：数据少一列的报错能在 GPU 机上排查半小时。checkpoints/ropstitch.pth → exports/ropstitch.onnx（数值自检误差应远小于 1e-3）→ bin/models/ 见 [OK]。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。请解读 e2e_gpu/training/ 的冒烟链：(1) train.py 的 --epochs/--batch 参数如何影响这次 2 轮小训练，输出 checkpoints/ropstitch.pth 里存了什么；(2) export_onnx.py 的数值自检怎么算误差、为什么要求远小于 1e-3；(3) deliver_model.bat 做了哪几步拷贝与校验。每条引用代码行号；我贴报错原文时请逐条给最小修复。

4 验收标准：满足全部条件才可结束迭代

- ☐ 四段完整输出：CPU 小训练日志 / ONNX 导出与数值自检 / deliver_model.bat [OK] / 引擎探针 PASS (5/5)

5 实验二报告需要写的内容

- 四环节记录与坑清单（哪个环节报什么错、怎么解决）

6 产生 / 更新文件后，需要更新与上传什么

- 实验二证据三件套归入「证据包-实验二」
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）

第 8 组 · 项目一 · 六目拼接（360Stitcher）

端到端优化组

实验二

2026-11-02 布置

1 实验二的任务（一句话）

跑通引擎模式 benchmark，采集逐帧计时，建立你组的第一本「耗时账」。

2 本组工作目录（文件夹）

本机 · `e2e_gpu/`（路线二部署与探针步骤同第 5 组）；任务夹 `training/group12_e2e_optim`

命令：`bin/360stitcher_e2e.exe --benchmark-stitch=bin/simulation --engine=ropstitch`（跑满 ≥30 帧）

交叉验证：第 6 组口径的 `benchmark.py`——两本账对得上才算数

模型加载（init）只发生一次，逐帧账里要单列；把加载算进每帧耗时是第一常见错。后台浏览器、杀毒扫描让计时抖动——关干净多跑几轮取稳定段。基线表必须注明测试图目录与帧数。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。请解读 e2e_gpu/cpp/src/ 引擎通道里 [BenchmarkFrame] 日志的产生代码：(1) 每帧各段耗时（预处理/引擎推理/后处理渲染）在哪里打点、怎么分界；(2) 模型加载耗时为什么单独一行；(3) 均值与 P95 该怎么由逐帧日志算出。每条引用代码行号。

4 验收标准：满足全部条件才可结束迭代

- □ 路线二部署日志（[Engine] loaded 与 engine active）
- □ ≥30 帧的完整 [BenchmarkFrame] 日志
- □ 逐帧计时基线表（均值 + P95，四段分列：加载/预处理/推理/渲染）
- □ 与第 6 组 benchmark.py 的交叉验证记录——两本账对得上

5 实验二报告需要写的内容

- 第一本耗时账：基线表 + 分析笔记（哪段最贵、波动来自哪）

6 产生 / 更新文件后，需要更新与上传什么

- 基线表存本组工作目录（性能预算书的数据底稿）
- 实验二证据三件套归入「证据包-实验二」
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）

第 9 组 · 项目二 · 水下双目自动标记（HYSY123 FPSO）

数据组

实验二
2026-11-02 布置

1 实验二的任务（一句话）

装通 ffmpeg / PySceneDetect / OpenCV 工具链，跑通「场景切割→段内抽帧→近重复去冗→按场景/日期切分」全流程，组装干跑数据集 v0.1 的雏形。

2 本组工作目录（文件夹）

本机 · `tasks/u09_data/`（实验一已建；主线 `code/` 只读）

验证命令：`ffmpeg -version` / `scenedetect version` / `python -c "import cv2"`——三条都有版本号才算装通

切割命令：`scenedetect -i demo.mp4 detect-content list-scenes`

抽出来的重复图没去冗——目录看着热闹全是双胞胎帧，模型等于反复吃一道菜。切分粒度是场景/日期不是帧（同一天同一场景只进一边，防数据泄漏）。每帧记溯源信息（哪段视频、哪一秒、哪个场景），否则评测组没法按场景分桶，返工的是你们自己。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。我要搭视频抽帧流水线。请解读：(1) PySceneDetect 的 detect-content 算法原理与阈值含义——“阈值先抄默认、再用体检数字调”调的是什么；(2) 感知哈希（pHash/dHash）做近重复剔除的原理、汉明距离阈值怎么选；(3) OpenCV 在这条流水线里承担哪几步。我贴安装报错原文时请逐条解释与最小修复；给我一段“场景切割→均匀抽帧→哈希去冗→按场景切分”的骨架代码，每行加中文注释。

4 验收标准：满足全部条件才可结束迭代

- □ 工具链三条版本号输出
- □ 场景切割与抽帧日志：抽了多少、去掉多少、留了多少
- □ 训练/验证划分清单与数据卡（每帧带溯源信息）

5 实验二报告需要写的内容

- 工具链安装记录
- 数据集 v0.1 统计：帧数、去冗比例、划分口径

6 产生 / 更新文件后，需要更新与上传什么

- 数据集 v0.1 与数据卡存 tasks/u09_data/（实验四联调对账的底稿）
- 实验二证据三件套归入「证据包-实验二」
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）

第 10 组 · 项目二 · 水下双目自动标记（HYSY123 FPSO）

部署组

实验二
2026-11-02 布置

1 实验二的任务（一句话）

按清单在 4080 上独立复现部署（资产包里已有可跑环境——你的价值是独立复现与压测），跑通 run_demo.py；UWStereo 50 对零样本冒烟，EPE/耗时/显存三项都记；受阻按降级链处置并记录。

2 本组工作目录（文件夹）

`4080 · campus13 · tasks/u10_deploy/`（本机建同名任务夹；主线 `code/` 只读；时段按本组值守制度预约）

验证命令：`python run_demo.py`（看到视差图输出才算部署成功）

降级链：`vitl → vits → IGEV++`（每级记录“卡在哪、换了什么、三项数字多少”）

flash-attn 装不上十有八九是版本与 CUDA 不匹配，别死磕一整晚——降级不是失败，无记录才是失败。权重文件没下全，报错长得像“模型坏了”——先校验文件完整性再怀疑人生；nvidia-smi 先看驱动支持的 CUDA 上限。显存爆就降分辨率重试，同样记录。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。我在 4080 上复现 FoundationStereo 部署。请解读资产包环境清单：(1) 各关键条目（torch/torchvision/flash-attn/CUDA 版本）的作用与相互约束；(2) 降级链 vitl→vits→IGEV++ 各级在精度/显存/速度上的预期差异；(3) run_demo.py 的输入输出与成功标志。我贴 nvidia-smi 输出与报错原文时，请先判断卡在驱动、CUDA、依赖还是权重哪一层，再给最小修复，不许猜。

4 验收标准：满足全部条件才可结束迭代

- □ conda list 全量存档
- □ run_demo.py 成功日志（视差图输出）
- □ UWStereo 50 对零样本三项数字表（EPE / 单对耗时 / 显存峰值）
- □ 降级链处置记录（如有）：每级卡在哪、换了什么、数字多少

5 实验二报告需要写的内容

- 部署记录：环境、权重校验、遇到的坑
- 50 对三项数字表
- 降级记录（如有）

6 产生 / 更新文件后，需要更新与上传什么

- 三项数字表与降级记录存 tasks/u10_deploy/（全队排班与后续评测的依据）
- 实验二证据三件套归入「证据包-实验二」
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）

第 11 组 · 项目二 · 水下双目自动标记（HYSY123 FPSO）

训练组

实验二
2026-11-02 布置

1 实验二的任务（一句话）

让一个 batch 走完全程——数据能进、loss 能算、梯度能传，为实验三扫清路障。

2 本组工作目录（文件夹）

`4080 · campus14 · tasks/u11_train/`（在第 10 组环境里激活 conda；改超参只改自己目录下的配置副本，主线 code/ 只读）

四步冒烟：取一个 batch 打印形状 → 前向一步记初始 loss → 反向一步 optimizer.step() → nvidia-smi 显存快照

数据路径大小写或后缀不对，dataloader 静默空转——“在训了”其实是空转，先确认日志里读到了多少对图。数据里有坏样本（全黑图、尺寸异常）会导致 loss 为 NaN——找第 9 组要体检报告提前排查。忘了 model.train()，跑通的是推理不是训练。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。我在第 10 组部署的环境里做训练冒烟。请解读资产包的 dataloader 与训练循环代码：(1) 一个 batch 从文件到张量经过哪几步（读取/裁剪/归一化/堆叠）；(2) 初始 loss 怎么算、量级多少算正常；(3) 怎么确认“梯度真的传了”——loss.backward() 与 optimizer.step() 前后该看什么。每条引用代码行号；我贴形状打印或 NaN 报错时请帮我定位到具体步骤。

4 验收标准：满足全部条件才可结束迭代

- □ batch 形状打印（左目/右目/视差真值的 shape 与数值范围）
- □ 前向一步：输出视差形状与初始 loss 数字记进日志
- □ 反向一步：优化器 step() 成功、loss 可反传
- □ nvidia-smi 显存快照（标注时刻）

5 实验二报告需要写的内容

- 冒烟记录四件：形状/初始 loss/反向成功/显存
- 遇到的坑与解决

6 产生 / 更新文件后，需要更新与上传什么

- 冒烟日志存 tasks/u11_train/（实验三开训前的路障清单）
- 实验二证据三件套归入「证据包-实验二」
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）

第 12 组 · 项目二 · 水下双目自动标记（HYSY123 FPSO）

评测组

实验二
2026-11-02 布置

1 实验二的任务（一句话）

跑通「推理→视差 float 存盘→可视化→EPE/bad-N 指标」链路，三步串成一键脚本，跑 10 对批量版。

2 本组工作目录（文件夹）

> `4080 · campus16 · tasks/u12_eval/`（第 10 组环境做推理：指标计算与画图本机 CPU 即可）

> 对象：`UWStereo 1` 对（打通）→ `10` 对（批量）；脚本放 `tasks/u12_eval/`

视差图存成 uint8 被截断——大于 255 的视差全变 255，指标全错，存 float 才稳。左右目路径接反——视差符号全反，EPE 大得离谱反而是提醒。可视化用线性配色：非线性配色会让小误差看着触目惊心，干扰人工归因。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。请解读资产包评测脚本的五个指标：EPE、bad-1、bad-2、bad-4、D1——(1) 各自的数学定义与单位；(2) 各衡量什么、什么量级算好；(3) 为什么分桶（按浑浊度/距离）后总和会撒谎。请对照评测代码逐条引用行号；再给我“推理→float 存盘→算分”一键脚本的骨架，每行加中文注释。

4 验收标准：满足全部条件才可结束迭代

- □ 1 对推理：视差图 float 存盘 + 可视化色图（肉眼检查锚链处视差连续、无大片空洞）
- □ 同一对五指标算出，并与中间文件手工核对一致
- □ 一键脚本（推理→保存→算分）已串通并放入 tasks/u12_eval/
- □ 10 对批量版完整日志

5 实验二报告需要写的内容

- 链路打通记录与一键脚本说明
- 10 对批量指标表

6 产生 / 更新文件后，需要更新与上传什么

- 一键脚本与批量日志存 tasks/u12_eval/（实验三两版权重比武就用它）
- 实验二证据三件套归入「证据包-实验二」
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）

第 13 组 · 项目三 · 数据基座治理（海洋石油平台）

数据结构化组

实验二

2026-11-02 布置

1 实验二的任务（一句话）

本地部署 Python 解析环境（PyMuPDF / pdfplumber / camelot 三选二装通），批量跑通 10 页的「抽取→锚点→登记」流水线，产出第一批结构化 JSON。

2 本组工作目录（文件夹）

本机 · `tasks/d01_struct/`（实验一已建：`data/` 放样例 PDF、`src/` 放脚本、`out/` 放产物）

环境：独立虚拟环境；三选二安装 `pip install pymupdf pdfplumber`（`camelot` 依赖 `Ghostscript`，装不上果断放弃，三选二即达标）

产物：`extract_pipeline.py`（输入一个 PDF，逐页「表格抽取→公式区域记录→锚点登记」输出 JSON）

中文路径报错——部分解析库对非 ASCII 路径支持不佳，样例统一放英文相对路径（如 data/sample1.pdf）最省事。跨页表格会被拆成两截：先如实登记“本表第 37 页结束、第 38 页有同列数疑似延续”，不强行合并——自动合并是加分项不是门槛。锚点 JSON 的字段口径（physical_page/bbox/confidence/flag）以讲义第 13 组任务书的产物示例为准。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。我要搭 PDF 结构化小流水线。请解读：(1) pdfplumber 与 PyMuPDF 各自靠什么认表格（线框？文字坐标？）、什么情况下 extract_tables 返回空（提示：扫描件没有文字层）；(2) 跨页表格为什么会被拆开、登记“可拼接线索”该记哪些字段；(3) 锚点里的 bbox 四个数怎么对应页面坐标。然后帮我写 extract_pipeline.py 骨架：逐页表格抽取→公式区域记录→按讲义示例字段输出锚点 JSON，每行加中文注释。我贴解析报错时请逐条解释。

4 验收标准：满足全部条件才可结束迭代

- ☐ 两个解析库的版本号输出（三选二装通）
- ☐ 10 页样例流水线的完整终端日志
- ☐ JSON 产物路径与统计：几张表、几段疑似公式、几条锚点、几个回退区域

5 实验二报告需要写的内容

- 流水线设计说明（选了哪两个库、为什么）
- 10 页产物统计与一处人工抽查核对

6 产生 / 更新文件后，需要更新与上传什么

- 流水线代码与 JSON 产物存 tasks/d01_struct/（out/）；中期第 14 组切你的产物问数
- 实验二证据三件套归入「证据包-实验二」
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）

第 14 组 · 项目三 · 数据基座治理（海洋石油平台）

智能问答与图检索组

实验二

2026-11-02 布置

1 实验二的任务（一句话）

搭通“一句话问数”小 demo：本地小模型 + 演示库 + 三个白名单工具 + SQL 安全检查——军规：模型只提出 SQL，程序执行 SQL；温度设 0。

2 本组工作目录（文件夹）

本机 · `tasks/d02_agent_kg/`（实验一已建：`demo/` 放问数服务、`kg/` 放图谱实验、`tests/` 放 30 题）

环境：`Ollama`（跑 `Qwen2.5-1.5B`，内存宽裕可 `3B`；温度设 `0`）+ `SQLite` 或 `DuckDB`（装下项目提供的四个精简查询模型 `CSV`）

三个白名单工具：`query_rows`（查明细）/ `agg_stats`（聚合）/ `describe_columns`（列名说明）——模型手里只有这三件

Ollama 模型没下载完就调用，报错还以为代码有问题。模型输出带解释文字不是纯 SQL，解析当场崩——输出格式在提示词里钉死。SQL 注入测试题自己先打自己：写一页攻击用例（改数据的、查系统表的、多语句的）逐条打自己的闸门。温度忘设 0，同一问题两次回答不一致会被同学当场抓包。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。我要搭“模型只提出 SQL、程序执行 SQL”的问数 demo。请帮我：(1) 设计三个白名单工具函数 query_rows/agg_stats/describe_columns 的签名与 docstring，并解释为什么模型手里只能有这三件；(2) 给出 SQL 安全检查的实现要点——只读拦截（INSERT/UPDATE/DELETE/DROP）与列名白名单，用 SQLGlot 或正则各自怎么写；(3) 解释 Ollama temperature=0 在问数场景为什么是硬要求；(4) 写一页我能用来打自己闸门的 SQL 注入测试用例清单。

4 验收标准：满足全部条件才可结束迭代

- □ ollama list 输出（模型已下载完）
- □ 建库行数统计（四个精简查询模型各自行数）
- □ 三个白名单工具函数与 SQL 安全检查代码
- □ 自测的注入用例逐条被拦截的记录
- □ 三个样例问答录屏（含生成的 SQL 与查询结果）

5 实验二报告需要写的内容

- demo 架构说明：模型/工具/检查/执行四层怎么衔接
- 三个样例问答记录与注入自测结论

6 产生 / 更新文件后，需要更新与上传什么

- demo 代码存 tasks/d02_agent_kg/demo/；测试题存 tests/（实验四扩成 30 题）
- 实验二证据三件套归入「证据包-实验二」
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）

第 15 组 · 项目三 · 数据基座治理（海洋石油平台）

虚假内容识别与对抗训练组

实验二

2026-11-02 布置

1 实验二的任务（一句话）

把实验一的三条规则扩成可配置的规则引擎，样本扩到 ≥100 条，训出 TF-IDF + 逻辑回归基线分类器——两条基线的成绩就是以后一切改进的对照物。

2 本组工作目录（文件夹）

本机 · `tasks/d03_adversarial/`（实验一已建：`fake/` 放造假产物、`dataset/` 放混装数据集、`model/` 放分类器）

环境：`pandas + scikit-learn`（README 锁版本）；实验一的 `15` 条样本是这次扩样的种子

流程：三条规则写成检查函数 → 按配方扩样 `≥100` 条（`AI` 按配额造假，开始记账）→ `TfidfVectorizer + LogisticRegression` 训练并留验证集

规则设得太严把真样本全判假，查准率崩盘——规则是闸门不是筛子。类别失衡却只看总体准确率，会被“全判真”骗过——看宏平均与混淆矩阵。sklearn 版本差异导致指标对不上——README 里锁版本。

3 第一轮提示词建议（发给 GLM，可直接复制）

▼ 复制以下整段，作为发给 GLM 的第一轮提示词

你是 GLM agent。我在做真假内容判别基线。请帮我：(1) 把“数值越界/单位错/时间戳矛盾”三条规则写成可配置的 pandas 检查函数（阈值集中在配置里）；(2) 给 TfidfVectorizer + LogisticRegression 的最小训练代码：train/val 划分、准确率/查准/查全/F1 与混淆矩阵一次算齐；(3) 解释宏平均为什么在本任务必看；(4) 给一版“AI 按难度配额造假”的提示词模板（易/中/难），我要扩到 100 条并记账。

4 验收标准：满足全部条件才可结束迭代

- □ 规则检查代码（三条规则可配置）
- □ ≥100 条样本（含实验一 15 条种子；AI 造假按配额并记账）
- □ 规则法与基线分类器各一份指标表（准确率/查准/查全/F1 同表）
- □ 盲区分析：哪类造假规则抓得住、哪类抓不住（纯文字造假大概率漏网——如实记下）

5 实验二报告需要写的内容

- 基线报告：两版基线指标表 + 盲区分析

6 产生 / 更新文件后，需要更新与上传什么

- 样本与模型存 tasks/d03_adversarial/（dataset/ 与 model/）；实验四接判官复核
- 实验二证据三件套归入「证据包-实验二」
- 线上提交：登录课程网站 → 实验与考核 → 「提交实验二报告」（截止 2026-11-27 周五 16:30，每组 ≤200MB，以最后一次上传为准）