

离线视频软件 MVP 操作手册

把现成视频输入系统并输出事件结果的操作顺序拆开写清楚。

这份手册解决什么问题

这份手册不是讲概念，而是讲操作顺序。

这份文档也不重复讲“每个开源项目为什么选它”。

如果你需要对照工具、组件和国内替代，优先看：

- 软件层 MVP 工具链与开源项目清单 (/toolchain/)

目标只有一个：

用现成泳池视频跑出一套可以复盘的软件结果，证明从视频输入到候选异常事件输出的整条链路成立。

这次 MVP 的输入和输出

输入

- 现成视频文件
- 每段视频的基本说明
- 一份泳道或区域配置

输出

- 事件结果表
- 带框和轨迹的输出视频
- 候选事件片段
- 一份人工复盘表

本手册适用的前提

- 先做离线视频，不做实时流
- 先用开源现成模型，不做自训练
- 先做单视角或少量视频，不做多机位融合
- 先做候选异常，不直接承诺“自动判定溺水”

角色分工

事项	非技术发起人	技术实施人
准备视频	负责	配合
说明场景	负责	配合
安装环境	不负责	负责
跑检测跟踪	不负责	负责
画泳道区域	解释业务含义	实现配置
看结果	负责	配合
调规则	提供业务反馈	负责

技术方交付给发起人的最小包

如果发起人不懂 GitHub，也不想先折腾开发环境，技术方至少应该交付下面这些东西：

1. 一个压缩包，例如 pool-drowning-mvp.zip
2. 一份 README.md
3. 一份 run_steps.txt
4. 一个 example_videos/ 目录
5. 一个 config/ 目录

理想状态下，发起人拿到包之后，至少能看懂：

- 原始视频放哪里
- 结果文件会输出到哪里
- 出错时把哪份日志发给技术方

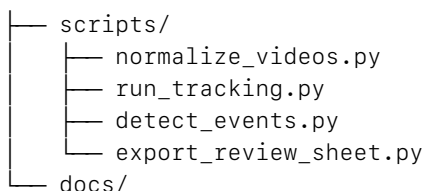
目录结构建议

建议从第一天起就把项目目录整理成这样：

```

pool-drowning-mvp/
├── README.md
├── data/
│   ├── raw_videos/
│   ├── normalized_videos/
│   ├── manifests/
│   └── review/
├── config/
│   ├── pool_zones.yaml
│   └── rules.yaml
├── outputs/
│   ├── tracks/
│   ├── events/
│   ├── clips/
│   └── annotated_videos/

```



最小运行顺序示意

如果技术方已经把环境准备好了，最小执行顺序可以长这样：

1. data/raw_videos/
2. scripts/normalize_videos.py
3. scripts/run_tracking.py
4. scripts/detect_events.py
5. scripts/export_review_sheet.py
6. outputs/events/events.csv outputs/annotated_videos/

如果技术方喜欢用命令行，也可以把 README 写成这种顺序：

```

python scripts/normalize_videos.py --input data/raw_videos --output data/normalized_videos
python scripts/run_tracking.py --input data/normalized_videos --config config/pool_zones.yaml
python scripts/detect_events.py --tracks outputs/tracks --rules config/rules.yaml --output outputs/tracks
python scripts/export_review_sheet.py --events outputs/events/events.csv --output data/review/review_sheet.csv

```

这些命令只是示意，关键是执行顺序要固定。

第一步：先确定第一批验证视频

不要一上来就塞几十小时。

第一轮建议只选：

- 3 到 5 段正常视频
- 2 到 5 段演练或复杂场景视频

总时长控制在：

- 30 分钟到 2 小时

第一轮的目的不是覆盖全面，而是跑通全流程。

第二步：建立视频清单

每段视频都写进清单。

建议字段：

字段	说明
video_name	文件名
pool_name	哪个泳池

字段	说明
camera_view	上方、水下、侧面
duration_sec	时长
lane_visible	泳道是否清楚
has_drill	是否是演练视频
note	备注

清单文件建议放在：

- data/manifests/video_manifest.csv

第三步：统一视频格式

这一阶段只做工程清洗。

目标是：

- 文件能稳定读取
- 格式尽量统一
- 输出目录清楚

常见动作：

- 转成统一 mp4
- 统一编码
- 裁掉太长的视频，拆成较短片段
- 记录分辨率和帧率

这一阶段的输出是：

- data/normalized_videos/

第四步：选择第一版检测和跟踪方案

第一版不要贪多。

推荐先只定 1 套主方案：

- 检测：YOLO
- 跟踪：Ultralytics Track 默认方案，或 ByteTrack

这一步的目标很简单：

- 看视频里的人能不能被持续看见

先别急着判断溺水。

第五步：先跑出轨迹

技术方需要先跑出下面这些中间结果：

- 每帧检测框
- 每个目标的 track_id
- 每个目标的中心点坐标
- 每帧时间戳

只有轨迹稳定，后面才有事件逻辑可谈。

建议至少导出：

- outputs/tracks/*.csv
- outputs/annotated_videos/*.mp4

第六步：画泳池区域和泳道

如果你要输出“第几泳道”，这一层必须显式配置。

建议至少画这几类区域：

- 整个泳池区域
- 每条泳道
- 深水区或重点风险区

建议配置文件长这样：

```
pool_name: pool_a
lanes:
  - lane_id: 1
    polygon: [[110, 80], [180, 80], [182, 620], [108, 620]]
  - lane_id: 2
    polygon: [[181, 80], [252, 80], [255, 620], [180, 620]]
risk_zones:
  - zone_id: deep_area
    polygon: [[108, 80], [420, 80], [422, 620], [106, 620]]
```

这一步的关键不是精美，而是可复用。

第七步：定义第一版事件规则

第一版建议只定义 2 到 4 条规则。

例如：

1. 长时间低速停留
2. 在高风险区域停留过久
3. 轨迹突然消失且长时间未恢复
4. 长时间处于近似静止状态

规则文件建议单独配置：

```
min_static_seconds: 8
min_low_speed_seconds: 6
max_missing_seconds: 4
event_score_threshold: 0.65
```

这一层的重点是：

- 规则简单
- 输出可解释
- 能快速重跑

第八步：导出事件

系统每跑完一段视频，都应该输出事件文件。

建议事件表字段：

字段	说明
event_id	事件编号
video_name	来源视频
event_type	事件类型
track_id	轨迹 ID
start_sec	开始时间
end_sec	结束时间
lane_id	泳道编号
score	规则分数
reason	触发原因
clip_path	对应片段

建议输出到：

- outputs/events/events.csv

- outputs/events/events.json

第九步：自动裁出候选片段

每条事件都应该能点开回看。

所以建议自动裁出：

- 事件前 10 秒
- 事件后 10 秒

保存到：

- outputs/clips/

如果没有片段，复盘效率会很低。

第十步：人工复盘

这一步必须有发起人参与。

建议每条事件都判断：

- 是否真的值得提醒
- 时间是否大致正确
- 泳道是否正确
- 误报原因是什么

复盘表建议字段：

字段	说明
event_id	对应事件
review_result	真阳性、误报、存疑
time_ok	时间定位是否可接受
lane_ok	泳道定位是否可接受
comment	备注

第十一步：只做小步调参

第一轮结果出来后，不要立刻进入自训练。

先只做这几类小步修正：

- 改阈值

- 改泳道区域
- 改事件触发时长
- 改最短片段长度
- 改视频标准化方式

第二轮建议扩充到：

- 10 到 20 段视频
- 至少覆盖不同时间段和不同拥挤度

第十二步：形成 MVP 验证报告

最后必须有一份报告。

建议最少回答这 6 个问题：

1. 哪类视频能跑通
2. 哪类视频效果明显差
3. 哪些规则最有价值
4. 哪些误报最多
5. 泳道定位是否可用
6. 是否值得进入第二阶段

第一阶段验收标准建议

如果下面这些条件大部分达成，就说明软件 MVP 有继续做的价值：

1. 能完整跑完至少 10 段视频
2. 每段视频都能输出带轨迹的结果视频
3. 大部分候选事件都能给出时间点
4. 大部分候选事件都能映射到泳道或区域
5. 人工复盘认为其中一部分事件确实有提醒价值

哪些情况说明现在还不该进入第二阶段

- 视频里连人都很难稳定检测到
- 跟踪 ID 频繁跳变，没法做规则
- 泳道区域根本无法可靠标定
- 绝大多数事件都是无意义误报

- 业务方看完后认为结果没有任何提醒价值

如果出现这些情况，先别讨论部署摄像头。

第二阶段再做什么

只有软件层 MVP 成立后，才建议进入：

- 现场实时接入
- 机位和摄像头方案
- 水下相机补强
- 标注和自训练
- 告警联动

这时再去看 室内泳馆现场试点技术规划 (/pilot-plan/) 才是合理顺序。