视频模态：结构化信息与情绪变量制作说明

多模态信息结构化提取与情绪变量构造系统 · 按当前代码整理

> **先回答一个问题：一条视频里到底发生了什么？说话内容、声音气氛和画面表情是一致的，还是彼此矛盾？**

一、理解版：从问题到结果

先把“多模态”理解成同一件事请三位观察员一起看：一位负责听懂说了什么，一位只注意声音听起来是什么感觉，另一位只观察画面中的表情和动作。视频分析的目的，就是把这三位观察员的记录放在同一张表里，研究它们是否在表达同一种情绪。

一部视频往往很长，直接从头看到尾，就像在一盘很长的录像带里找某一句话，既慢又不容易定位。因此系统先找出“有人说话”的时间段。可以把这一步想成给视频装了一个说话指示灯：有人讲话时灯亮，完全安静时灯灭。它只负责找位置，不负责听懂具体说了什么。

找到说话位置以后，系统像剪辑师一样，在停顿附近把长视频剪成一段一段的小片。每个小片都有自己的开始时间和结束时间。这样研究者以后看到某个结果时，可以马上回到原视频的对应片段，而不是重新翻找整部视频。

接着请一位“既能看画面、又能听声音、还能读懂语言”的助手看每个小片，并回答几个固定问题：里面讲了什么？出现了谁和什么东西？发生了什么事？整体更像开心、平静还是紧张？这一步把原本散在视频里的内容，整理成可以放进表格的文字。

同一个小片还会被再看一遍，但这一次分成两个问题：如果只听声音，不看说话内容，语气像什么情绪？如果只看画面，不听声音，表情和动作像什么情绪？这样就能区分“嘴上说得很平静，但声音很紧张”这类情况。

最后像让三位观察员对答案：如果三种判断完全一致，就保留这个情绪；如果互相冲突，就把“存在分歧”记录下来，并暂时标为中性。这个中性不是说视频真的没有情绪，而是提醒研究者：三种线索没有达成一致，应该回看原片。

> **边界提醒：**视频结构化和两类非语言情绪目前都依赖 Qwen API；FunASR 在视频项目里只做 VAD。不要把 dialogue_text 当成 FunASR 的逐字稿，也不要把冲突时的 neutral 当成模型真正判断出的中性情绪。

二、学术版：可直接放入方法部分的表述

> 本项目首先使用 ffmpeg 将视频音轨转换为单声道 16 kHz WAV，并由 FunASR FSMN-VAD 识别语音活动区间，在约 30 秒目标长度、8–45 秒边界约束下合并区间并生成视频片段；随后将片段分别输入 Qwen

> qwen3.6-plus：阶段 1 提取对白、摘要、实体、关键词、事件及语义情绪，阶段 2 在不使用对白语义的条件下提取声学与视觉情绪；最后以三对标签相等指示变量构造跨模态一致性分数，并在三模态完全一致时保留标签、否则记为中性。

三、实际使用的包、模型与职责

组件/模型	作用	本项目实际怎么用
ffmpeg / ffprobe	媒体处理工具	读取视频时长、抽取 16 kHz 单声道音频、按时间戳生成视频片段；超过 14 MiB 的 API 输入先缩放到 640 像素宽并压缩为 H.264/AAC 代理视频。
FunASR FSMN-VAD	语音活动检测模型	只标出有人说话的起止毫秒，不做文字转写；默认本地目录为 speech_fsmn_vad_zh-cn-16k-common-pytorch。
Qwen qwen3.6-plus（阶段 1）	视频结构化提取	接收短视频，严格 JSON 输出 dialogue_text、summary、entities、keywords、events、emotion_label 和 emotion_reason。
Qwen qwen3.6-plus（阶段 2）	非语言情绪识别	接收同一短视频，分别输出 audio_emotion_label 与 visual_emotion_label；提示词禁止用对白语义代替声学/视觉线索。
Python csv / json / pathlib	结果整理与融合	写出片段清单、结构化变量、情绪变量和一致性表，并清理下载结果中的服务器绝对路径。

四、按代码复现：每一步输入什么、做什么、得到什么

步骤	人话说明与复现动作	得到什么
1. 准备输入	上传 .mp4、.mov、.avi、.mkv 或 .webm。网页任务会把文件放到临时目录；模型只接收短片段，原始长视频不会直接反复发送。	原始视频文件
2. 抽音频与 VAD	ffmpeg 输出 mono/16 kHz/PCM WAV；FunASR FSMN-VAD 返回毫秒区间 [[start_ms, end_ms], ...]。相邻区间按目标 30 秒合并，超过 45 秒再拆开。	语音区间
3. 生成片段	用 ffmpeg 的 -ss、-t 和 -c copy 按区间生成 segment_path，并记录 segment_start_sec、segment_end_sec。	`video_segment_variables.csv`
4. 语义结构化	阶段 1 以 video_url + JSON schema 调用 Qwen。返回结果抓取第一个 JSON 对象，列表字段在 CSV 中用“|”连接。	`video_variables.csv` / `per_video_structured/*.json`

步骤	人话说明与复现动作	得到什么
5. 声学与视觉情绪	阶段 2 使用非语言提示词再次调用 Qwen，音频和画面各得一个标签及理由；失败时当前代码保留 neutral 回退值并记录 API 错误。	`video_emotion_variables.csv`
6. 一致性与最终标签	阶段 3 计算 text/audio、text/visual、audio/visual 三个相等指示变量；三者都为 1 才保留情绪，否则 final_emotion_label=neutral。	`video_emotion_consistency.csv`

五、公式、权重与融合规则

$$Consistency_i = \frac{\mathbb{1}(T_i=A_i) + \mathbb{1}(T_i=V_i) + \mathbb{1}(A_i=V_i)}{3}$$

$$FinalLabel_i = \begin{cases} T_i，若\ T_i=A_i=V_i \\ \text{neutral}，否则 \end{cases}$$

LaTeX 写法（可复制到论文编辑器）

```
\operatorname{Consistency}_i = \frac{\mathbb{1}(T_i=A_i)+\mathbb{1}(T_i=V_i)+\mathbb{1}(A_i=V_i)}{3}

\operatorname{FinalLabel}_i = \begin{cases}T_i, & T_i=A_i=V_i\\ \text{neutral}, & \text{otherwise}\end{cases}
```

一致性： $Consistency_i = [I(T_i=A_i) + I(T_i=V_i) + I(A_i=V_i)] / 3$。T、A、V 分别是文本、声学、视觉情绪标签，I(条件) 成立取 1，否则取 0。

最终标签： 当三个指示变量均为 1 时，$FinalLabel_i=T_i$；否则 $FinalLabel_i$=neutral。这里没有连续权重，研究者报告时应把它写成规则融合。

公式中的英文变量名与实际输出列保持一致；若某一步是规则优先级而不是连续加权，文中已明确标出。

六、关键输出变量：英文名与中文解释

英文变量名	中文解释
`source_video_id`	原始视频编号（文件名去掉扩展名）
`segment_id`	视频片段编号

英文变量名	中文解释
segment_index	片段在原视频中的顺序，从 0 开始
segment_start_sec / segment_end_sec	片段在原视频中的开始/结束秒数
dialogue_text	片段中的对白或语言内容，由 Qwen 从视频中整理
summary	片段内容的一句话摘要
entities_raw	人物、组织、地点、产品等实体，多个值用 | 分隔
keywords_raw	关键词列表
events_raw	事件或动作列表
text_emotion_label	根据语言/语义判断的情绪标签
audio_emotion_label	根据音调、语速、响度、停顿和韵律判断的情绪
visual_emotion_label	根据面部表情和可见行为判断的情绪
consistency_score	三对情绪标签相等的比例，取值 0、0.3333、0.6667 或 1
final_emotion_label	三模态完全一致时的标签；冲突时为 neutral

七、结果文件与研究用途

结果文件	用途
video_segment_variables.csv	切片清单和时间边界
video_variables.csv	Qwen 阶段 1 的结构化变量与语义情绪
video_emotion_variables.csv	文本、声学、视觉三类情绪及理由
video_emotion_consistency.csv	三对一致性指标和最终标签
video_batch_results.json / video_run_summary.json	逐片段原始结果、成功/失败/重试摘要

可按 segment_id 研究视频片段的主题、事件和情绪一致性，例如比较不同来源视频的跨模态一致性均值，或把 final_emotion_label 转成分类变量。正式分析前要保留原始片段，抽样核对对白、声学理由和视觉理由。

本篇说明作为项目复现参考，更多字段、原始响应和运行日志请结合下载结果中的 JSON、CSV、progress.json 与 config_snapshot.json 一起核对。