✅ 视频标注整体流程

```
原始视频 (mp4)
   ↓
[Step 1] 按 10s 切割视频
   ↓
10s 子视频 (.mp4 × N)
   ↓
[Step 2] 构造 jsonl 数据集（video + prompt）
   ↓
videotest_10s.jsonl
   ↓
[Step 3] 配置环境（conda + requirements）
   ↓
[Step 4] 跑 exp_forrest.py
   ↓
模型加载 → 视频抽帧 → prompt → LLM 推理
   ↓
[Step 5] 输出 jsonl（每 10s 一个情绪描述）
```

🟦 Step 0：目录结构（非常重要）

假设你的项目根目录是：

```
/mnt/dataset0/lzs/ViLAMP-main
```

最终你应该有这些关键路径：

```
ViLAMP-main/
├── exp_forrest.py
├── llava/
├── models/
│   └── ViLAMP-llava-qwen/
├── icml/
│   ├── videotest.mp4              # 原始视频
│   ├── videotest_10s/              # 10s 切片视频
│   ├── videotest_10s.jsonl         # 数据集
│   └── videotest_out/              # 模型输出
```

🟦 Step 1：按 10 秒切割视频

1.1 为什么要切?

- ViLAMP / LLaVA **不是流式模型**

- 情绪是时间变化的 → **用短时间窗口**
- 10s 是一个非常合理的 affective window（论文级）

1.2 切割代码

✅ 直接终端输入

```
ffmpeg -i icml/videos/videotest.mp4 \
  -c copy -map 0 \
  -f segment -segment_time 10 -reset_timestamps 1 \
  icml/videotest_10s/videotest_%05d.mp4
```

输出：

```
videotest_00000.mp4
videotest_00001.mp4...
videotest_00026.mp4
```

🟦 Step 2：构造 jsonl 数据集（最关键）

2.1 jsonl 是什么?

一行 = 一个样本

每一行告诉模型：

- 视频在哪
- 问什么问题（prompt）

2.2 生成jsonl的prompt（在下面终端输入的代码里）

（后续可以改）：

```
You are an expert in affective computing.
Watch the video and describe the emotions expressed in the scene.
Focus on facial expression, body language, scene tone, lighting, motion and pacing.
Output ONLY JSON with keys:
dominant_emotion, secondary_emotions, valence, arousal, description.
dominant_emotion must be one of
[neutral,happiness,sadness,anger,fear,disgust,surprise,calm].
valence in [-1,1], arousal in [0,1].
```

2.3 生成 jsonl 的脚本

✅ 终端输入

```
python - <<'PY'
import glob, json, os
prompt = """You are an expert in affective computing.
Watch the video and describe the emotions expressed in the scene.
Focus on facial expression, body language, scene tone, lighting, motion and pacing.
Output ONLY JSON with keys:
dominant_emotion, secondary_emotions, valence, arousal, description.
dominant_emotion must be one of
[neutral,happiness,sadness,anger,fear,disgust,surprise,calm].
valence in [-1,1], arousal in [0,1]."""
videos = sorted(glob.glob("icml/videotest_10s/*.mp4"))
out_path = "icml/videotest_10s.jsonl"
with open(out_path, "w", encoding="utf-8") as f:
    for v in videos:
        f.write(json.dumps({"video": os.path.abspath(v), "query": prompt},
ensure_ascii=False) + "\n")
print("wrote", out_path, "num_videos=", len(videos))
PY
```

生成：

```
icml/build_videotest_jsonl.py
```

2.4 检查数据集（一定要做）

```
head -n 2 icml/videotest_10s.jsonl
```

你应该看到类似：

```
{"video": "icml/videotest_10s/videotest_00000.mp4", "query": "..."}
```

🟦 Step 3：配置运行环境（关键复现点）

3.1 建议：

单独一个环境

```
conda create -n vilamp python=3.10 -y
conda activate vilamp
```

3.2 requirements（最后跑通的版本）

```
transformers==4.45.0
torch
torchvision
einops
opencv-python
pandas
av
decord
sentence_transformers
accelerate==0.29.3
peft==0.13.2
diffusers==0.32.2
tyro
wandb==0.18.7
deepspeed==0.15.4
```

安装：

```
pip install -r requirements.txt
```

3.3 验证依赖（非常重要）

```
python - <<'PY'
import torch, transformers, cv2, decord, pandas, einops
print("torch:", torch.__version__)
print("transformers:", transformers.__version__)
print("env OK")
PY
```

🟦 Step 4：运行 exp_forrest.py（核心）

4.1 命令

```
cd /mnt/dataset0/lzs/ViLAMP-main

python exp_forrest.py \
  --dataset_path icml/videotest_10s.jsonl \
  --output_dir icml/videotest_out \
  --version ViLAMP-llava-qwen \
  --split 1_1 \
  --max_frame_num 120
```

4.2 exp_forrest.py 内部做了什么（理解级）

```
jsonl →
  读视频 →
  每秒抽 1 帧 →
  CLIP + Vision Encoder →
  Qwen LLM →
  文本生成 →
  写回 jsonl
```

🟦 Step 5：输出是什么？

运行完成后：

```
ls icml/videotest_out
```

你会得到：

```
ViLAMP-llava-qwen-1_1.json
```

内容就是你刚才贴出来的那样：

```
{
  "video": "...videotest_00012.mp4",
  "query": "...",
  "response": "{... emotion json... }"
}
```

👉 **一行 = 一个 10 秒视频的情绪描述**

🧠 最终一句总结

> 我们把一个长视频切成 10 秒片段
>
> → 每个片段当成一个情绪样本
>
> → 用 ViLAMP-llava-qwen 生成结构化情绪描述
>
> → 得到一条“时间对齐的情绪序列”