

# 第三通道技术规格书：逻辑与事实推理检测

模块名称：Channel 3: Logical Reasoning Engine (VLM-CoT)

文件路径：channel\_3\_reasoning/

定位：系统逻辑层防线，处理 CLIP 无法识别的细粒度属性冲突与常识谬误。

## 1. 核心技术目标

本模块旨在构建一个具备\*\*“深度认知”能力的 AI 审判官，通过视觉大模型 (VLM)\*\* 与 **思维链 (Chain of Thought)** 技术，解决语义层（通道二）无法覆盖的细粒度逻辑冲突。

### 1.1 与通道二 (CLIP) 的关键区别 (Critical Distinction)

- 通道二 (语义一致性)**：解决 "Topic Alignment" (主题是否一致)。
  - 能力边界**：只能判断“图和文是不是在说同一件事”。对于细节（如天气、时间、数量）不敏感。
- 通道三 (逻辑推理)**：解决 "Fact Verification" (事实是否冲突)。
  - 能力边界**：在主题一致的前提下，通过 **VLM (视觉转译)** 和 **LLM (逻辑比对)**，寻找时空、因果、常识上的矛盾。

### 1.2 覆盖范围 (The Reasoning Space)

- 细粒度属性冲突 (Fine-grained Attribute Conflict)**
  - 时间**：图(正午) vs 文(深夜)。
  - 天气**：图(晴天) vs 文(暴雨)。
  - 数量**：图(空地) vs 文(人山人海)。
- 实体/地标错位 (Entity Mismatch)**
  - 地标**：图(东方明珠) vs 文(东京塔)。
  - 文字**：图中的路牌/横幅文字与新闻内容矛盾 (OCR能力)。
- 常识因果谬误 (Common Sense Error)**
  - 物理常识**：图(夏天短袖) vs 文(大雪纷飞)。

## 2. 技术原理：VLM-CoT (Chain of Thought)

本模块采用 **Visual Chain of Thought (视觉思维链)** 技术路线：

- Step 1: 视觉转译 (Captioning)**
  - 利用 VLM (如 Moondream) 将图片转化为**结构化元数据**。
  - Prompt**: "Describe the image focusing on: time of day, weather, location landmarks, and quantity of people."
- Step 2: 逻辑比对 (Reasoning)**
  - 利用 LLM 将“视觉描述”与“新闻文本”进行 NLI (自然语言推理) 任务。
  - Logic**:  $\text{Premise}(\text{Image}) \rightarrow \text{Hypothesis}(\text{Text})$  ?

**工程实现注记**：鉴于演示环境算力限制，系统默认开启 **Mock Mode (模拟模式)**，即通过读取预处理的元数据 (Excel Ground Truth) 模拟 VLM 的输出，以确保演示的

低延迟与高准确率。

### 3. 接口定义 (Interface Specification)

本模块代码放置于 channel\_3\_reasoning/ 目录下。

#### 3.1 实际目录结构要求

```
MultiChannel-Reasoning-System/
├── data/
│   ├── images/          # [数据仓]
│   └── YuanJing_AI_Data_Standard.xlsx # [知识库] 提供 Meta 字段作为推理依据
├── channel_3_reasoning/  # [本模块工作区]
│   ├── logic_engine.py   # [主代码] 包含 LogicReasoner 类
│   ├── utils.py          # [工具类]
│   └── main_demo.py      # [调用方]
```

#### 3.2 输入输出规范

维度	参数名	类型	描述
Input 1	image_path	str	图片相对路径。
Input 2	text	str	新闻文本。
Input 3	meta_data	dict	<b>关键输入。</b> Excel 中的元数据行。在 Mock 模式下，它是推理引擎的直接依据。
Output 1	is_conflict	bool	True (逻辑冲突 / Fake) / False (逻辑自洽)。  <b>该结果具有一票否决权。</b>
Output 2	reason	str	<b>推理证据。</b> 例如： "Conflict: Image shows 'Sunny', Text says 'Rain'."

4. Excel 数据字段对代码逻辑的影响

为了体现与通道二的区别，第三通道的测试数据必须\*\*\*“看起来像真的，细看是假的”\*\*\*。Meta 字段在这里充当了 "Oracle" (全知视角) 的角色。

4.1 关键字段解析

字段名称	对 Channel 3 代码逻辑的影响
Meta_Time	<p><b>时间真值。</b>取值：Day / Night。</p> <p>代码将对比此字段与文本中提取的时间词（如"深夜"），若矛盾则报警。</p>
Meta_Scene	<p><b>场景真值。</b>取值：Sunny / Rain / Indoor 等。</p> <p>代码将对比此字段与文本中提取的环境词（如"暴雨"），若矛盾则报警。</p>
Meta_Object	<p><b>实体真值。</b></p> <p>代码将检查文本中提及的关键实体是否与此字段冲突（如地标冲突）。</p>

4.2 数据构造与验收标准 (SOP)

Type A: 属性陷阱样本 (Attribute Trap)

- **特点：**图文主题完全一致（骗过通道二），但核心属性相反。
- **构造：**
  - 图：白天的森林火灾。
  - 文：“昨晚深夜，森林发生大火...”
  - Excel设置：Meta\_Time = Day。
  - 预期：Channel 2 (Pass); **Channel 3 (Conflict, Fake)。**

Type B: 地理陷阱样本 (Geo Trap)

- **特点：**地标相似但不同。
- **构造：**
  - 图：伦敦桥。
  - 文：“金门大桥通车仪式...”
  - Excel设置：Meta\_Object = London Bridge。
  - 预期：Channel 2 (Pass); **Channel 3 (Conflict, Fake)。**

## 5. 开发里程碑

1. **Mock 逻辑实现**：优先完成基于 Excel 元数据的规则比对逻辑，确保 Demo 能跑通 Type A 和 Type B 样本。
2. **Prompt 设计**：设计用于从文本中提取时间、地点实体关键词的正则表达式或简单逻辑。
3. **VLM 接口预留**：代码结构中需预留 `call_vlm_api()` 函数接口，为后续接入真实模型做准备。