Solstice: Technical Architecture

Computer Vision + Multi-LLM Pipeline for Medical Document Verification

1 Technical Implementation

The Solstice pipeline processes medical documents through computer vision-based layout detection followed by multi-stage LLM analysis. Each stage builds on the previous, creating a traceable verification chain from raw PDF to final evidence report.

2 Pipeline Architecture

1. **Layout Detection**: Detectron2 model identifies bounding boxes for text blocks, tables, figures, and captions
2. **Content Extraction**: Parser converts detected regions into structured JSON with text content and metadata
3. **Claim Processing**: User-provided claims are processed through orchestrated LLM pipeline
4. **LLM Execution**: Four specialized LLM calls analyze documents with specific prompts
5. **Result Caching**: All intermediate outputs stored in hierarchical cache structure for reproducibility

3 Implementation Details

3.1 Layout Detection Pipeline

The layout detection uses Detectron2 with ResNet-50 backbone. The model outputs bounding boxes with class labels:

The detection pipeline:

- Processes PDFs page-by-page at 400 DPI resolution
- Runs inference with confidence threshold of 0.2
- Merges overlapping boxes to reduce duplicates
- Outputs JSON with bbox coordinates and class labels

3.2 Multi-Stage LLM Processing

The claim verification uses four LLM calls (Claude-3.5-Sonnet):

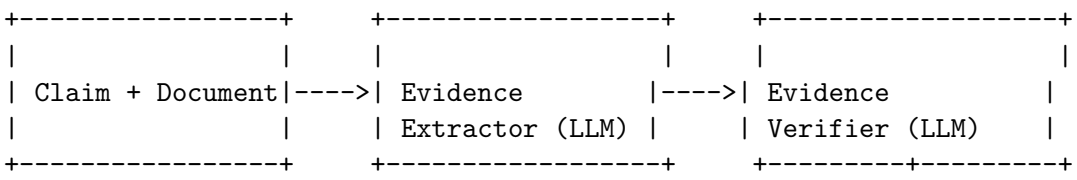
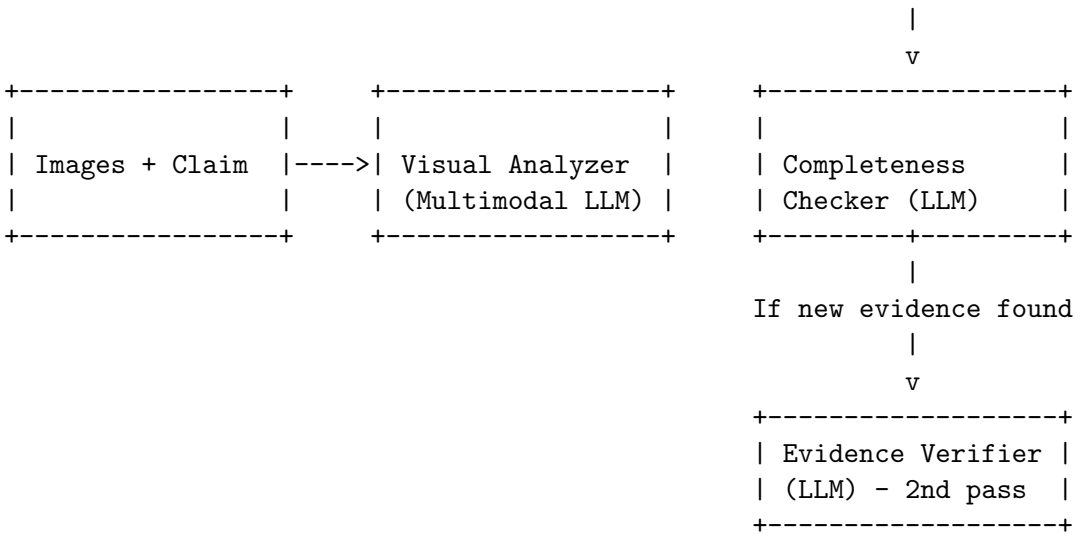


Figure 1: Detectron2 output showing bounding box detection on clinical trial paper. Colors indicate detected classes: text (blue), table (green), figure (red), caption (yellow).



1. **Evidence Extractor**:
   - Input: Claim text + full document content
   - LLM extracts quotes that support the claim
   - Returns JSON with quotes and relevance explanations
2. **Evidence Verifier**:
   - Input: Extracted evidence + original document
   - Validates quote accuracy and relevance
   - Checks for missing context or cherry-picking
3. **Completeness Checker**:
   - Input: All previous LLM outputs
   - Searches for additional supporting evidence
   - If found, new evidence is passed to Evidence Verifier
4. **Visual Analyzer**:
   - Input: Extracted tables/figures as PNG images + claim text
   - Uses multimodal LLM to analyze if image supports the claim
   - Returns binary support decision with extracted evidence

3.3 Cache Structure

All outputs are stored in a hierarchical cache:

```
data/cache/{document_name}/
    extracted/
        content.json      # Structured text content
        document.md       # Markdown representation
        figures/          # Extracted images
    agents/        # Named for legacy reasons, stores LLM outputs
        claims/
```

```
        claim_{id}/
            evidence_extractor/output.json
            evidence_verifier/output.json
            completeness_checker/output.json
            image_evidence_analyzer/
                {figure_id}/output.json
visualizations/       # Layout detection overlays
```

3.4 Marketing Pipeline Adaptation

Marketing materials require modified processing due to different layout patterns:

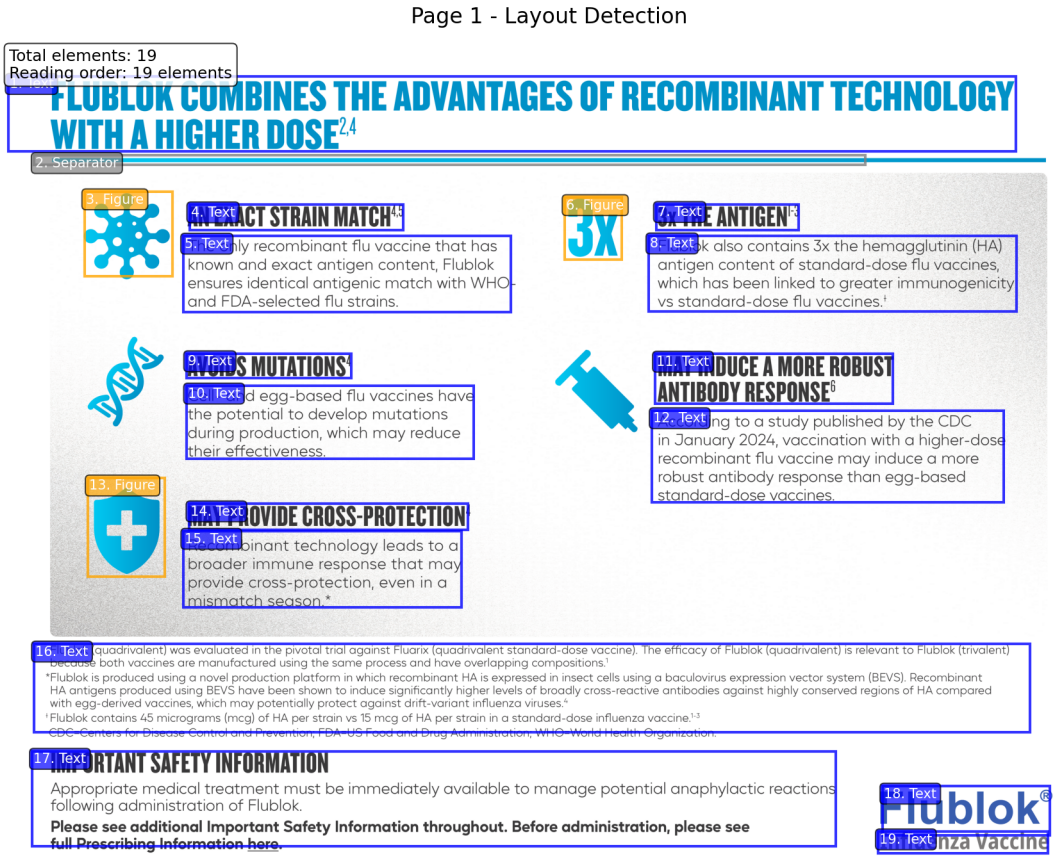


Figure 2: Marketing layout detection shows emphasis on visual hierarchy and promotional elements. The pipeline adapts confidence thresholds for better detection of design-heavy layouts.

Key differences:

- Lower confidence threshold (0.1 vs 0.2) for creative layouts
- Additional post-processing for overlapping design elements
- Separate cache directory (data/marketing_cache/)
- Same LLM prompts used for both document types