Technical Design Document (TDD): Teach by Doing (TbD) V3.0 - "Insight"

Status: Approved **Author:** Greg Burns / System Architect **Date:** November 23, 2025 **Previous Version:** V2.1 (Cloud Native MVP)

1. Executive Summary

While V2.0 successfully established a scalable, asynchronous infrastructure on Google Cloud Platform (GCP), the quality of the data extraction (V1 logic) remains basic. V3.0 focuses on the "Brain" of the system, replacing simplistic heuristics with robust Computer Vision techniques and Generative AI to maximize data fidelity.

Primary Goal: Increase ActionNode confidence > 85% and eliminate "UNKNOWN_TEXT" errors.

2. Architecture Updates

The V2 infrastructure (Dispatcher/Worker/PubSub/GCS) remains unchanged. The upgrades are strictly within the **Worker Service's processing pipeline**.

2.1 The V3 Pipeline Flow

1. **Ingest:** (Same as V2) Download video from GCS.
2. **Segment:** (Enhanced) Use PySceneDetect, but refine thresholds based on motion energy.
3. **Active Region Discovery (New):**
    - Compare *Frame N* (Start of segment) vs *Frame N-1* (End of previous segment).
    - Compute **SSIM (Structural Similarity)** difference map.
    - Apply contours to the difference map to generate a specific bounding box `[x, y, w, h]` where the screen changed.
4. **Spatial OCR (New):**
    - Run Tesseract on the full frame.
    - **Filter:** Discard any text that does not geometrically intersect with the *Active Region* bounding box.
    - *Result:* We only "read" what the user interacted with.
5. **Action Classification (New):**
    - Calculate **Optical Flow** vectors between frames.
    - If vectors show consistent vertical movement -> `Action: Scroll`.

    - If vectors show localized high energy -> `Action: Click`.
    - If keyboard heatmap (future) or rapid text appearance -> `Action: Type`.
6. **Semantic Enrichment (GenAI - Updated):**
    - Call **Vertex AI (Gemini 2.5 Pro)**.
    - Input: Keyframe Image + Raw OCR Text + Action Type.
    - Prompt: *"Describe the user's action in this UI screenshot. The user clicked '{ocr_text}'. Context: Software Tutorial."*
    - Output: Human-readable `description` field.

3. Implementation Specs

3.1 Unified Vision Pipeline (`pipeline.py`)

The `build_pathway` function will be refactored to include all vision modules (SSIM, OCR, and Optical Flow).

Python
```
def detect_active_region(frame_before, frame_after):
    # Convert to grayscale
    grayA = cv2.cvtColor(frame_before, cv2.COLOR_BGR2GRAY)
    grayB = cv2.cvtColor(frame_after, cv2.COLOR_BGR2GRAY)

    # Compute Structural Similarity Index (SSIM) between the two images
    # ensuring we get the difference image
    (score, diff) = compare_ssim(grayA, grayB, full=True)
    diff = (diff * 255).astype("uint8")

    # Threshold the difference image to find regions of change
    thresh = cv2.threshold(diff, 0, 255, cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]
    contours = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    # Return the largest bounding rect as the Active Region
    # ... (logic to pick largest contour)
    return x, y, w, h

def classify_action_optical_flow(prev_frame, curr_frame):
    # Use Farneback optical flow to detect movement patterns
    flow = cv2.calcOpticalFlowFarneback(prev_frame, curr_frame, ...)
    # Logic to distinguish Scroll vs Click based on flow vectors
    return action_type
```

3.2 Gemini Integration (Vertex AI)

We will add `google-cloud-aiplatform` to `requirements.txt` and use the 2.5 Pro model.

Python
```
from vertexai.preview.generative_models import GenerativeModel, Part

def generate_semantic_description(image_bytes, ocr_text):
    # UPDATED: Using Gemini 2.5 Pro for maximum robustness as requested
    # Note: This assumes the endpoint alias 'gemini-2.5-pro-preview' or similar is available
    model = GenerativeModel("gemini-2.5-pro-preview")
    response = model.generate_content(
        [
            Part.from_image(image_bytes),
            f"The user clicked on the UI element labeled '{ocr_text}'. Write a one-sentence
description of this step for a tutorial."
        ]
    )
    return response.text
```

4. Data Schema Updates (PAD v0.2)

We will expand the `ActionNode` schema to support richer data.

Python
```
class ActionNode(BaseModel):
    # ... existing fields ...

    # New V3 Fields
    action_class: str  # e.g., "navigation", "input", "confirmation"
    semantic_description: str # AI-generated summary
    active_region_confidence: float # How sure are we about the click location?
    screen_context: str # e.g., "Settings Menu", "Main Dashboard" (from LLM)
```

5. Release Scope (Version 3.0)

Version 3.0 "Insight" will be delivered as a single, unified release containing the complete feature set. There is no phased rollout; the goal is to move from "Video Parsing" to "Task Understanding" in one deployment cycle.

- **Core Vision Upgrade:** SSIM differencing for precise active region discovery and Spatial OCR to eliminate "UNKNOWN_TEXT" errors.
- **Action Classification:** Optical Flow analysis to distinguish between clicks, scrolls, and typing events.
- **AI Integration:** Full integration with **Vertex AI (Gemini 2.5 Pro)** for semantic description generation.

6. Risks & Mitigation

- **Latency:** Calling Gemini for every node will add latency.
    - *Mitigation:* Run Gemini calls in parallel using `asyncio` after the main video processing loop is done.
- **Cost:** Vertex AI calls cost money per image.
    - *Mitigation:* Use Gemini 1.5 Flash if Pro becomes cost-prohibitive, or batch requests.