Updated Hackathon Workflow: Vision-Based Document Analysis

Overall Strategy

This workflow uses a two-stage pipeline. **Round 1A** employs a pure computer vision approach to identify a document's structure directly from page images. **Round 1B** then uses this highly accurate structural output to perform a semantic analysis and find content relevant to a user's specific needs.

Workflow for Round 1A: Vision-Based Structure Extraction

The goal of this stage is to extract a structured outline by visually analyzing the document layout, making it robust for various document types, including scanned images. This approach specifically targets the multilingual bonus.

Models Used:

- **YOLOv10:** A model fine-tuned on a custom dataset (annotated in Roboflow) for document layout detection.
- **Tesseract OCR:** For accurate, multilingual text extraction from image regions.

Step-by-Step Process:

[Input: Single PDF File] ↓ **1. PDF to Image Conversion:** The system iterates through the input PDF, converting each page into a high-resolution image. ↓ **2. Layout Detection:** The fine-tuned **YOLOv10** model processes each page image. It detects and classifies all relevant layout elements, outputting bounding boxes and labels for each one (e.g., `title`, `heading_1`, `paragraph`, `list`). ↓ **3. Text Extraction (OCR):** For each bounding box identified by YOLO, the corresponding image region is cropped. This cropped image is then passed to the **Tesseract OCR** engine, which extracts the text content within that box. This step is configured to handle multiple languages. ↓ **4. JSON Assembly:** The label from the YOLO model (e.g., `heading_1` is mapped to H1) and the corresponding text from Tesseract are combined. This process is repeated for all detected elements to build the final, structured JSON output required by the hackathon specifications.

↓ **[Output: Final JSON File]**