Coding Test: Office Work Desk Edge Detection and Coordinate Retrieval

Objective Given an image of an office work desk, identify, plot, and retrieve the coordinates for the edge of the desk that is facing or close to the person's side. This test aims to assess the ability to accurately locate and define key structural elements within an office environment for potential applications in ergonomic analysis or spatial awareness.

[Blue line indicates the front edge of the desk closer to the person]

Inputs and important considerations:

- Image of an office work desk in standard formats (PNG, JPEG, WEBP, etc.)
- The desk may or may not be rectangular and straight (might have curves or corner desks)
- Workstation configuration may vary (e.g., presence of monitors, keyboards, other desk items).
- Lighting conditions, shadows, or partial occlusions may affect visibility.
- Variability in desk design, material, or color may affect detection.

Definitions and considerations:

1. **Desk Edge (Person-Facing)** The primary edge of the desk surface that would typically be closest to a seated or standing person using the desk. This is generally the longest horizontal edge of the desk facing the user.
    - **Identification**: This edge is crucial for defining the user's immediate workspace and can be used for ergonomic assessments (e.g., reach zones, proximity to input devices).
    - **Coordinate Retrieval**: The task requires identifying a series of points along this edge to define its shape and position in the image. This could be a line segment (two points) or a more complex curve (multiple points).

Task:

A. Detection & Analysis You are free to use any technique to analyze the image and identify the person-facing desk edge based on the definitions above. The output should include:

- The identified desk edge is plotted on the image.
- A list of coordinates (e.g., pixel coordinates [x, y]) representing the identified edge. For a straight edge, two points (start and end) are sufficient. For a curved or irregular edge, multiple points may be necessary.

B. Output Processing

- Generate frame-level JSON output containing the retrieved coordinates and any relevant metadata (e.g., confidence score, detection method).
- Include annotated images with the detected desk edge highlighted and debug states that illustrate the detection process.

C. Basic UI Use Streamlit or Gradio or any other UI frameworks of your choice:

- Allow image upload.
- Display:
    - The original image with the detected desk edge plotted.
    - The retrieved coordinates in a readable format.
    - Any relevant debug visualizations.

D. Include test cases: Use images where the person-facing desk edge is clearly visible. Include:

- A standard rectangular desk or corner desk or curved desk.
- A desk with some common occlusions (e.g., monitor, keyboard).
- A desk with varying lighting conditions.

Validate that:

- The desk edge detection is accurate.
- The retrieved coordinates consistently represent the identified edge.
- Edge cases (e.g., complex desk shapes, severe occlusions) are reasonably handled.

Submission Guidelines:

- Source code with setup instructions.
- Clearly documented logic and assumptions.
- Development environment and dependency list.
- Explanation of:
    - Model or algorithm choice for desk edge detection.
    - How the person-facing edge was distinguished from other edges.
    - Challenges in coordinate retrieval and plotting.
    - Suggestions to improve accuracy.
- Optional: deploy UI on Hugging Face Spaces or Vercel.

Evaluation Criteria:

- **Accuracy of Detection and Coordinate Retrieval**: How well the solution identifies and plots the desk edge, and the precision of the retrieved coordinates.
- **Test Cases Implemented**: Test cases written to validate the detection and coordinate retrieval across various scenarios.
- **Creativity and Implementation Approach**: Use of innovative methods or techniques to solve the problem.
- **User Interface**: Clear and user-friendly presentation of the analysis results.
- **Documentation and Explanation**: Quality of the explanation of the algorithms, development process, and choices made during the assignment.

Sample images: