Project Overview

AI Pipeline for Image Segmentation and Object Analysis

Project Objective:

The goal of this project is to develop an AI pipeline that processes an input image to segment, identify, and analyze objects within the image. The pipeline then outputs a summary table with mapped data for each object.

Pipeline Steps and Deliverables:

1. **Image Segmentation:**
    - **Task:** Segment all objects within an input image.
    - **Deliverables:**
        - Implement or use a pre-trained model (e.g., Mask R-CNN, DETR).
        - Code to input an image and output segmented regions.
        - Visual output showing segmented objects.
2. **Object Extraction and Storage:**
    - **Task:** Extract each segmented object and store them with unique IDs.
    - **Deliverables:**
        - Code to extract and save objects as separate images.
        - Assign unique IDs for objects and a master ID for the original image.
        - Save object images and metadata in a file system or database.
3. **Object Identification:**
    - **Task:** Identify and describe each object.
    - **Deliverables:**
        - Implement or use a pre-trained model (e.g., YOLO, Faster R-CNN, CLIP).
        - Code to generate descriptions for each object.
        - Document containing identified objects and descriptions.
4. **Text/Data Extraction from Objects:**
    - **Task:** Extract text or data from each object image.
    - **Deliverables:**
        - Implement or use a pre-trained model (e.g., Tesseract OCR, EasyOCR).

        - Code to extract and store text/data from each object.
        - Document containing extracted text/data.

5. **Summarize Object Attributes:**
    - **Task:** Summarize the nature and attributes of each object.
    - **Deliverables:**
        - Code to generate a summary of attributes for each object.
        - Document containing summarized attributes.
6. **Data Mapping:**
    - **Task:** Map all extracted data and attributes to each object and the master input image.
    - **Deliverables:**
        - Code to map unique IDs, descriptions, text/data, and summaries.
        - Data structure (e.g., JSON, database schema) representing the mapping.
7. **Output Generation:**
    - **Task:** Output the original image with a table containing all mapped data for each object.
    - **Deliverables:**
        - Code to generate the final output image with annotations.
        - Table summarizing all mapped data.
        - Final visual output showing the original image with segmented objects and accompanying table.

Page 2: Implementation Details

Folder Structure:

kotlin

Copy code

project_root/

├── data/

| ├── input_images/

| ├── segmented_objects/

| └── output/

```
├── models/
|   ├── segmentation_model.py
|   ├── identification_model.py
|   ├── text_extraction_model.py
|   └── summarization_model.py
├── utils/
|   ├── preprocessing.py
|   ├── postprocessing.py
|   ├── data_mapping.py
|   └── visualization.py
├── streamlit_app/
|   ├── app.py
|   └── components/
├── tests/
|   ├── test_segmentation.py
|   ├── test_identification.py
|   ├── test_text_extraction.py
|   └── test_summarization.py
├── README.md
├── requirements.txt
└── presentation.pptx
```

Key Components:

1. **Preprocessing:**
    - Load and preprocess images.
    - Resize and convert images to appropriate formats for model input.
2. **Models:**
    - Implement or integrate models for segmentation, identification, text extraction, and summarization.
    - Use pre-trained models where applicable.
3. **Postprocessing:**

    - Process model outputs to generate visual results.
    - Save metadata and generate annotated images.
4. **Data Mapping:**
    - Map extracted data to corresponding objects and the master image.
    - Structure the data in a JSON or database format.
5. **Visualization:**
    - Generate visual representations of segmented objects and final outputs.
    - Display metadata and summaries in a user-friendly format.
6. **Streamlit App:**
    - Provide a user interface for uploading images and viewing results.
    - Allow users to interact with each step of the pipeline.

Project Requirements:

- Ensure code is well-documented and modular.
- Provide test cases to verify functionality.
- Create a presentation summarizing the project approach, implementation, and results.
- Implement a Streamlit UI for easy interaction with the pipeline.

Name- Kartik Tiwari

Email-kartikdtiwari555@gmail.com

Mob no. - +918112534516