Computer Vision Project Documentation

Introduction

This project is focused on computer vision tasks implemented using Python and the tkinter library for the graphical user interface (GUI). The project includes functionalities for image processing such as edge detection, image stitching, and applying morphological operations. It utilizes libraries such as OpenCV, NumPy, Matplotlib, and PIL (Pillow) for image processing and visualization.

Files Overview

The project consists of two main files:

Main GUI File (`FinalProject.py`):

- This file contains the main GUI application implemented using tkinter.
- It provides a user-friendly interface for performing image processing tasks and displaying results.
- Features include image selection, edge detection using Canny and DoG methods, image stitching, and AI-based human edge detection (Which I did not have enough time to complete).

Canny Class File (`tester.py`):

- This file defines the `CannyApp` class responsible for managing image processing functionalities.

- It uses OpenCV for image processing tasks such as applying morphological operations (e.g., Canny edge detection) and displaying processed images on the GUI.

Dependencies

The project requires the following libraries to be installed:

- tkinter (for GUI)
- OpenCV (cv2)
- NumPy
- Matplotlib
- PIL (Pillow)

These dependencies can typically be installed using package managers such as pip.

Usage Instructions

Running the Application:

- Execute the main GUI file (`FinalProject.py`) to launch the application.
- The GUI will provide buttons for various image processing tasks.

Edge Detection:

- The "Edge Detection" button offers options for edge detection using Canny or DoG (Difference of Gaussians) methods.
- Adjusting Kernel Size:
  - Use the slider labeled "Kernel Size" to adjust the size of the kernel used in morphological operations.
  - After selecting images and adjusting the kernel size, click the "Process Images" button to perform edge detection.
- Upon selecting a method, the user can choose an image for edge detection.

Image Stitching:

- The "Stitching" button allows the user to select two images for stitching.
- The selected images will be stitched together using OpenCV's stitching functionality.
- The stitched image will be displayed alongside the original images.

AI-based Human Edge Detection (Planned):

- My application includes a button for AI-based human edge detection (not yet implemented).

Thanks

By : Zainab Atwa