

# Project Document: AI-Enhanced Image Stitching and Edge Detection

## Goal

The goal of this project is to develop an application that allows users to select a group of images, which the application will then stitch together into a single panoramic image. This stitched image will undergo multiple edge detection processes using different techniques: Canny Edge Detection, Difference of Gaussians (DoG) with an adjustable morphological operation, and an AI-based edge detector specifically tuned to recognize human figures with a confidence level of more than 50%. The application will provide an interactive interface for users to adjust parameters and visually compare the results of different edge detection techniques.

## Requirements

### Functional Requirements:

#### **Image Selection and Stitching:**

- The application must include a user interface (UI) component to allow users to select a group of images from their computer.
- Stitch the selected images into a single panoramic image.
- Display both the individual images and the stitched panoramic image within the application on a first window.

#### **Edge Detection Implementation:**

- Apply Canny Edge Detection to the stitched image and display the result.
- Implement Difference of Gaussians (DoG) edge detection followed by a morphological operation to clean the results.
- Provide a user interface component (e.g., a slider) to adjust the kernel size of the morphological operation. The kernel shape can be any form selected by the developer.
- Display both results and the slider on a second window.

### **AI-based Human Edge Detection:**

- Implement an AI-based object detection model capable of identifying human figures within the stitched image.
- The model should filter and display only those detections with a confidence level above 50%.
- The results should be visually represented on the stitched image, clearly marking detected human figures on a third window.

### **User Interface:**

- Develop a simple, intuitive user interface that allows users to upload images, view processed results, and adjust Morphological Operations' parameters dynamically.

## **Non-functional Requirements:**

**Performance:** The application should process images and perform edge detection efficiently, ensuring a responsive user experience.

**Usability:** The interface should be user-friendly, with clear instructions and feedback for users to easily navigate and use the application's features.

**Scalability:** The design should accommodate future enhancements, such as the addition of new edge detection algorithms or support for larger image sets.

## **Marking Criteria**

### **1. Functionality (40%)**

- Image Stitching: Successful stitching of multiple images into a panoramic image. (10%)
- Edge Detection Accuracy: Accurate edge detection using Canny, DoG with morphological operations. (10%)
- The AI-based detection's ability to correctly identify human figures. (10%)
- Parameter Adjustment: Effective implementation of adjustable parameters for DoG morphological operations and their impact on edge detection results. (10%)

### **2. User Interface and Experience (20%)**

- Ease of Use: Clarity and intuitiveness of the user interface. (10%)
- Interactivity: Smoothness and responsiveness of parameter adjustments and image processing. (10%)

### 3. Code Quality and Documentation (20%)

- Readability: Clear, well-commented, and organized code. (10%)
- Documentation: Comprehensive documentation, including comments in code, usage instructions, and a brief overview of algorithms used. (10%)

### 4. Innovation and Creativity (20%)

- Algorithm Implementation: Creativity in the application of edge detection algorithms and the integration of the AI model for human detection. (10%)
- UI/UX Design: Originality and effectiveness of the user interface design, especially in facilitating user interaction with the application's features. (10%)

## Deliverables

- Source code of the application.
- A report documenting the design decisions, algorithms used, challenges encountered, and solutions implemented.

## Submission Deadline

All deliverables must be submitted by **October 17, 2024**

## Additional Notes

- Students are encouraged to use open-source libraries such as OpenCV for image processing and PyTorch or TensorFlow for implementing the AI-based edge detection.
- While implementing the morphological operation, consider exploring different shapes (e.g., rectangular, elliptical) and their impacts on edge detection results.
- For the AI-based human detection, pretrained models such as YOLO (You Only Look Once) or SSD (Single Shot MultiBox Detector) can be utilized for the specific task of human detection.
- Documentation of the code and the project process is crucial. Ensure to comment your code adequately and maintain a project log that details your design decisions, challenges faced, and how they were overcome.