Date: 12.06.2024

Lab Project Report

Note: You can access the code via this GitHub repository.

Introduction:

The provided code consists of classes and their associated functions related to a user interface for image operations. These classes facilitate various image processing tasks such as conversion, edge detection, and segmentation, among others.

Objective:

The objective of these classes and functions is to provide a user-friendly interface for performing different image operations efficiently. These operations include but are not limited to editing, saving, exporting, and applying various filters to images.

Class Functionalities:

1. **UI_Interface Class:**
    - Manages the user interface for the application.
    - Inherits functionalities from the Image_Operations class.
    - Handles initialization, button states, visibility, and interactions.
    - Provides methods for loading, saving, exporting images, and performing various image operations.

2. **Image_Operations Class:**
    - Handles image-related operations and functionalities.
    - Inherits functionalities from the Image class.
    - Manages source and output images.
    - Provides methods for undoing, redoing modifications, and performing image processing actions such as conversion, edge detection, and segmentation.
3. **Image Class:**
    - Represents an image object.
    - Provides basic functionalities for image handling.
    - Stores image data, manages image channels, and provides methods for converting and saving images.

Name Surname: Umut Deniz
Student ID: 151220192021

Date: 12.06.2024

Class Hierarchy:

- **UI_Interface** inherits from **Image_Operations**:
    - **UI_Interface** extends the functionality of **Image_Operations** by adding user interface components and interaction handling.
    - This means that **UI_Interface** has access to all methods and properties of **Image_Operations**, in addition to its own.
- **Image_Operations** inherits from **Image** (Composition):
    - **Image_Operations** builds upon the basic image handling capabilities provided by **Image**.
    - This allows **Image_Operations** to perform more complex image processing tasks using the foundational methods from **Image**.
- **Image**:
    - Serves as the base class that provides fundamental image handling functionalities.
    - This class encapsulates the basic operations related to storing, setting, and retrieving image data, and serves as the foundation for the other classes.
- **Progress_Show_Function:**
    - An external component for displaying progress during lengthy operations.
    - Provides a function or module to show progress bars or status indicators.
    - Enhances user experience by providing feedback on the progress of operations4

Essential Dependencies:

1. **NumPy (np):**
    - Used for numerical operations on image data.
    - Provides support for large, multi-dimensional arrays and matrices.
2. **PyQt5 (or similar library):**
    - Used for creating the graphical user interface.
    - Provides classes for widgets and other UI components.
3. **OpenCV (cv2) or Similar Image Processing Library:**
    - Used for image processing tasks like conversion, edge detection, and segmentation.
    - Provides a wide range of functions for image manipulation.

Name Surname: Umut Deniz
Student ID: 151220192021