151248620: OBJECT-ORIENTED PROGRAMMING II
2022-2023 SPRING SEMESTERS
LAB FINAL
Due: June 7, 2023, Wednesday, 17:00

Remarks:

- Module code should be cleverly commented
- You should write a report explaining the design and implementation issues as well as anything a person may need to know while testing your program
- You are expected to make a good design and exploit object-oriented programming principles (abstraction, inheritance, exception handling, etc.) as often as possible.
- You are expected to make a good design and exploit built-in data structures (list, dictionary, and set) and libraries (numpy, pandas, and matplot) as often as possible.
- For lab final, you need to send the project workspace including your .py files. You must organize all this material in a single .zip or .rar archive, name the file as "yourID_Name_ProjectID.zip/rar", and load ESOGU UZEM.
- Grades will be given according to
    - 50% Class design and availability of object-oriented programming principles,
    - 15% code quality (commented and well formatted),
    - 15% documentation,
    - 20% functionality (implemented and correctly running functionalities).

In this assignment, you will design an interface to perform some operations on an image. An example interface is given in Figure 1. **It is important to note that the given interface is an example, and you have to design your interface.** In your backhand codes, you have to use the scikit-image library (Please visit the website: https://scikit-image.org/). You do not need to download and install the scikit-image library if you use WinPython distribution, which is used in our course. You can easily check that the library is available in your system by typing import skimage in interactive mode.

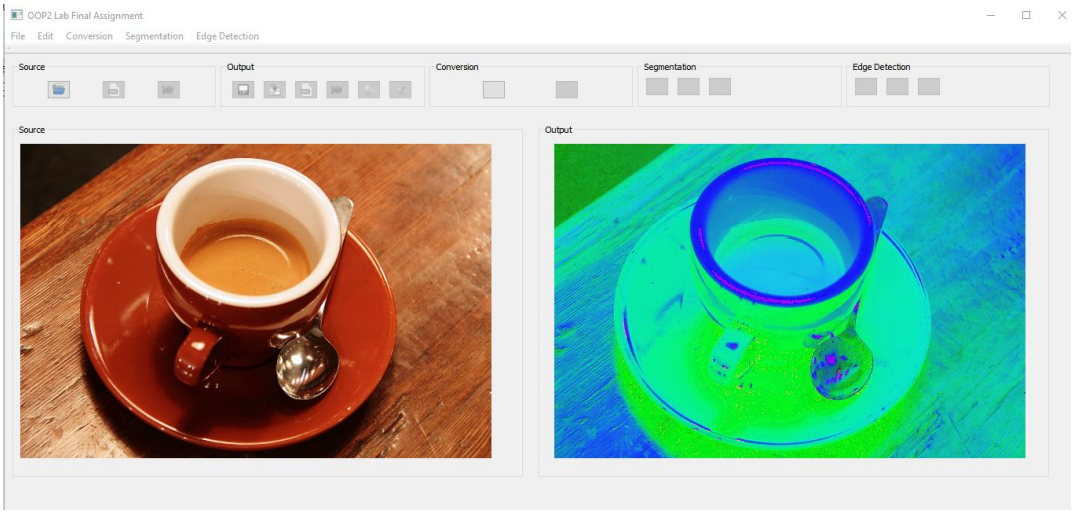
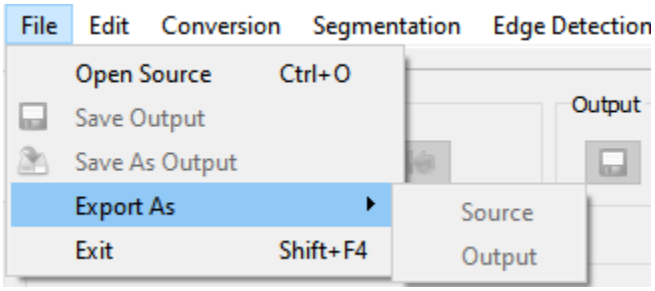


Figure 1. An example interface.

Your design must meet the following criteria:

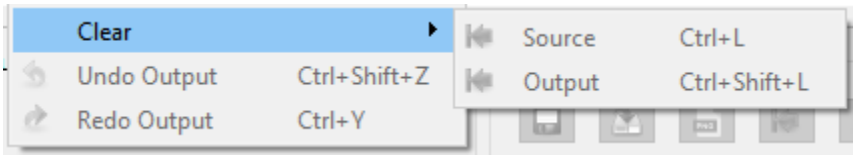
- As seen from the figure, the interface must include two parts: One for source and the other for output.
- The functionalities must be collected under five menus: File, Edit, Conversion, Segmentation, and Edge Detection. Also, each functionality must have a button. You can use toolbar or group boxes for that purpose.
- Each functionality must have an appropriate shortcut, status bar tip, and icon.
- Do not use default object names that are given in Qt Designer. Each object in the interface must have an appropriate and unique name.

File Menu



In the file menu, there are five functionalities: Open Source, Save Output, Save As Output, Export as for both Source and Output, and Exit. In the beginning, all functionalities except Open Source must be disabled. When the user selects Open Source functionality in the File menu or clicks the corresponding button, a dialog window must be opened. The user can only open the files with .jpg and .png extensions. After the source file is opened, functionalities and also corresponding buttons for Conversion, Segmentation, and Edge Detection menus must be enabled. At this point, Export as functionality for the Source part must also be enabled. Save Output, Save As Output must be enabled after an operation is performed from Conversion, Segmentation, and Edge Detection menus. Save Output functionality save the file with the same name and extension in the same folder. Save As Output functionality must open a dialog window, and the user can alter the name and current folder of the output. Lastly, Export as functionality must change the type of the file. For example, the file with .jpg extension must be saved as .png extension and vise versa.

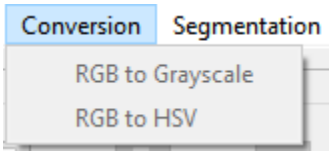
Edit Menu



In the edit menu, there are three functionalities: Clear for both Source and Output, Undo Output, and Redo Output. In the beginning, all functionalities must be disabled. When the user opens a source file Clear Source functionality must be enabled. If the user selects Clear Source functionality in the Edit menu or clicks the corresponding button, Clear Output functionality must also be invoked. Undo Output and Redo Output must be enabled after an

operation is performed from Conversion, Segmentation, and Edge Detection menus. You have to implement a polymorphic class hierarchy for undo and redo functionalities.

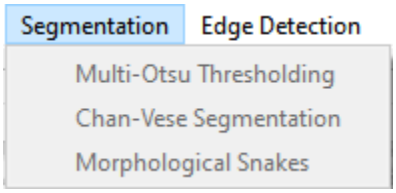
Conversion Menu



In the conversion menu, there are two functionalities: RGB to Grayscale and RGB to HSV. To implement these functionalities, please visit the website: https://scikit-image.org/docs/stable/auto_examples/index.html. Use the coffee image, which is given in Figure 2, to test the functionalities. After an operation is employed, the result must be shown in the Output part.

Figure 2. The coffee image (You can import data module to use the image).

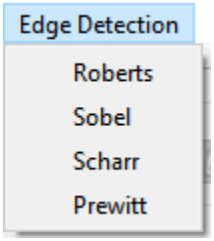
Segmentation Menu



In the segmentation menu, there are three functionalities: Multi-Otsu Thresholding, Chan-Vese Segmentation, and Morphological Snakes. To implement these functionalities, please visit the website: https://scikit-image.org/docs/stable/auto_examples/index.html. Use the camera image, which is given in Figure 3, to test the functionalities. After an operation is employed, the result must be shown in the Output part.

Figure 3. The camera image (You can import data module to use the image).

Edge Detection Menu



In the edge detection menu, there are four functionalities: Roberts, Sobel, Scharr, and Prewitt. To implement these functionalities, please visit the website: https://scikit-image.org/docs/stable/auto_examples/index.html. Use the horse image, which is given in Figure 4, to test the functionalities. After an operation is employed, the result must be shown in the Output part.

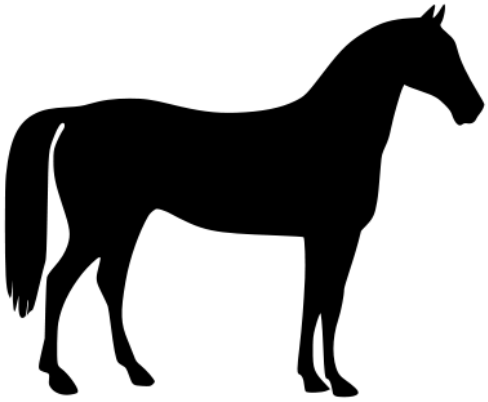


Figure 4. The horse image (You can import data module to use the image).