151228620/151248620: OBJECT-ORIENTED PROGRAMMING II

2021-2022 SPRING SEMESTERS

LAB FINAL Report

151220172030: Heval Can Aslan ÖZEN

Overview:

When we run the program, a screen consisting of 1 menu bar and 3 main headers appears (*Figure1*). All operations in the headers are also available in the menu bar. The first of these three header is the 'source' header (*Figure3)*, where the source file will be retrieved (*Figure4*), and some basic operations will be performed on this imported file. The second header is the 'operations' header, which contains the main features of the program. It contains the operations that allow us to make changes to the imported source file and show the result under the next heading. Final header the 'output' header contains the final version of the modified source file and a few simple actions.

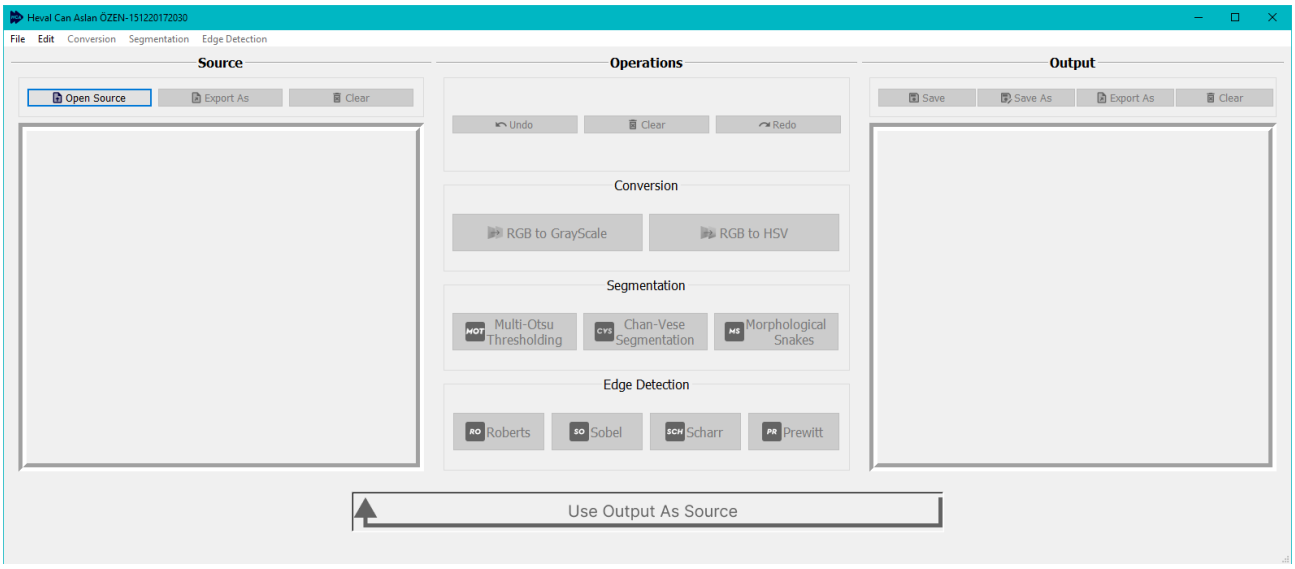

Figure1.

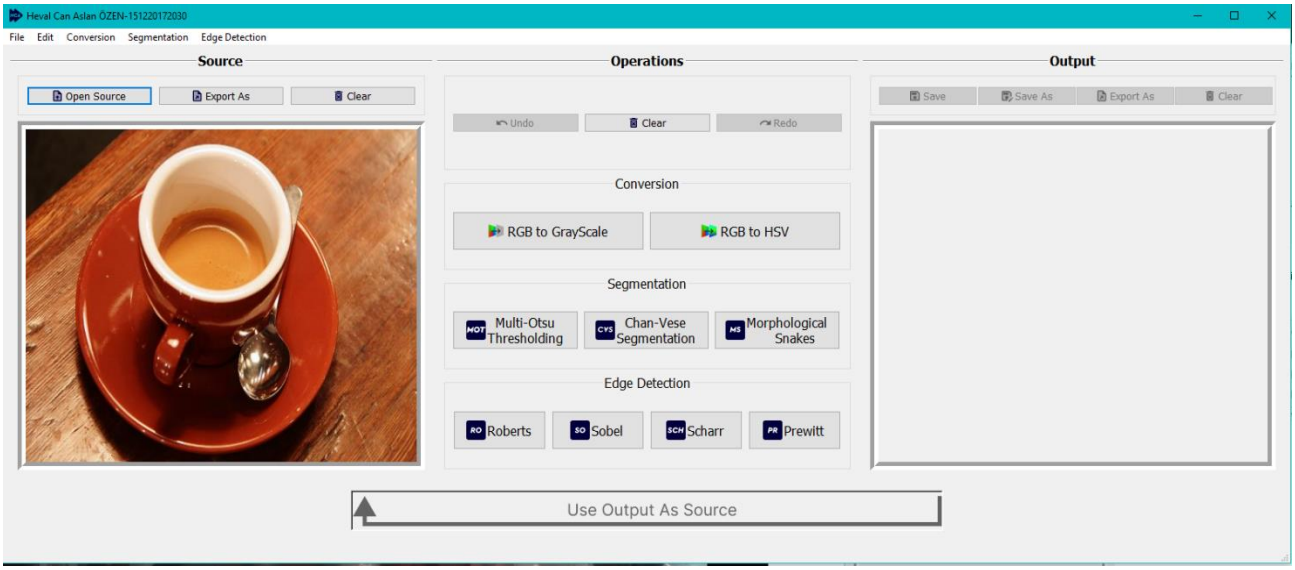

Figure2.

With the first run, only the exit and open-source buttons are available (*Figure1)*. Other operations become available once the user selects the source file (*Figure2)*.

Figure3.

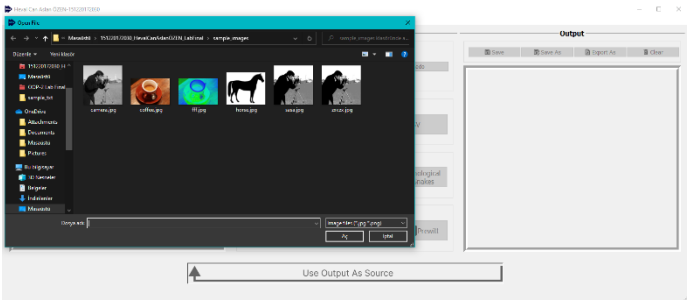

Figure4.

Under the 'operations' heading, there are ‘undo’, ‘redo’, and ‘clear all’ buttons and conversion, segmentation, and edge detection button groups. If the selected image is not suitable for the operation to be performed, an error message is received (*Figure5)*.

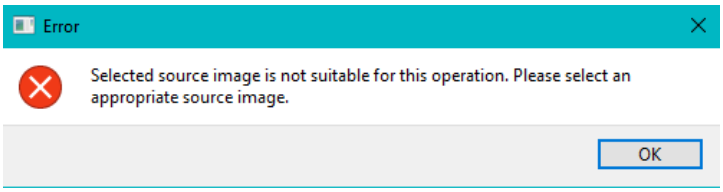

Figure5.

With the selection of a source file, the export as and clear buttons under the 'source' header become available. The Export as button enables saving by changing the extension of the source file. If the source file has .jpg extension, we can save it as .png and vice versa (*Figure6)*.

After any operation is performed, the result is displayed under the output header (*Figure7)* and the buttons under this header become available. In addition, because an operation is implemented, the undo button under the operations header becomes available.

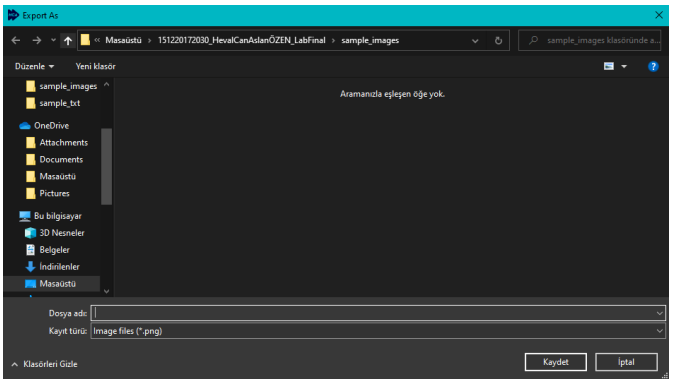

Figure6.

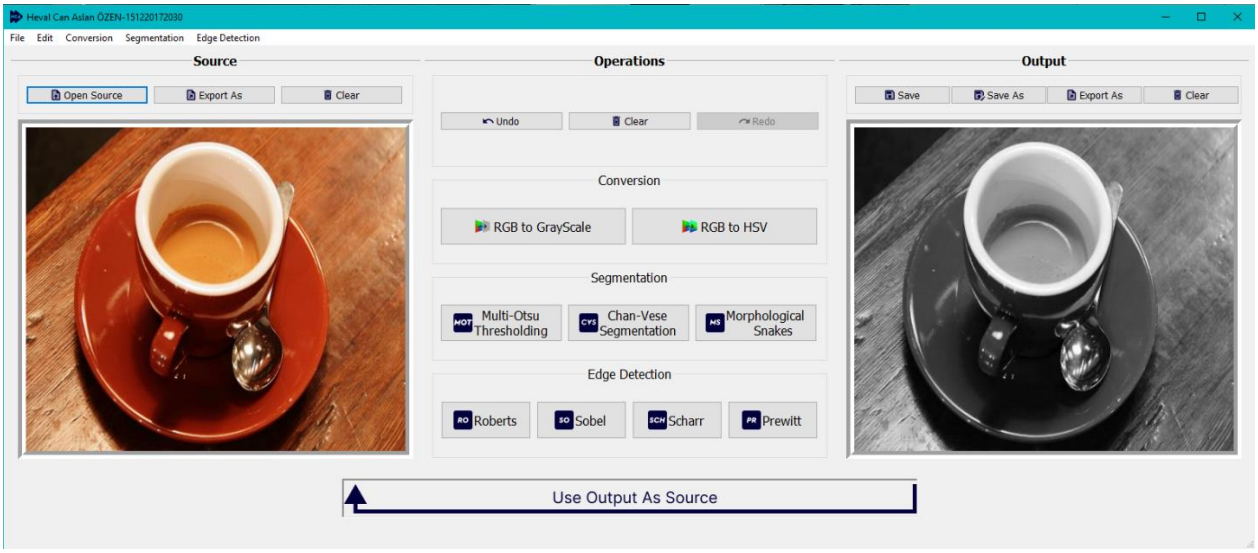

Figure7.

If the user wants to exit the application without performing any output recording operation after performing an operation, it is asked whether it is desired to save when exiting (*Figure8)*. If a save operation under the output header has been applied, it will be asked whether to exit the application (*Figure9)*.

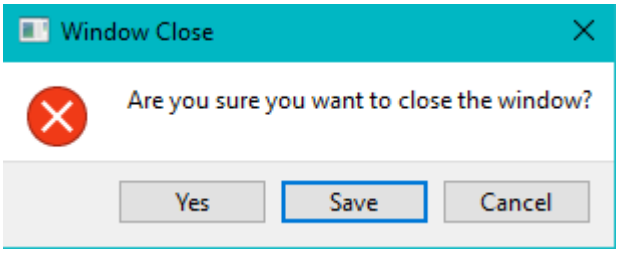

Figure8.

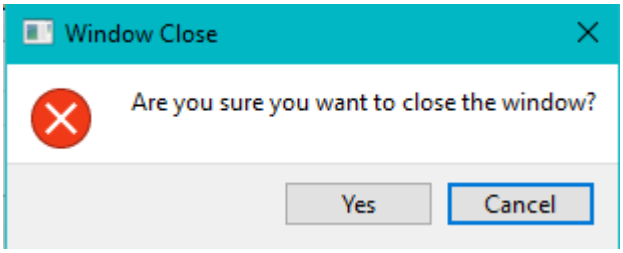

Figure9.

Menubar:

All operations that can be performed at that time are available in the menu bar. operation names as well as shortcut keys can be seen here (*Figure10)*.

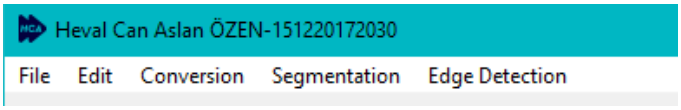

Figure10.

Buttons and Button Groups:

When you hover over the buttons, a more detailed description of that button appears at the bottom left (*Figure11)*.

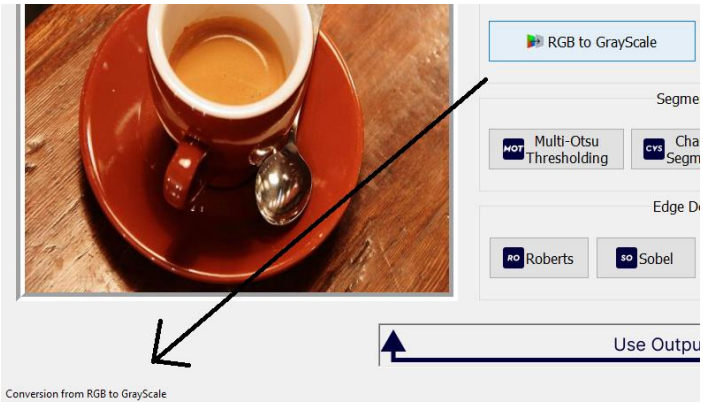

Figure11.

Source button group:

Under this header (*Figure12)*, there is an 'open source' button to select the source file, an 'Export As' button to save the source file with a different extension, and a 'clear' button to delete the source file.

Figure12.

Operations button group:

This header has 3 buttons and 3 sub-headers (*Figure13, Figure14, Figure15and Figure17)*. The buttons are respectively; the 'undo' button which is not available unless any operation is performed, the 'Clear' button which is not available unless a source is selected, and the 'redo' button which is not available unless the 'undo' button is used once. The 'undo' button allows to undo the last performed operation. The 'redo' button re-executes the last undone operation, and the 'clear' button deletes the source file and the output image, if any.

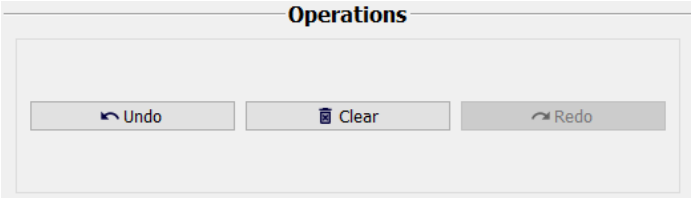

Figure13.

There are 2 buttons under the Conversion sub-header (*Figure14)*. These buttons are 'RGB to GrayScale' and 'RGB to HSV' respectively. If the source file is not an rgb image, these operations reflect a warning to select an appropriate image.

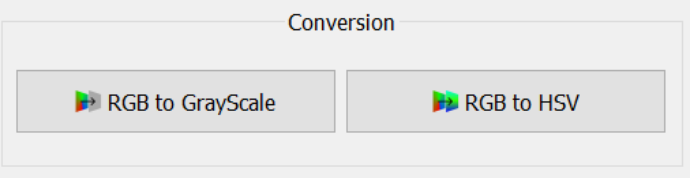

Figure14.

There are 3 buttons under the segmentation sub-header (*Figure15)*. These buttons are 'Multi-Otsu Thresholding', 'Chan-Vese Segmentation' and 'Morphological Snakes' respectively. If the source file is not suitable for these operations, these operations reflect a warning to select an appropriate image. In addition, if the operation will take a long time, they reflect a warning to the user to wait for it (*Figure16)*.

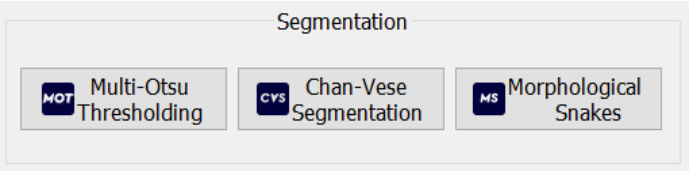

Figure15.

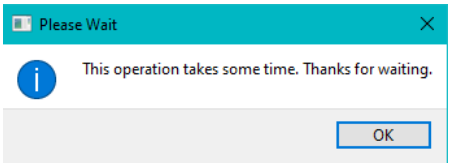

Figure16.

There are 4 buttons under the edge detection sub-header (*Figure17)*. These buttons are 'Roberts', 'Sobel', 'Scharr' and 'Prewitt' respectively. If the source file is not suitable for these operations, these operations reflect a warning to select an appropriate image.

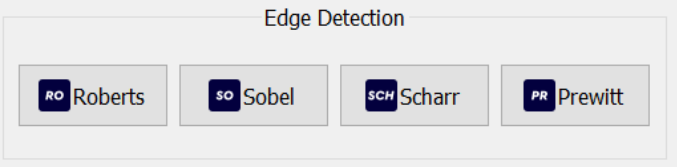

Figure17.

Output button group:

The buttons under this heading (*Figure18)* only operate on the output image. The 'Save' button saves the output image with the changes made over the source file. The 'Save As' button allows to save the output image as a different file. 'Export As' button saves the source file with a different extension, and 'Clear' button delete the output image.

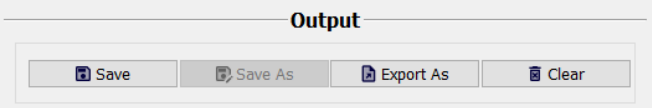

Figure18.

Use Output as Source button:

This button (*Figure19)* takes the output image to the source section to use as a source file without the need to save it. Every operation after this operation affects the new resource file.

Figure19.

Implementation Issues:

Undo and Redo:

In order to enable the use of undo and redo buttons, a label had to be kept for each operation performed and the operation for this label had to be repeated according to the undo or redo operation. For this, two list; called history[] and last[], and history_position were defined to determine our place in this list. Provided that the redo operation is false every time an operation is performed, the name of the operation is added to this history list and the position is decreased by 1. The redo button is not accessible until an undo operation is performed, as a redo operation cannot be performed without any undo operation. When an undo operation is performed, the history_position is decremented by one, the last undone operation is removed from the history[] and added to the last[], and the previous operation is called back with the eval function. If a redo is done, all operations on last[] are appended to the end of history[], starting with the last one added to last[], and then the last[] is cleared. The history_position is increased by one and the operation at that position is called with eval. and so on again.

Check Repetition:

As repetitively applying the same operation to a resource will not yield any different results, repeated operations are blocked to avoid this. If the current function name in history[] is the same as the previous function name, the last is removed from history[].

Ask before Exit:

If a source file has been taken and an operation has been performed, the buttons containing the save options for the output image become active. When any save operation takes place, the button of that transaction becomes deactivated and remains in the state until a new operation is performed. By evaluating these conditions, if an operation has been performed and no save has been made, the user is asked whether he wants to save before exiting the application. Otherwise it will only ask if it wants to Exit.