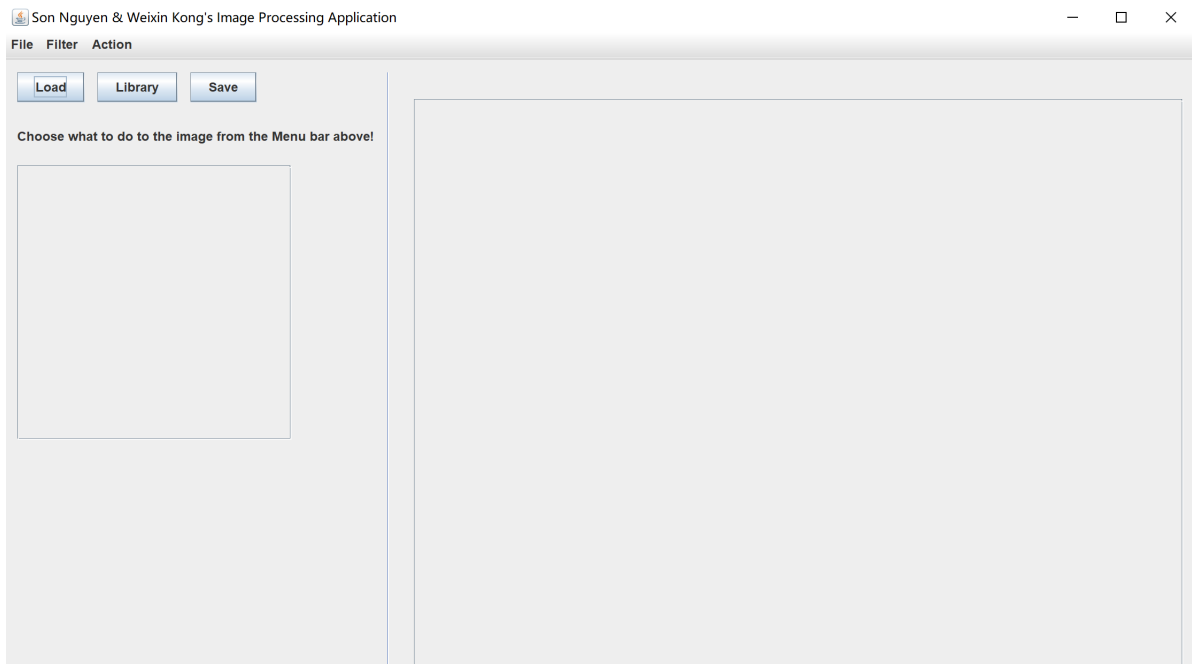


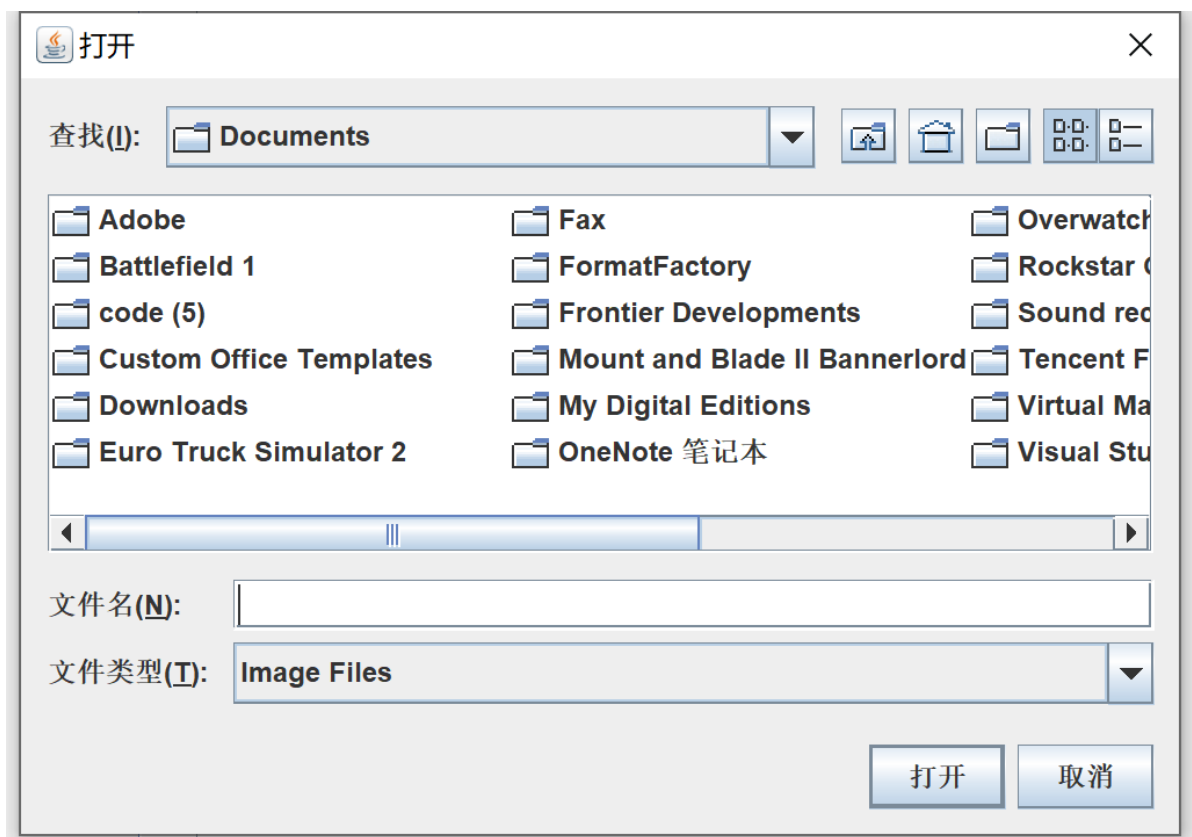
ImageProcessing - USEME

Instruction for GUI



This is the entering image of the GUI. Most commands are contained in the bar at the top of the window. Load, Library and Save are both contained in the bar (in File part) and on the window.

Want to load an image, click the Load button.

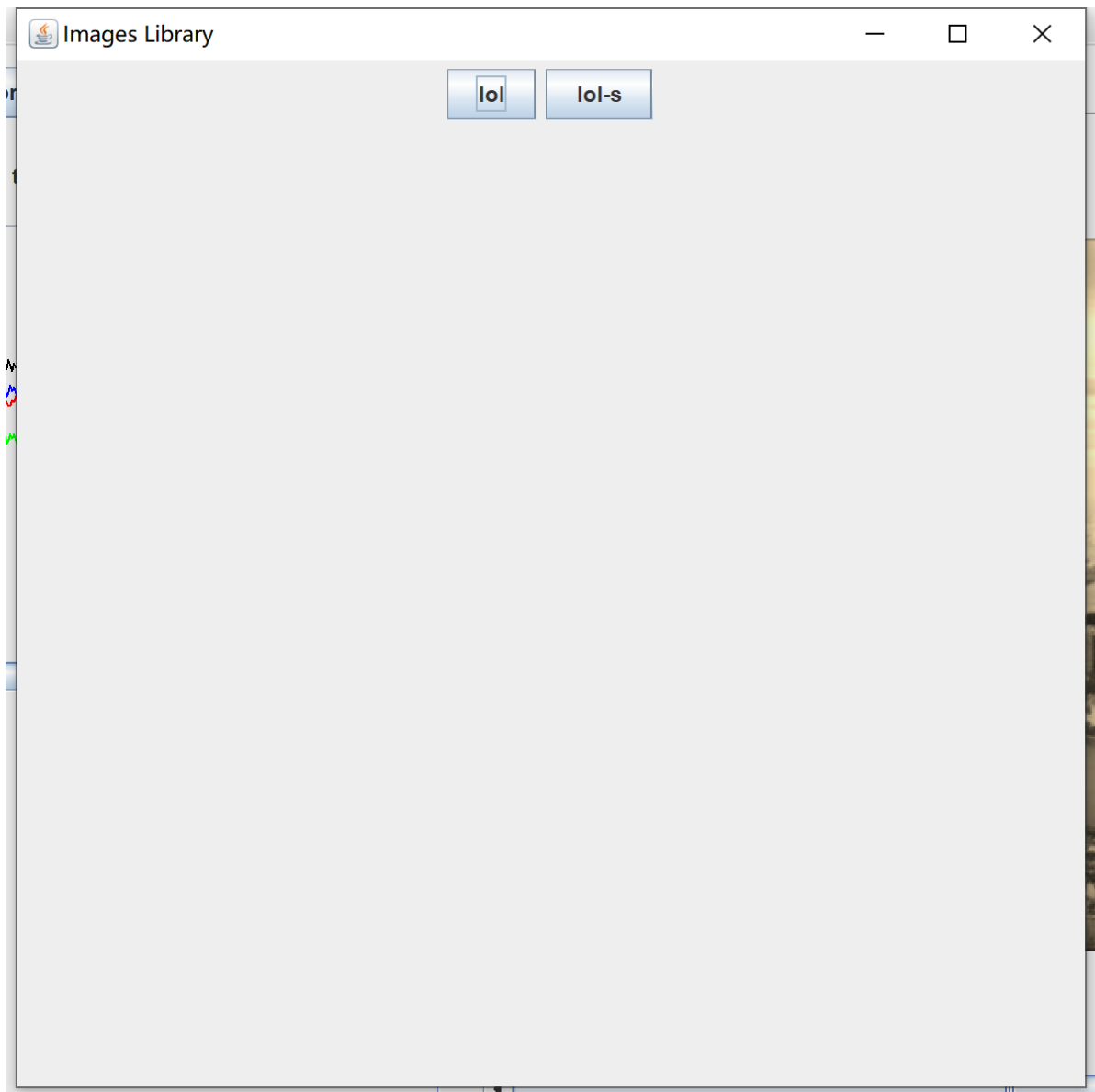


This is the pop-up window of the loading that allows you to choose a file from you System.



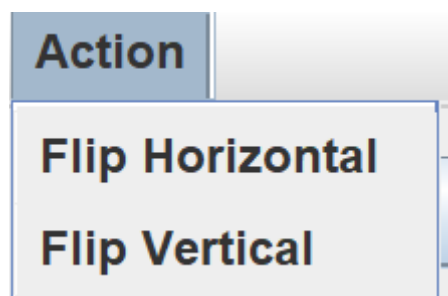
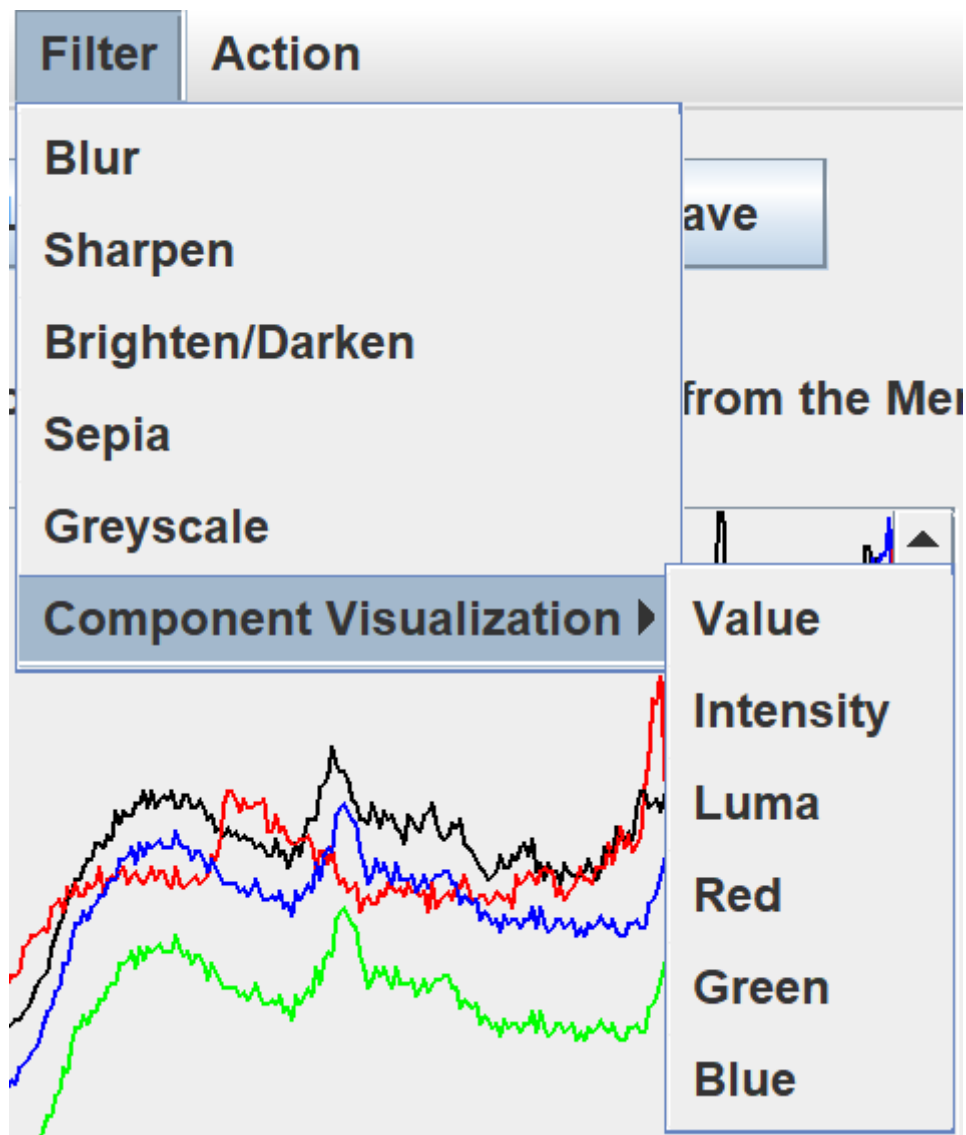
After entering the file, it will ask you to rename the file and store the image with the given name into library.

By clicking Library, we can find out all images we store (including image with changes).

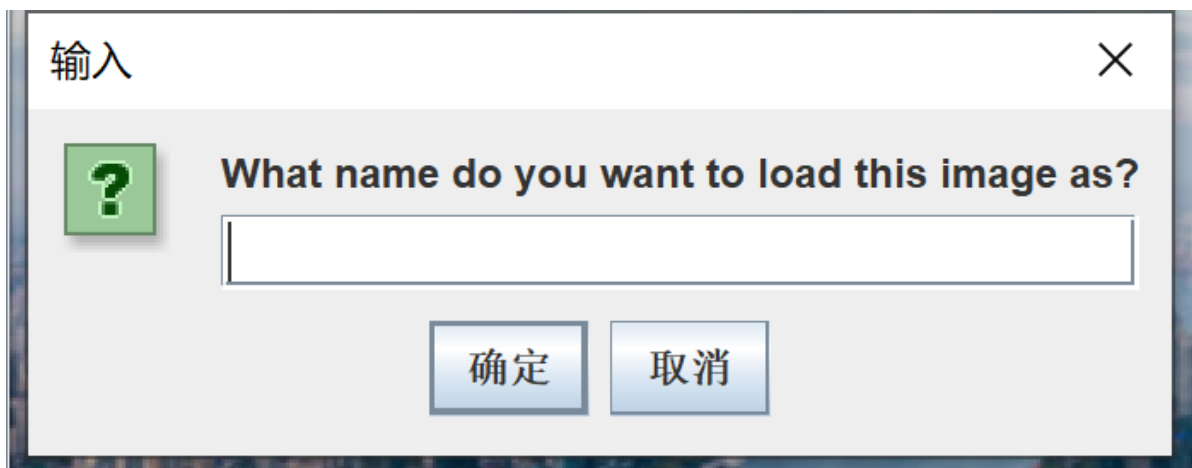


By clicking on the name of the button, it will change the image display on the window.

All changes to the image are put in the Filter and Action bar.

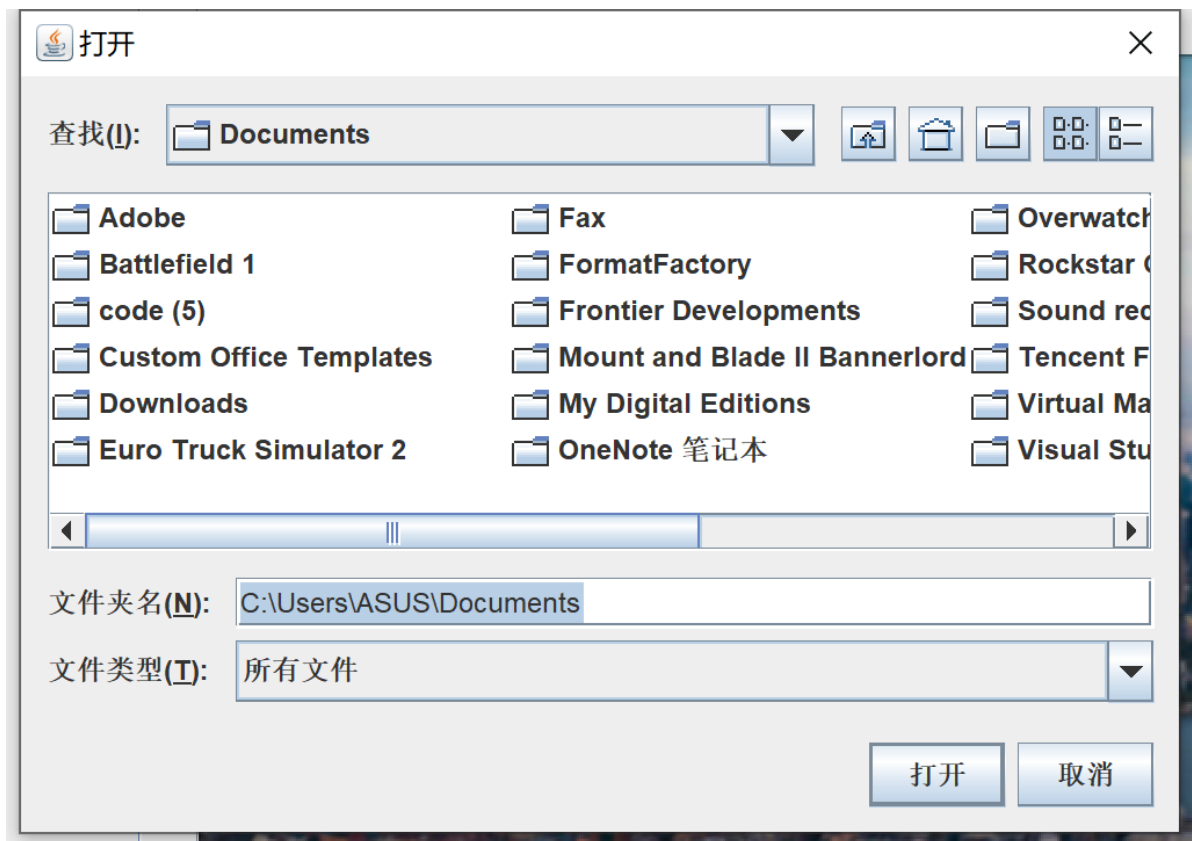


By clicking each button (changing functions), it will pop-up a window to ask you to rename it and store it in the library.

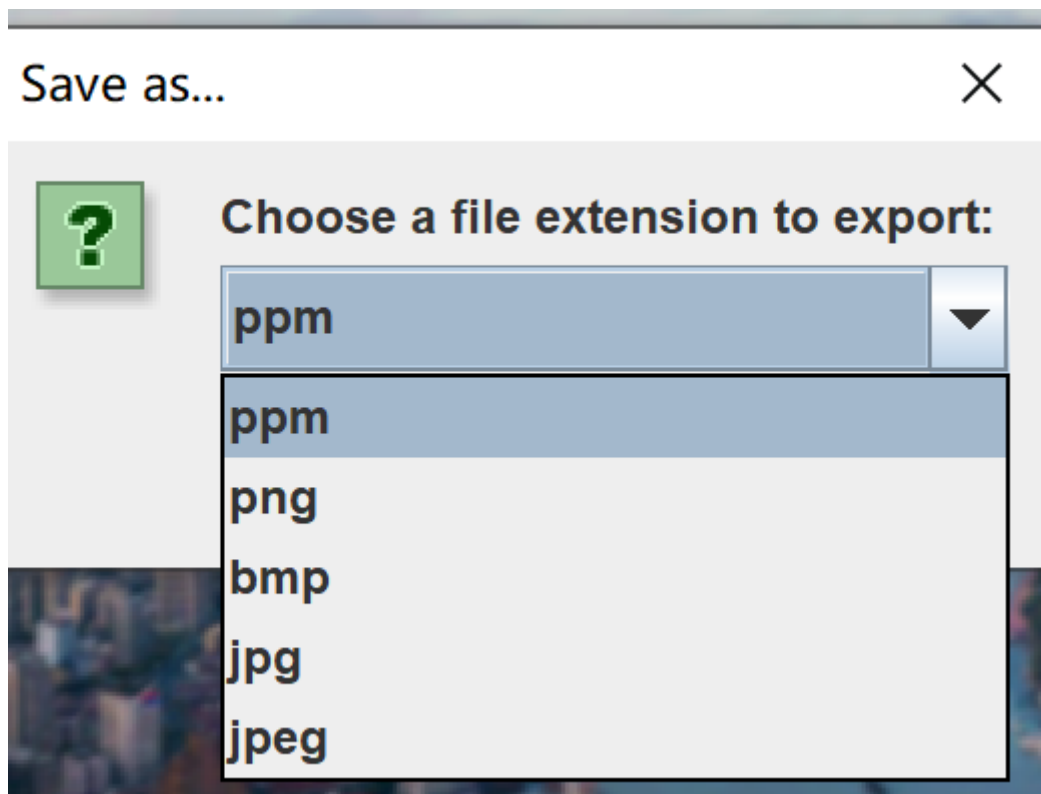


To Save a file, select the image you want to save in the library, and click save button.

This will pop-up a window to enable the user to locate the file saving path by opening a file package.



Then, user will be able to choose the file format output they want.



After confirming it, the GUI will save the file as given information .

Instruction for script using:

Step 1: Taking the given script.txt file.

download the script.txt to your system.

Step 2: Edit the Configuration

Go to the ImageProcess File. And find the "Edit Configurations."
In the box of Program Arguments, we have few options to type in.

1. To run the given script by entering
 - file [The path for script.txt]
 - f [The path for script.txt]
2. To get helping information
 - help / -h
3. To run own command lines in the terminal from IDEA
 - text
4. To run GUI frame by calling nothing
 - [Input Nothing, !!!!!!! Leave it blank]

Error Handling:

Put all commands that are not included above will receive a error message.

Step 3: Run the ImageProcess

Run the ImageProcess.

1. For using the script.txt file input, it will do these commands.
 - a. Load the given ood.ppm image as the name ood stored in the library
(!!! The path needs to be changed to the users' path of the ood.ppm
The given path is like this for windows on my laptop
C:\Users\ASUS\Desktop\ImageProcessing\Assignment4\res\ood.ppm)
 - b. Brighten the ood by 50 and save as ood-brighter.
 - c. Greyscale the ood as blue-component and save as ood-blue.
 - d. Make the ood as sepia-filter and save as ood-sepia.
 - e. Sharpen the ood save as ood-sharpen.
 - f. Save the ood-blue to a output path as ppm file.
 - g. Save the ood-blue to a output path as png file.
 - h. Save the ood-sepia to a output path as jpg file.
 - i. Save the ood-sharpen to a output path as ppm file.
 - j. Load the given oodjpg.jpg as fun.
 - k. Brighten the fun by 50 and save as fun-brighter.
 - l. Greyscale the fun as red-component and save as fun-red.
 - m. Flip vertically the fun-red and save as fun-v.
 - n. Make the fun as sepia-filter and save as fun-sepia.
 - o. Blur the fun for five times and save as fun-blur.
 - p. Save the fun-blur to a output path as jpg file.
 - q. Save the fun-sepia to a output path as png file.
 - r. Save the fun-v to a output path as ppm file.
 - s. Load the given \oodpngtransparent.png as png.
 - t. Brighten the png by 50 and save as png-brighter.
 - u. Greyscale the png as red-component and save as png-red.
 - v. Flip vertically the png-red and save as png-v.
 - w. Make the png as Greyscale(luma)-filter and save as png-greyscale.
 - x. Sharpen the png and save as png-sharpen.

- y. Save the png-sharpen to a output path as png file.
- z. Save the png-greyscale to a output path as png file.
- a1. Save the png-v to a output path as png file.
- a2. Save the png-v to a output path as jpg file.
- a3. Save the png-greyscale to a output path as ppm file.
- 2. For using the self typed command lines in the console or terminal, see the instructions below (Functions Instructions).
- 3. For using the help, the system will return helpful messages.

Functions Instruction:

IMPORTANT!!!

1. The naming of the [name-in-library] cannot be the same with the commands.

2. Including: "load", "brighten", "save", "horizontal-flip", "vertical-flip", "red-component", "green-component", "blue-component", "luma-component", "intensity-component", "value-component", "blur", "sharpen", "greyscale", "sepia".

3. Attempts on naming the image as the name of the commands will cause the fail of the action. !!!

4. Loading images from a system path and add it into the program library.

load [image-path] [name-in-library]

5. Brighten the image and save it with a name into the program library.

brighten [Amount of changing] [Image-name] [Output-Name-In-Library]

6. Vertically flip the image and save it with a name into the program library.

vertical-flip [Image-name] [Output-Name-In-Library]

7. Horizontally flip the image and save it with a name into the program library.

horizontal-flip [Image-name] [Output-Name-In-Library]

8. Use the red component to make a greyscale image and save it with a name into the program library.

red-component [Image-name] [Output-Name-In-Library]

9. Use the green component to make a greyscale image and save it with a name into the program library.

green-component [Image-name] [Output-Name-In-Library]

10. Use the blue component to make a greyscale image and save it with a name into the program library.

blue-component [Image-name] [Output-Name-In-Library]

11. Use the value component to make a greyscale image and save it with a name into the program library.

value-component [Image-name] [Output-Name-In-Library]

12. Use the intensity component to make a greyscale image and save it with a name into the program library.

intensity-component [Image-name] [Output-Name-In-Library]

13. Use the luma component to make a greyscale image and save it with a name into the program library.

luma-component [Image-name] [Output-Name-In-Library]

14. Save the image with an output system path.

save [Image-name] [Output-Path-In-System]

15. Blur the image by kernel projection and save it with a name into the program library.

blur [Image-name] [Output-Name-In-Library]

16. Sharpen the image by kernel projection and save it with a name into the program library.

sharpen [Image-name] [Output-Name-In-Library]

17. Make a greyscale style image by filter and save it with a name into the program library.

greyscale [Image-name] [Output-Name-In-Library]

18. Make a sepia style image by filter and save it with a name into the program library.

sepia [Image-name] [Output-Name-In-Library]