USEME (Assignment 5 and Assignment 6)

Command-line argument:

-file path-of-script-file: the program would open the script file, execute it and shut down. The application only accepts a txt file.
-text: the program would open in an interactive text mode, allowing the user to type the script and execute it one line at a time.
nothing in the command-line input: the program would open the graphical user interface.

All the Commands Supported by Our Application:

- load image-path image-name: Load an image from the specified path by the given image name.
- save image-path image-name: Save the image with the given name to the specified path.
- red-component image-name dest-image-name mask-image-name: Visualize the red-component of the image.
- green-component image-name dest-image-name mask-image-name: Visualize the green-component of the image.
- blue-component image-name dest-image-name mask-image-name: Visualize the blue-component of the image.
- value-component image-name dest-image-name mask-image-name: Visualize the value of the image.
- luma-component image-name dest-image-name mask-image-name: Visualize the luma of the image.
- intensity-component image-name dest-image-name mask-image-name: Visualize the intensity of the image.
- horizontal-flip image-name dest-image-name: Flip an image horizontally.
- vertical-flip image-name dest-image-name: Flip an image vertically.
- brighten increment image-name dest-image-name mask-image-name: brighten the image by the given increment.
- darken decrement image-name dest-image-name mask-image-name: darken the image by the given decrement.

- blur image-name dest-image-name mask-image-name: blur an image.
- sharpen image-name dest-image-name mask-image-name: sharpen an image.
- greyscale image-name dest-image-name mask-image-name: Create a greyscale image with the luma of the image.
- sepia image-name dest-image-name mask-image-name: Convert a normal color image into a sepia-toned image.
- quit: quit the image processor.
- Enter an image name for the mask-image-name if you want to manipulate the image partially. Enter 'none' otherwise.

Usage and Conditions:

You can choose to input commands step by step or use a script file that has all the operations you want to execute, **but the script file has to be in txt format.** In the script file or command inputs, first, you need to load an image first when you start the application. Then, you can use the commands to process the image. The user should input image-name before dest-image-name. The user is able to process an image multiple times. Finally, you can save the image to a specified valid path. We only support the following image formats: png, jpg, bmp, and ppm.

Our application supports partial image manipulation. The user is required to input mask-image-name for those commands that support partial image manipulation. Only load, save, horizontal-flip, and vertical-flip don’t require entry for mask-image-name. The user could input “none” if they don’t want to partially process the image.

Examples:

1. load an image:
    - load a new image (jpg/png/bmp/ppm) to the application:
        - load res/example.png img
        - load res/example.ppm img
        - load res/example.jpg img
        - load res/example.bmp img

```
    - override an image:
        - load res/example2.png img

2. visualize the red-component of an image:
    - the command "red-component" can be in mixed cases:
        - red-component img img-red none
        - RED-component img img-green none
    - override an image:
        - red-component img img none(now img is a red-component
greyscale image)

3. visualize the green-component of an image:
    - the command "green-component" can be in mixed cases:
        - green-component img img-green none
        - grEEN-component img img-green none
    - override an image:
        - green-component img img none (now img is a
green-component greyscale image)

4. visualize the blue-component of an image:
    - the command "blue-component" can be in mixed cases:
        - blue-component img img-blue none
        - blue-COMPonent img img-blue none
    - override an image:
        - blue-component img img none (now img is a
blue-component greyscale image)

5. visualize the value-component of an image:
    - the command "value-component" can be in mixed cases:
        - value-component img img-value none
        - value-compoNENt img img-value none
    - override an image:
        - value-component img img none (now img is a
value-component greyscale image)

6. visualize the luma-component of an image:
    - the command "luma-component" can be in mixed cases:
        - luma-component img img-luma none
        - LUMA-component img img-luma none
    - override an image:
        - luma-component img img none (now img is a
luma-component greyscale image)

7. visualize the intensity-component of an image:
    - the command "intensity-component" can be in mixed cases:
        - intensity-component img img-intensity none
        - intensity-COMPONENT img img-intensity none
    - override an image:
        - intensity-component img img none (now img is a
value-component greyscale image)

8. flip an image horizontally:
    - the command "horizontal-flip" can be in mixed cases:
        - horizontal-flip img img-hf
        - horizontal-FLIP img img-hf
    - override an image:
        - horizontal-flip img img (now img is a horizontally
flipped image)
    - support flip horizontally again:
        - horizontal-flip img-hf img-hf2 (img-hf2 should look
like the original picture)

9. flip an image vertically:
    - the command "vertical-flip" can be in mixed cases:
        - vertical-flip img img-vf
        - vertical-FLIP img img-vf
    - override an image:
        - vertical-flip img img (now img is a vertically flipped
image)
    - support flip vertically again:
        - vertical-flip img-vf img-vf2 (img-vf2 should look like
the original picture)

10. brighten an image:
    - the command "brighten" can be in mixed cases:
        - brighten 30 img img-b-30 none
        - brIGHTen 30 img img-b-30 none
    - the increment can be in anywhere after "brighten":
        - brighten img 30 img-b-30 none
        - brighten img img-b-30 30 none
    - override an image:
        - brighten 90 img img none(now img is a brightened by 90
image)
    - support brightening again:
        - brighten img-b-30 30 img-b-60 none (img-b-60 is
brightened by 60 from the original picture)

11. darken an image:
    - the command "darken" can be in mixed cases:
        - darken 30 img img-d-30 none
        - darKEN 30 img img-d-30 none
    - the decrement can be in anywhere after "darken":
        - darken img 30 img-d-30 none
        - darken img img-d-30 30 none
    - override an image:
        - darken 90 img img none (now img is a darkened by 90
image)
    - support darkening again:
        - darken img-d-30 30 img-d-60 none (img-d-60 is darkened
by 60 from the original picture)

12. blur an image:
    - the command "blur" can be in mixed cases:
        - blur img img-blur none
        - BLur img img-blur none
    - override an image:
        - blur img img none (now img is a blurred image)
    - support blurring again:
        - blur img-blur img-blur-2 none (img-blur-2 is blurred
twice from the original picture)

13. sharpen an image:
    - the command "sharpen" can be in mixed cases:
        - sharpen img img-sharpen none
        - SHArpen img img-sharpen none
    - override an image:
        - sharpen img img none (now img is a sharpened image)
    - support sharpening again:
        - sharpen img-sharpen img-sharpen-2 none (img-sharpen-2
is sharpened twice from the original
        picture)
```

```
14. convert an image to greyscale using luma:
    - the command "greyscale" can be in mixed cases:
        - greyscale img img-greyscale none
        - GREYscale img img-greyscale none
    - override an image:
        - greyscale img img none(now img is a greyscale image)
    - support greyscale again:
        - greyscale img-greyscale img-greyscale-2 none
(img-greyscale-2 is greyscale-d twice from the
        original picture)

15. convert an image to a sepia-toned image:
    - the command "sepia" can be in mixed cases:
        - sepia img img-sepia none
        - SEPIA img img-sepia none
    - override an image:
        - sepia img img none (now img is a sepia image)
    - support sepia again:
        - sepia img-sepia img-sepia-2 none (img-sepia-2 is
sepia-d twice from the original picture)

16. save an image:
    - save an image as jpg/png/bmp/ppm file
        - save res/img.png img
        - save res/img.bmp img
        - save res/img.ppm img
        - save res/img.jpg img

17. Partially manipulate an image:
    - visualize the red-component of a part of an image:
        - load res/img.ppm img
        - load res/img-mask.ppm img-mask
        - red-component img img-part-red img-mask
     Similar syntax for
green/blue/value/intensity/luma-component, blur, sharpen,
greyscale, and sepia commands.
```

Note:

Our application can handle an error that happens if the command is not entered in the correct order, the program can still work. For example, the increment comes after "brighten" can be in any order, before two string inputs, in between, or after the second one. Moreover, if any error occurs during the process, such as, invalid command, invalid image path, non-existing image, invalid extension, and so on, the application would display a message to remind the user the inputs are not right and the user can input again.

Our application only supports jpg, png, bmp, and ppm. ppm has to begin with P3.

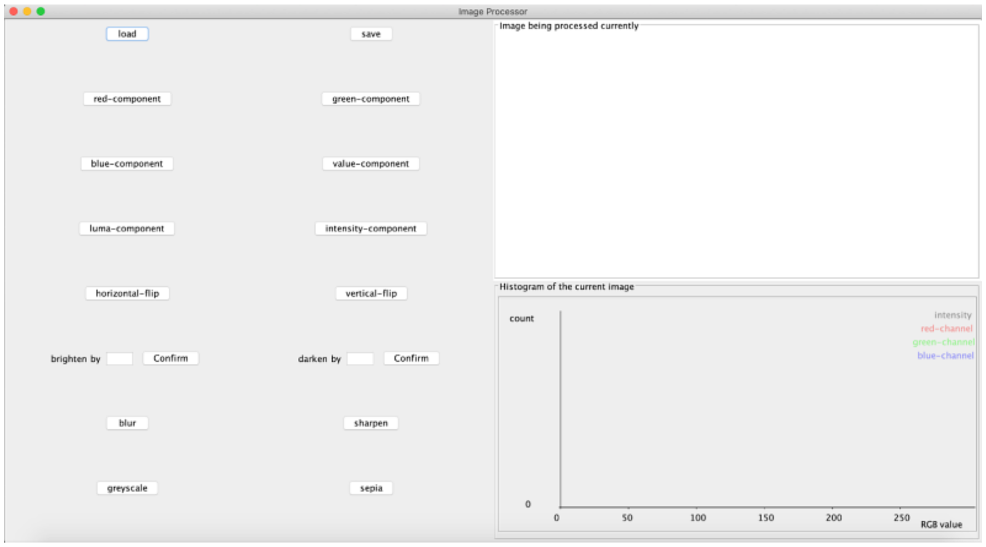
When using a command that requires image-name and dest-image-name, image-name must be typed before dest-image-name. Image-name must be typed before mask-image-name for commands that support partial image manipulation. Mask image must have the exact same size as the image.

GUI USEME:

Note: GUI doesn't support partial image manipulation.

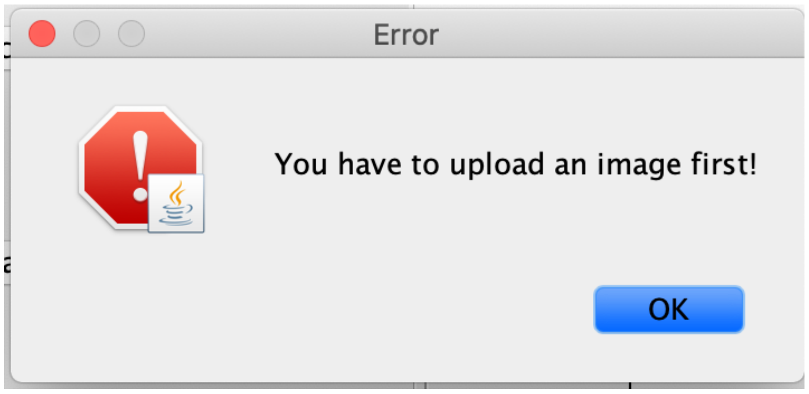
- Load an image.
    - Only supports loading jpg, png, ppm, and bmp images.
    - Override the current image by loading an image again.
- Apply operations to manipulate the image.
- Save the image.
    - Only supports saving jpg, png, ppm, and bmp images.

When the GUI application is started, it looks like this:



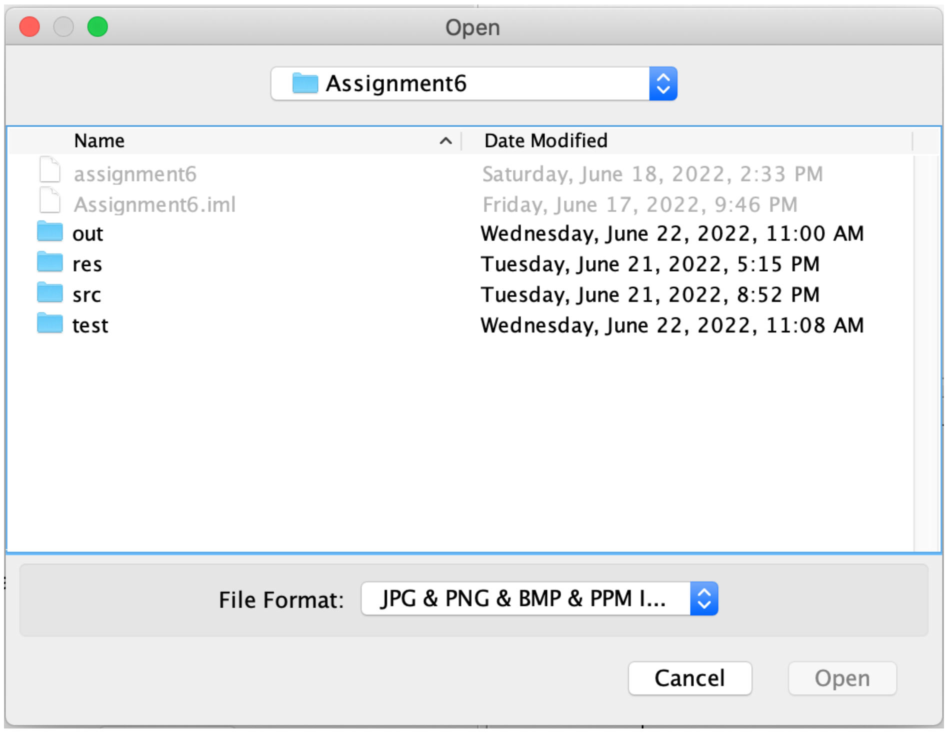
The left side contains all the supported operations as buttons. The top right displays the image currently being manipulated. The bottom right displays the histogram.

When the program detects an error, it will pop up a window to tell the user. For example, if the user tries to apply operations when there's no image, it will tell the user to load an image first.


Error

You have to upload an image first!

If the user press the "load" button,

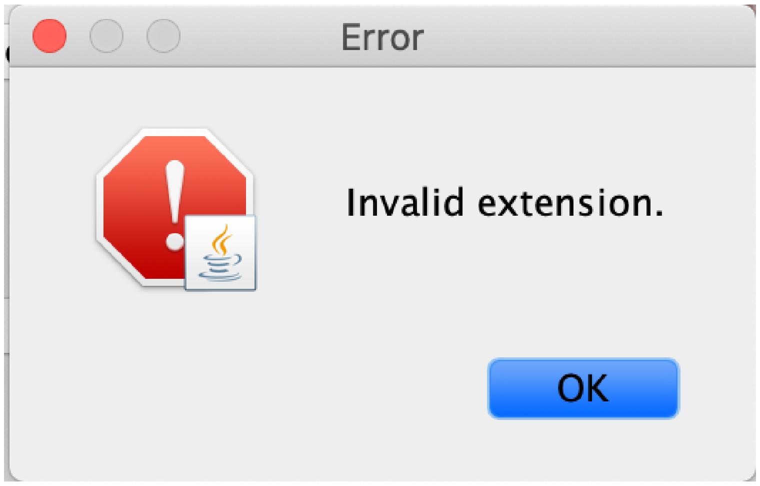


Name	Date Modified
assignment6	Saturday, June 18, 2022, 2:33 PM
Assignment6.iml	Friday, June 17, 2022, 9:46 PM
out	Wednesday, June 22, 2022, 11:00 AM
res	Tuesday, June 21, 2022, 5:15 PM
src	Tuesday, June 21, 2022, 8:52 PM
test	Wednesday, June 22, 2022, 11:08 AM

File Format: JPG & PNG & BMP & PPM I...

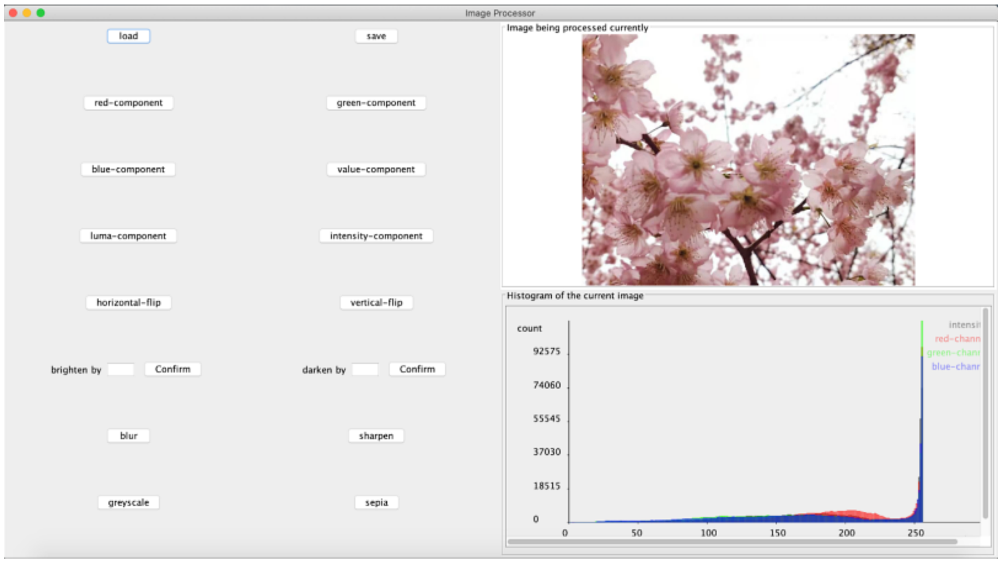
An open window will pop up to allow the user to choose a file.

If the user chooses a file that is not valid, the application will pop up a window,

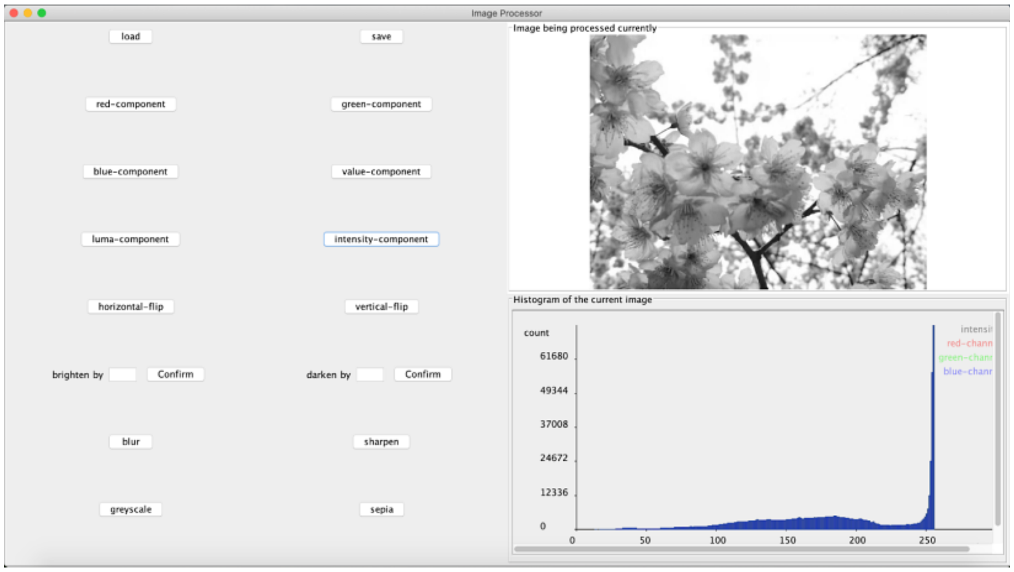


Invalid extension.

Once the user chooses a valid file, the application will display the image and the histogram of that image.

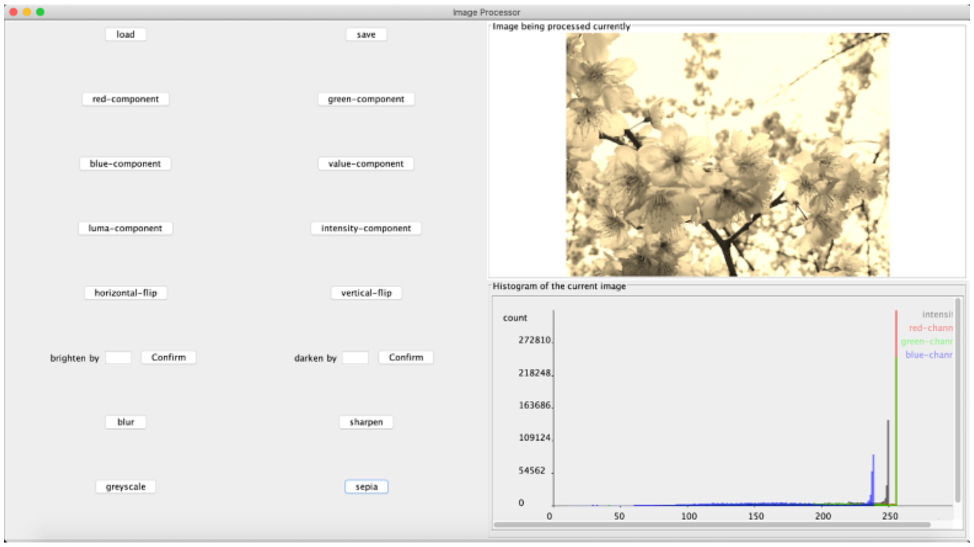


Once there's an image, the user can apply operations on it. Both the image display and histogram display will be refreshed. If I press "intensity-component",

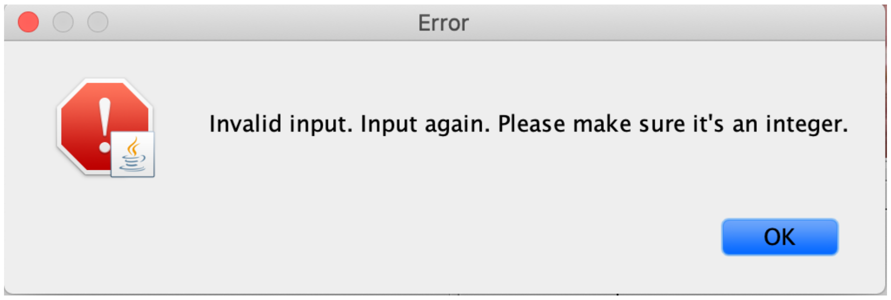


The user is able to scroll the histogram display.

If the user loads example.png again and chooses sepia for example, the image and histogram will be refreshed again,

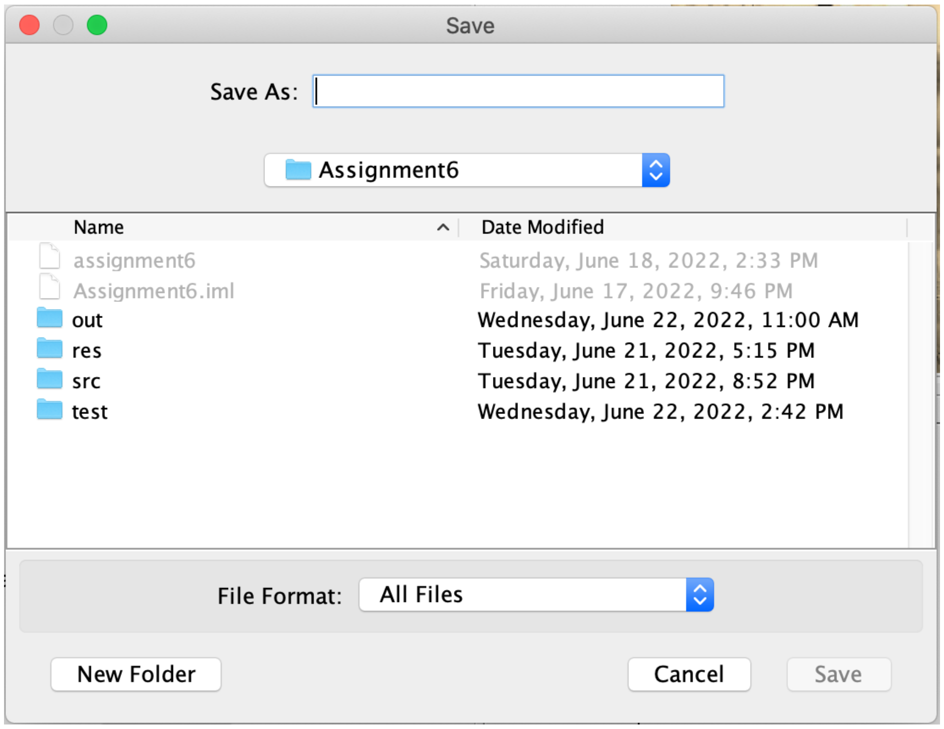


If the user wants to brighten/darken an image but they input something invalid, the application will pop up a window with an error message,



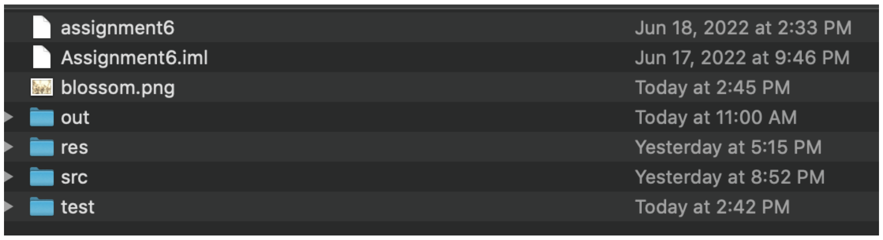
Invalid input. Input again. Please make sure it's an integer.

When the user presses "save", a window will pop up to allow the user to choose a directory to save the file.



Name	Date Modified
assignment6	Saturday, June 18, 2022, 2:33 PM
Assignment6.iml	Friday, June 17, 2022, 9:46 PM
out	Wednesday, June 22, 2022, 11:00 AM
res	Tuesday, June 21, 2022, 5:15 PM
src	Tuesday, June 21, 2022, 8:52 PM
test	Wednesday, June 22, 2022, 2:42 PM

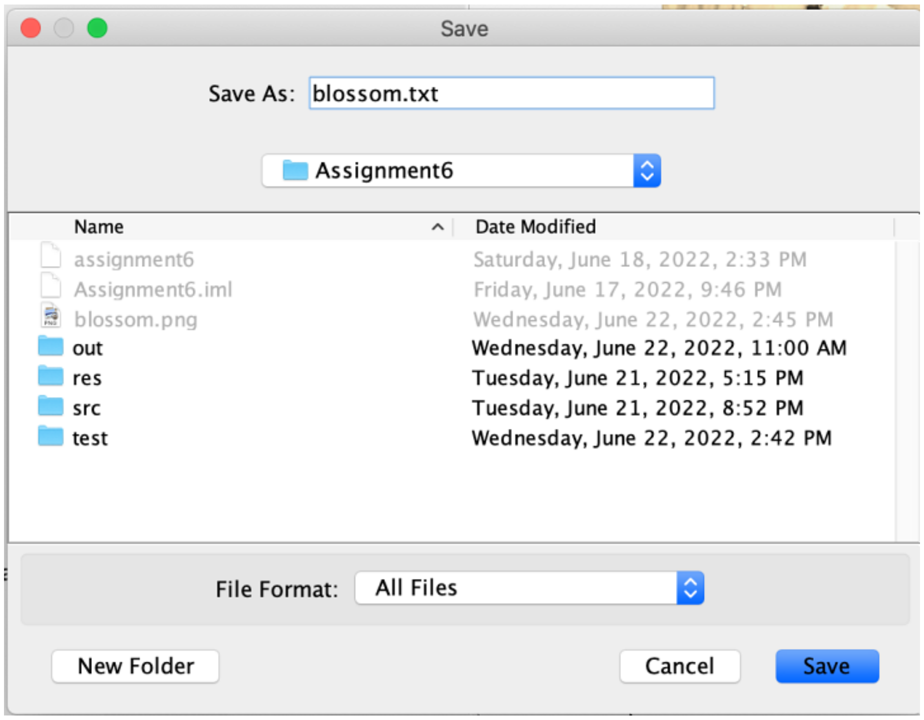
The user needs to input the name of the image and the extension correctly. Suppose I save this image as blossom.png in this directory, the image blossom.png is saved in my computer.



assignment6	Jun 18, 2022 at 2:33 PM
Assignment6.iml	Jun 17, 2022 at 9:46 PM
blossom.png	Today at 2:45 PM
out	Today at 11:00 AM
res	Yesterday at 5:15 PM
src	Yesterday at 8:52 PM
test	Today at 2:42 PM

After saving the image, that image will still be displayed on the screen for the user if they want to process it further. The user just needs to load another image if they don't want to work on that image anymore.

However, if I save it as an unsupported extension, there will be a pop-up window reminding me the extension is invalid,

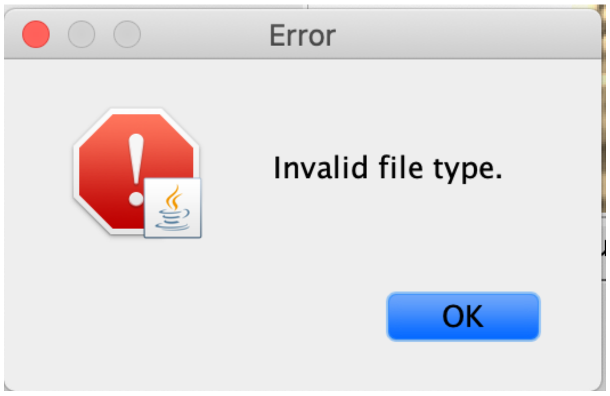

Save

Save As: blossom.txt

Assignment6

Name	Date Modified
assignment6	Saturday, June 18, 2022, 2:33 PM
Assignment6.iml	Friday, June 17, 2022, 9:46 PM
blossom.png	Wednesday, June 22, 2022, 2:45 PM
out	Wednesday, June 22, 2022, 11:00 AM
res	Tuesday, June 21, 2022, 5:15 PM
src	Tuesday, June 21, 2022, 8:52 PM
test	Wednesday, June 22, 2022, 2:42 PM

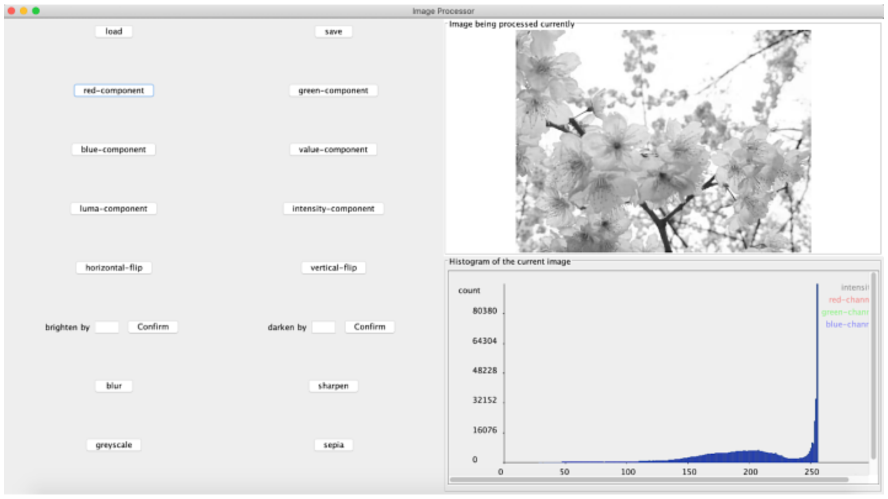
File Format: All Files


Error

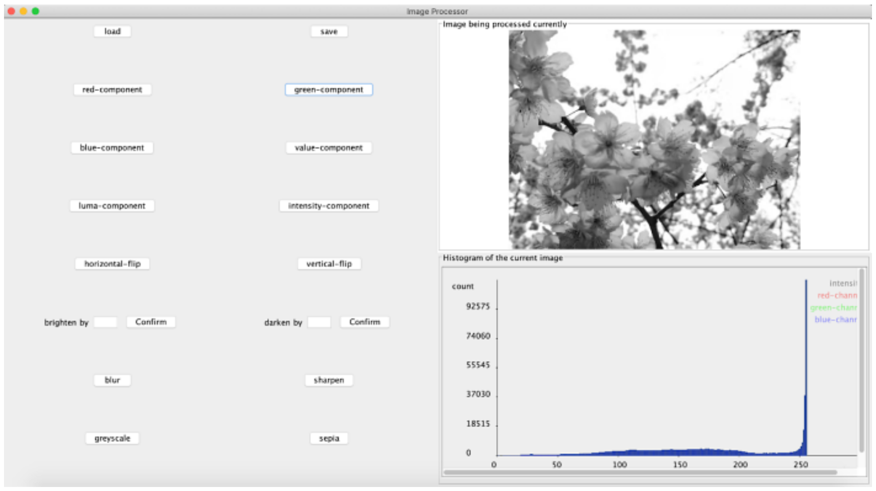
Invalid file type.

Examples of every command **(We loaded example.png before every operation!):**

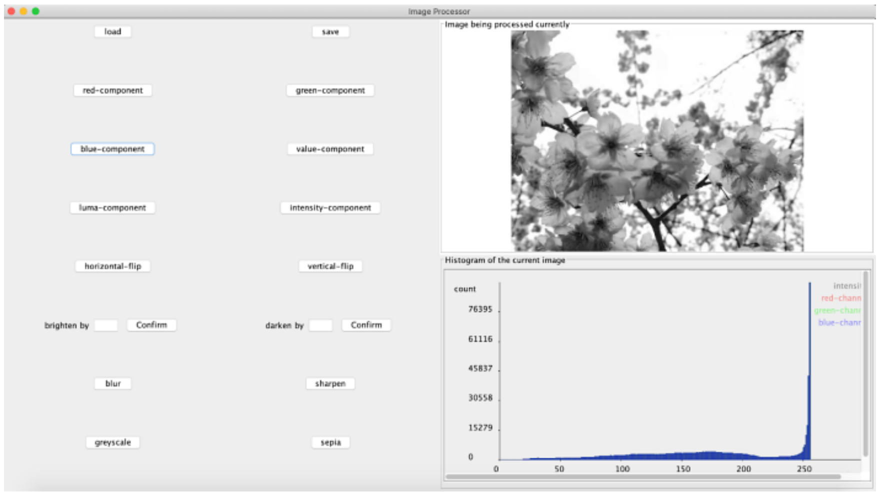
- red-component:



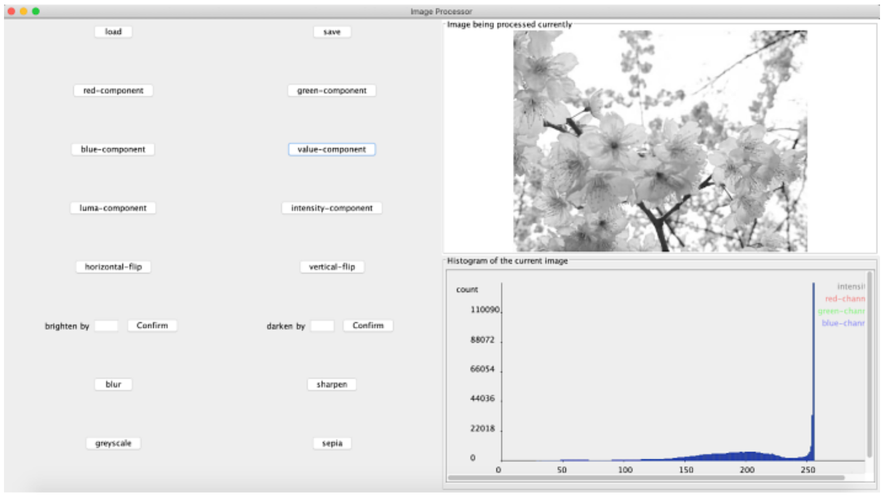
- Green-component:



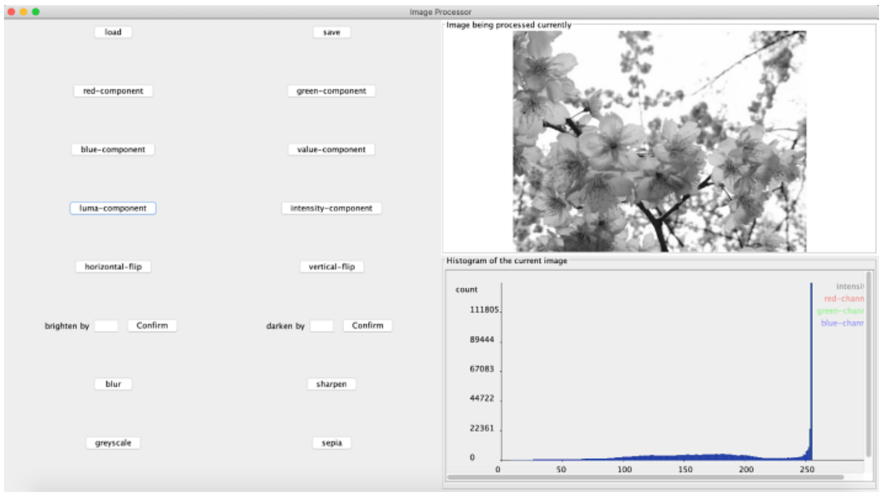
- blue-component:



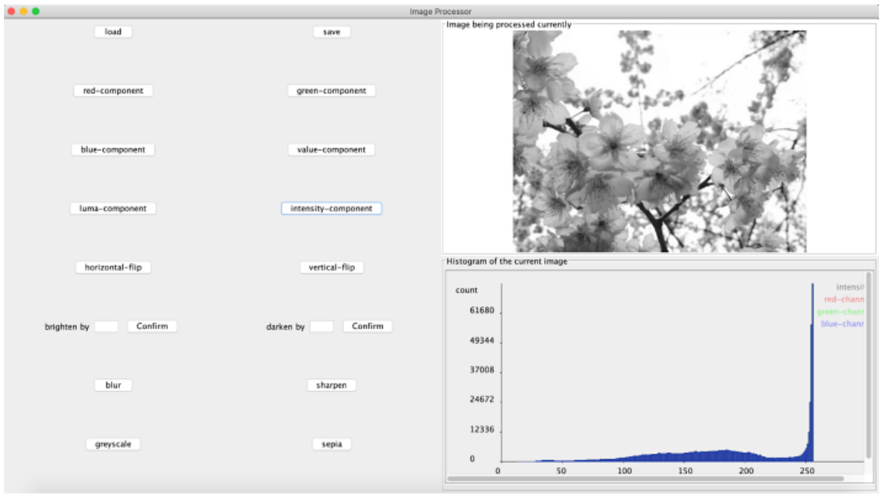
- Value-component:



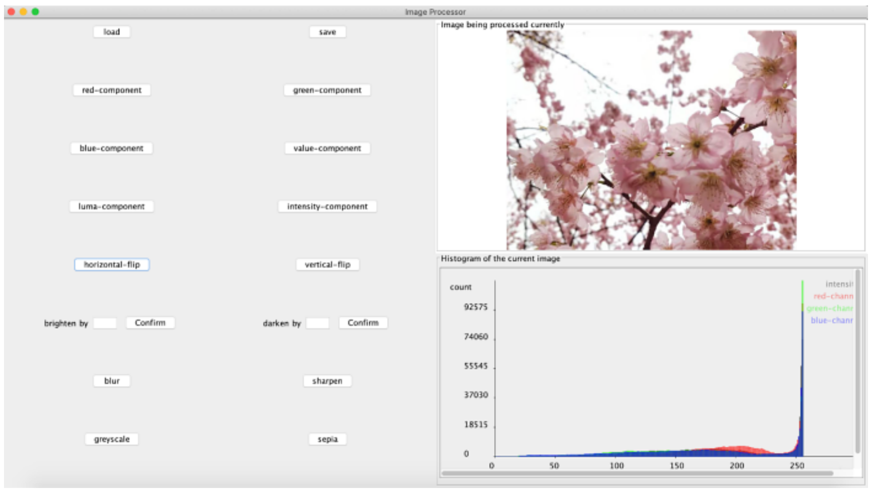
- luma-component:



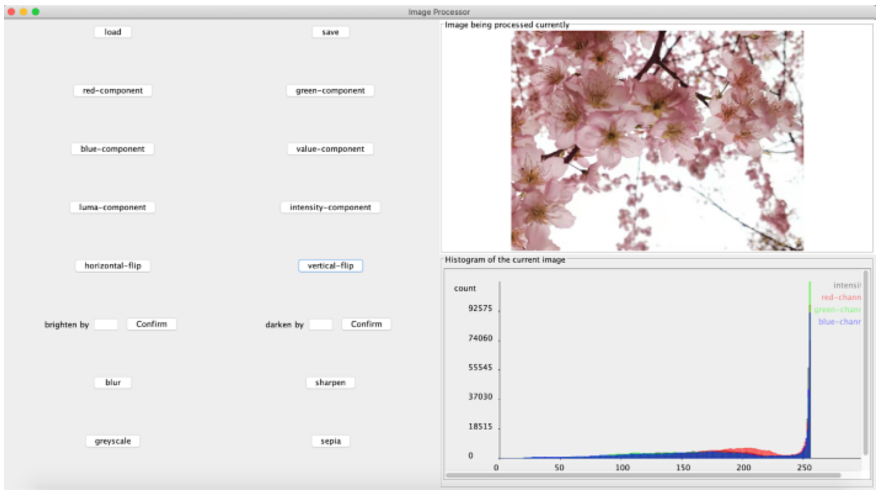
- Intensity-component:



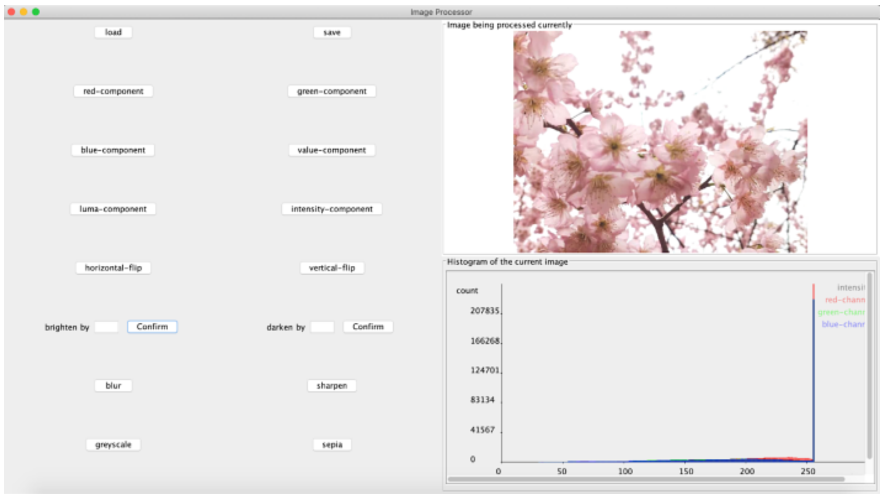
- horizontal-flip:



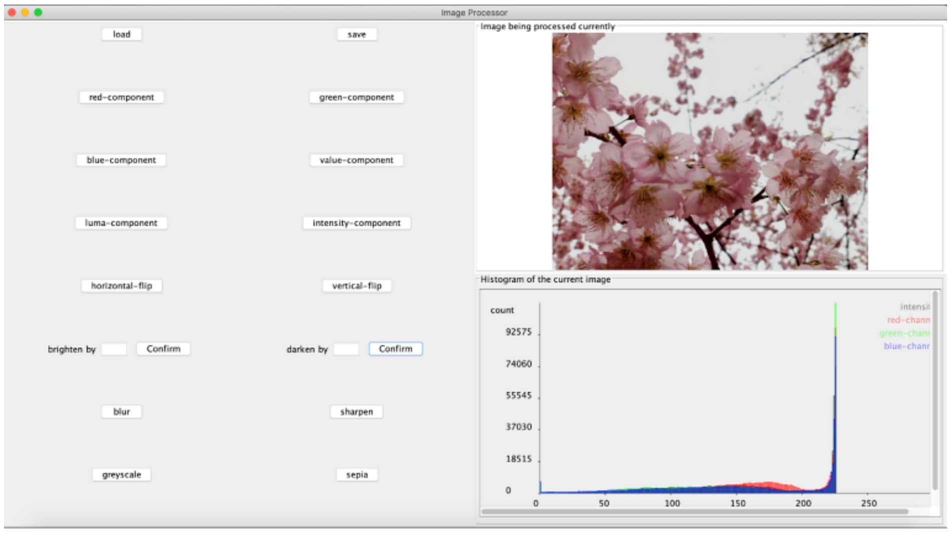
- Vertical-flip:



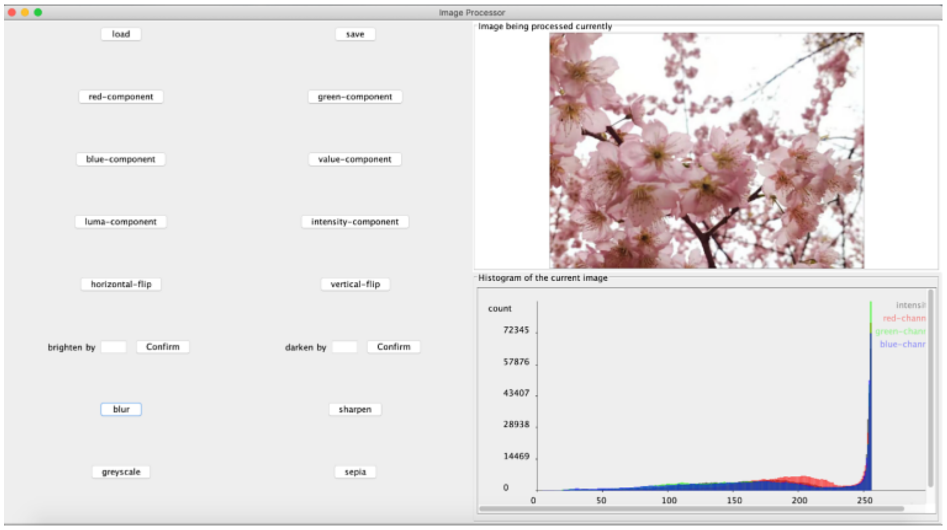
- Brighten by 30:



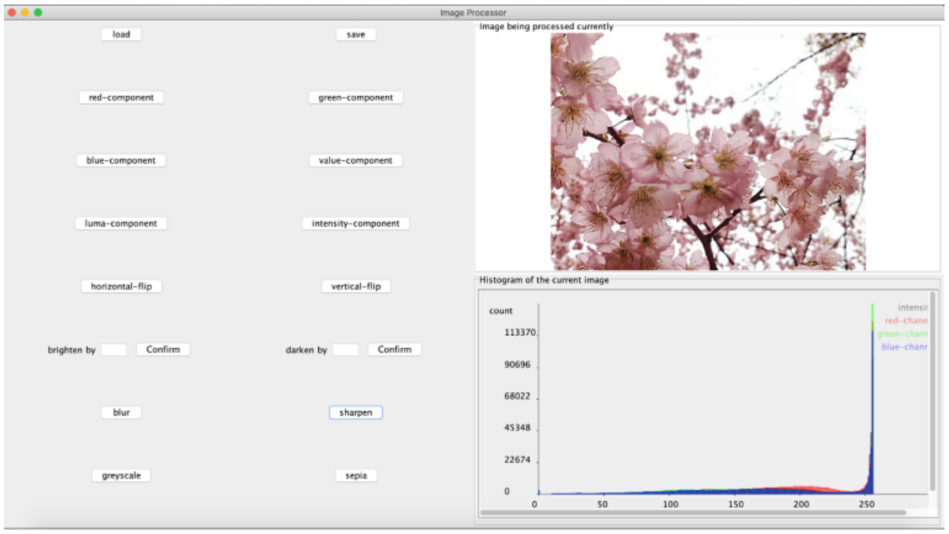
- Darken by 30:



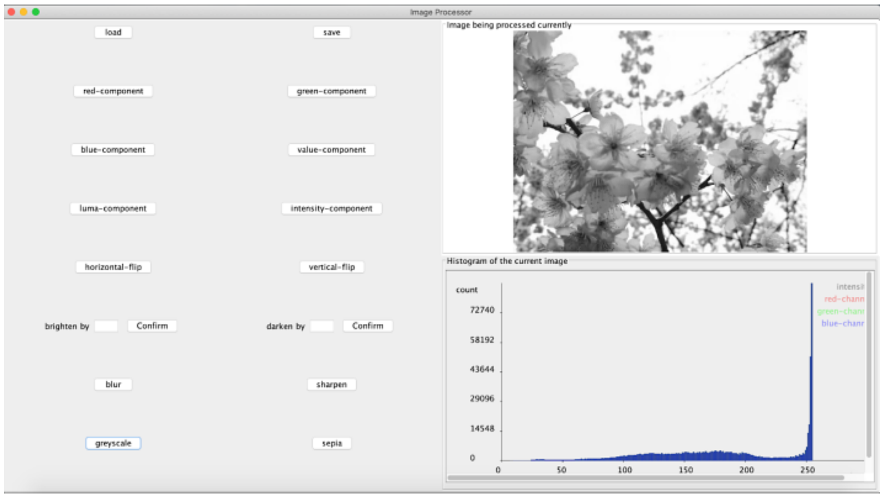
- blur:



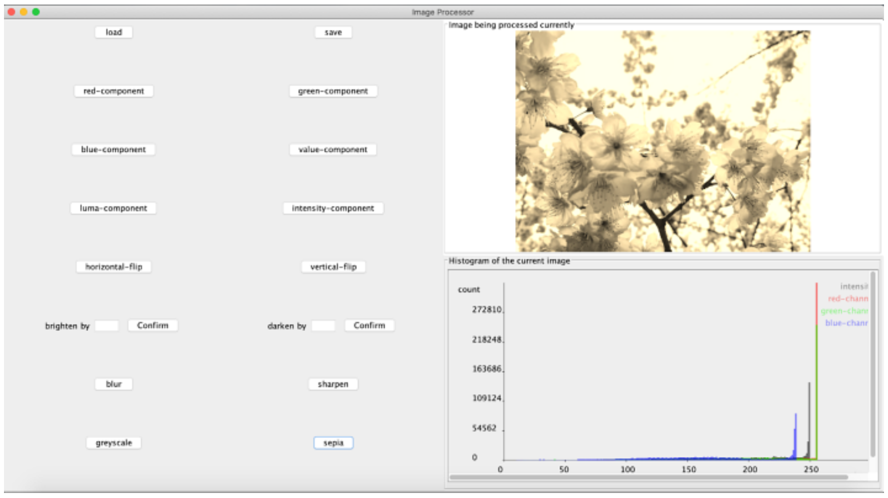
- Sharpen:



- Grayscale:

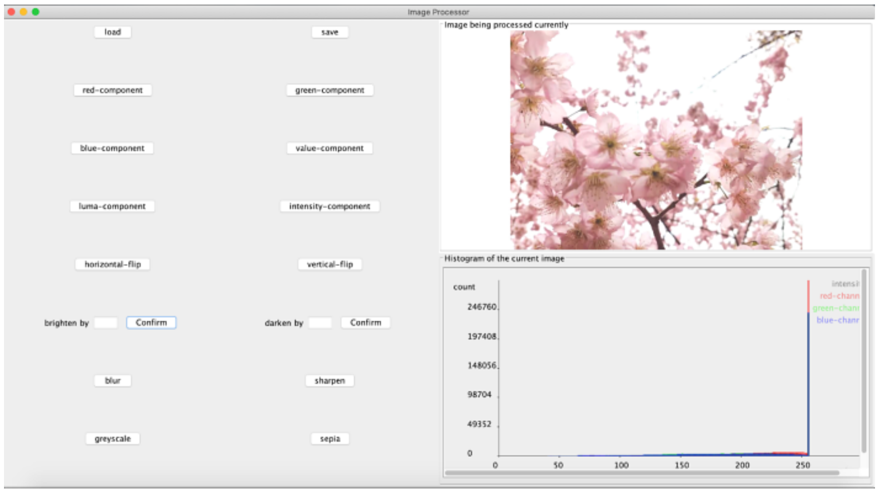


- sepia:



The user can also apply operations on an image multiple times.

- Load res/example.png
- Brighten by 40



- `vertical-flip:`

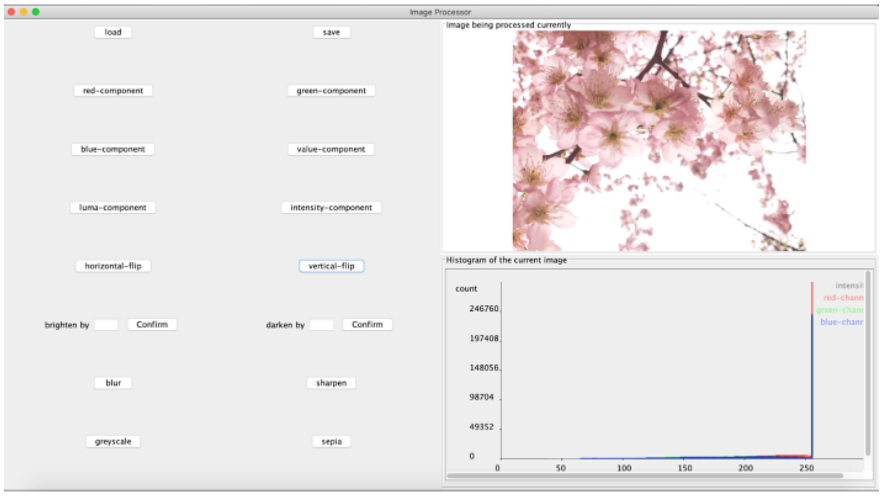


Image Processor
load
save
red-component
green-component
blue-component
value-component
luma-component
intensity-component
horizontal-flip
vertical-flip
brighten by
Confirm
darken by
Confirm
blur
sharpen
greyscale
sepia
Image being processed currently
Histogram of the current image
count
246760
197408
148056
98704
49352
0
0
50
100
150
200
250
intensi
red-chann
green-chann
blue-chanr