mehtapriya05 /
ImageApp

Code Issues Pull requests Actions Projects Security Insights

ImageApp / USEME.md

mehtapriya05 Add files via upload a14ea65 · 1 minute ago

516 lines (405 loc) · 16.8 KB

Preview Code Blame Raw

Image Processor App - USEME.md

This application enables users to process images by executing commands directly via a command prompt or through a script file. It offers various functionalities, including color adjustments, image transformations, and enhancement effects. Additionally, users can preview split views for particular manipulations, where only a portion of the image is altered.

Prerequisites

1. **Java Development Kit (JDK)** installed. Check installation by running:

```
java -version
```

2. **Compile** the Java file before running:

```
cd path/to/directory
javac src/controller/*.java src/model/*.java src/utils/*.java src/*.
```

3. **Run** the application:

```
java -cp src ImageProcessorApp
```

This Program mainly runs in 3 different ways

1. **Script Mode**: To run the script mode of the file we assume that the text file to be run is in the res folder. All the paths are set relative to that location.

```
    java -jar jar-file-name -file absolute-path-to-script-file
```

2. **Interactive Mode**: To run the file in user based interactive mode then the user needs to enter the following commands:

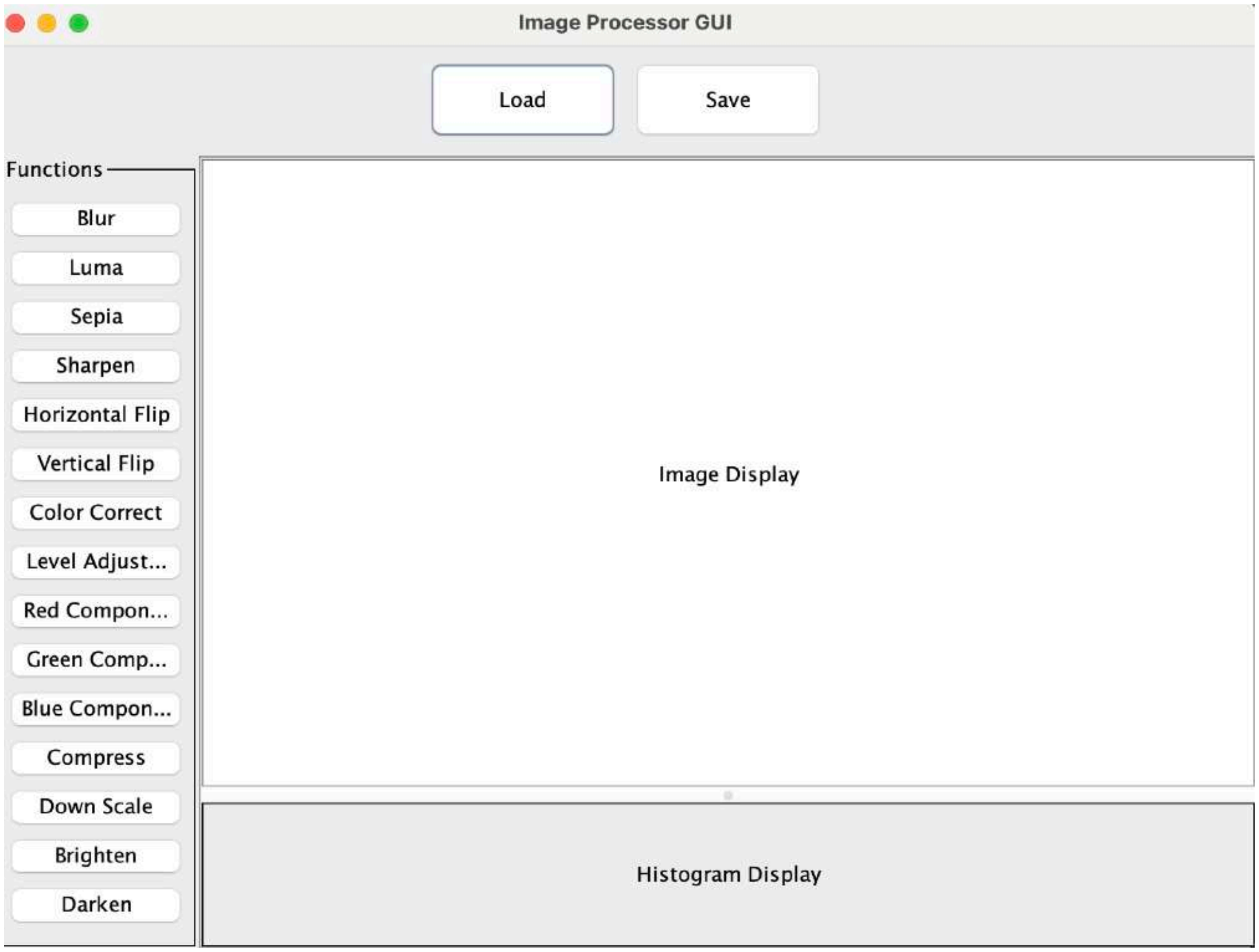
```
        java -jar jar-file-name -text
```

3. **GUI Mode**: To run in interactive mode the following command needs to be given:

```
java -jar jar-file-name
```

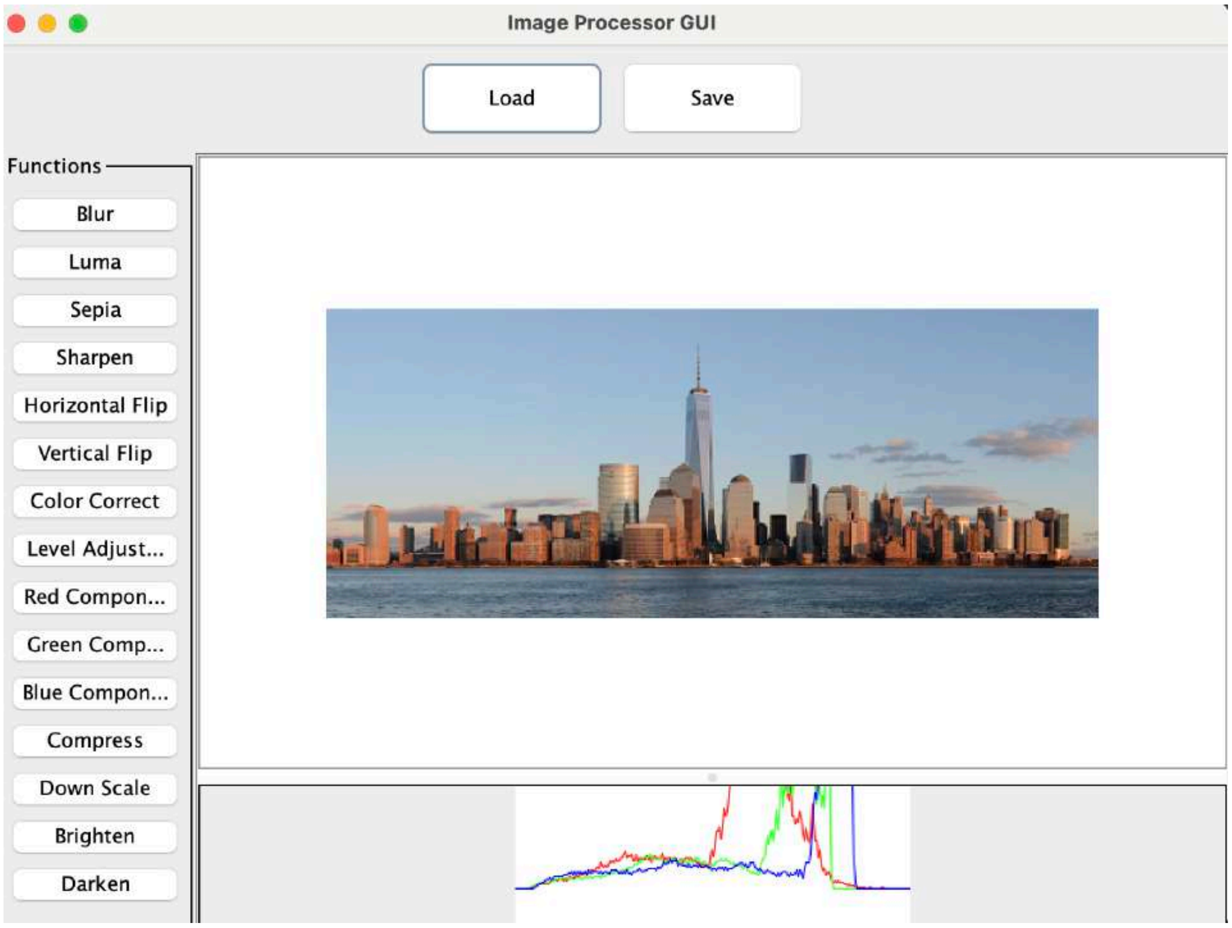
GUI Model

When the user runs the JAR file, a window will open that looks like this:

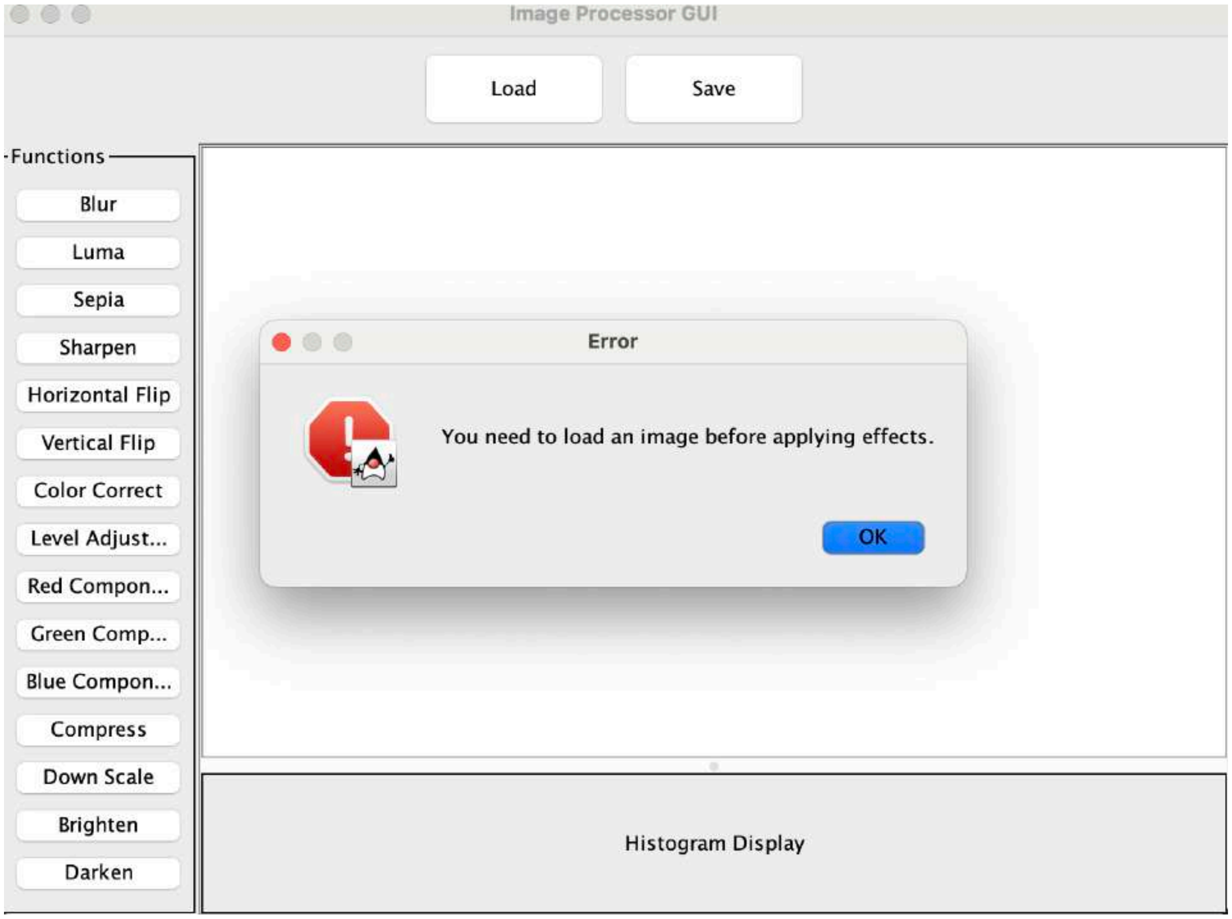


1. **Load Button** When the load button is pressed the user is asked to load the file. The user can enter either JPEG,PNG,JPG,PPM files anywhere from the device. **Note1:** There should be no spaces in the filepath. **Note2:** Without loading any image, none of the functions nor save would work so first load an image.

On successful image load, success message will be displayed. The loaded image and its histogram would be displayed.

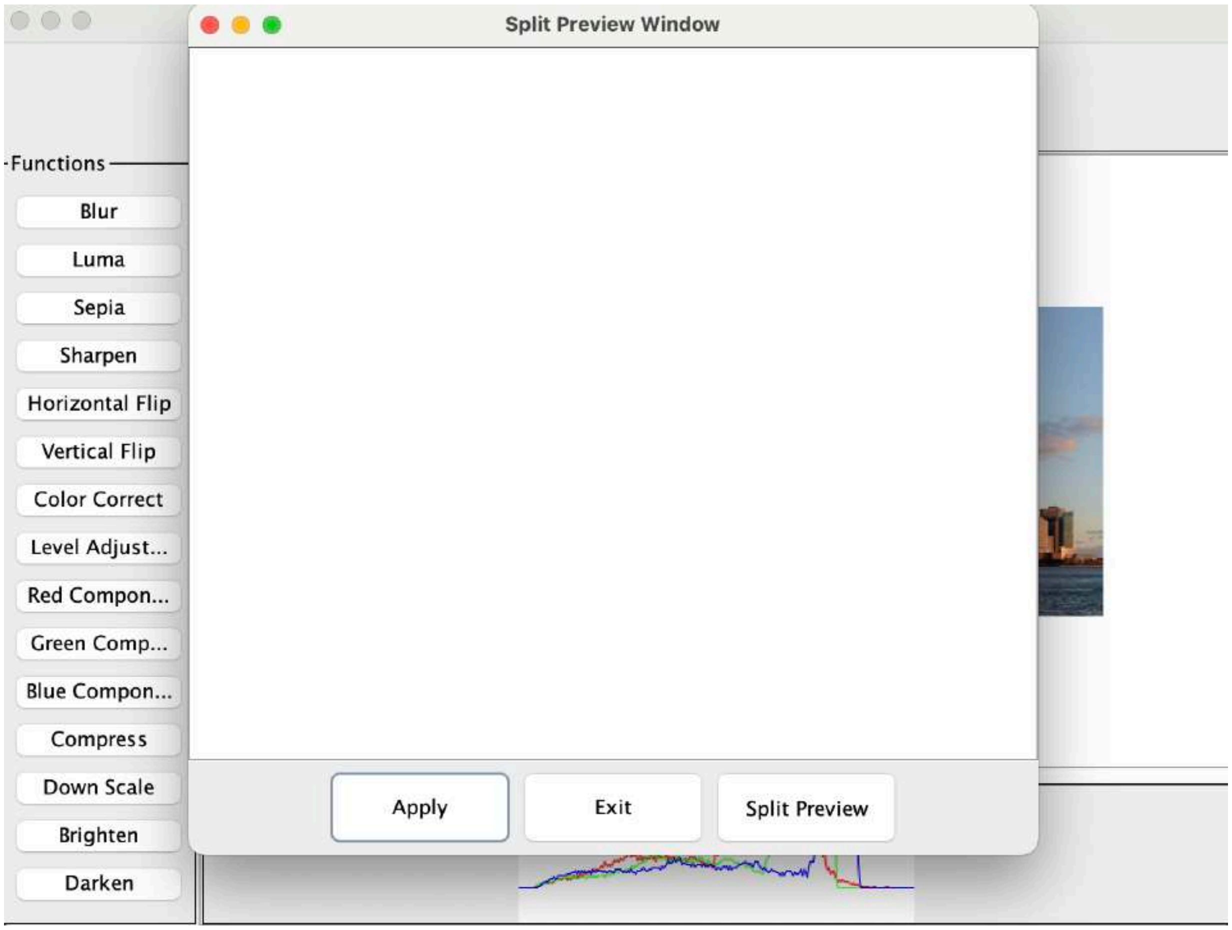


2. **Blur Button** When pressed checked if the image is present on the canvas, if it is present then only it will open the popup window or else will throw an error message

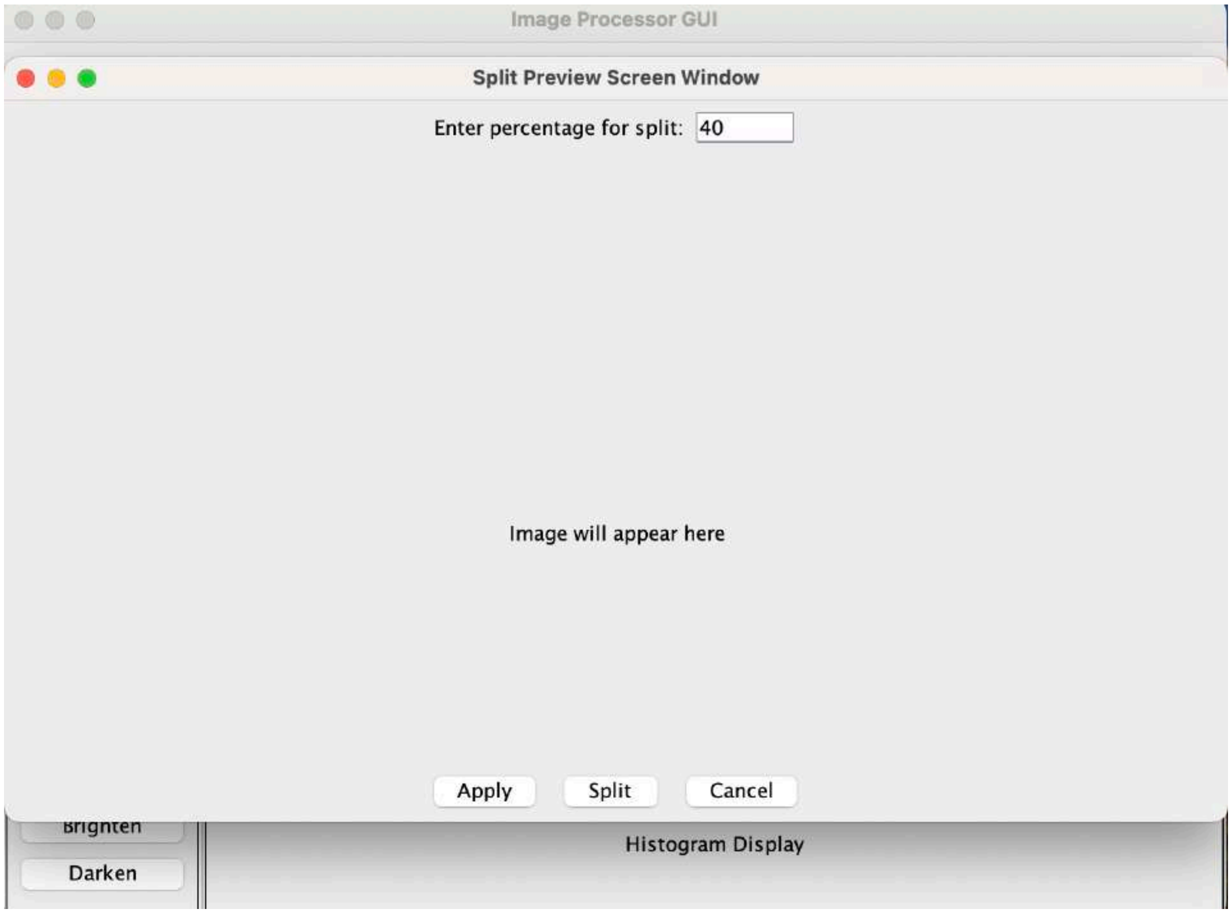


You need to load an image before applying effects.

If image is present then popup window will open containing three buttons for split preview, apply and exit.



On clicking split preview button, user can enter integer values from 0-100. No negative nor alphabets are allowed.



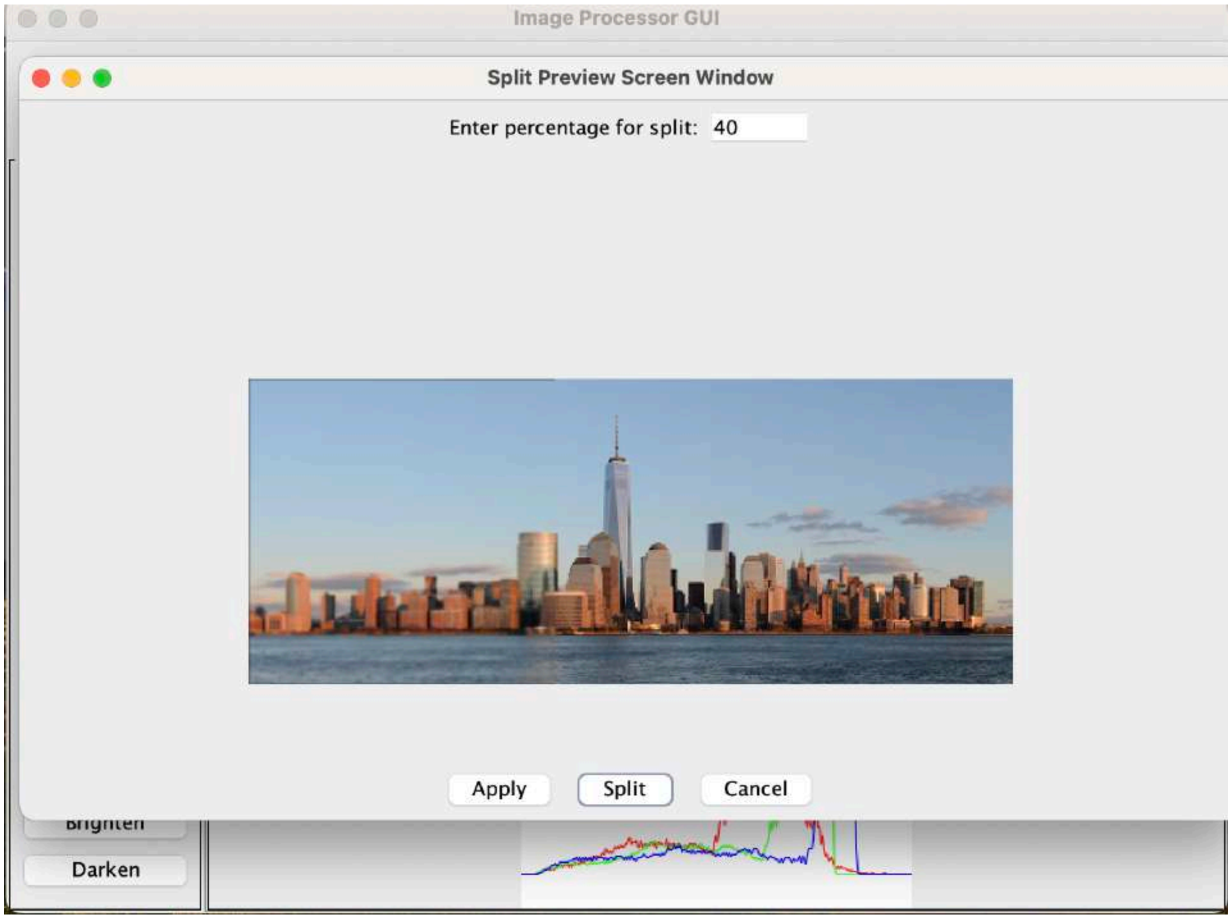
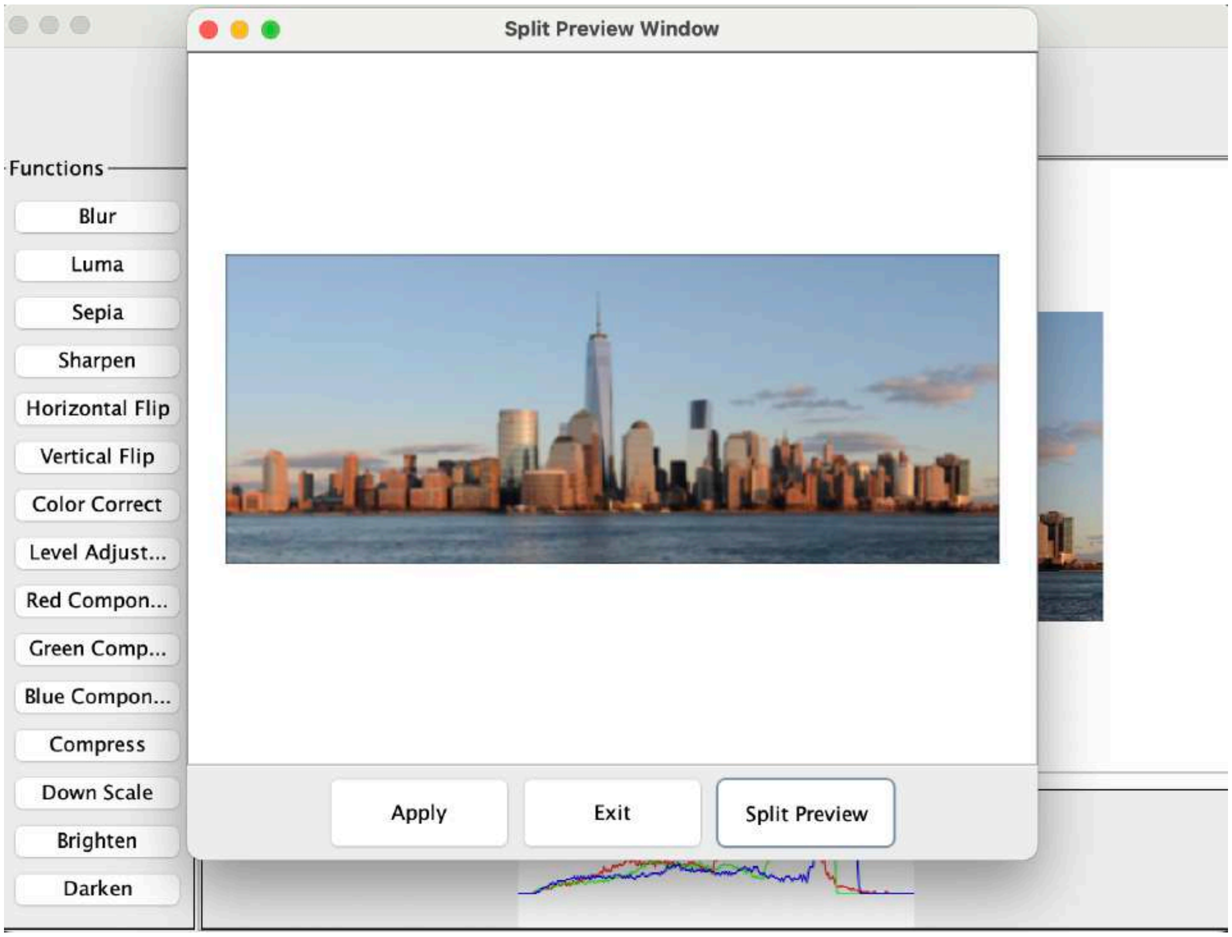
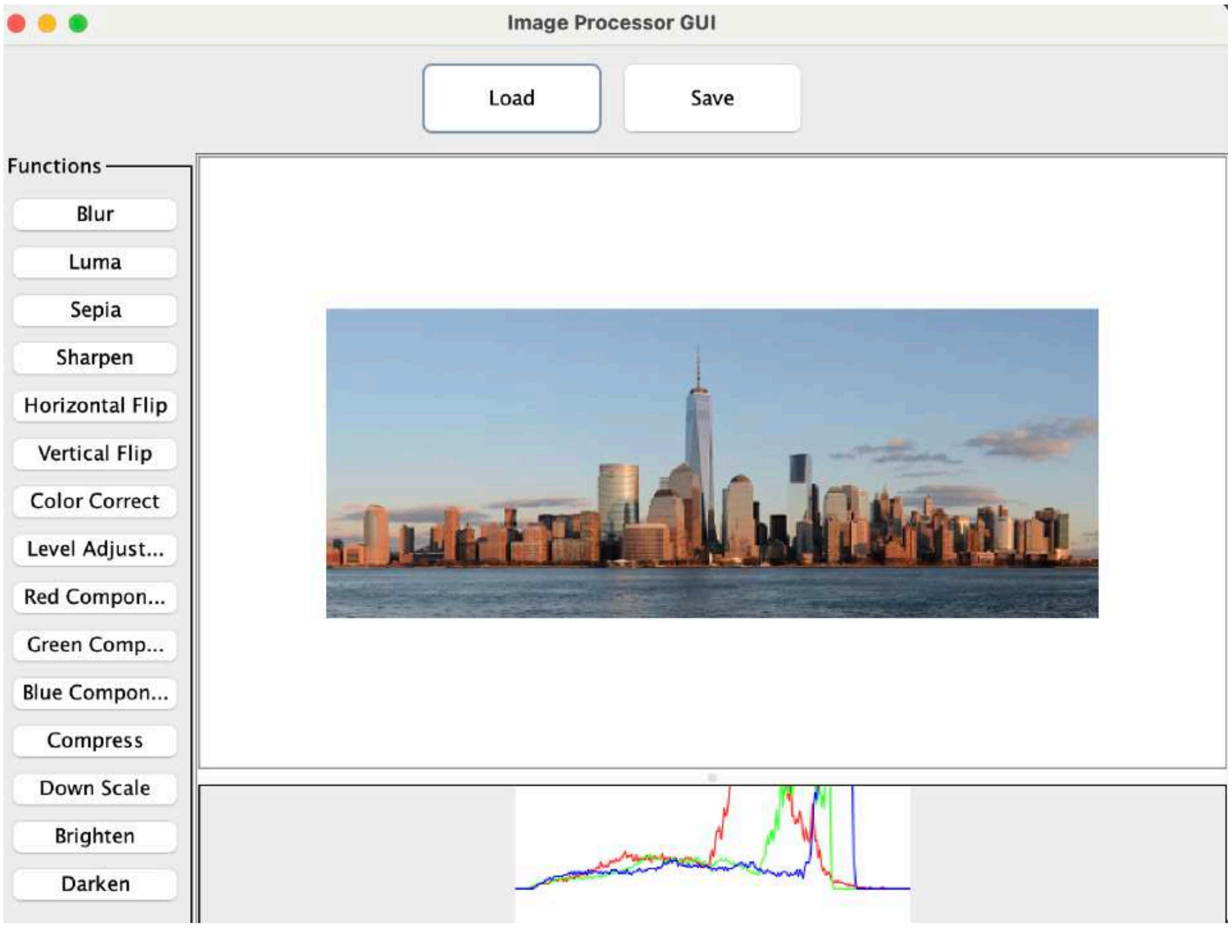
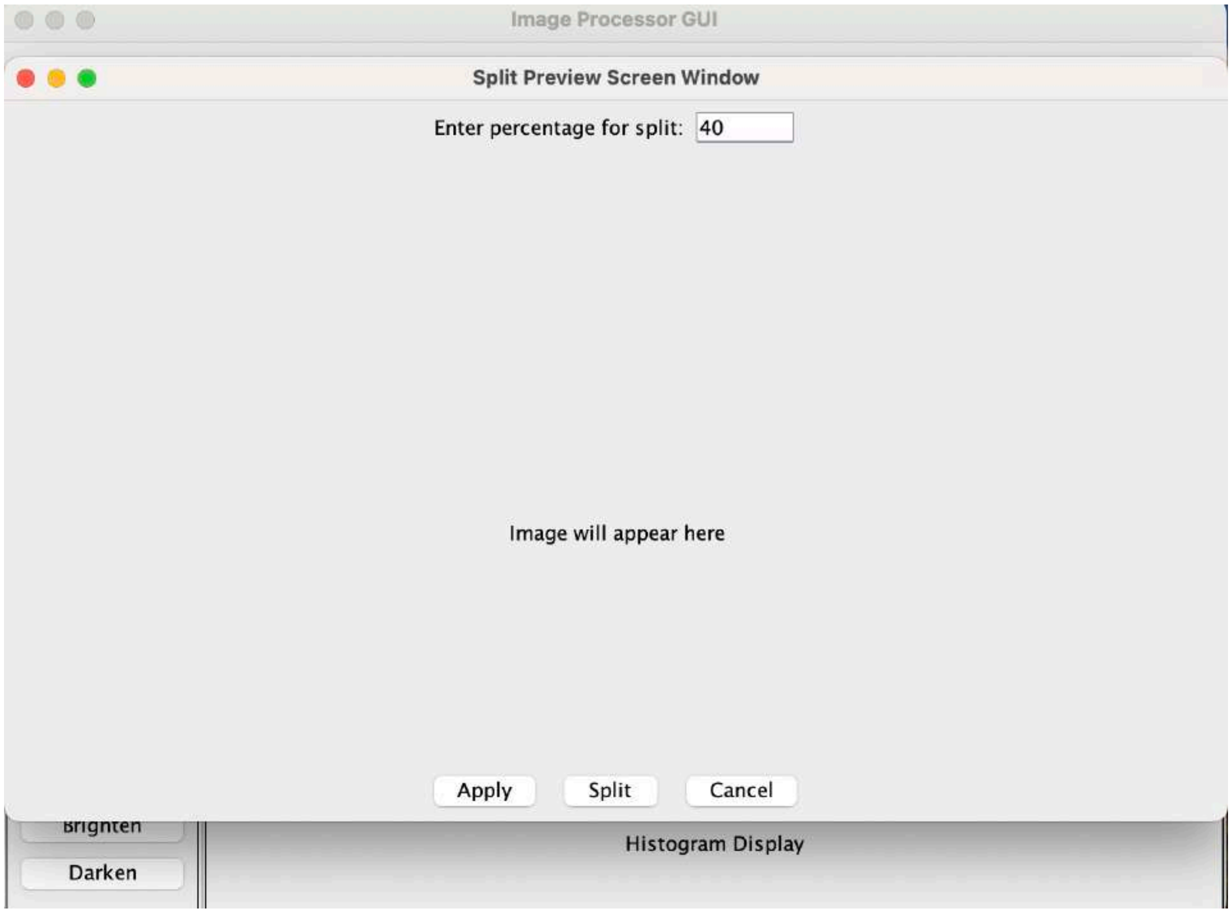


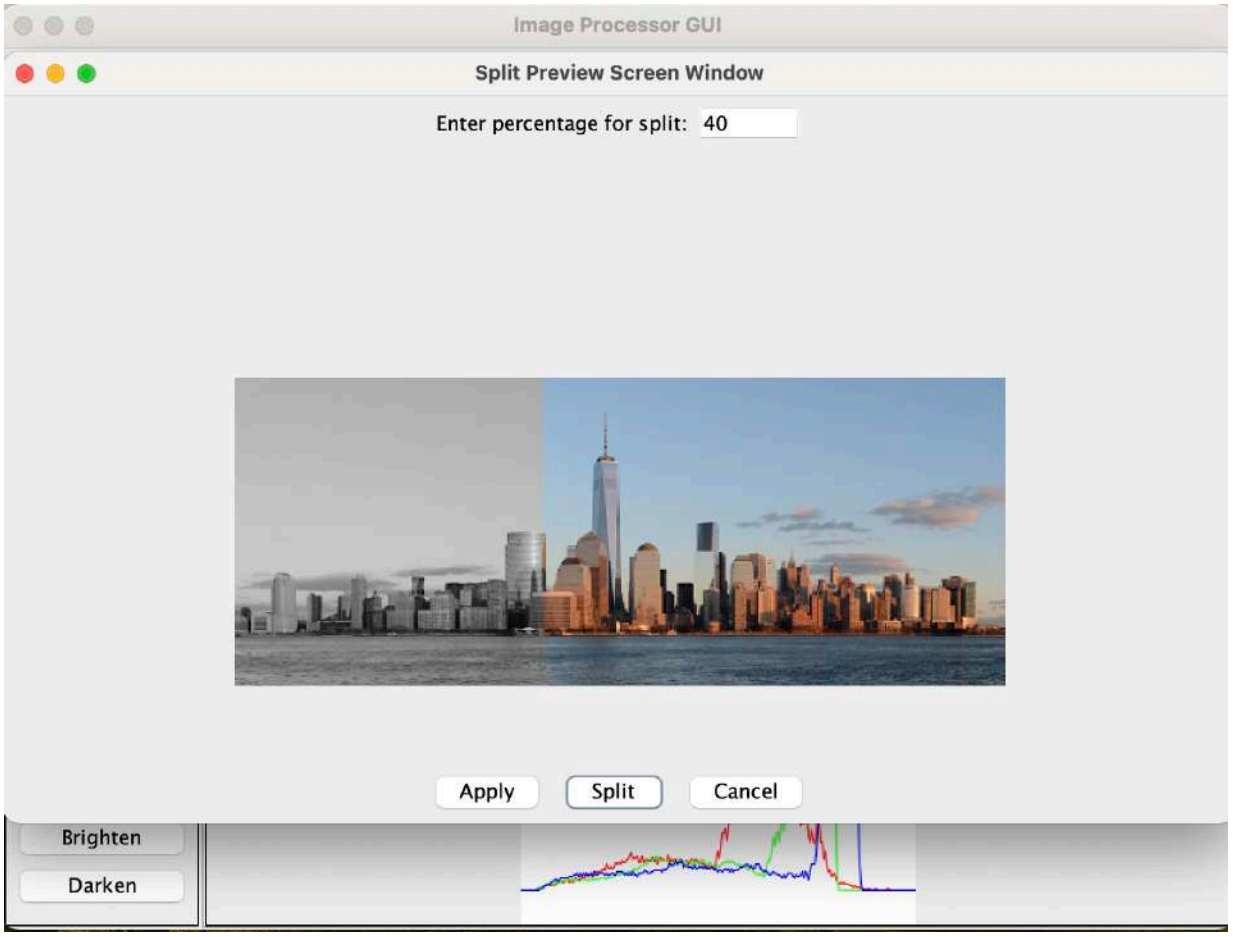
Image for split preview.



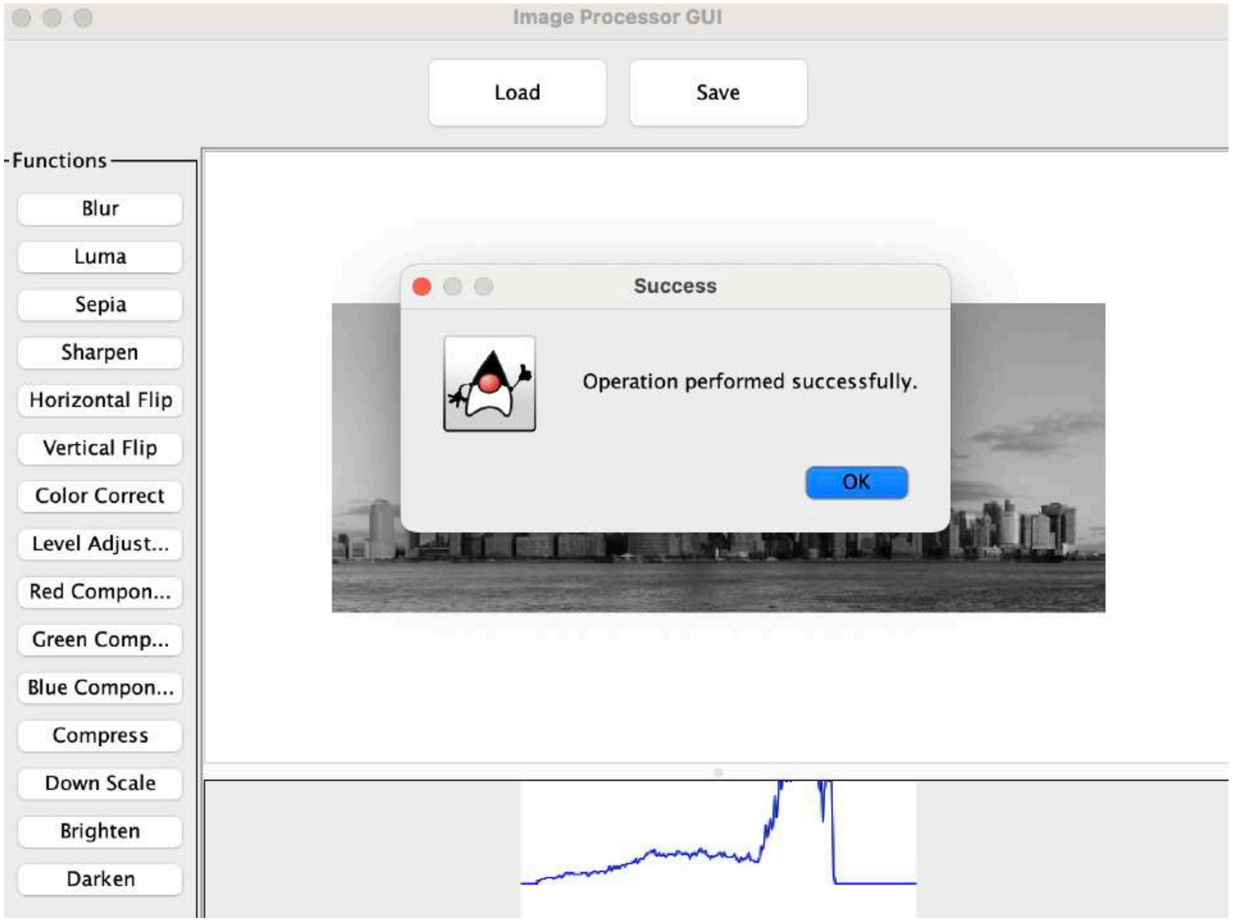
3. **Luma Button** When pressed checked if the image is present on the canvas, if it is present then only it will open the popup window or else will throw an error message.





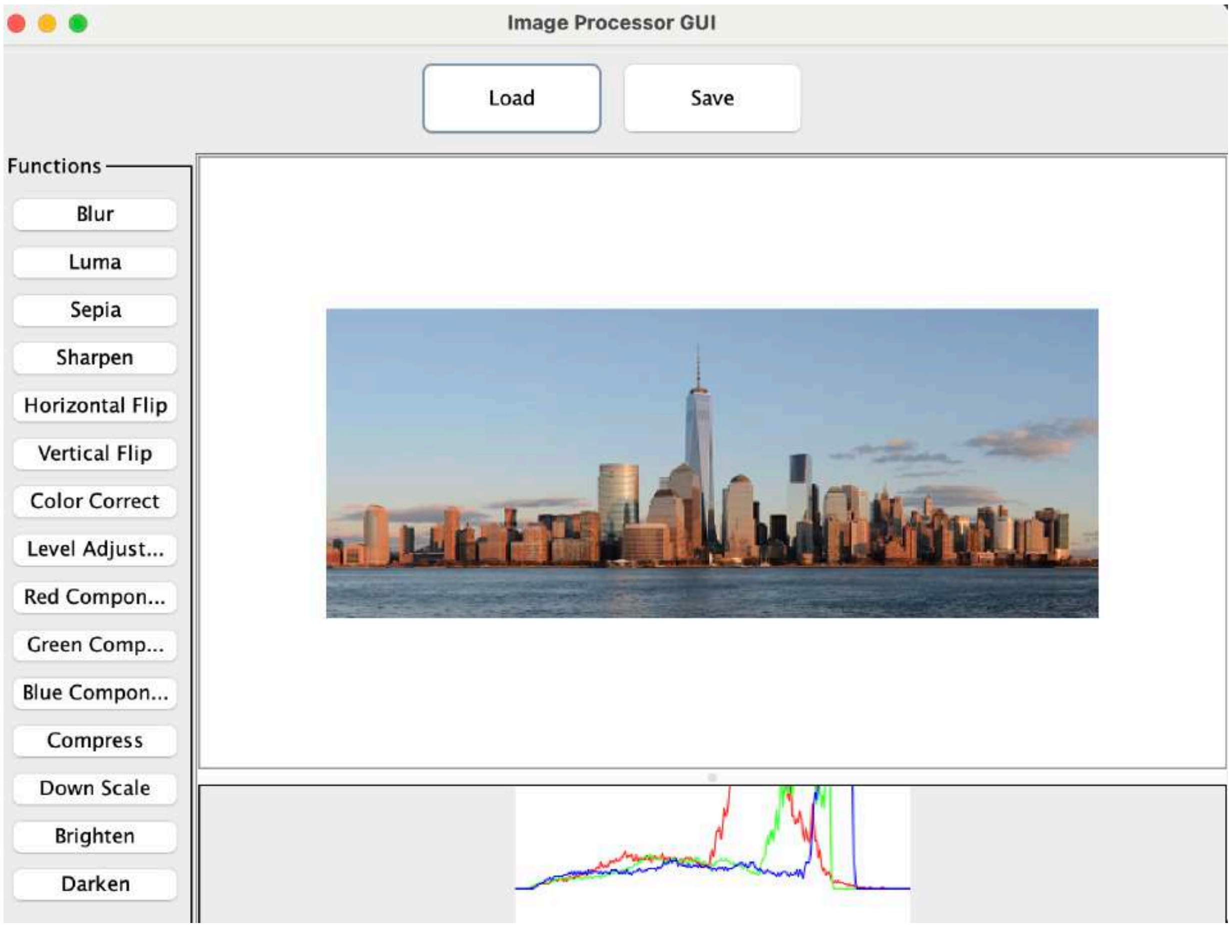


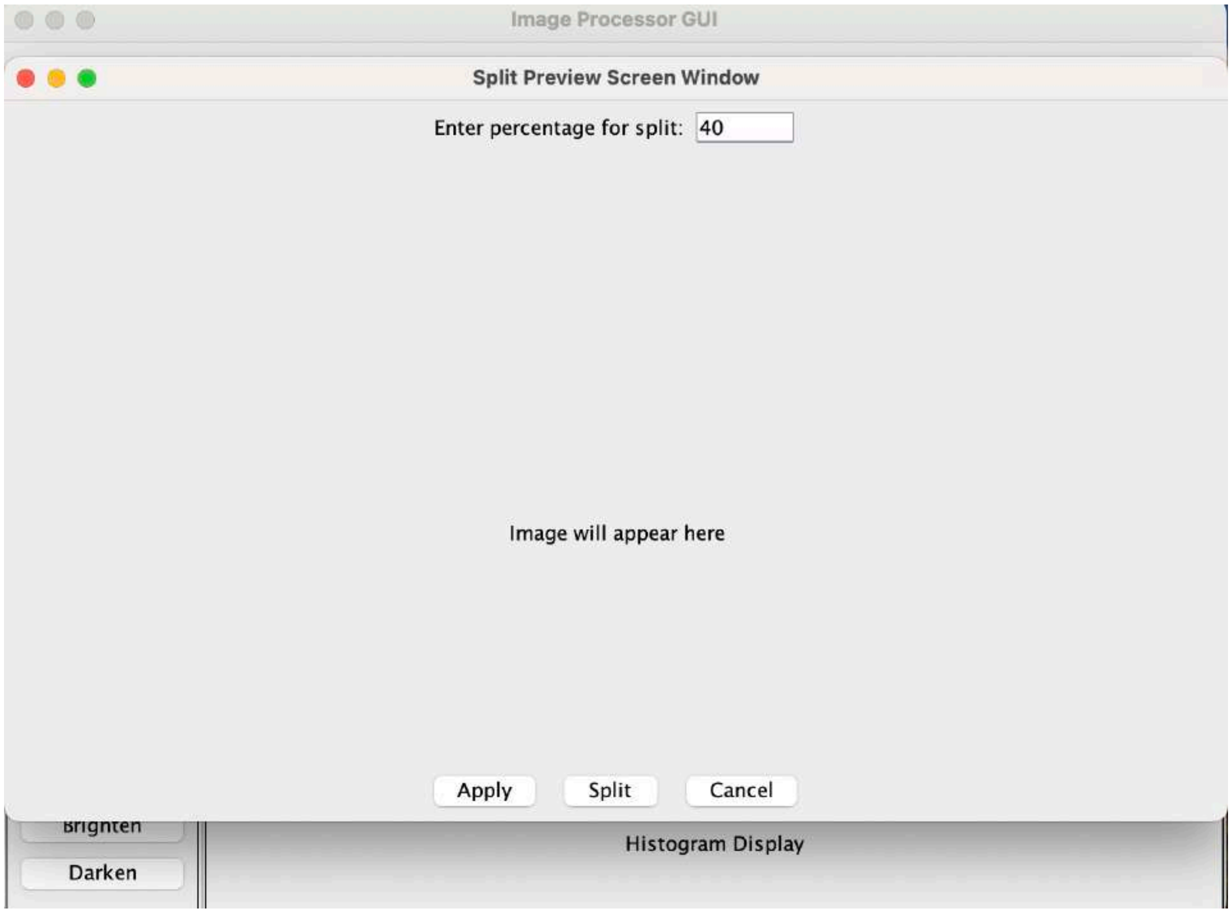
On clicking apply button, success message is shown and functionality is applied and seen in image and histogram.



Operation performed successfully.

4. **Sepia Button** When pressed checked if the image is present on the canvas, if it is present then only it will open the popup window or else will throw an error message.





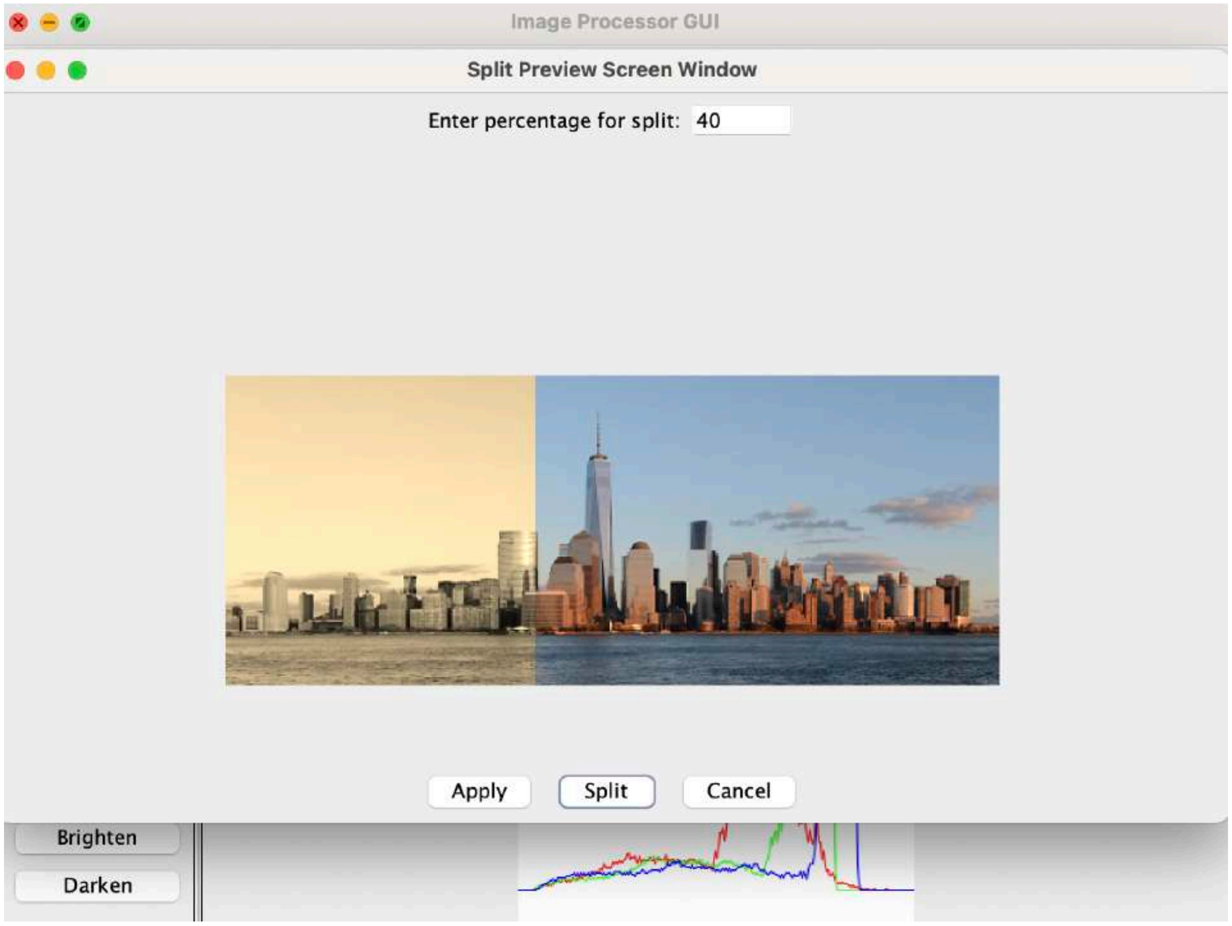


Image Processor GUI
Split Preview Screen Window
Enter percentage for split: 40
Apply
Split
Cancel
Brighten
Darken

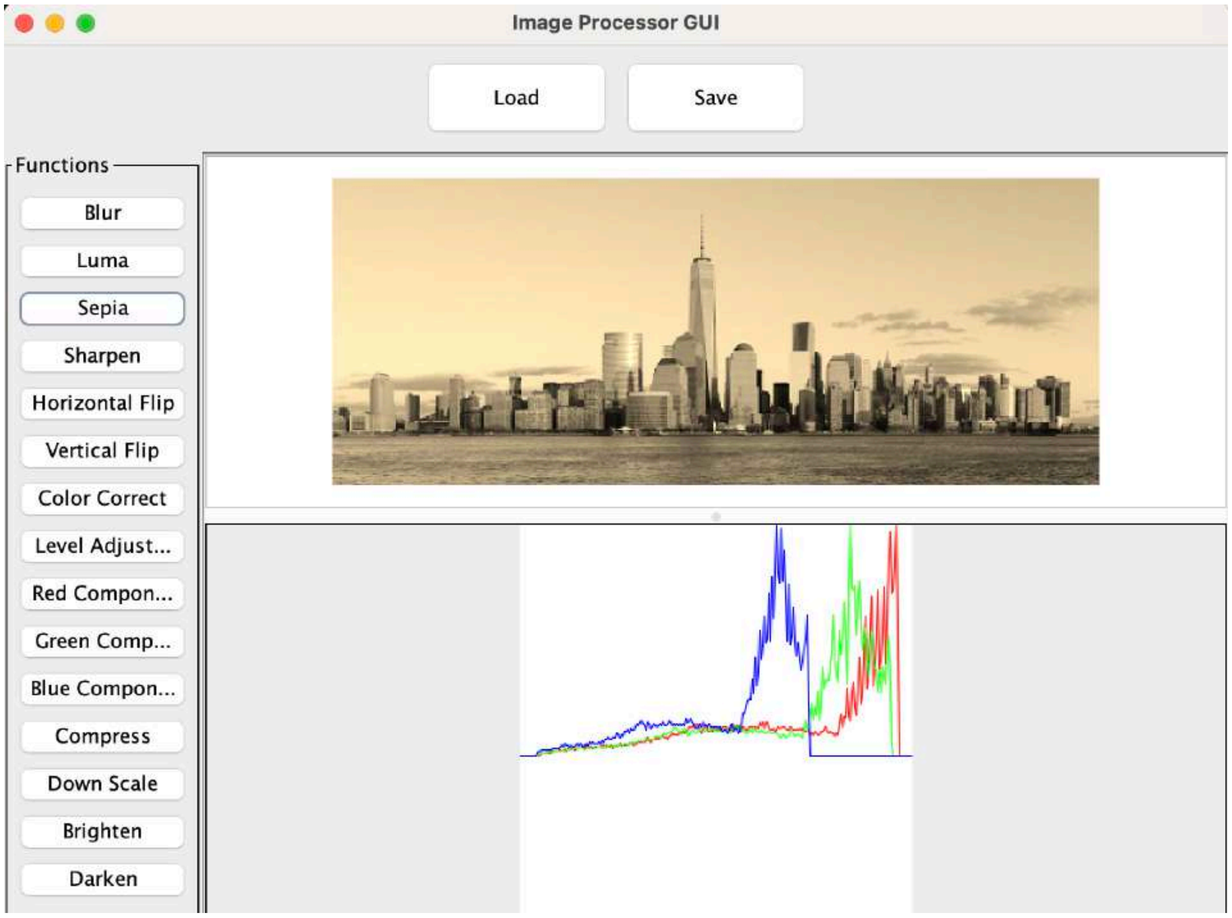
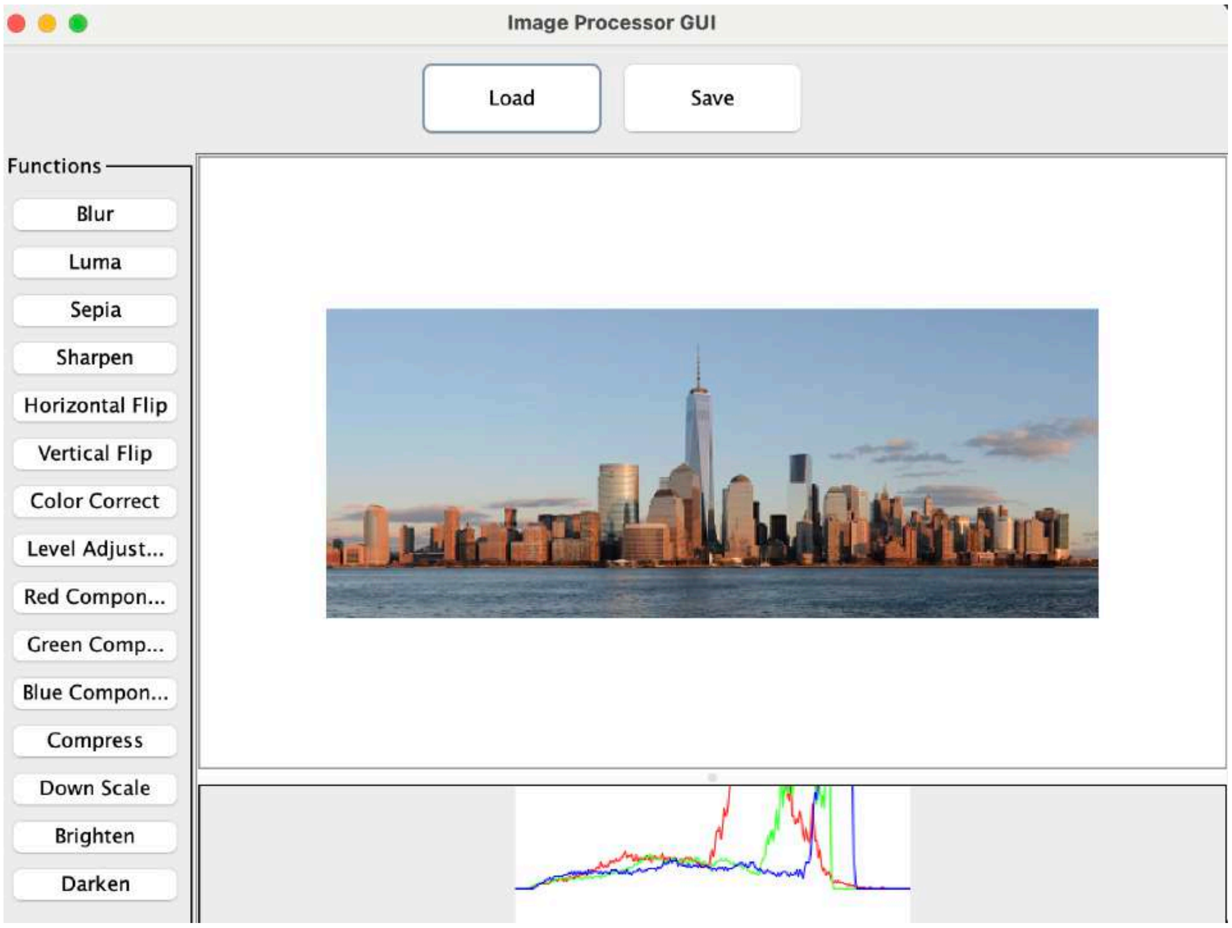
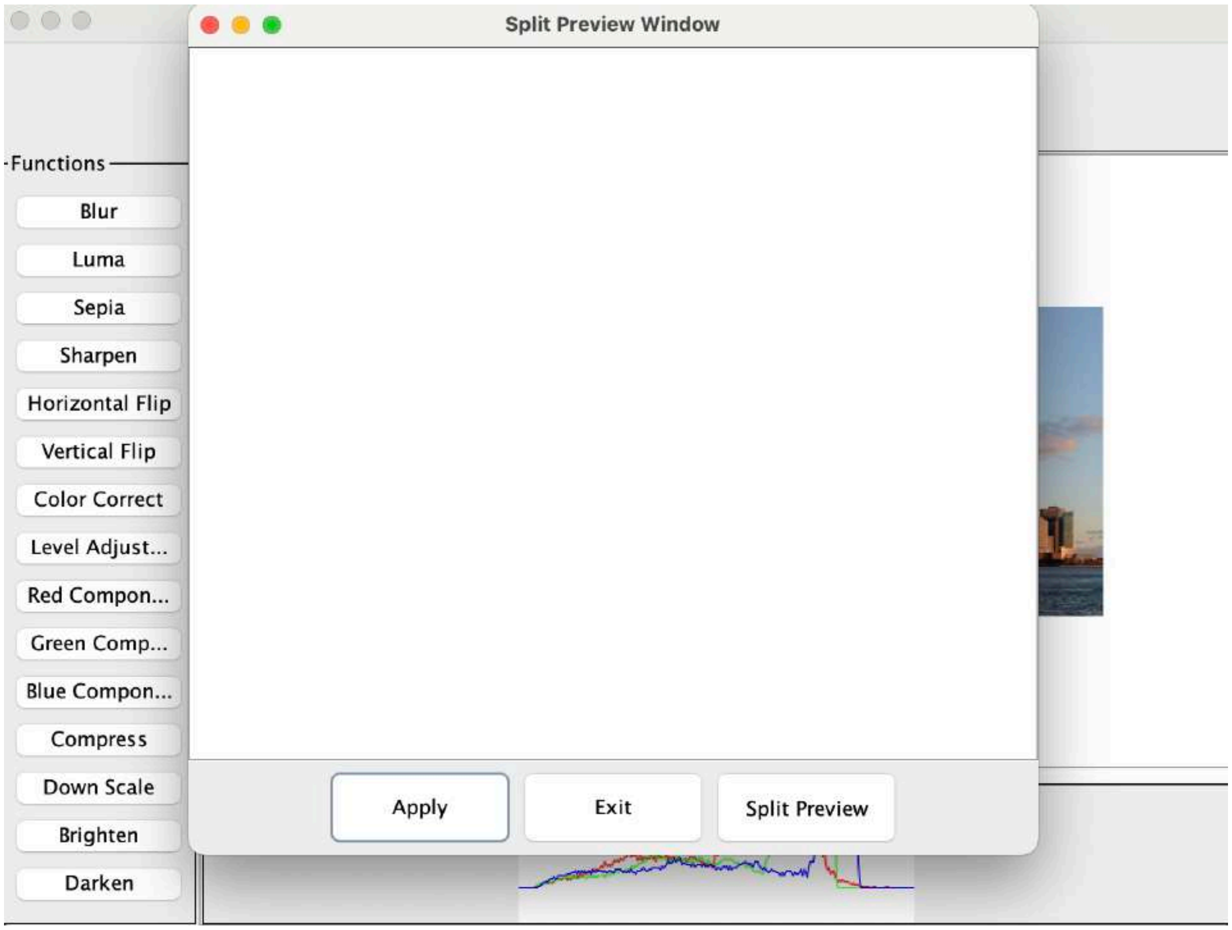


Image Processor GUI
Load
Save
Functions
Blur
Luma
Sepia
Sharpen
Horizontal Flip
Vertical Flip
Color Correct
Level Adjust...
Red Compon...
Green Comp...
Blue Compon...
Compress
Down Scale
Brighten
Darken

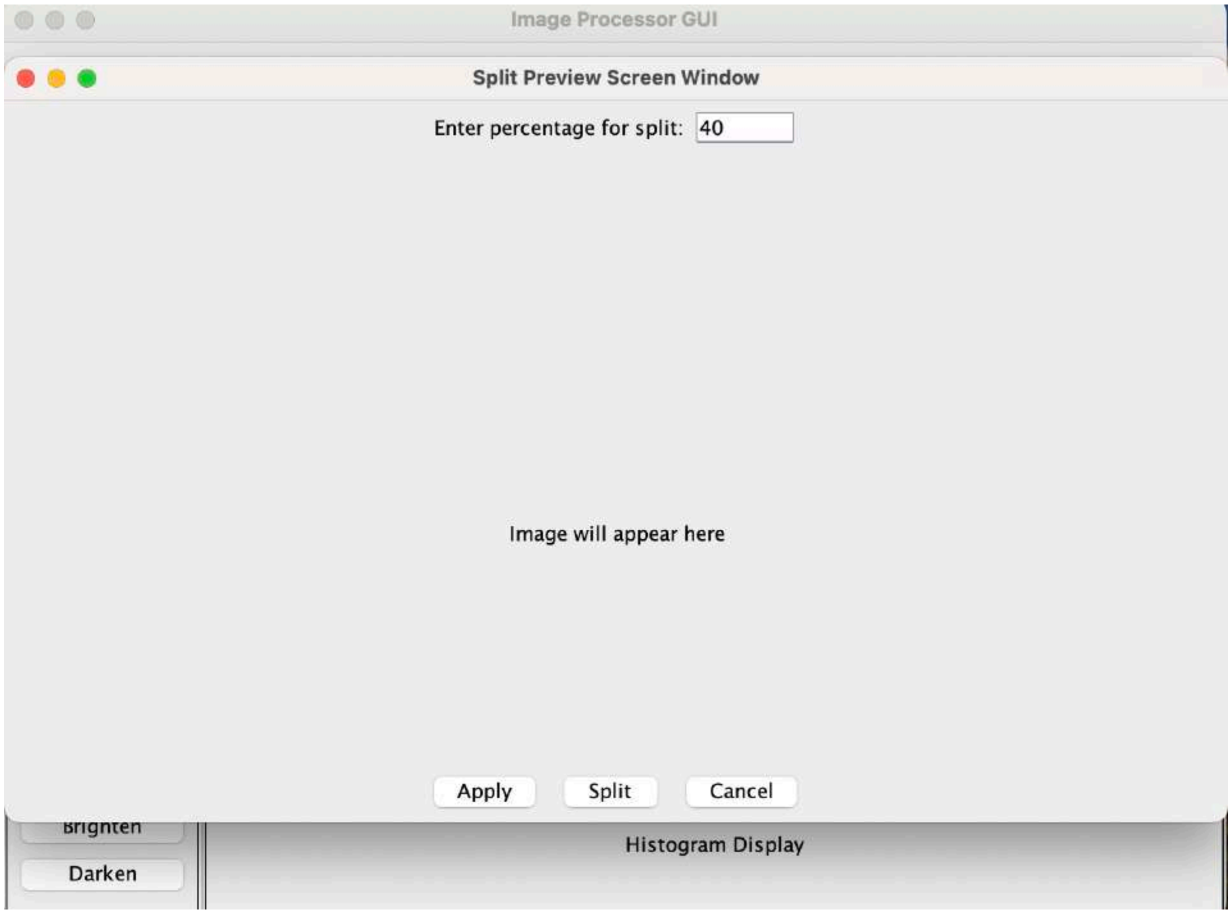
5. **Sharpen Button** It when pressed checked if the image is present on the canvas, if it is present then only it will open the popup window or else will throw an error message.



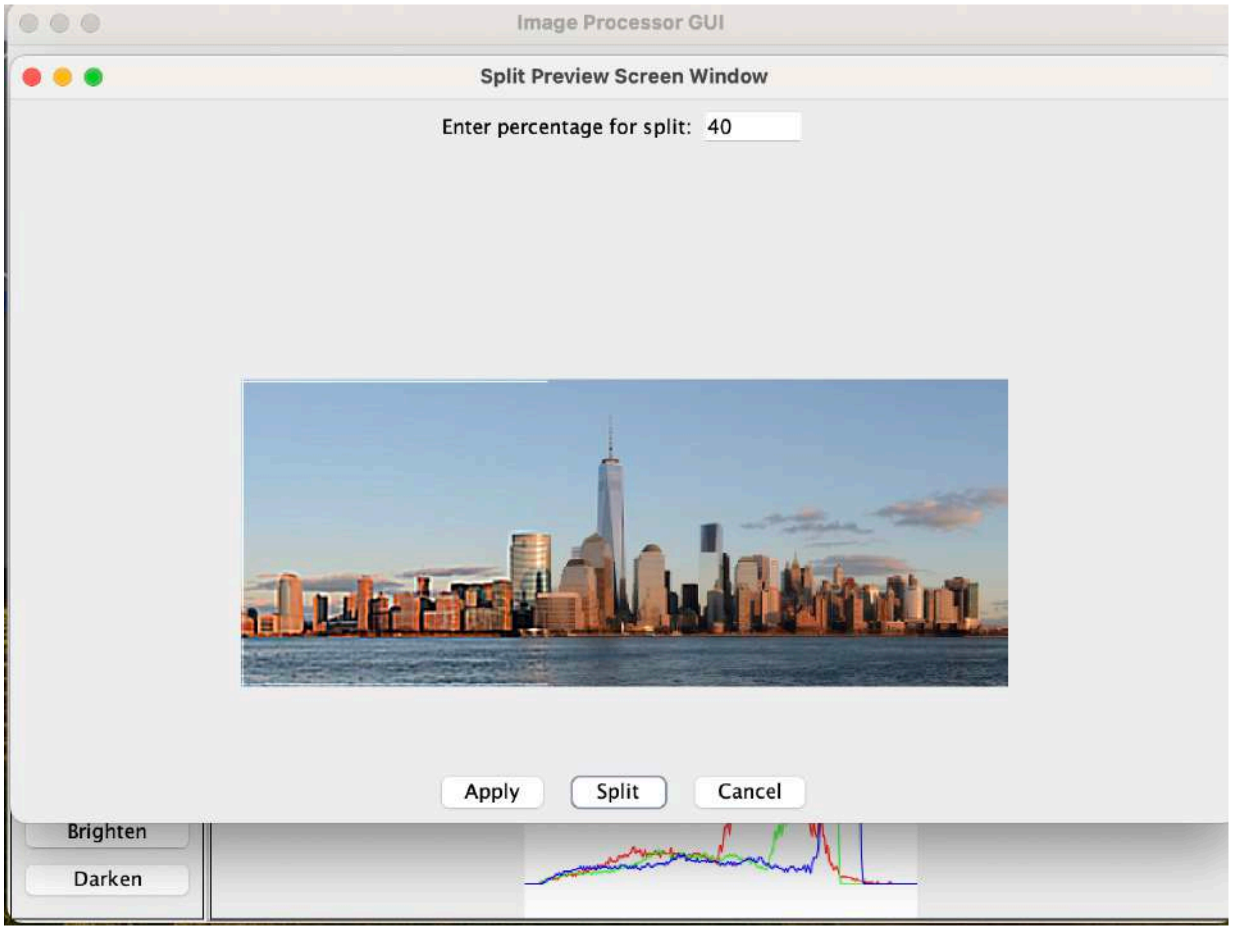
When Sharpen Button is clicked.

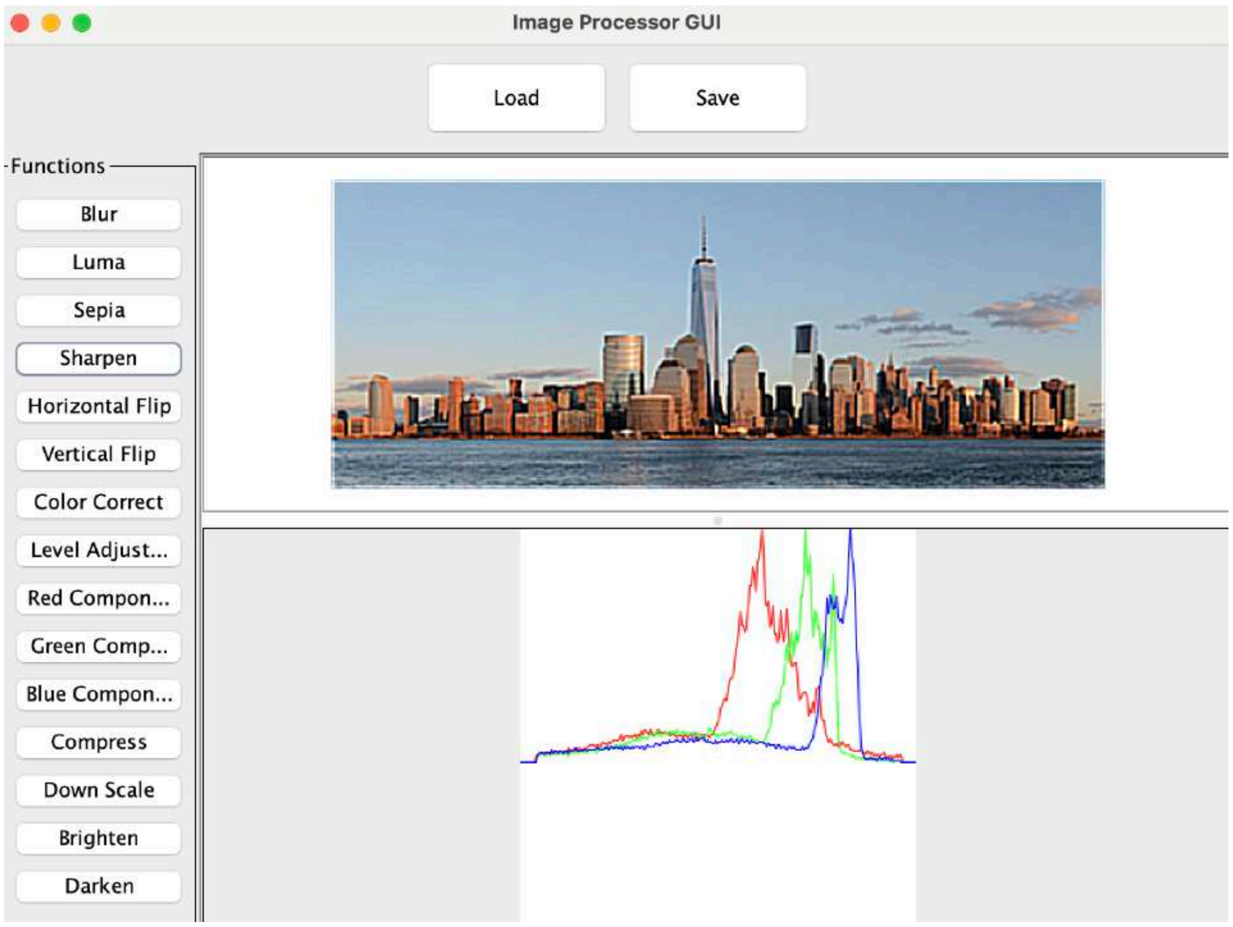


When Split PreviewButton is clicked

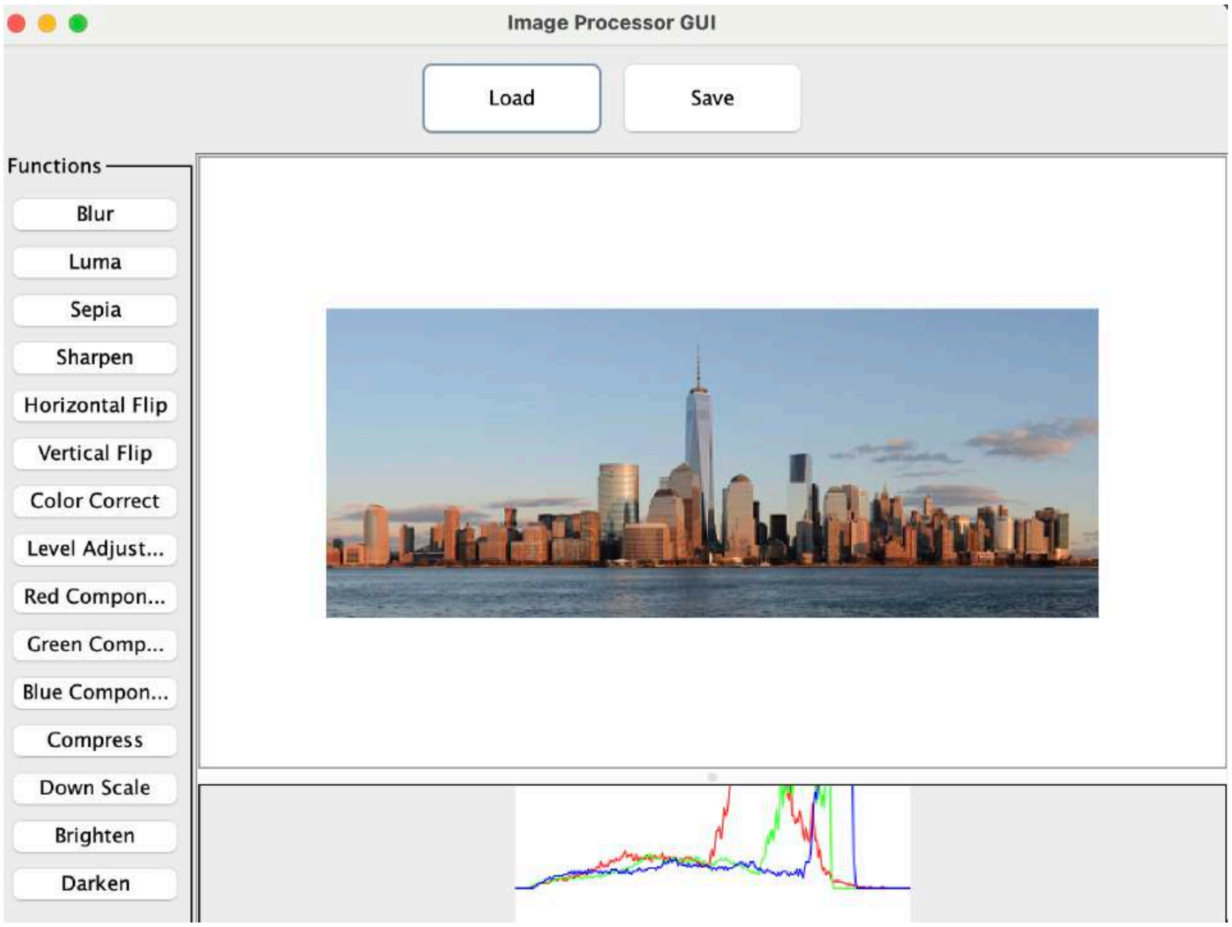


When OK is pressed the split preview can be seen. If cancel is pressed it undo the effect of split preview for that function.

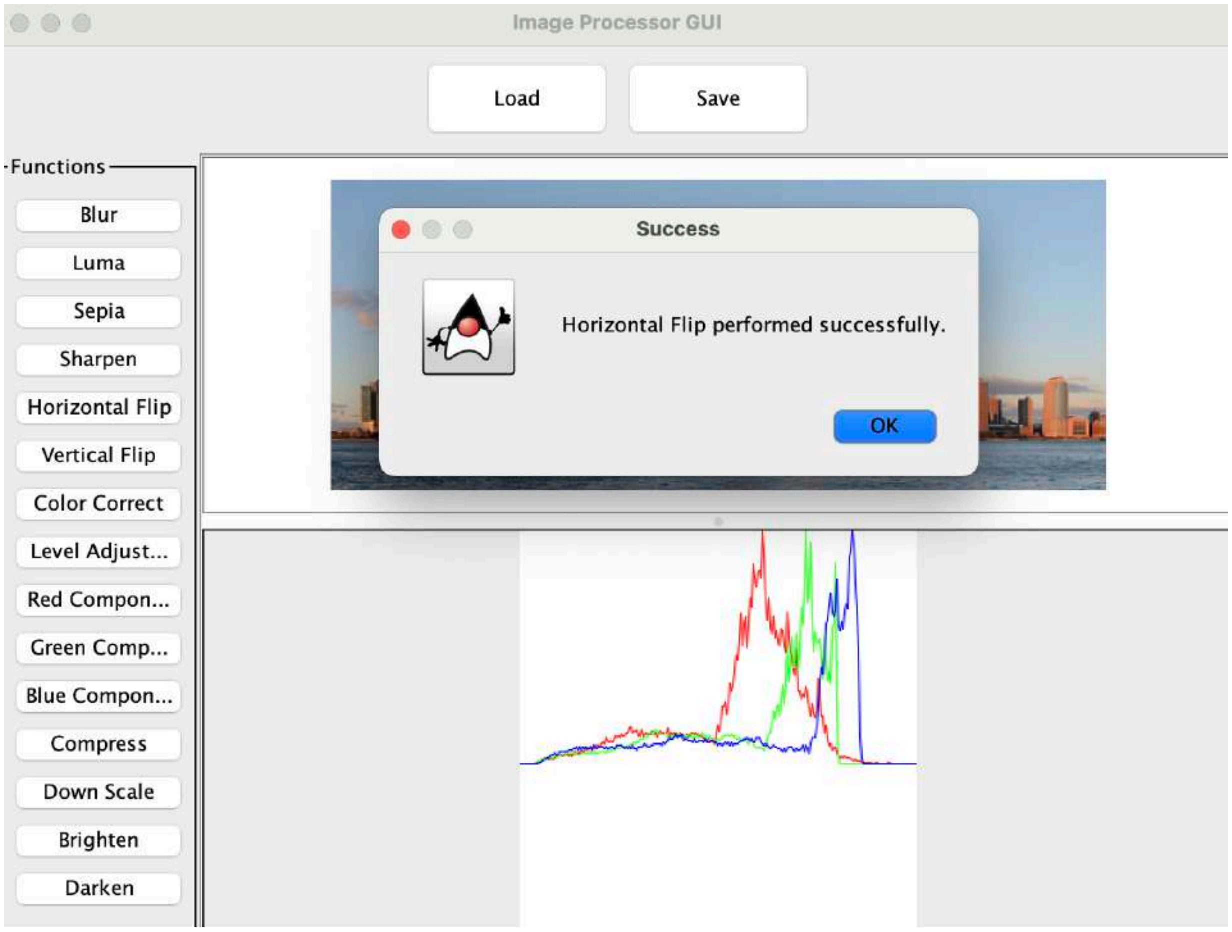




6. **Horizontal-Flip Button** It when clicked will simply rotate the image horizontally.

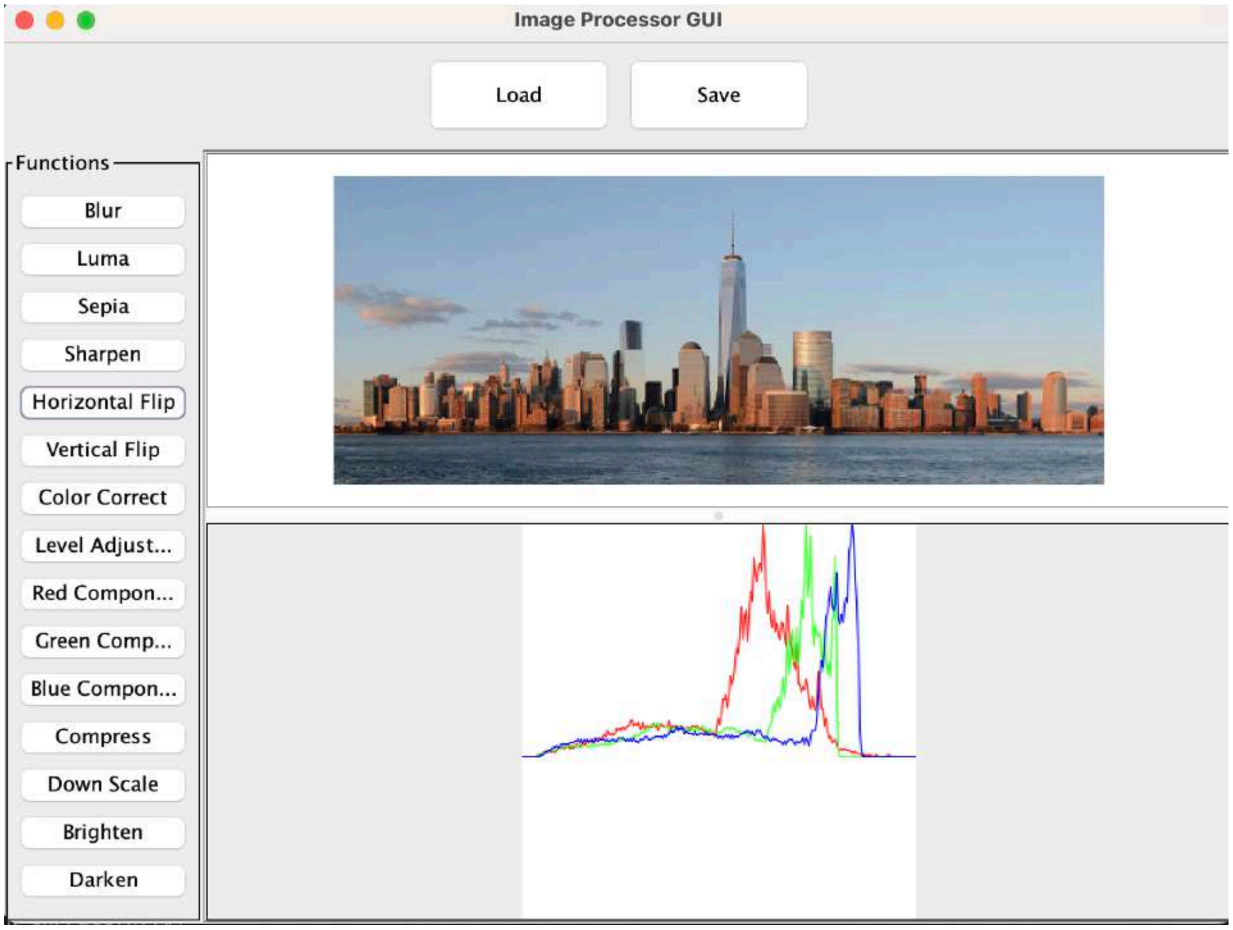


When Horizontal-Flip Button is clicked.

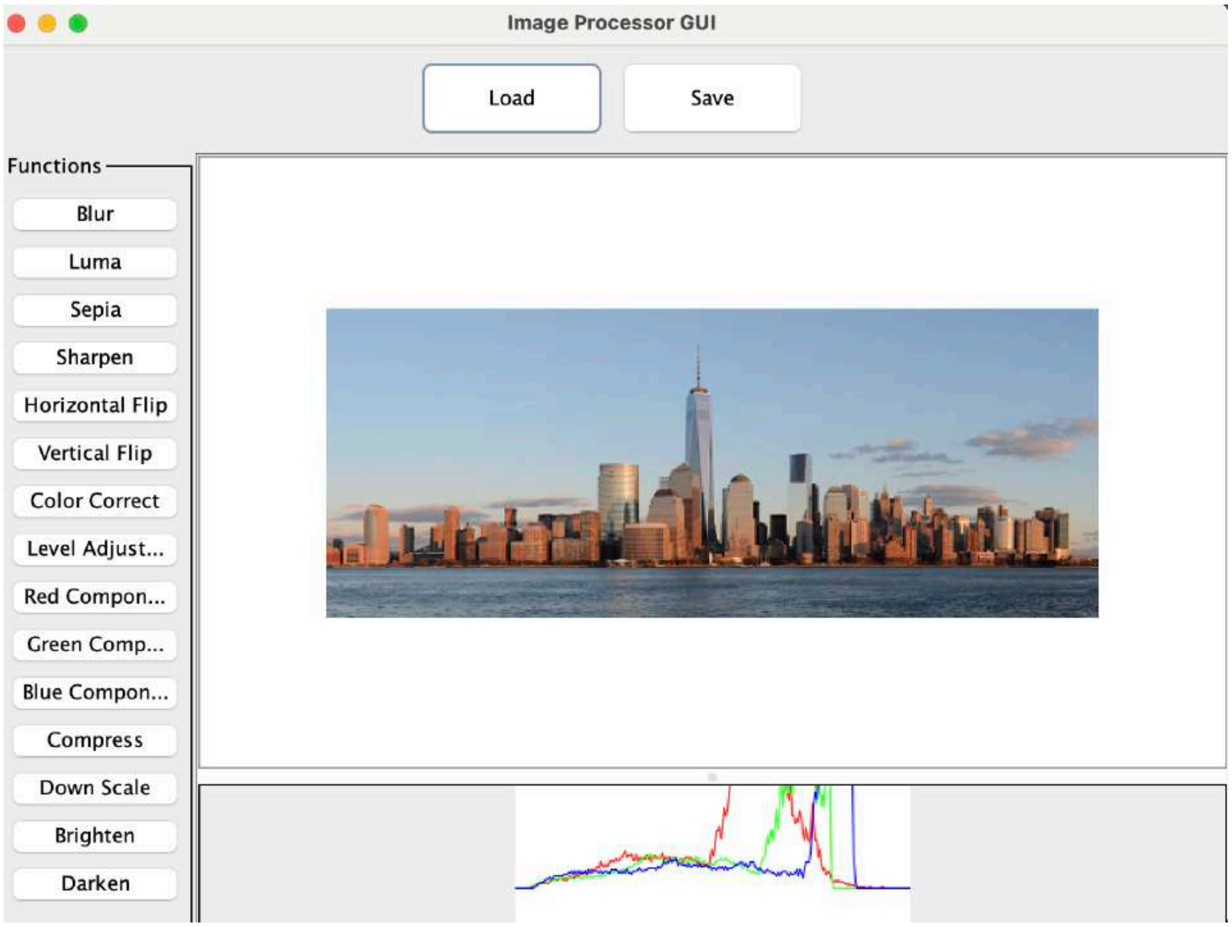


Horizontal Flip performed successfully.

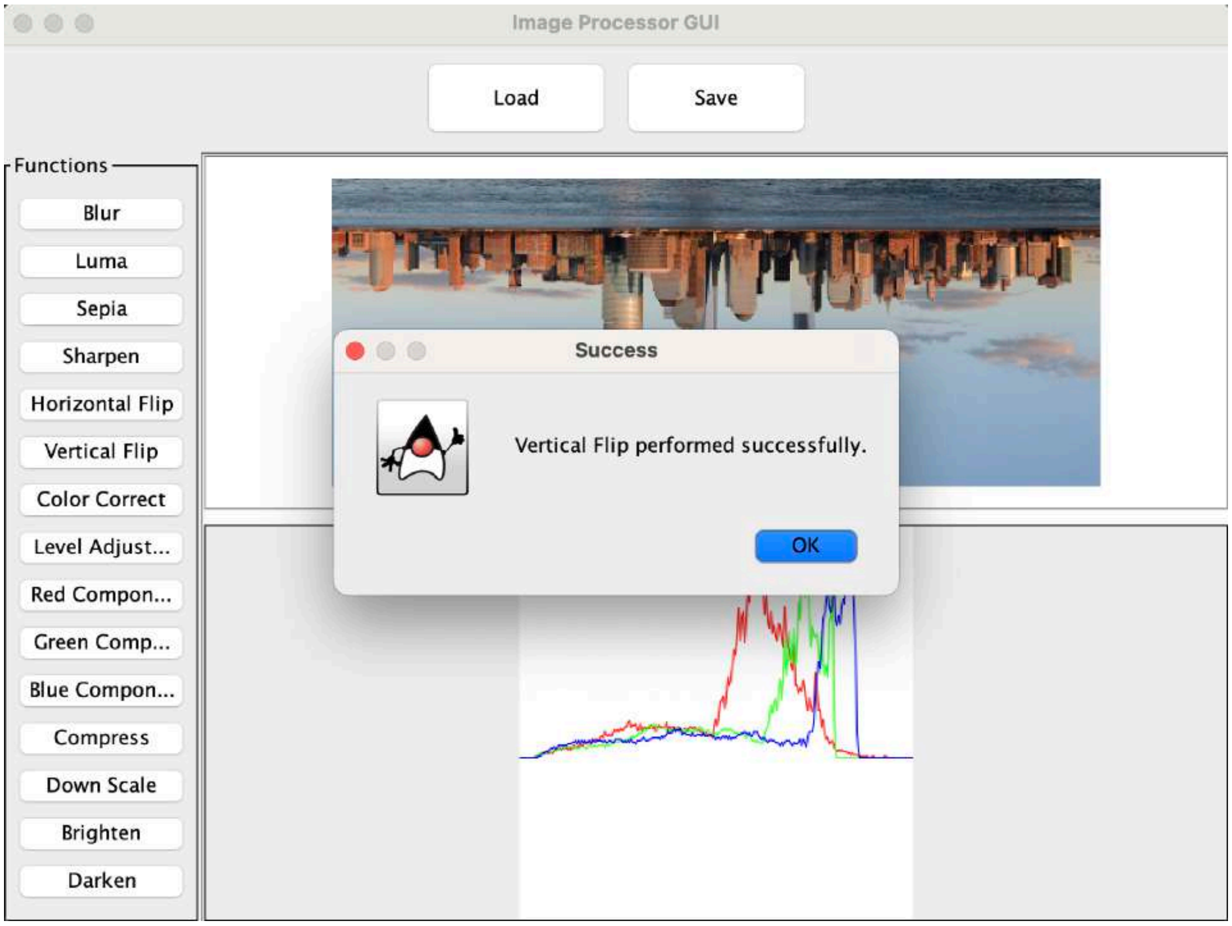
When OK is pressed it closes the dialogue Box.



7. **Vertical-Flip Button** It when clicked will simply rotate the image vertically.

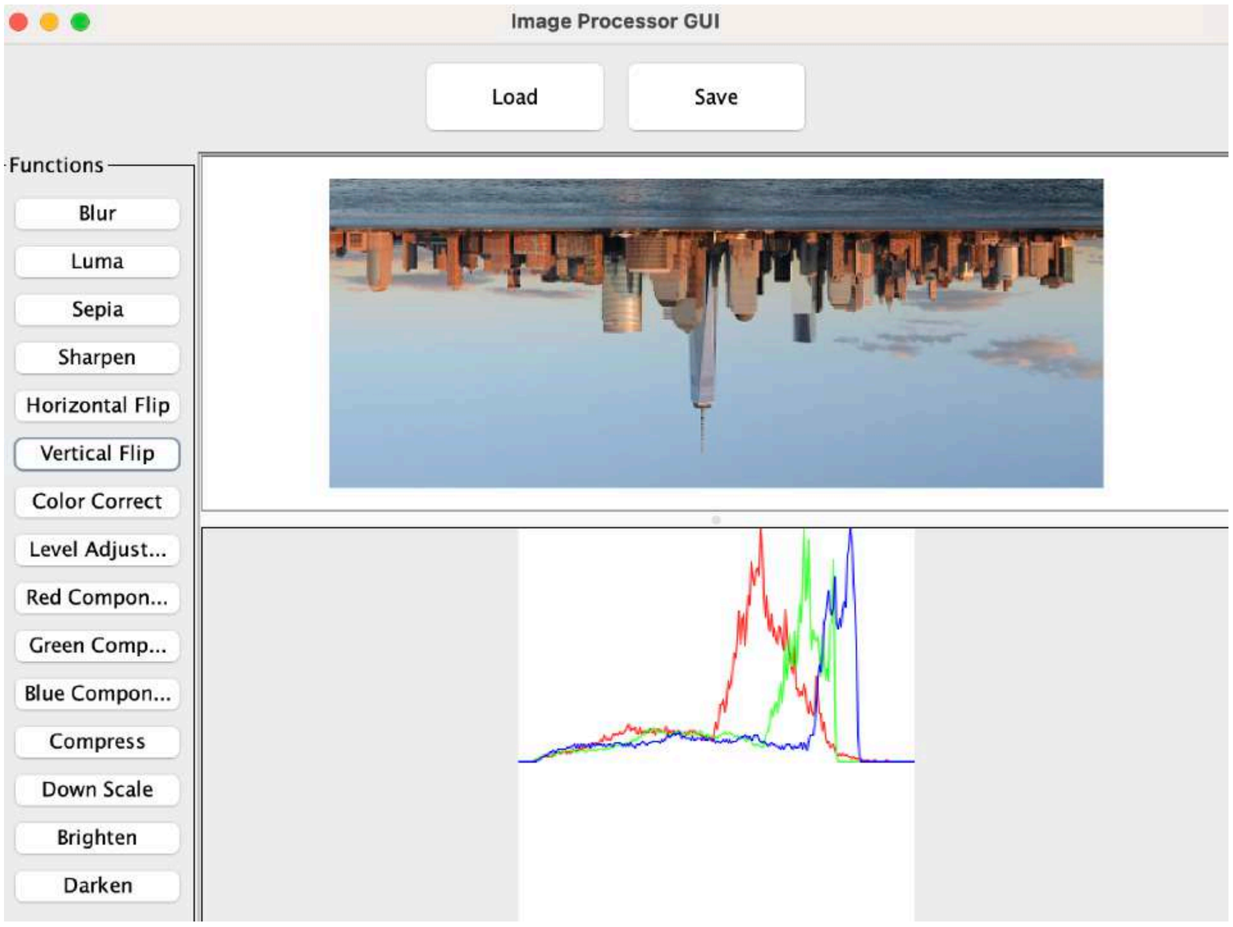


When Vertical-Flip Button is clicked.

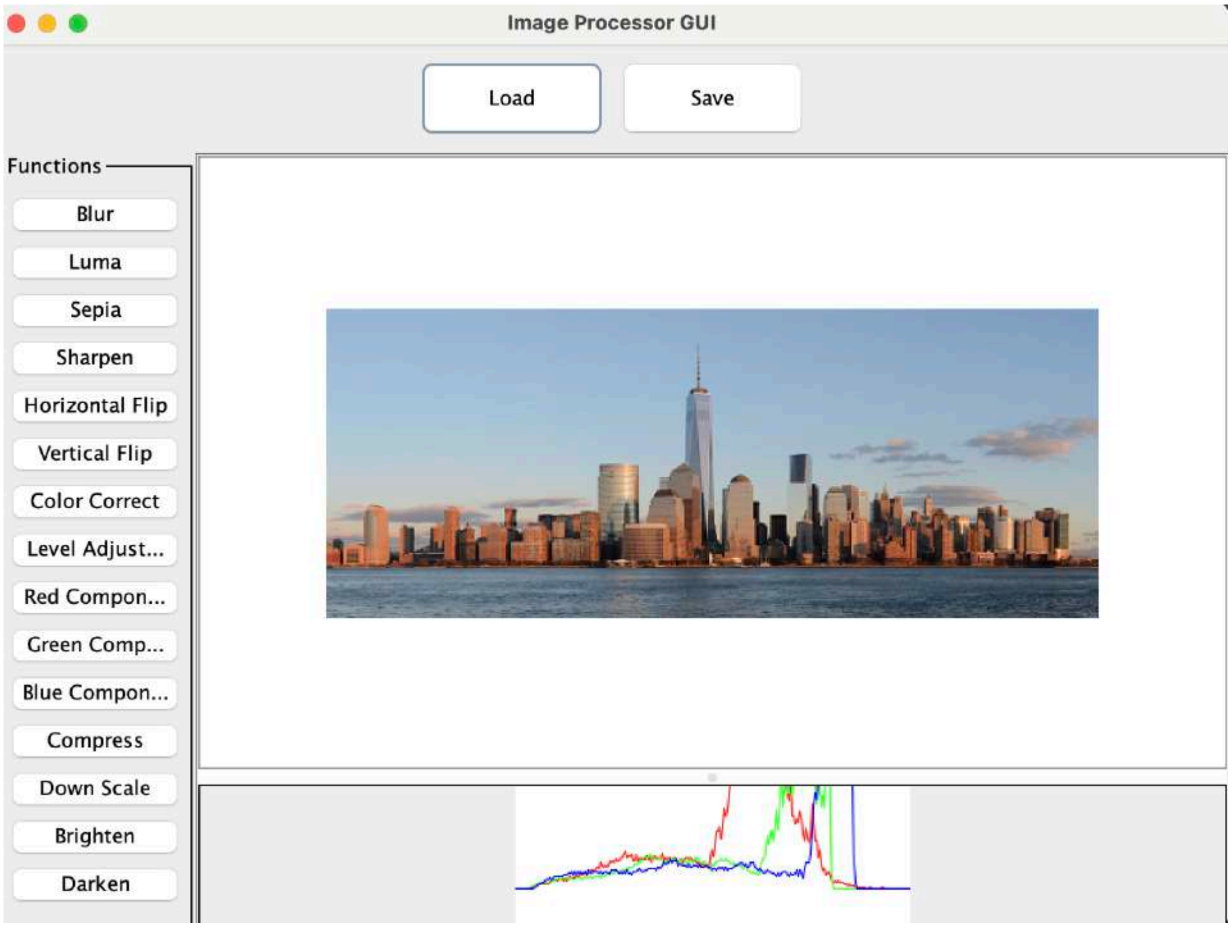


Vertical Flip performed successfully.

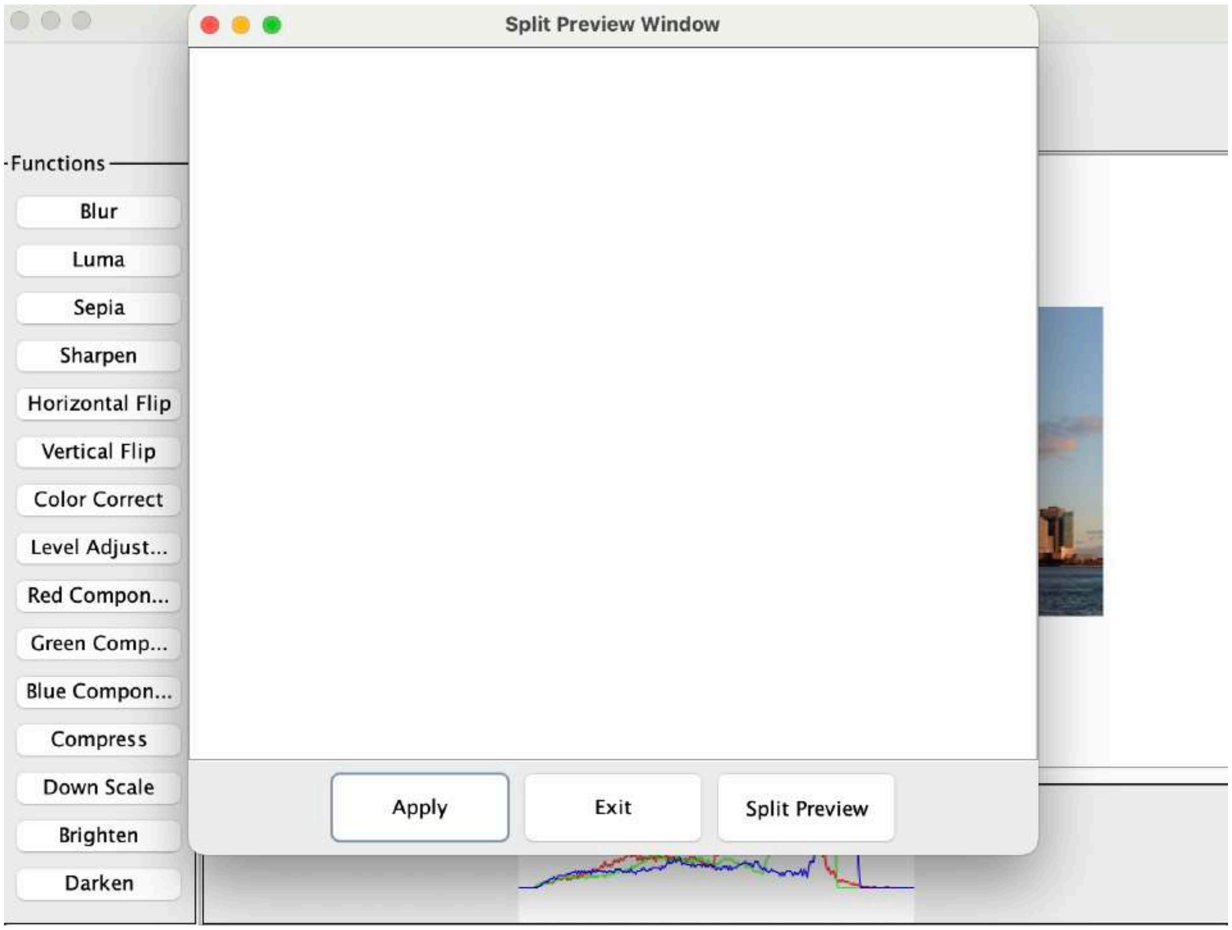
When OK is pressed it closes the dialogue Box.



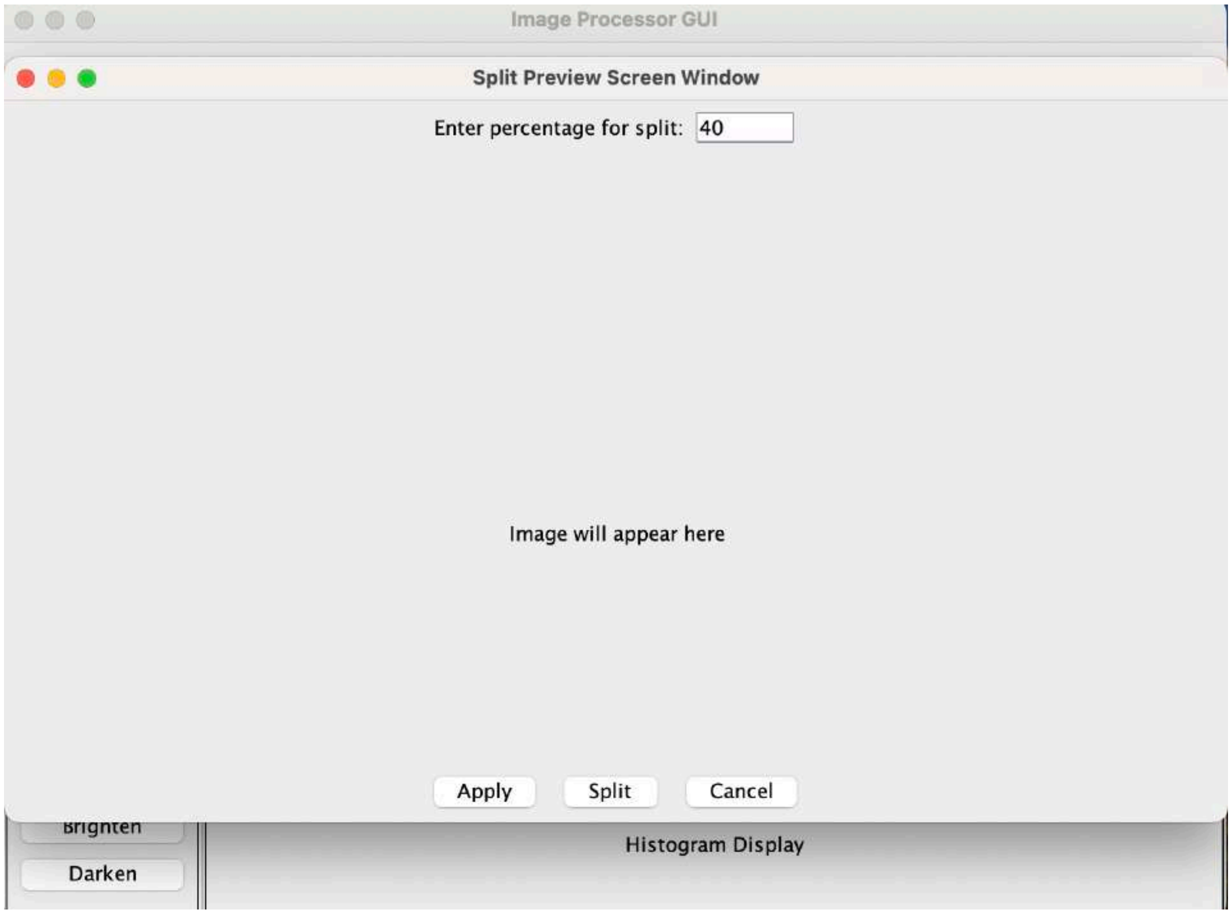
8. **Color-Component Button** It when clicked will open the popup window to ask to apply the function, split preview or exit.

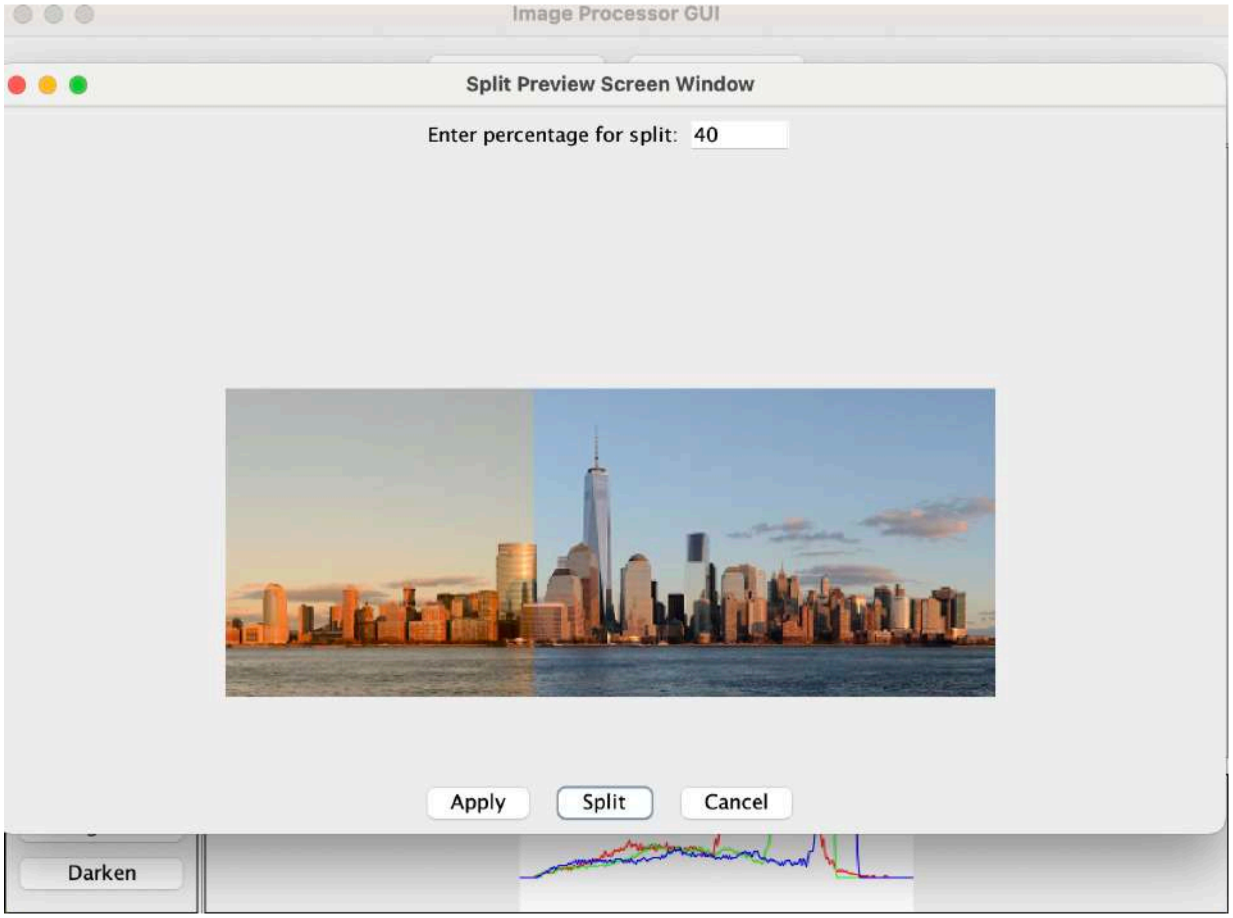


When Color-Correct Button is clicked.

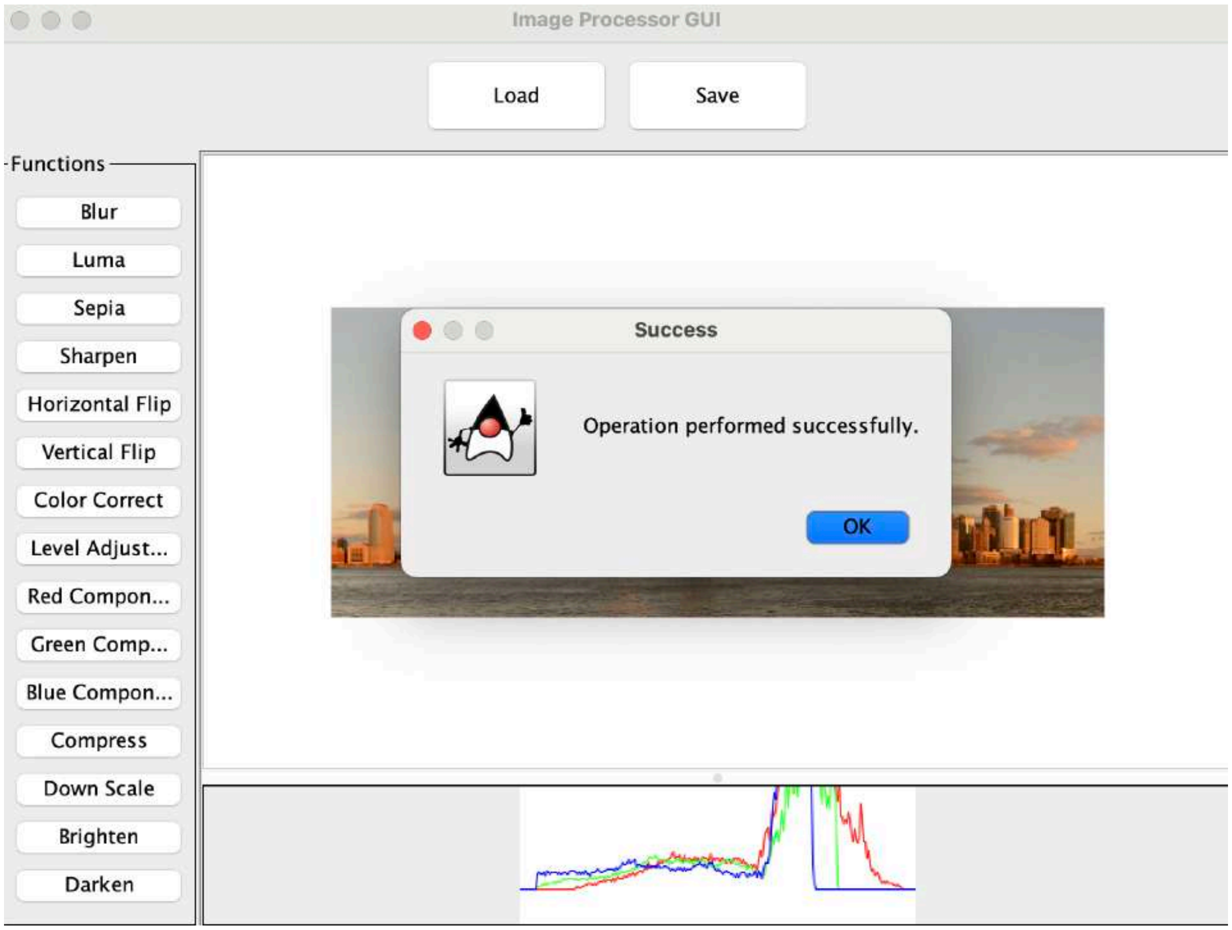


When Split Preview Button is clicked

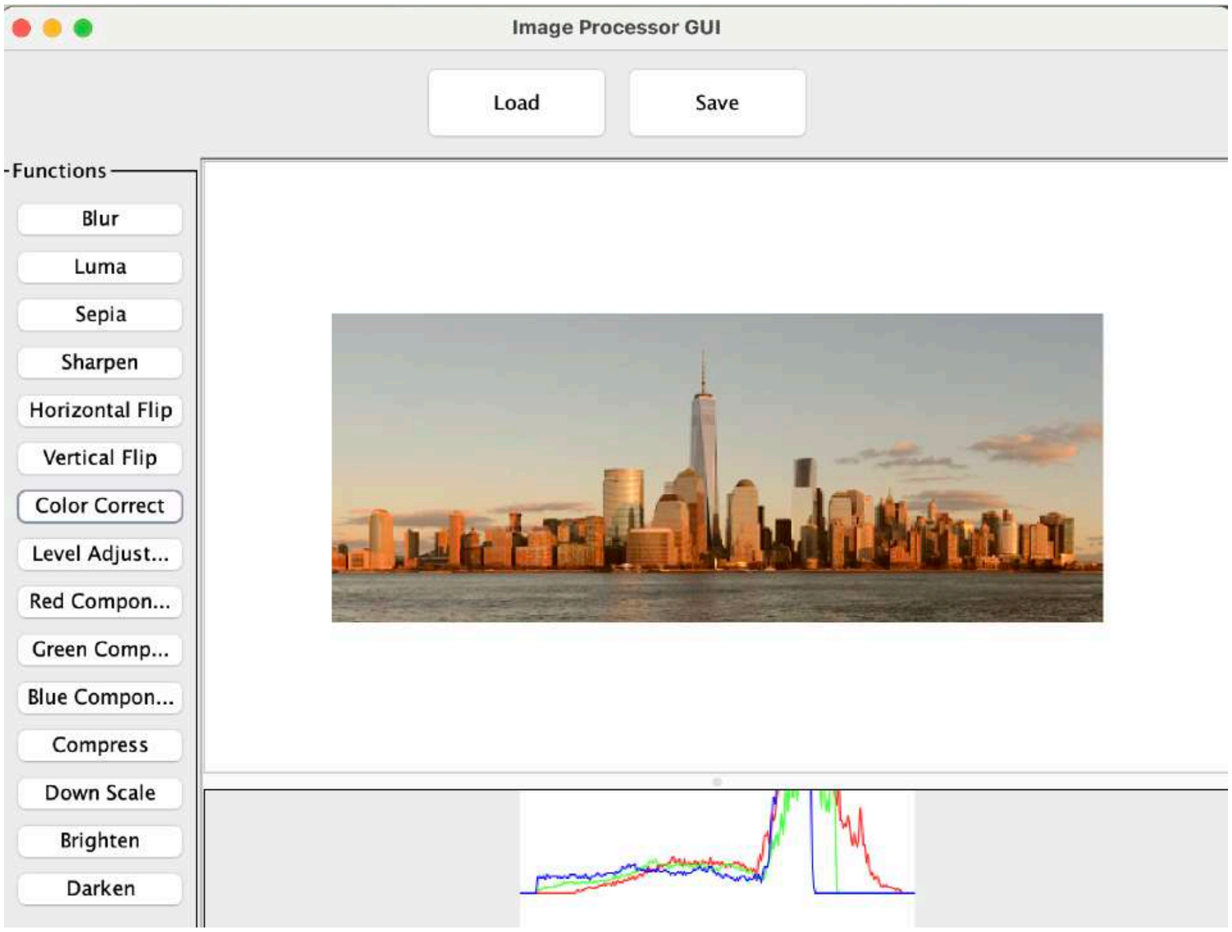




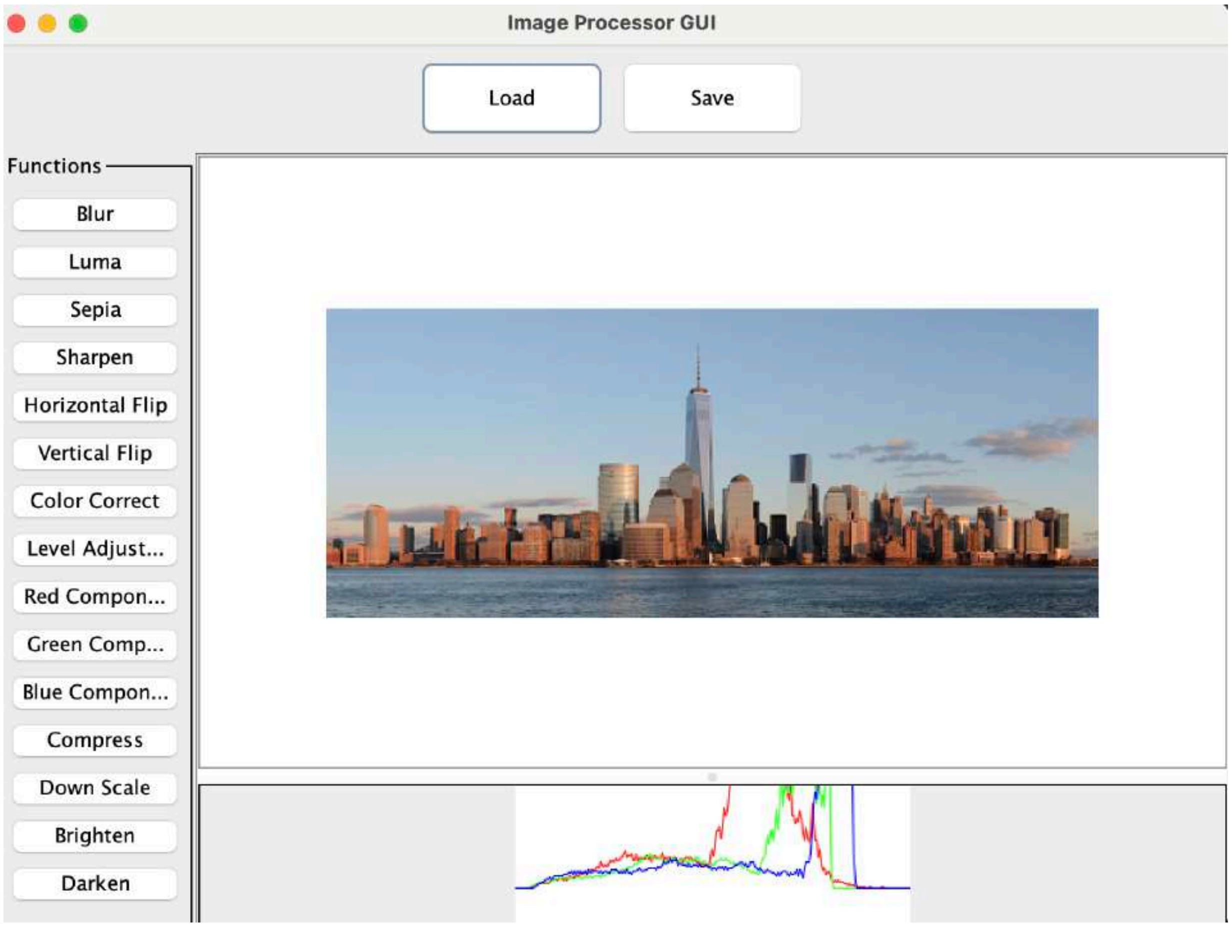
When apply button is clicked



Operation performed successfully.



9. **Level-Adjust Button** When the level adjust button is clicked a pop-up window for split is opened.The popup first asks for the percentage input and then ask for the black, mid and white values, which must be in ascending order and between 0 and 255.



When the LevelAdjust button is clicked

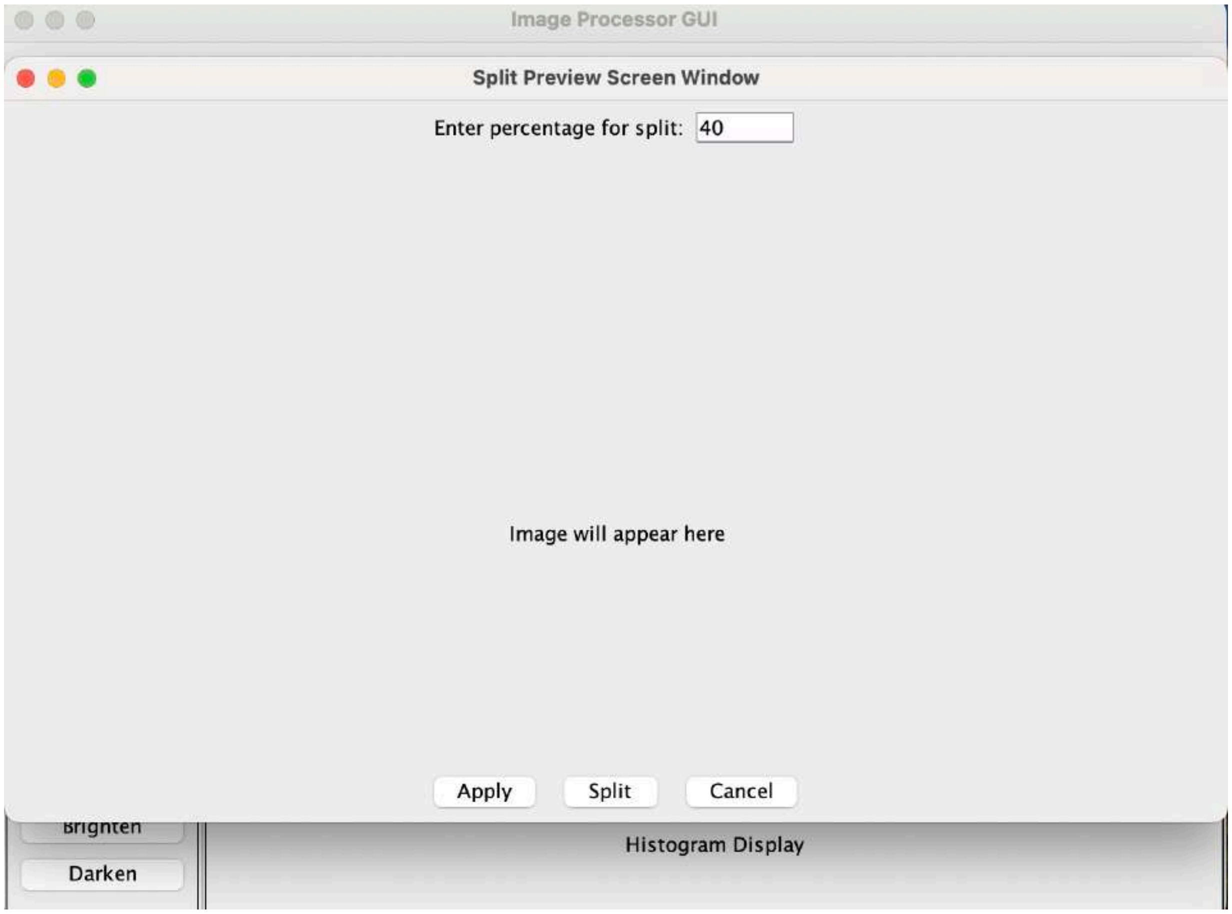


Image Processor GUI
Split Preview Screen Window
Enter percentage for split: 40
Image will appear here
Apply
Split
Cancel
Brighten
Darken
Histogram Display

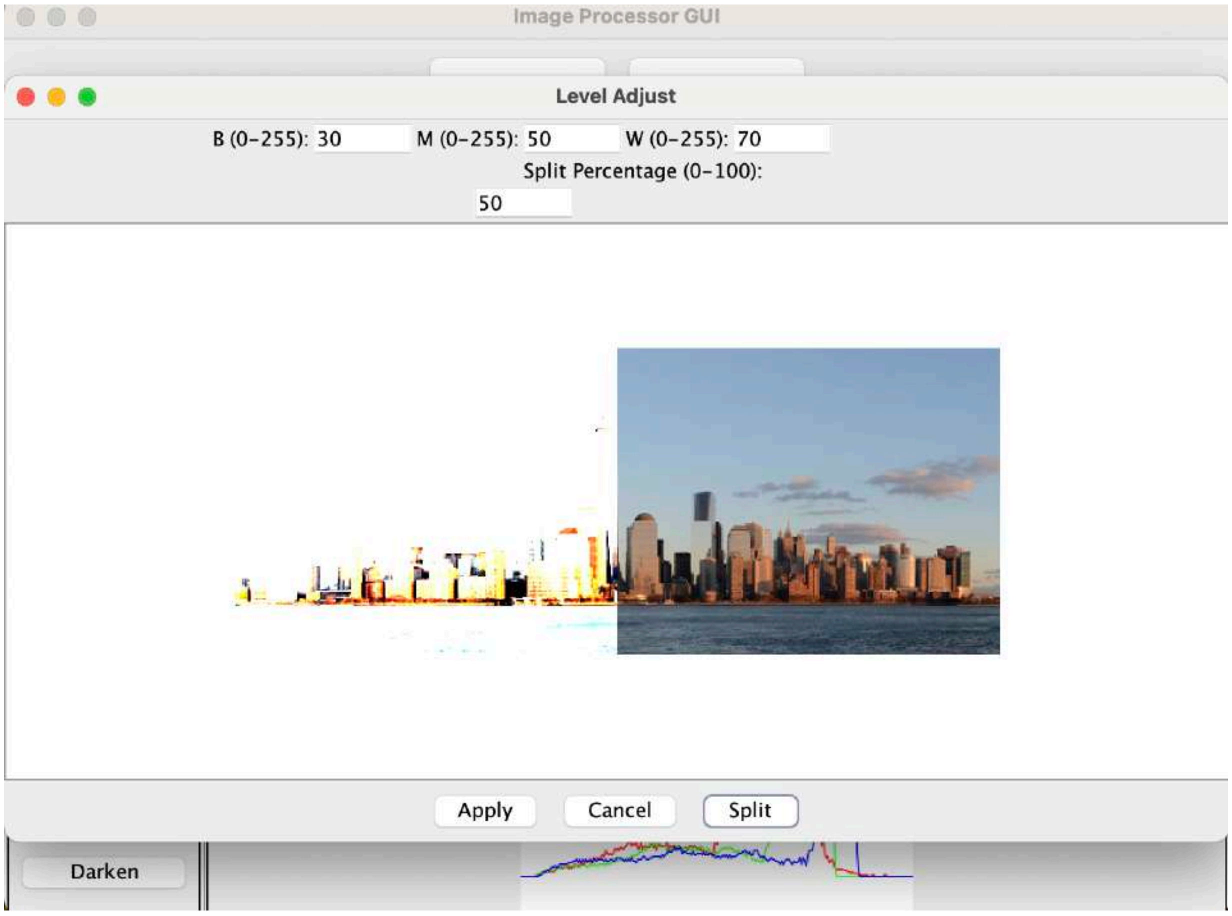
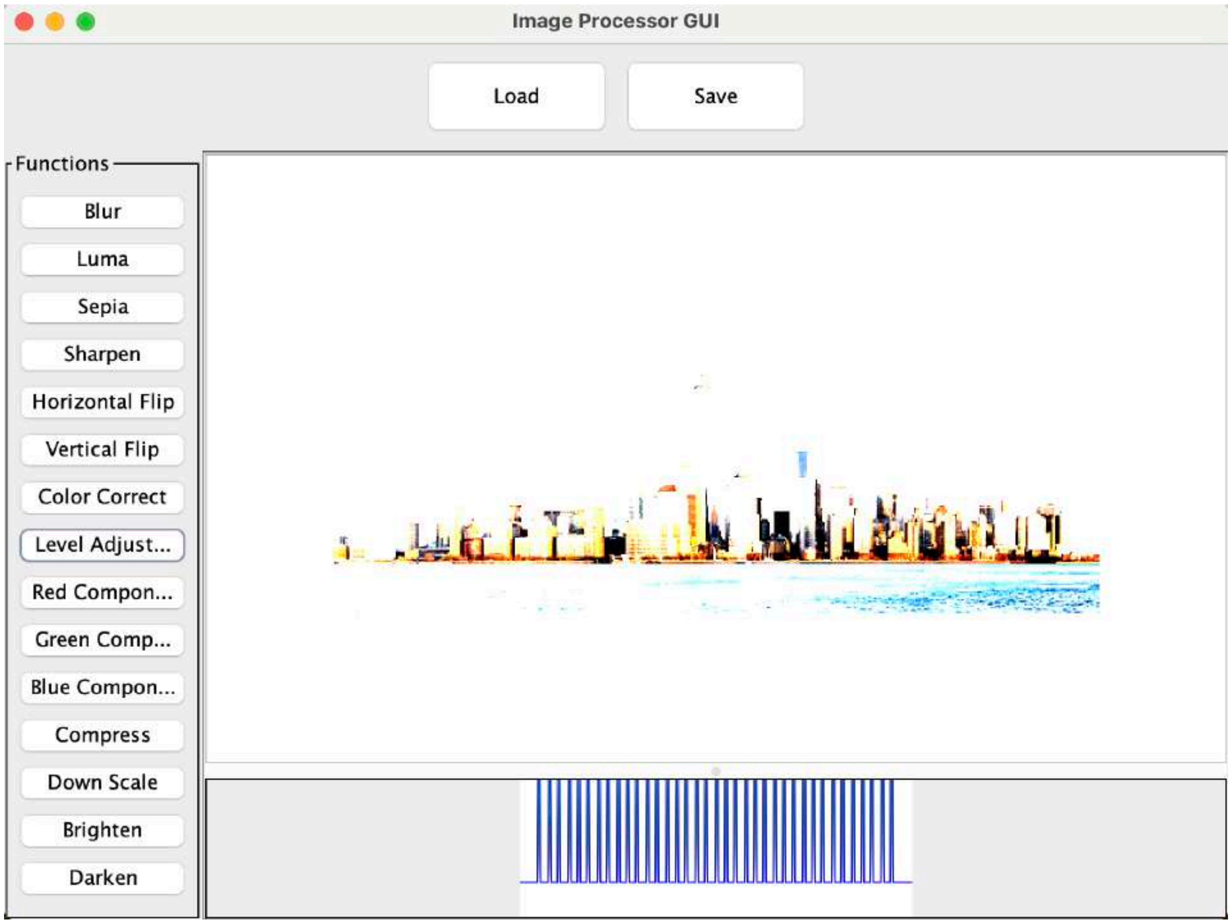
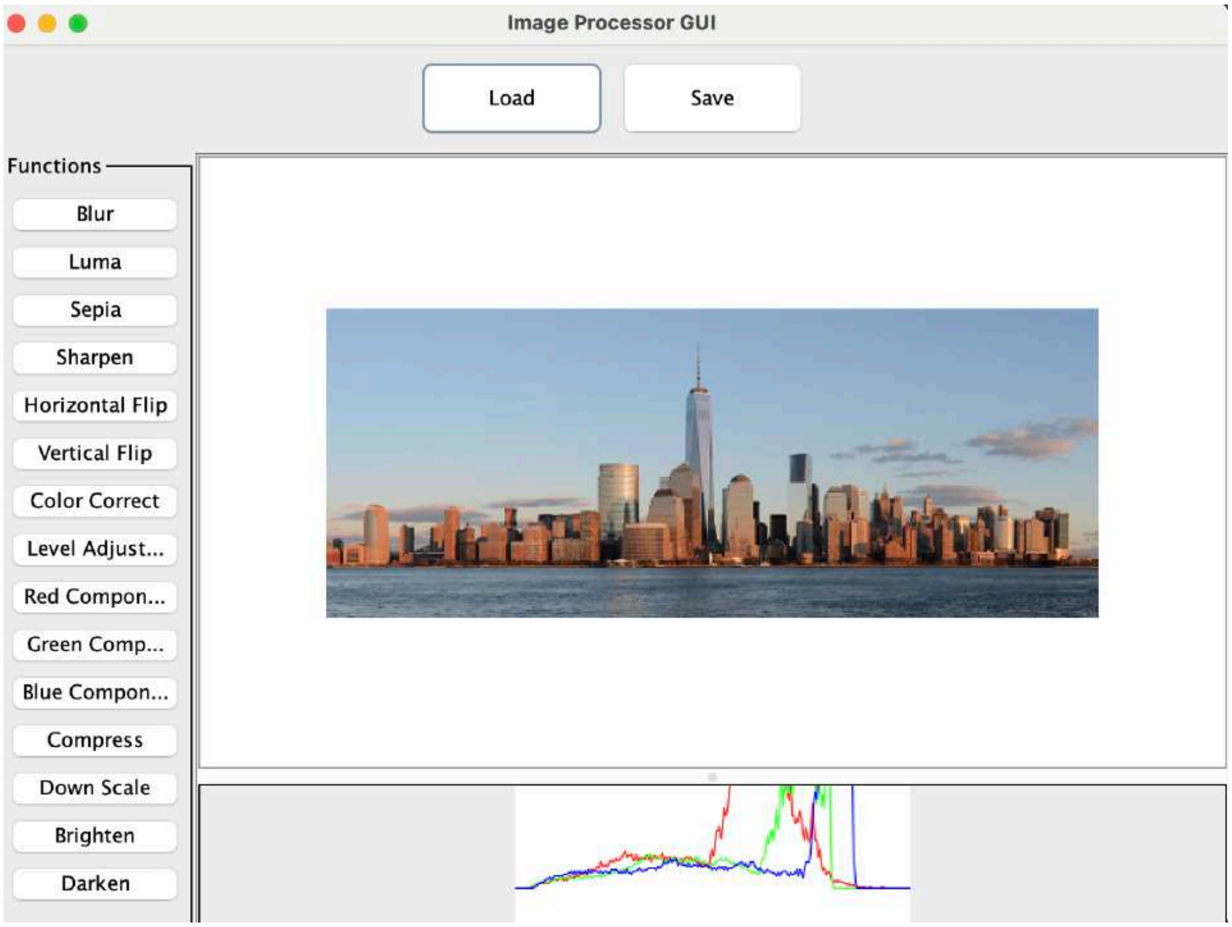


Image Processor GUI
Level Adjust
B (0-255): 30
M (0-255): 50
W (0-255): 70
Split Percentage (0-100):
50
Apply
Cancel
Split
Darken

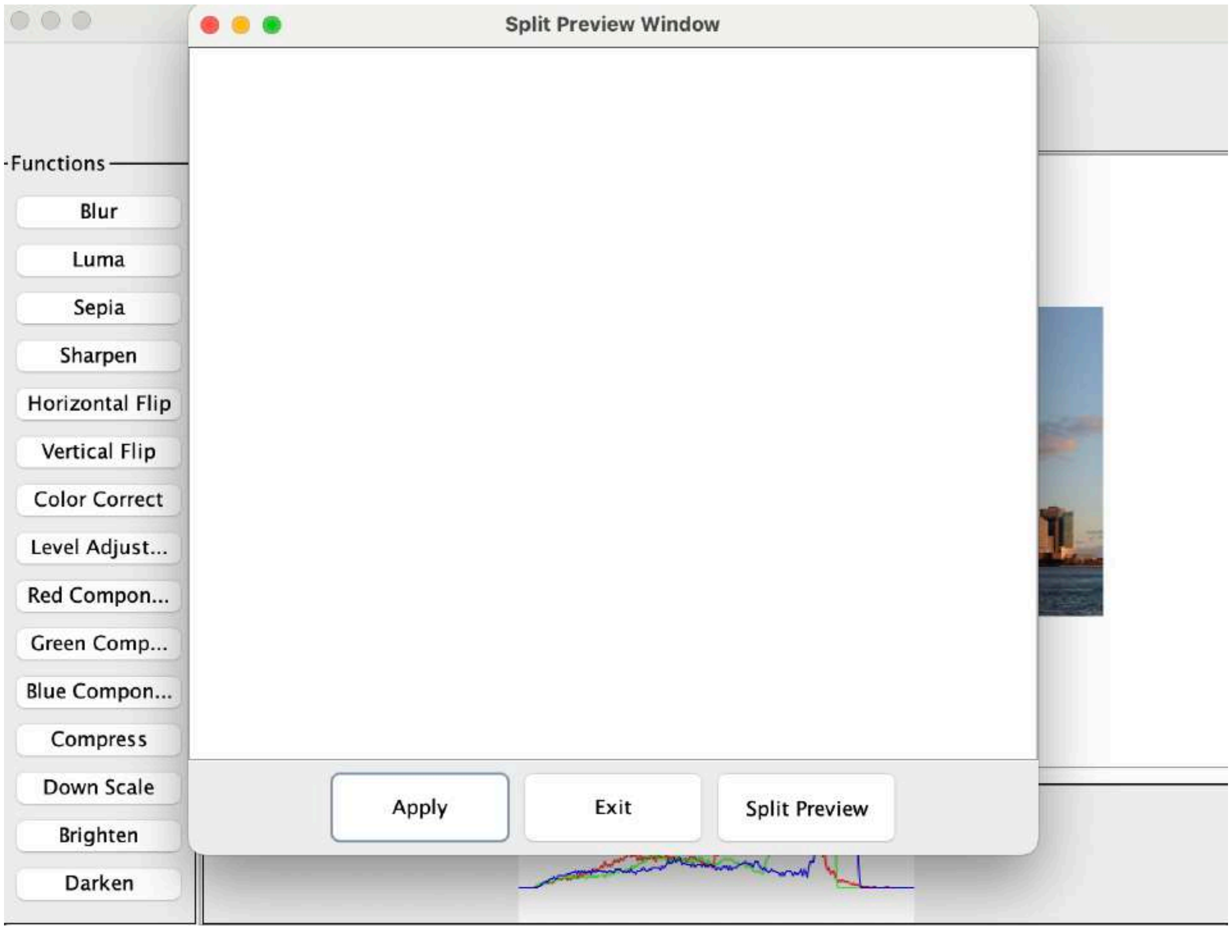
When Apply button is clicked.



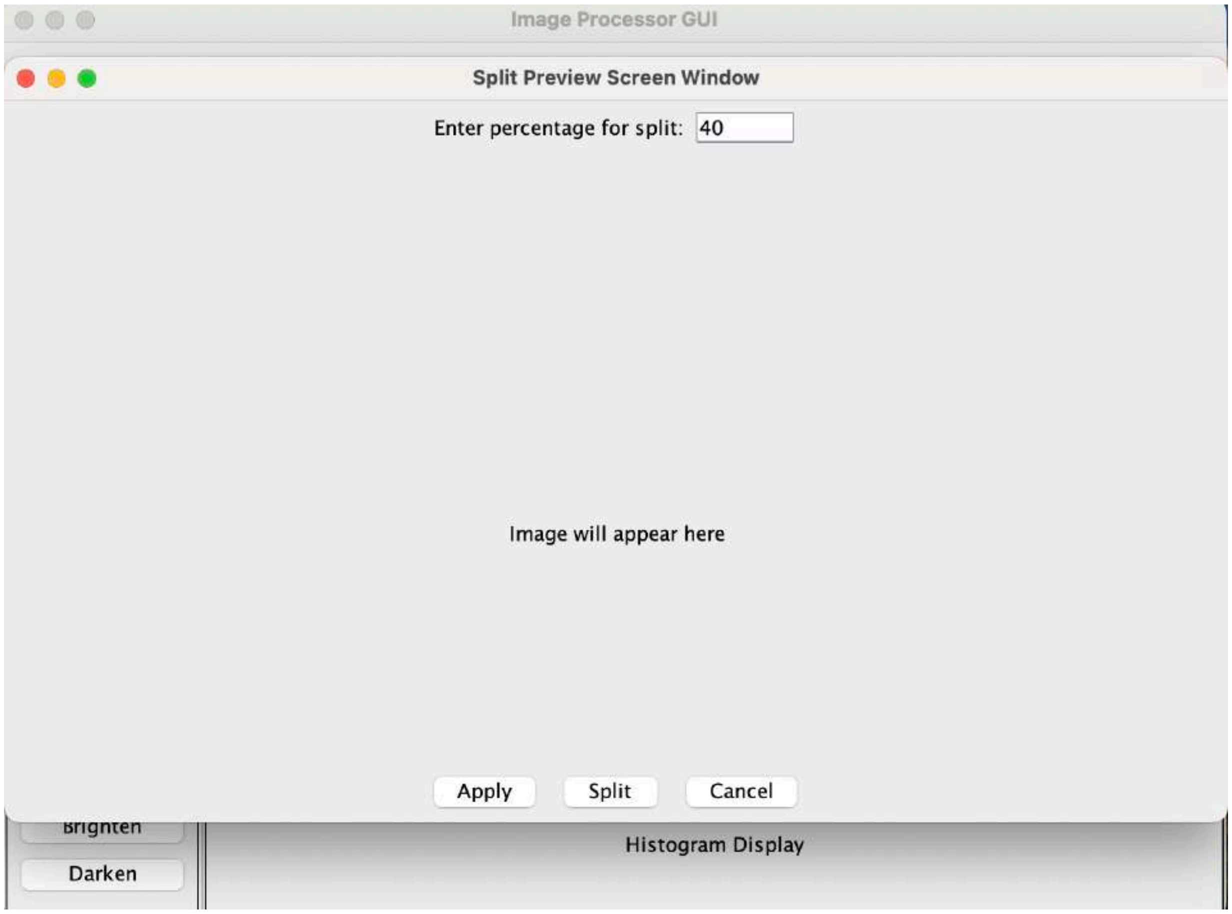
10. **Red-Component Button** When red component button is pressed a popup opens which shows the split preview of the images and has options to apply it to the whole image or exit.



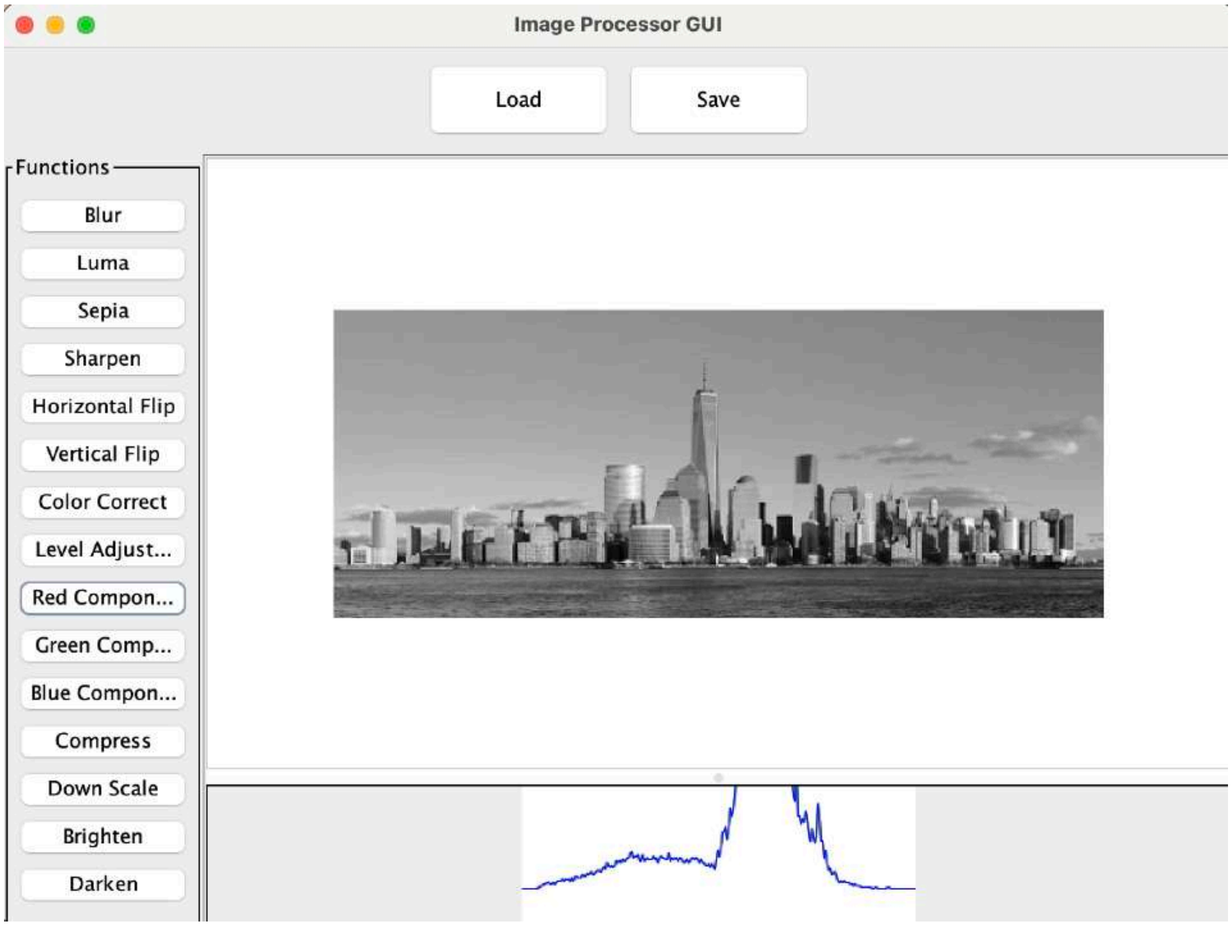
When the LevelAdjust button is clicked



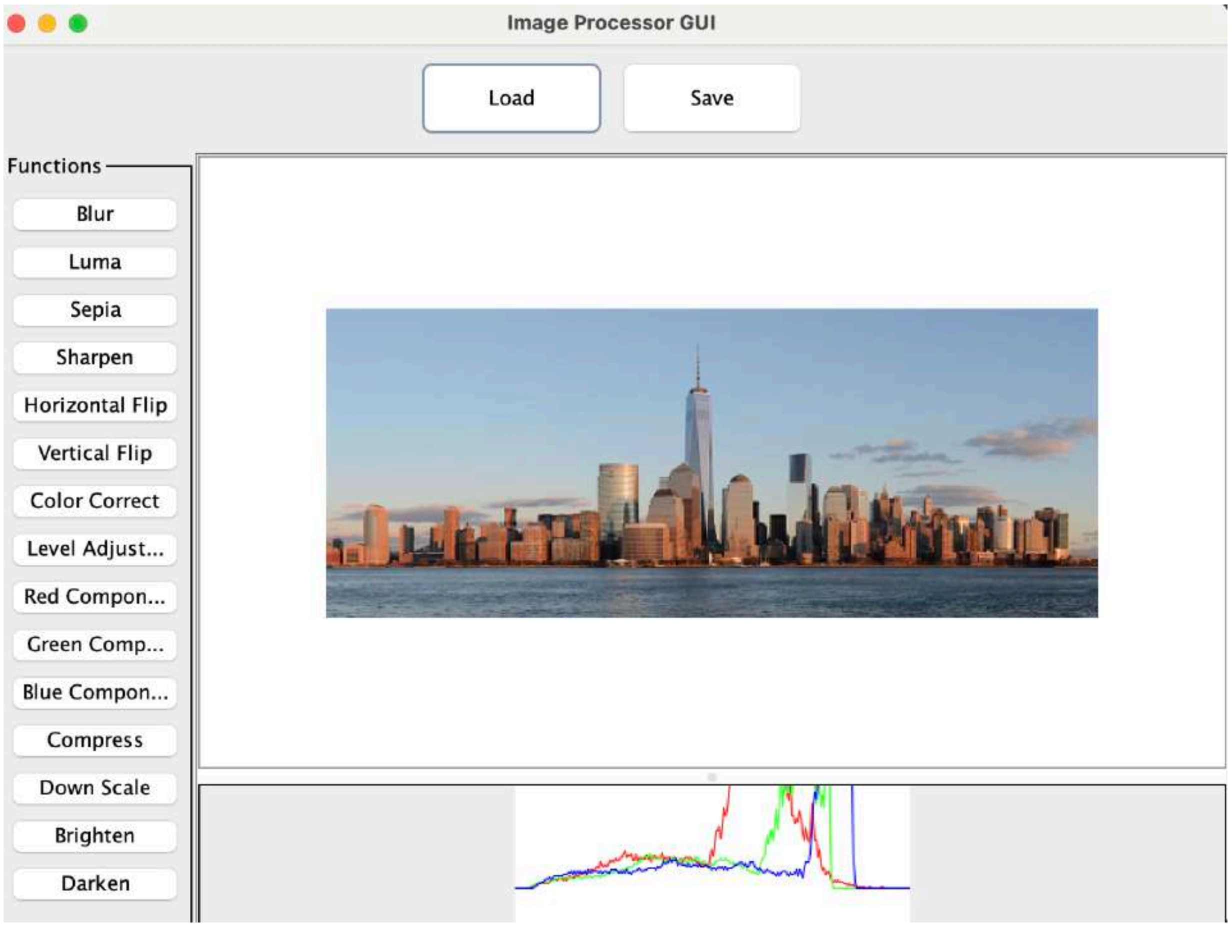
Prompt for entering the percentage is shown when clicked on Split Preview



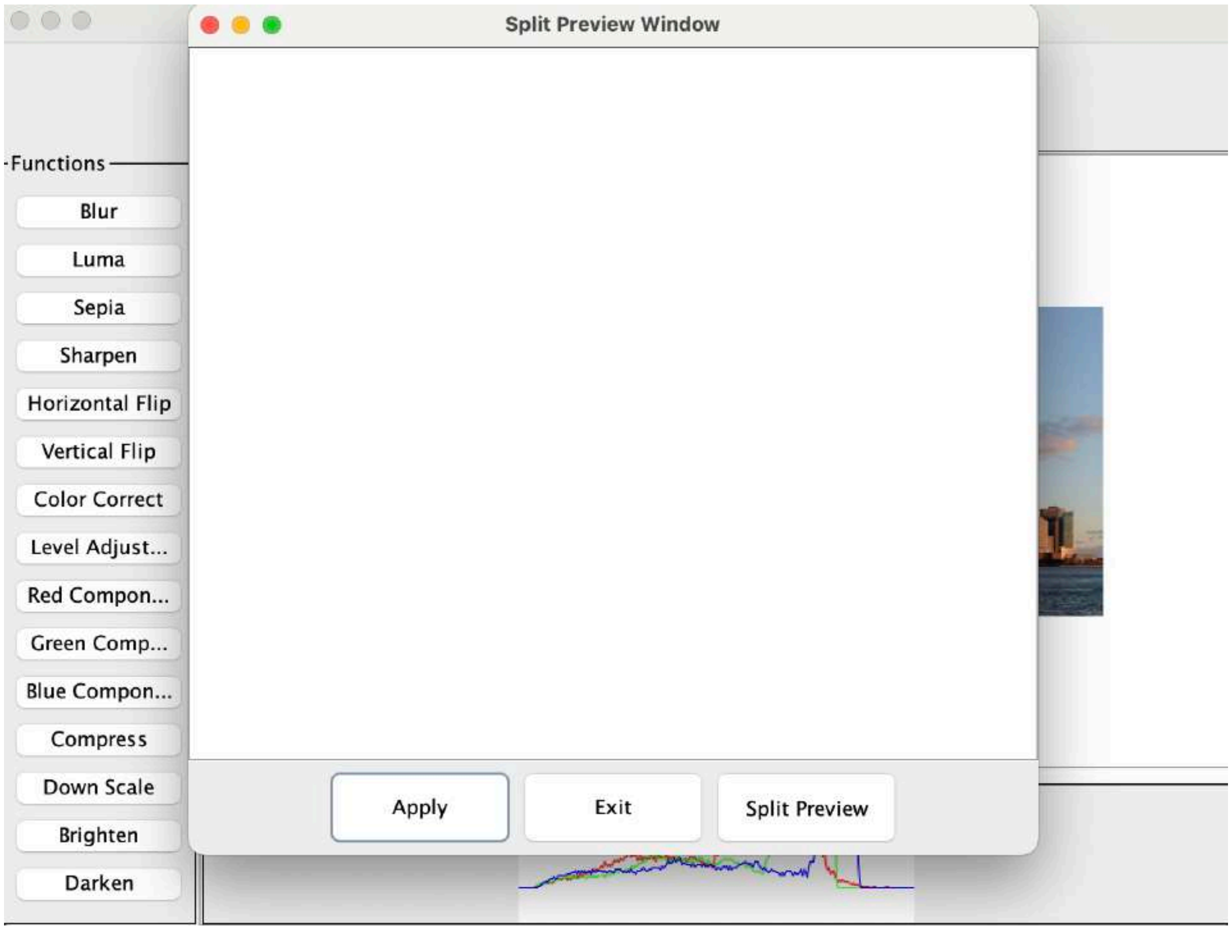
When Clicked on Apply Button



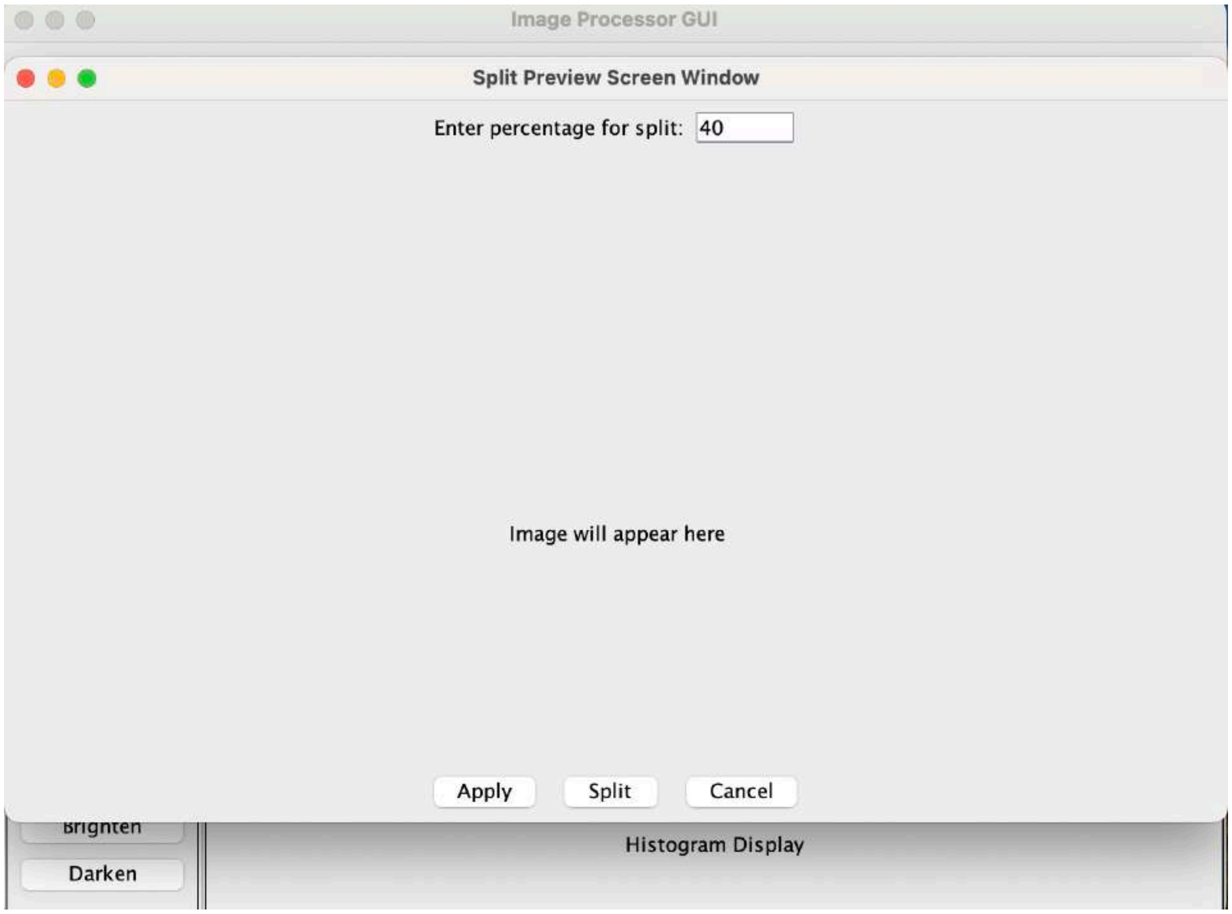
11. **Green-Component Button** When green component button is pressed a popup opens which shows the split preview of the images and has options to apply it to the whole image or exit.



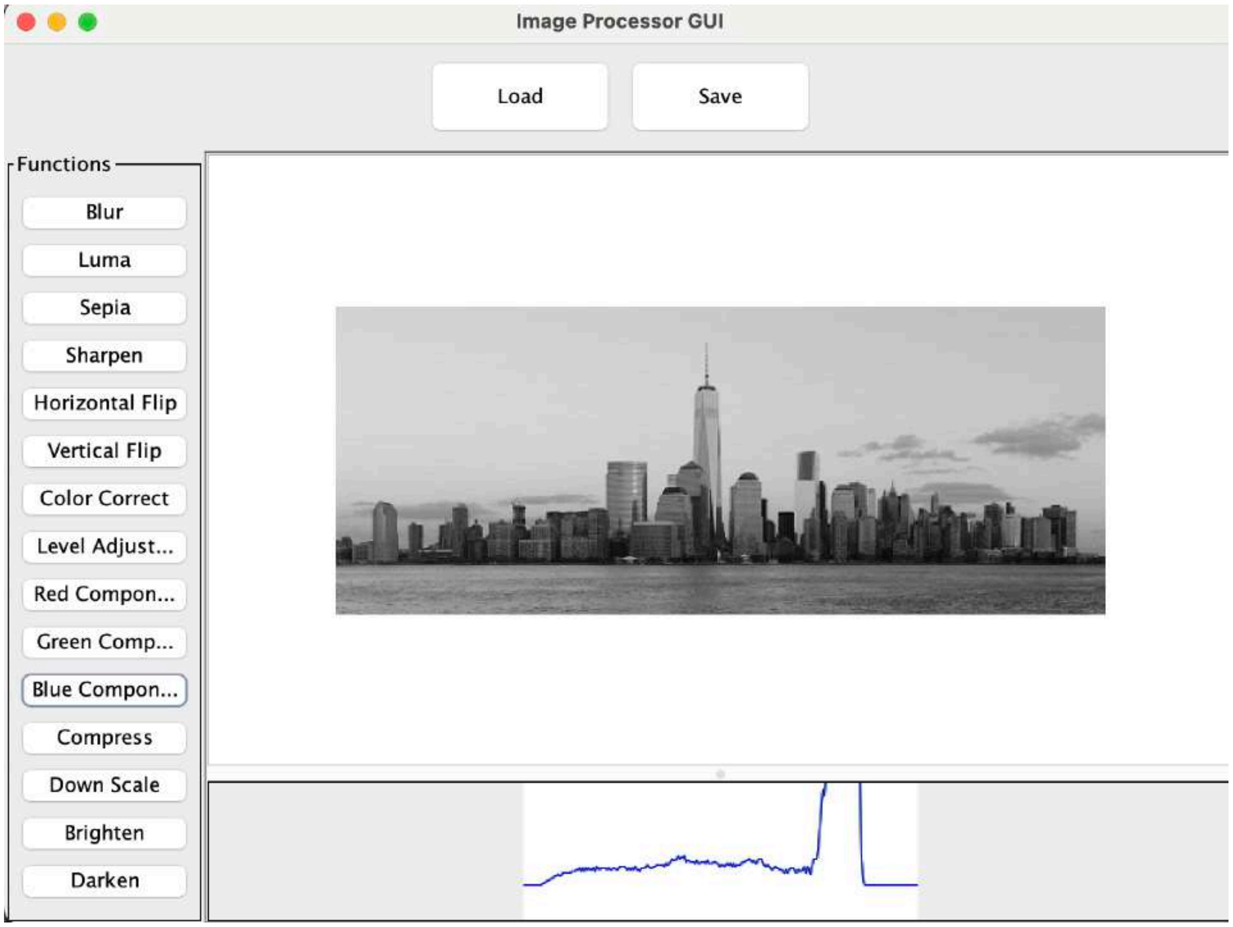
When the LevelAdjust button is clicked



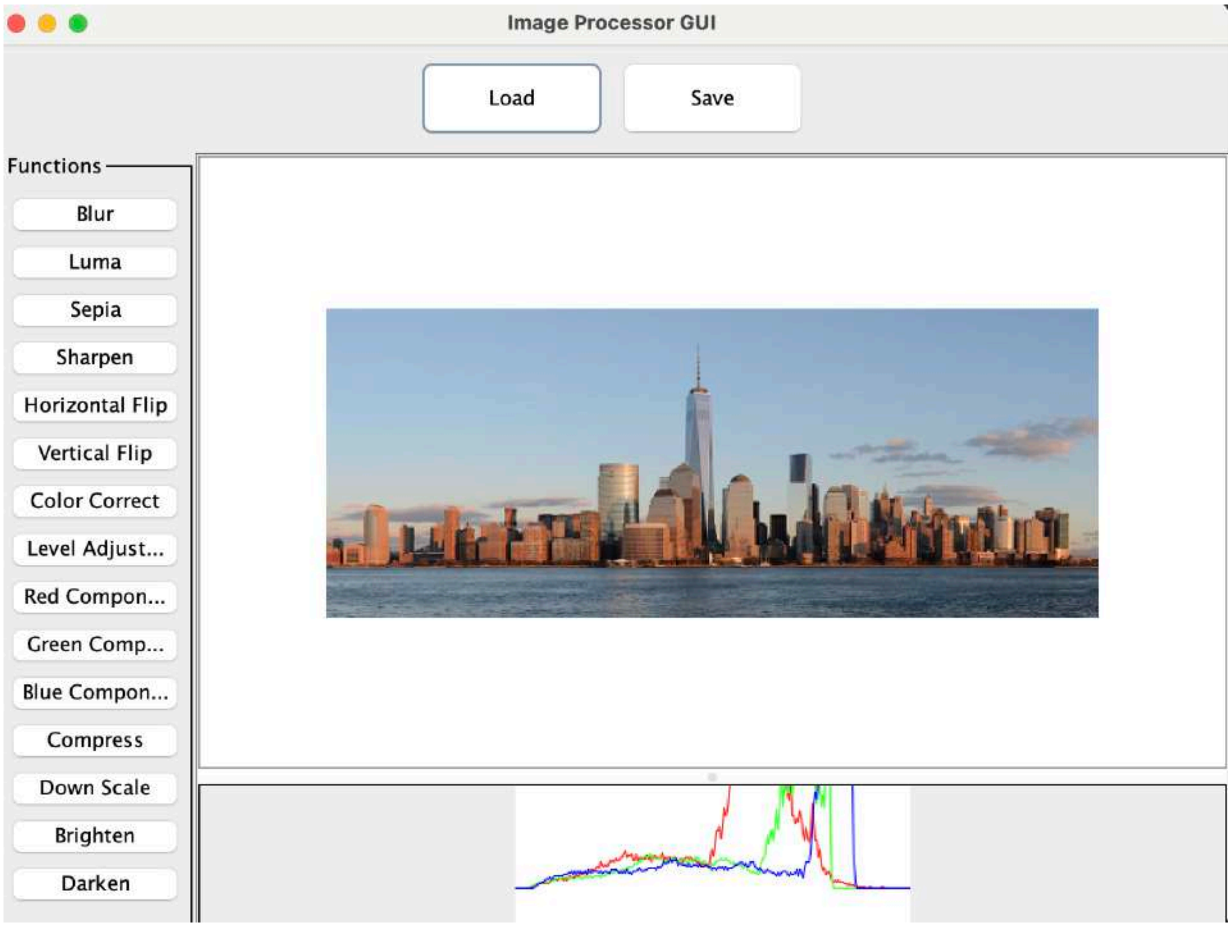
Prompt for entering the percentage is shown when clicked on Split Preview



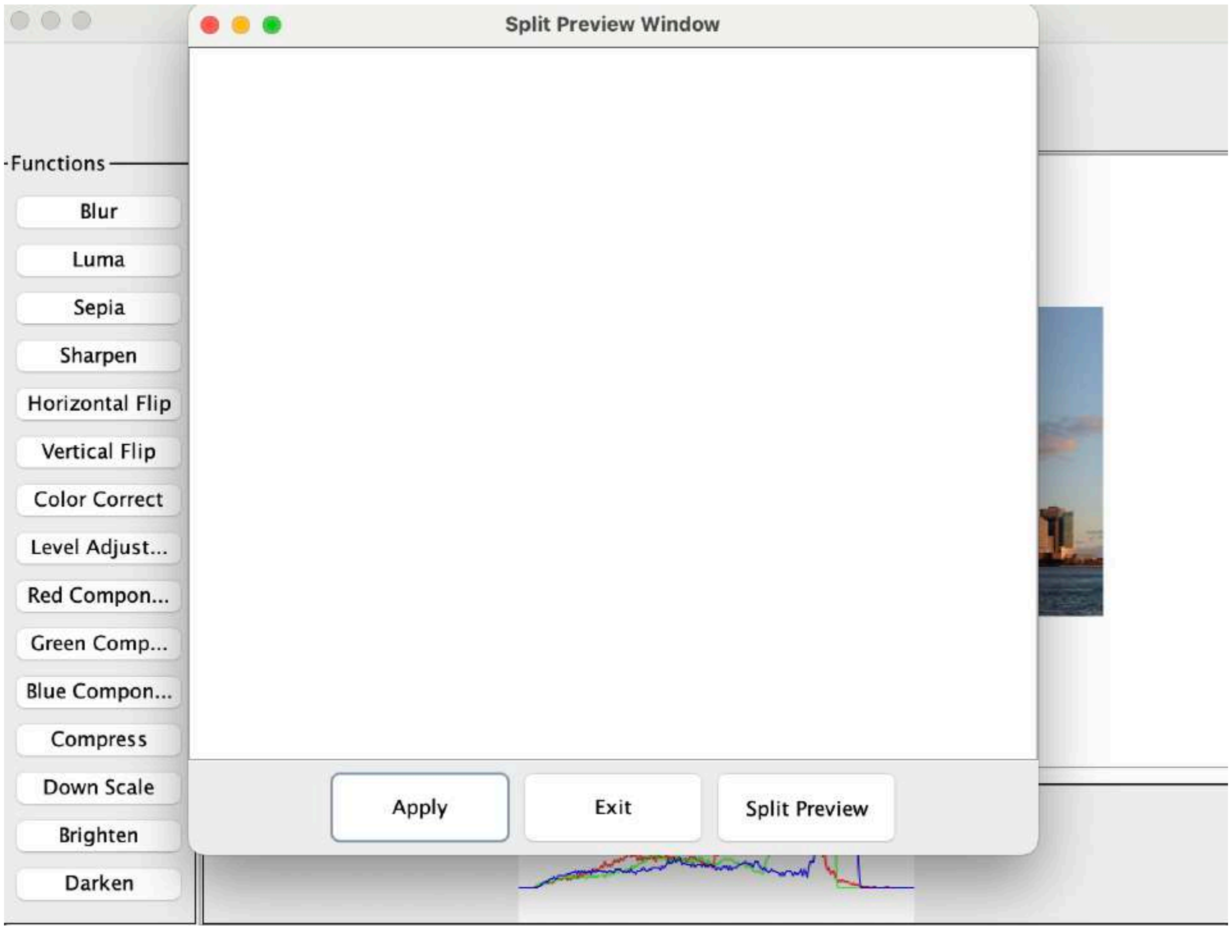
On successful apply for green component.



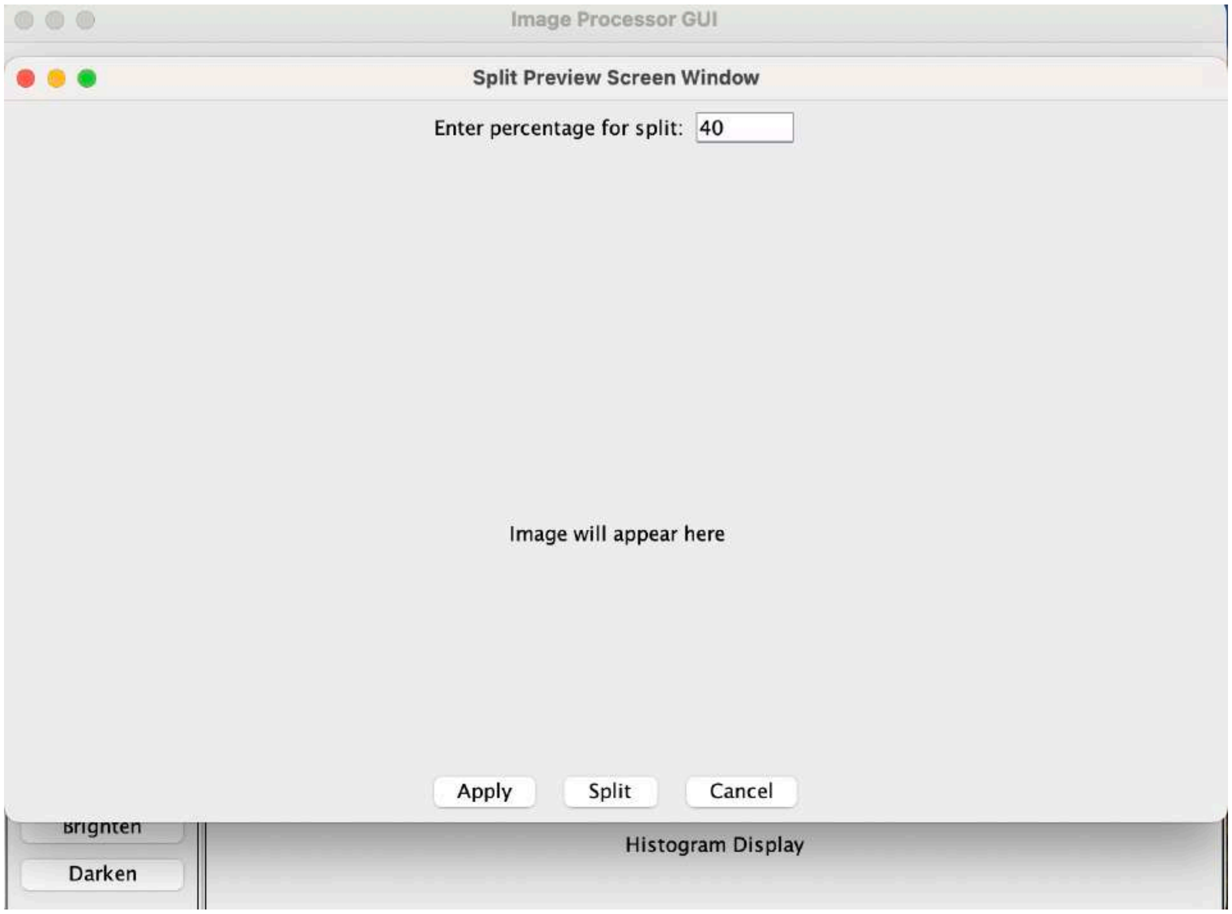
12. **Blue-Component Button** When blue component button is clicked a popup opens which shows the split preview of the images and has options to apply it to the whole image or exit.



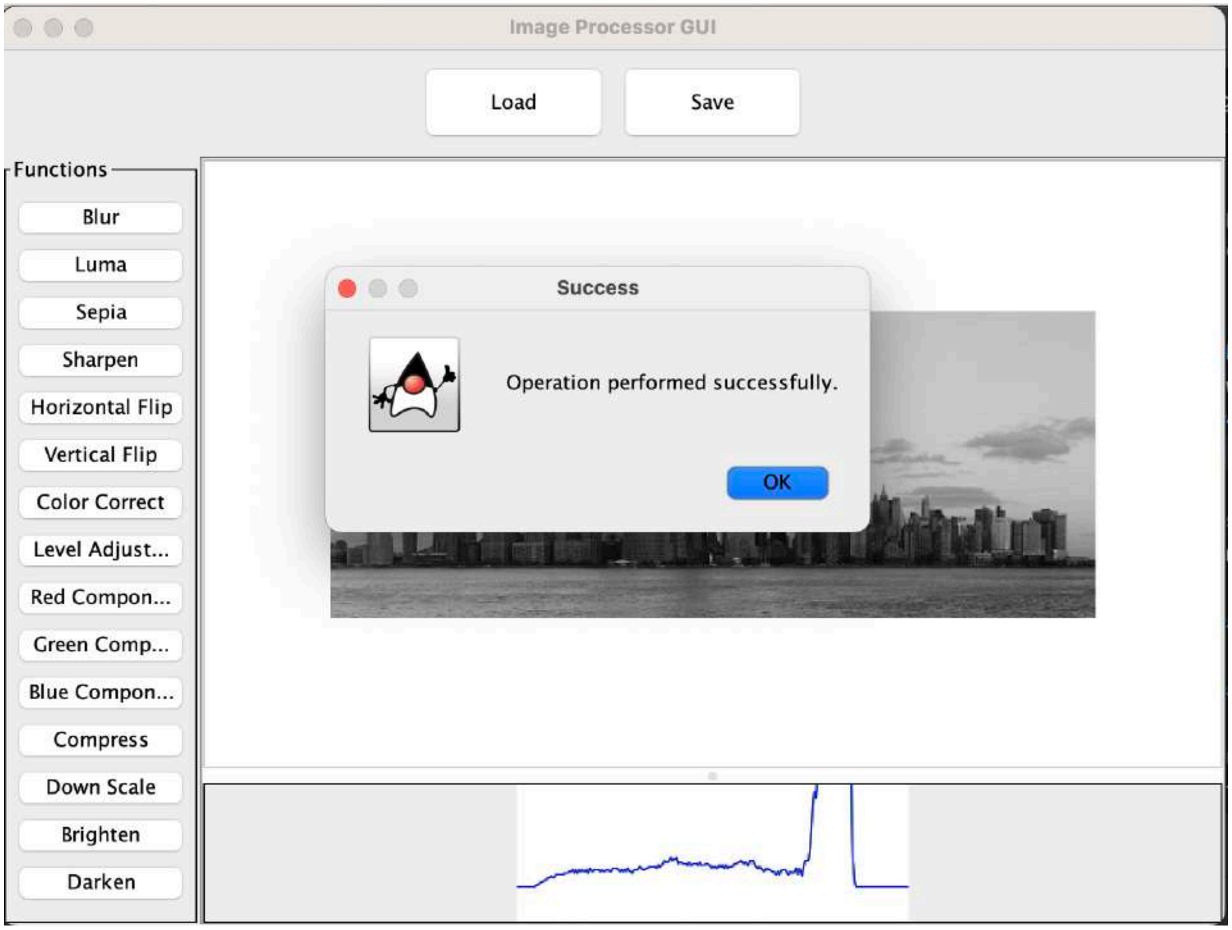
When the LevelAdjust button is clicked



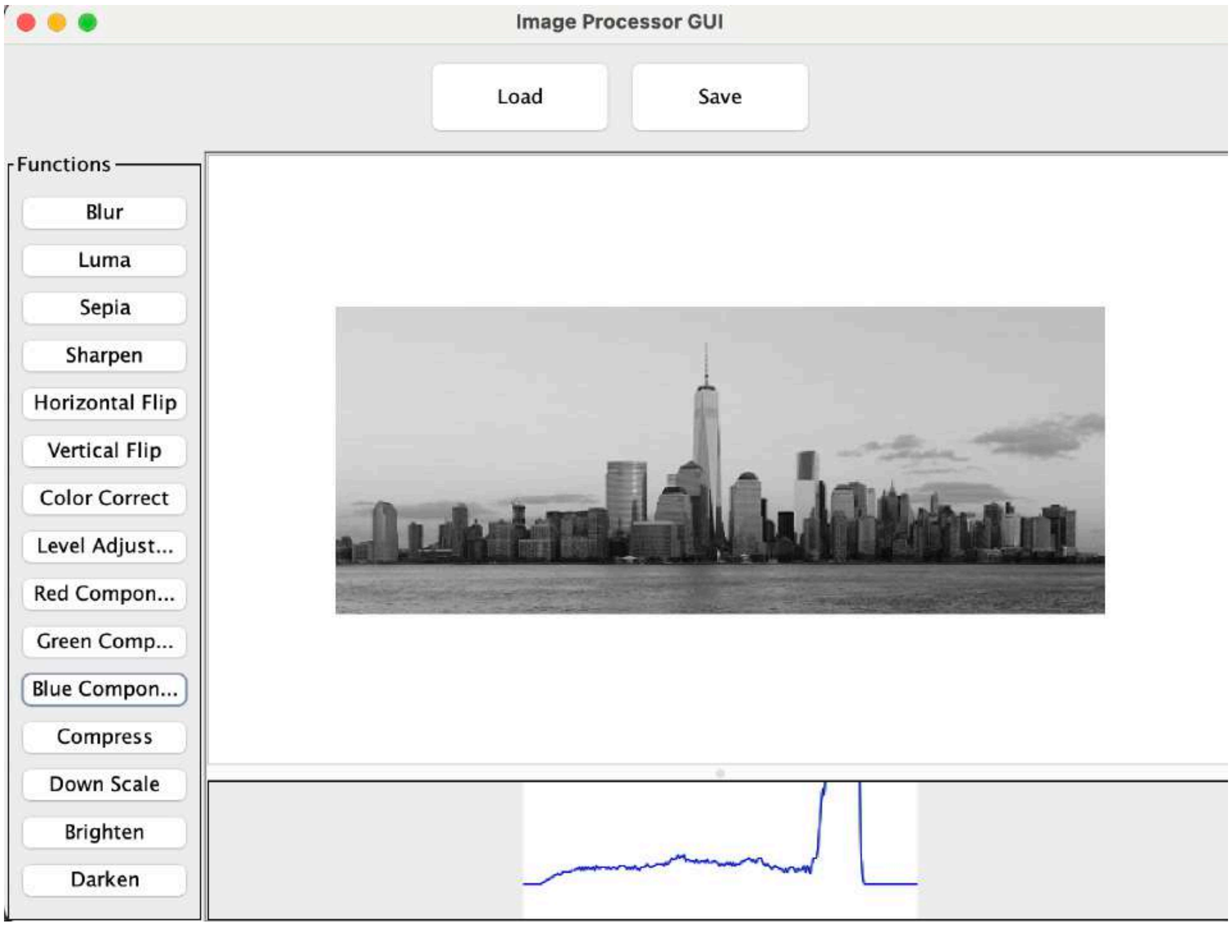
Prompt for entering the percentage is shown when clicked on Split Preview



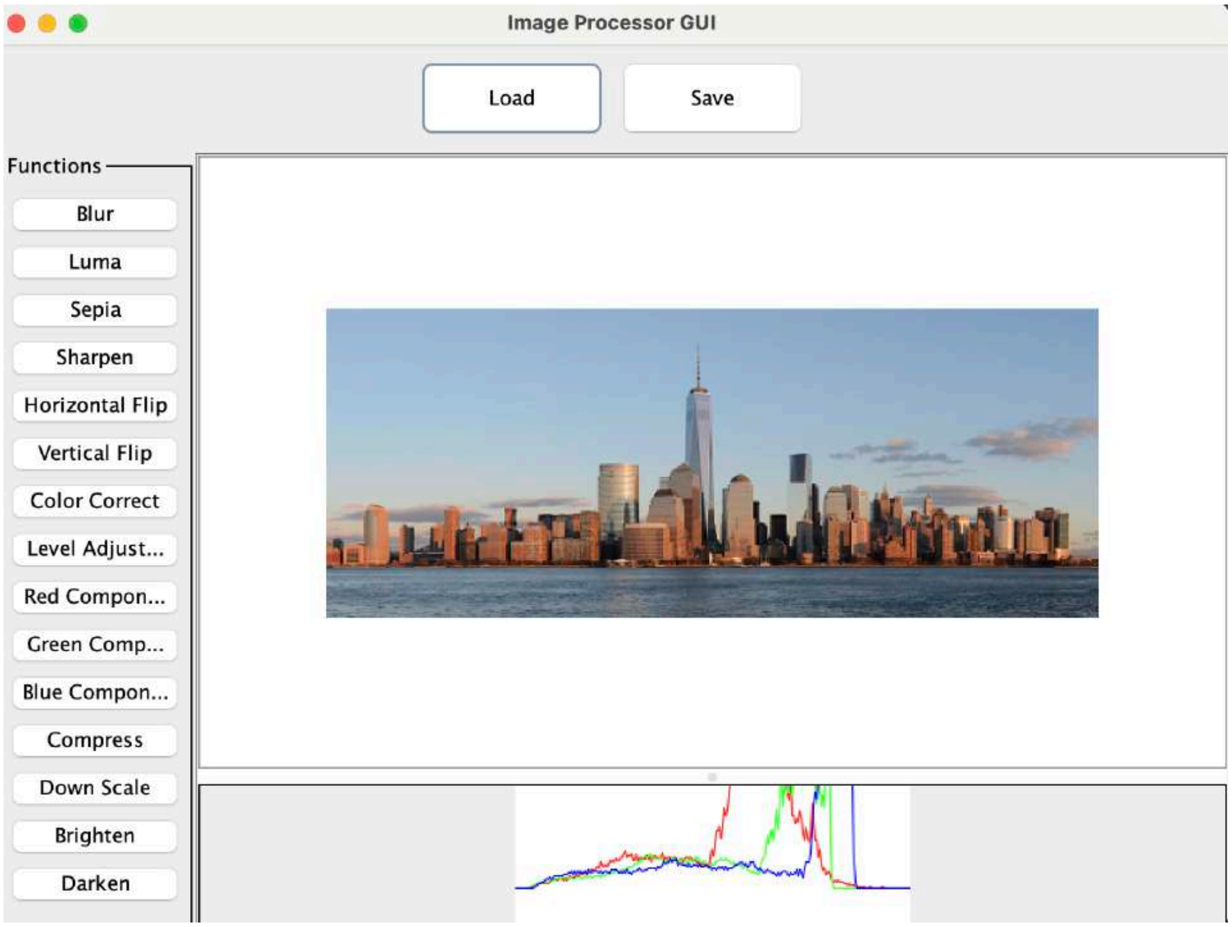
When Apply button is clicked.

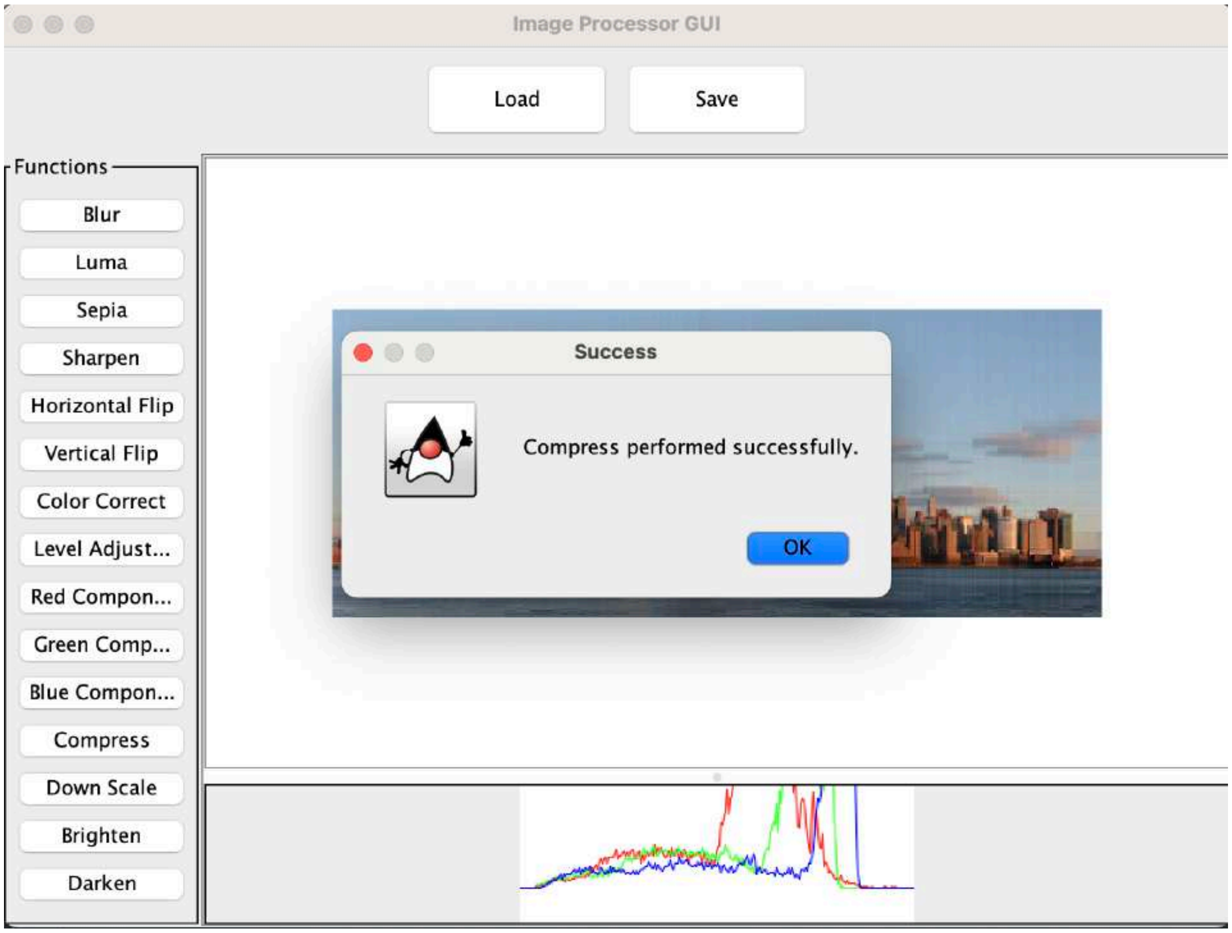


Operation performed successfully.



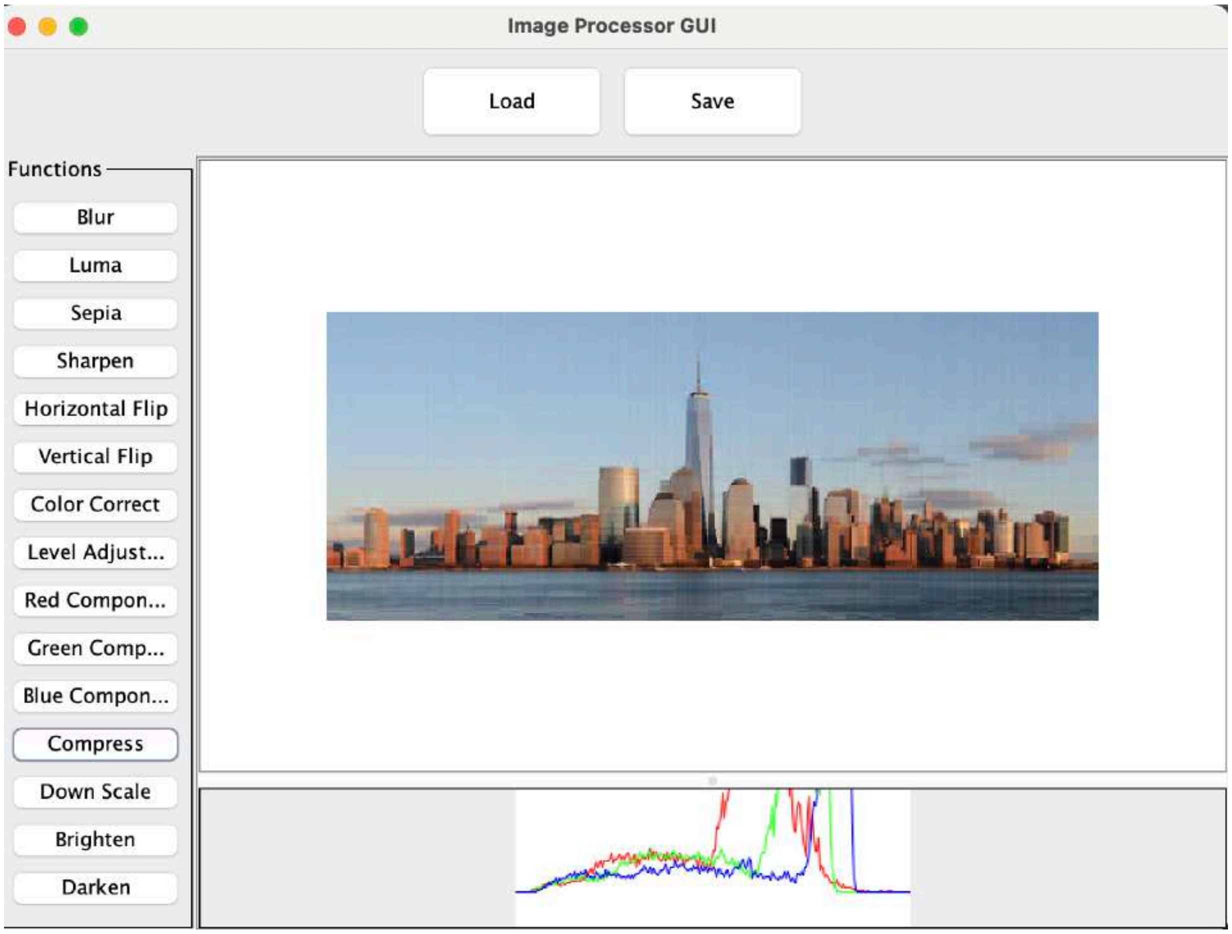
13. **Compress Button** When Compress button is clicked it opens a user input window to let the user enter the percentage of compression hence the user can enter integer value which must be between 0 and 100.



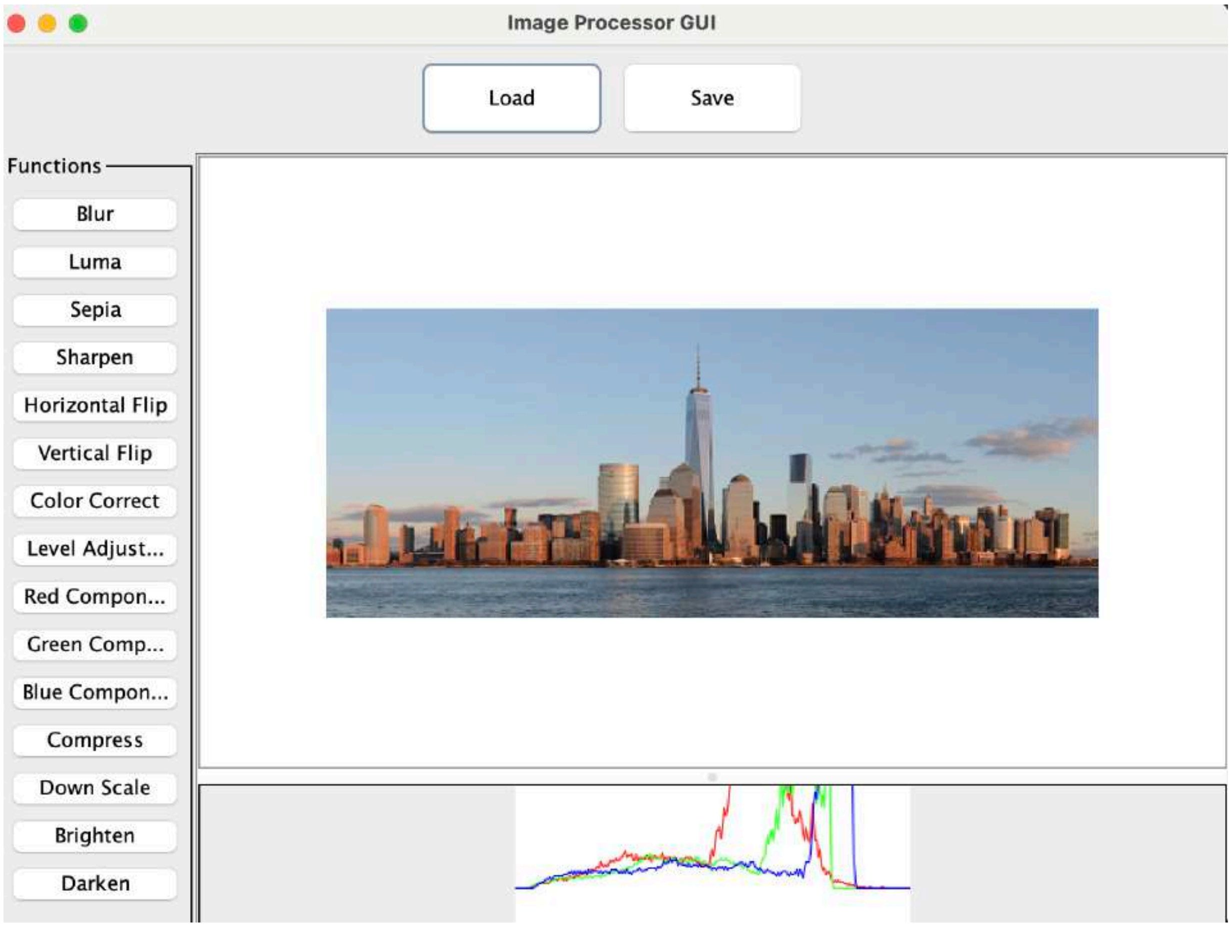


Compress performed successfully.

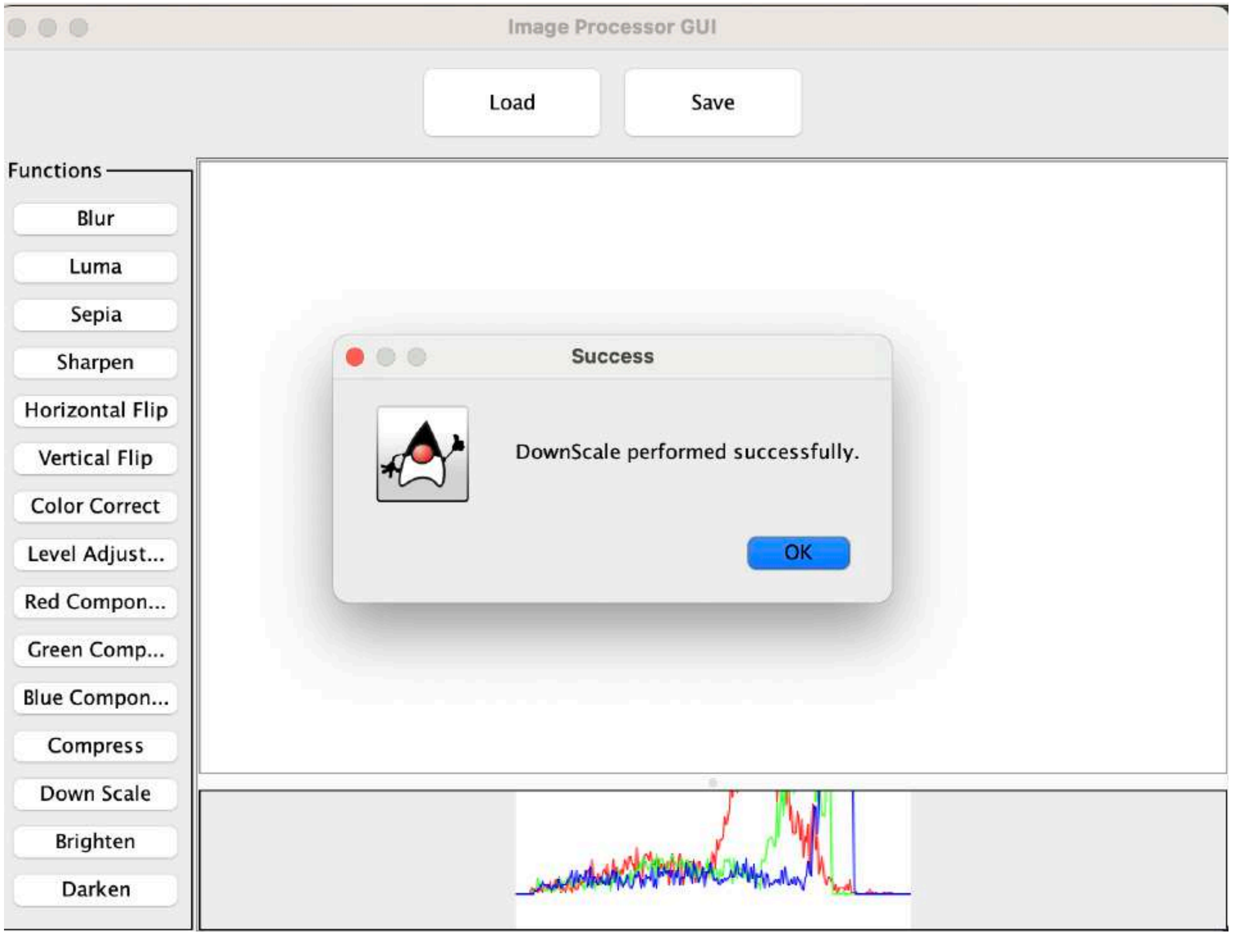
This image is compressed by 20%.



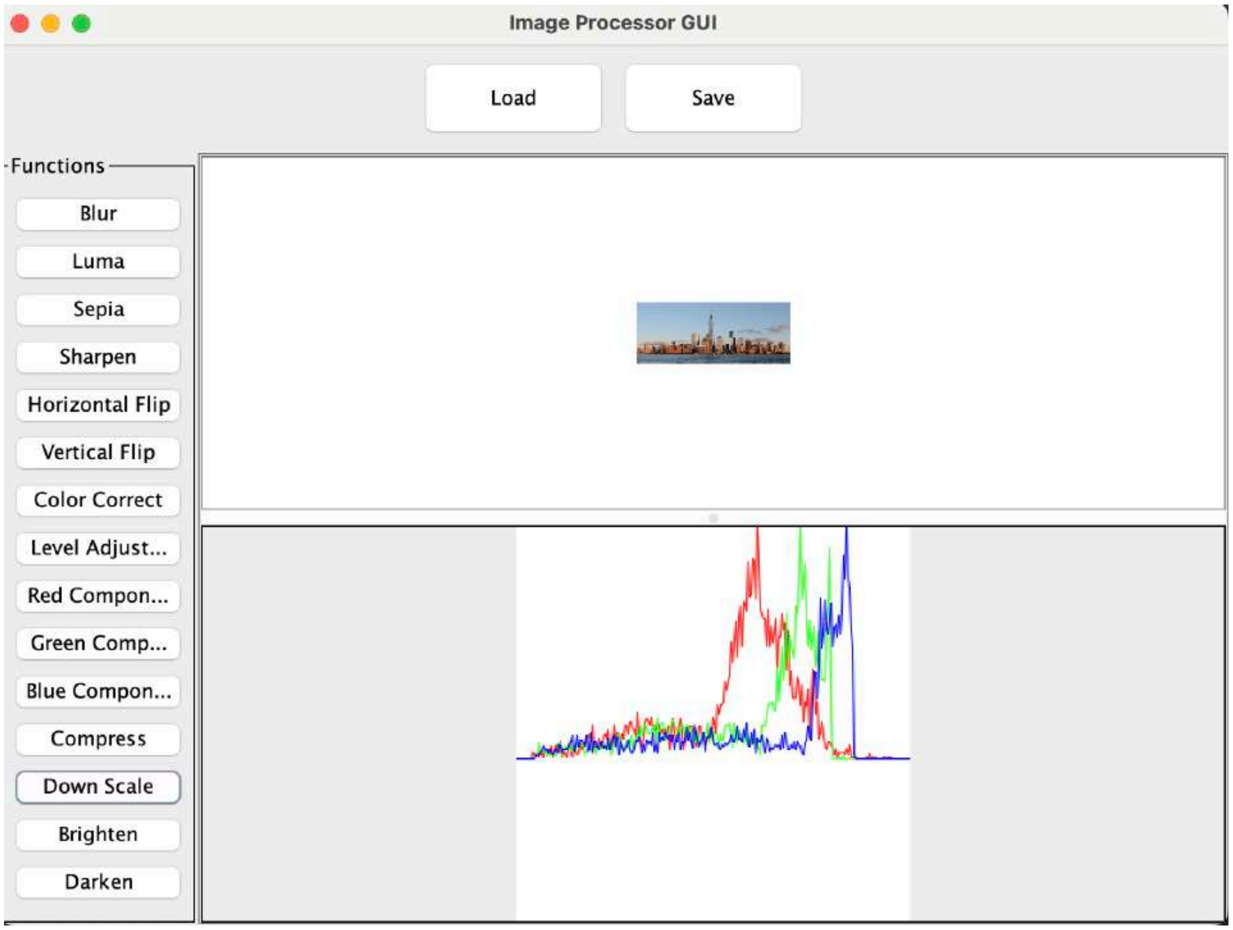
14. **DownScale Button** When the downscale button is clicked the user is given 2 prompts one after the other for width and height input, the user must ensure that the input are non-zero positive integers only.



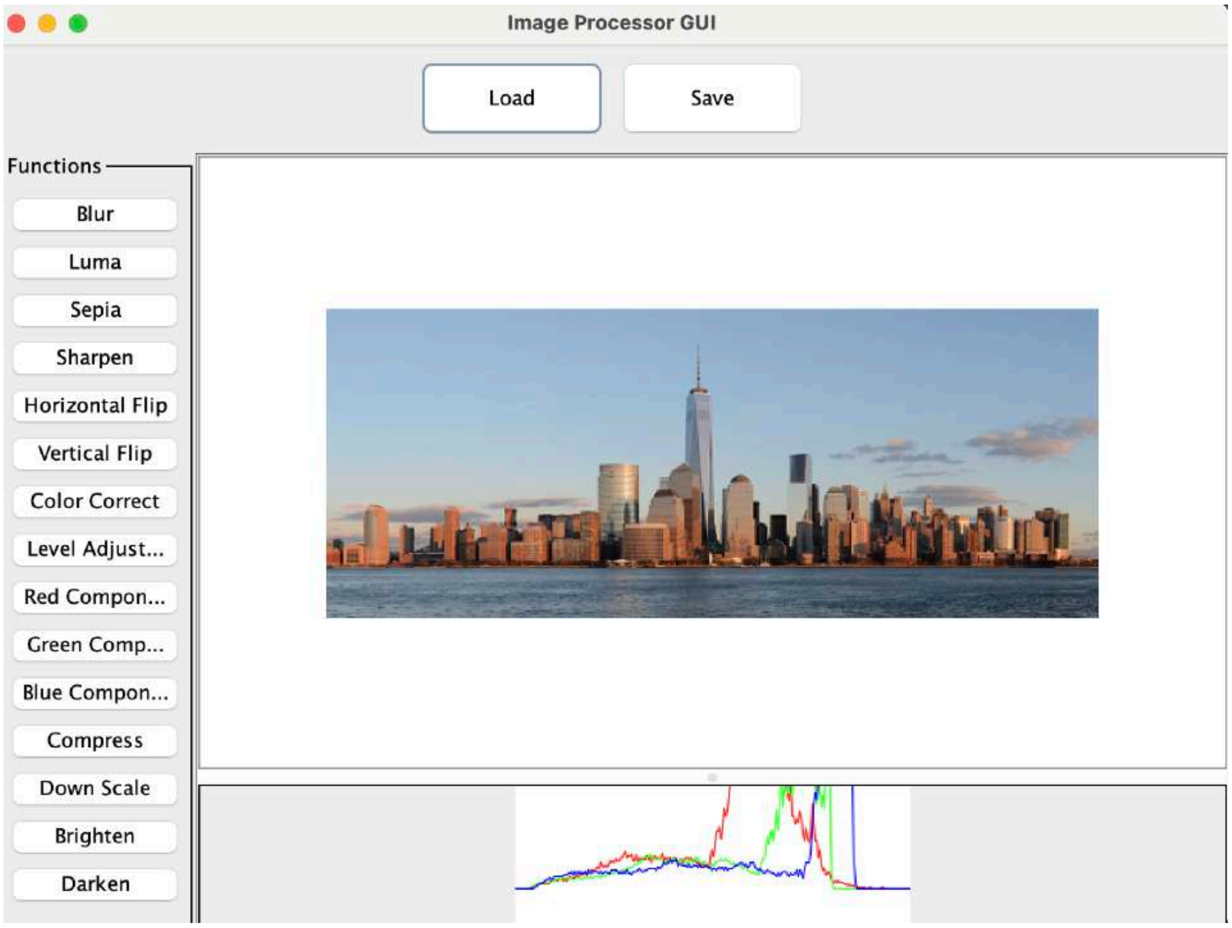
When downscale button is pressed. First width is asked, and upon entering valid width, then height is asked.When entered as needed a success message is shown



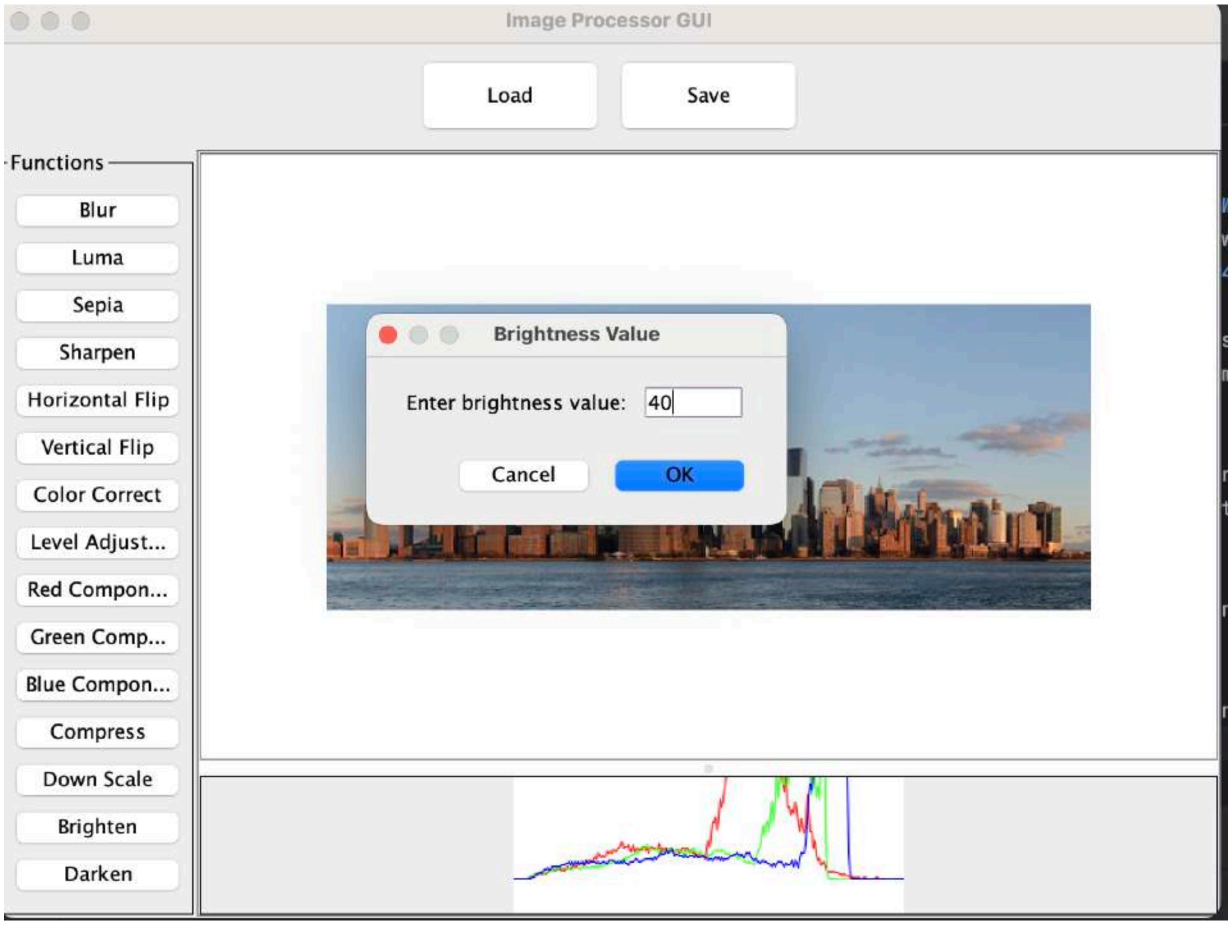
DownScale performed successfully.



15. **Brightness Button** When this button is clicked the user is given a prompt to enter the level of brightness needed.

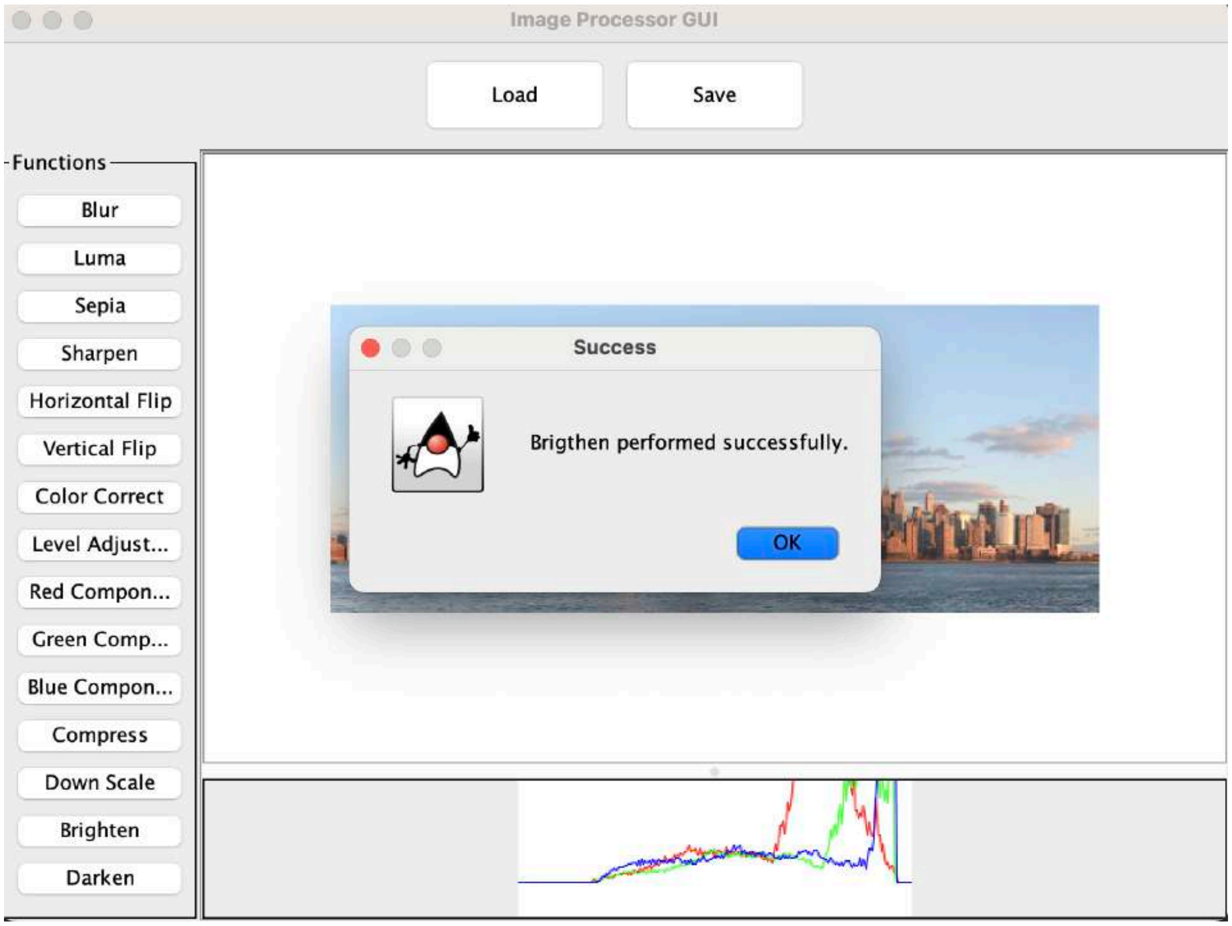


When the brightness button is clicked the user is prompted for brightness value input which should be positive integer only. Also, upon hovering on the text field a message is shown for what to enter for input.

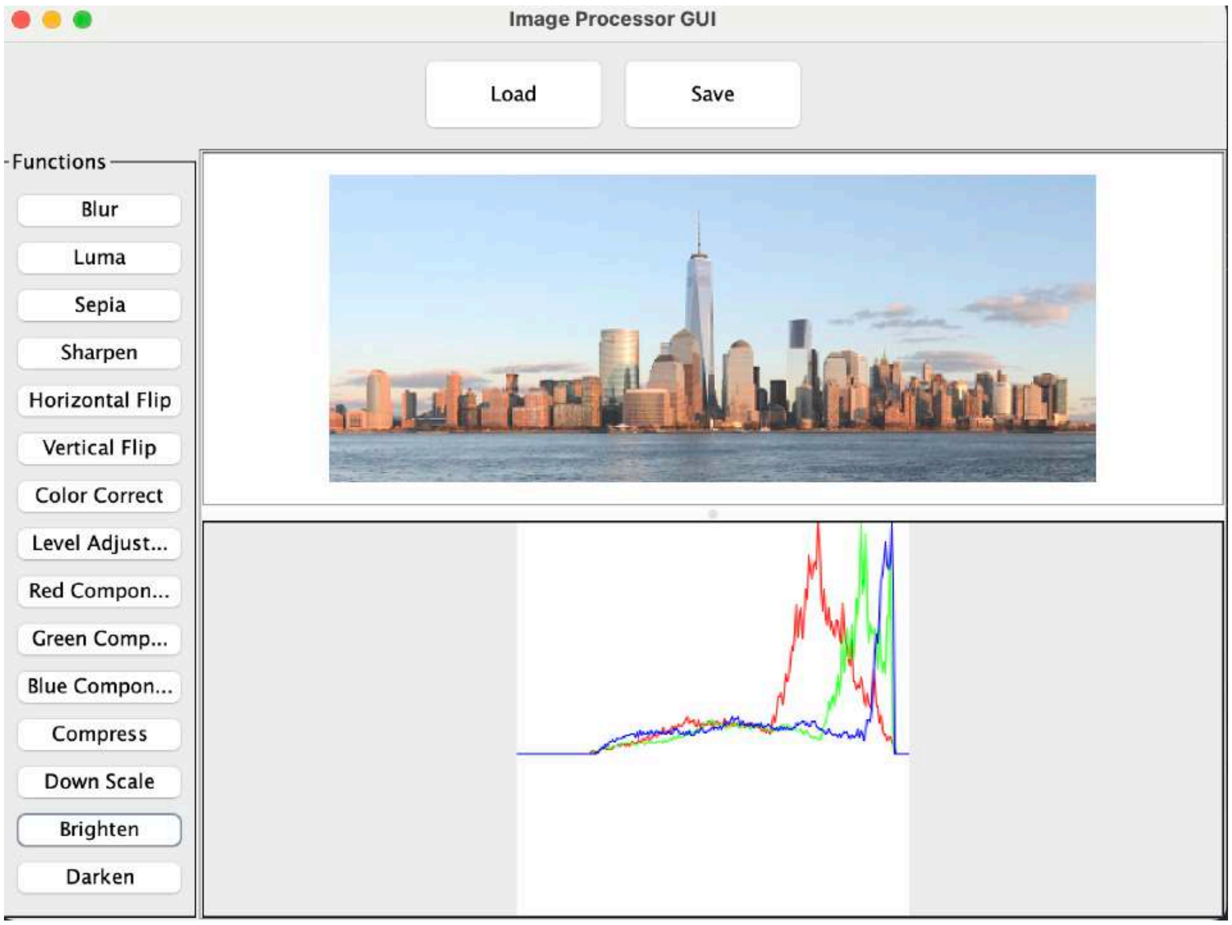


Enter brightness value: 40

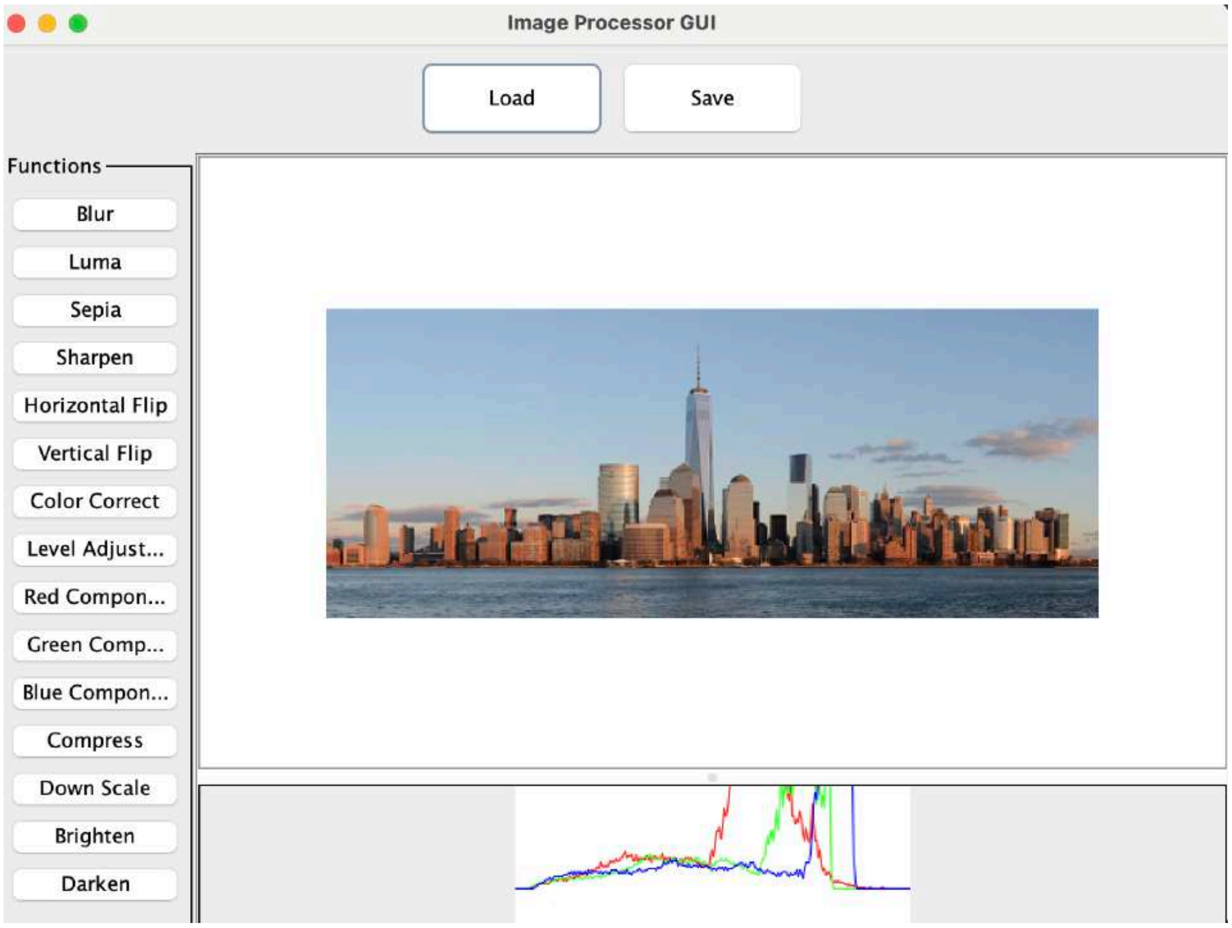
When pressed ok a success message and the brightened image is displayed



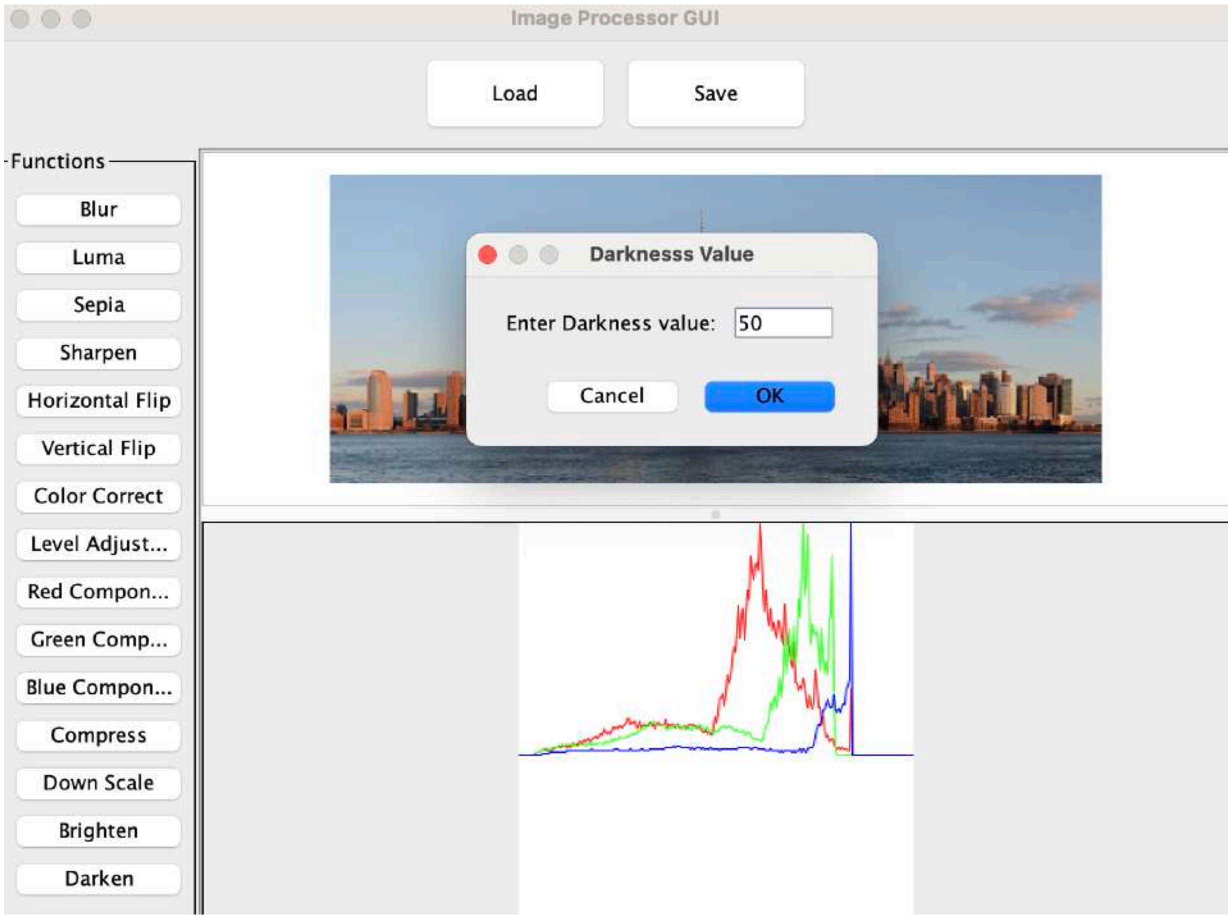
Brigthen performed successfully.



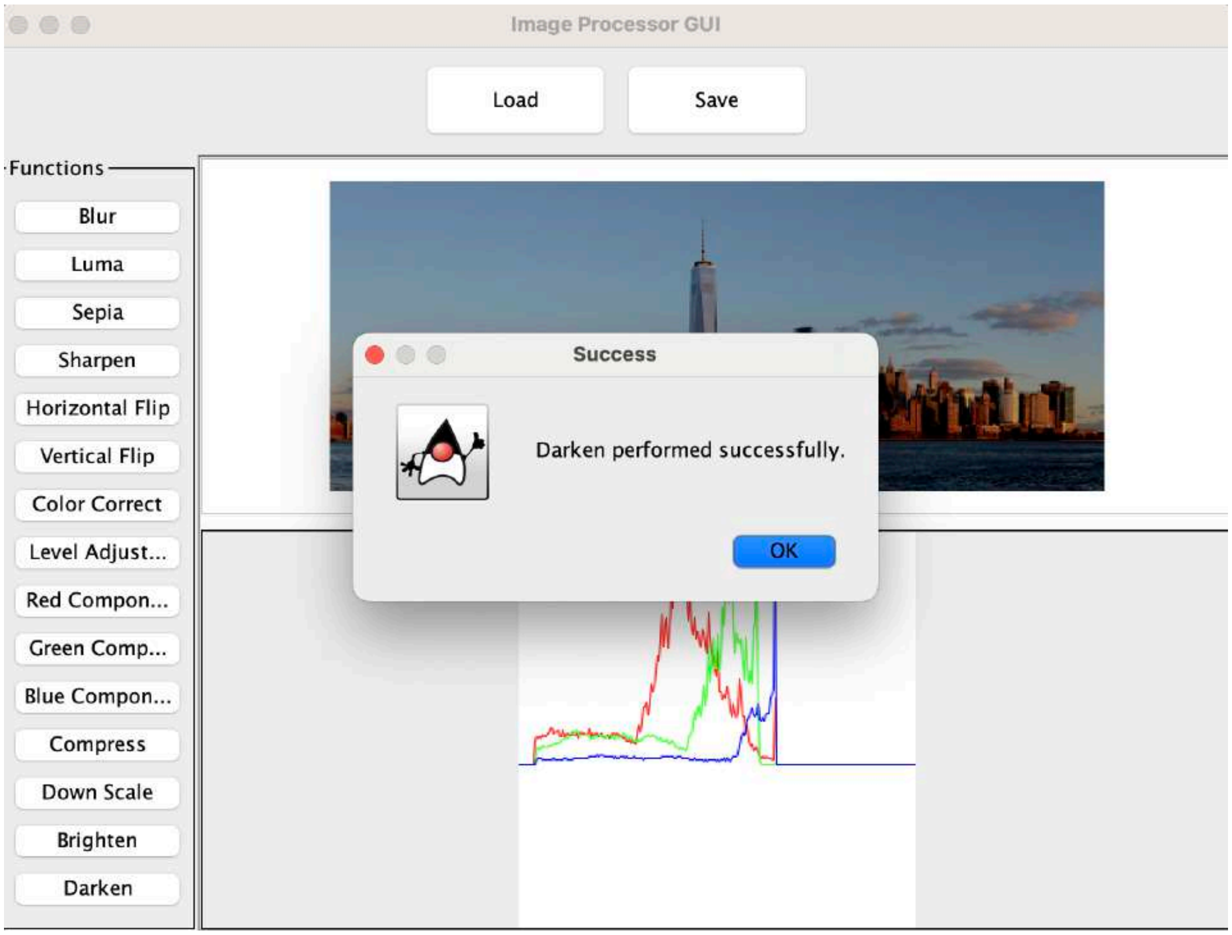
16. **Darken Button** When this button is clicked the user is given a prompt to enter the level of brightness needed.



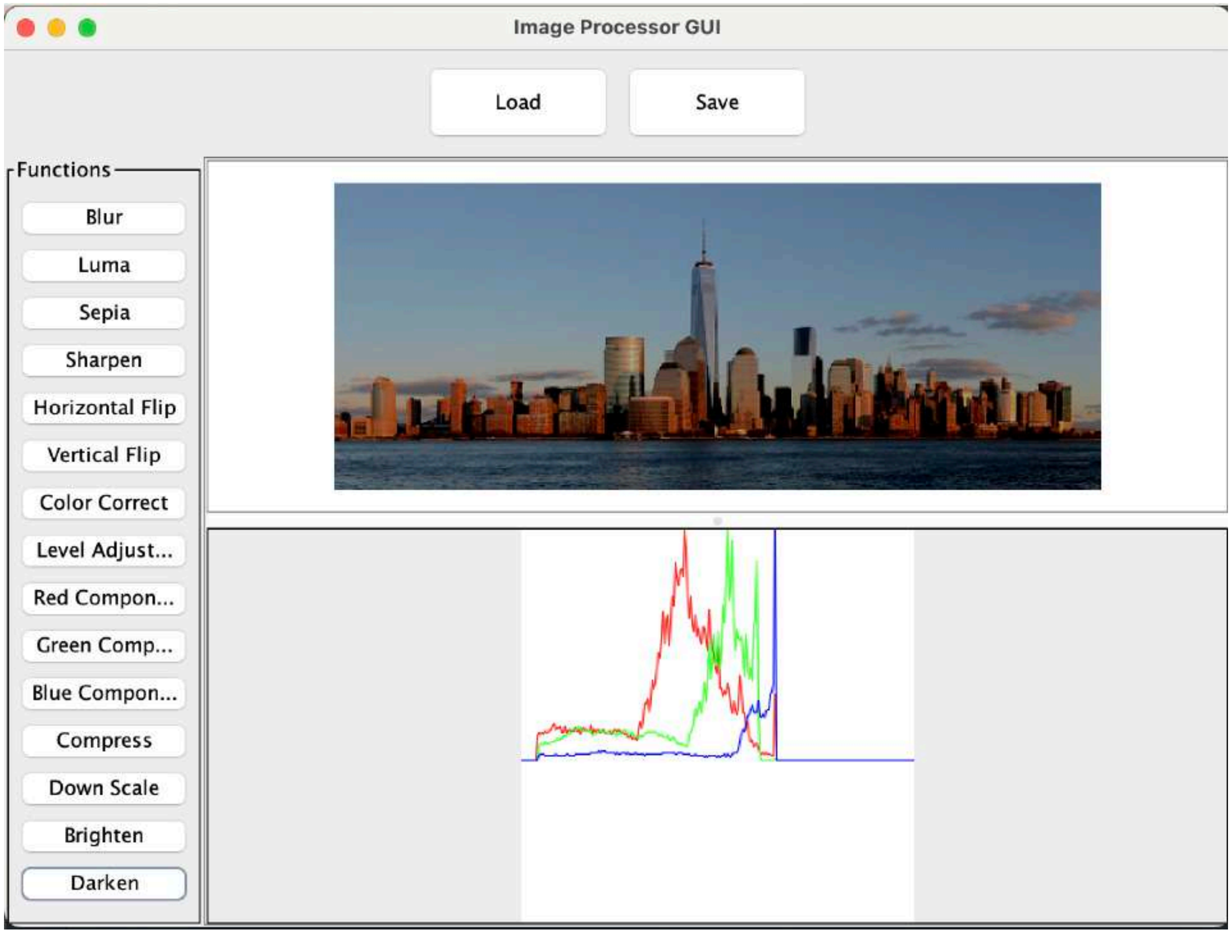
When the darken button is clicked the user is prompted for entering the value by which the image must be darkened. It should be positive integer only.



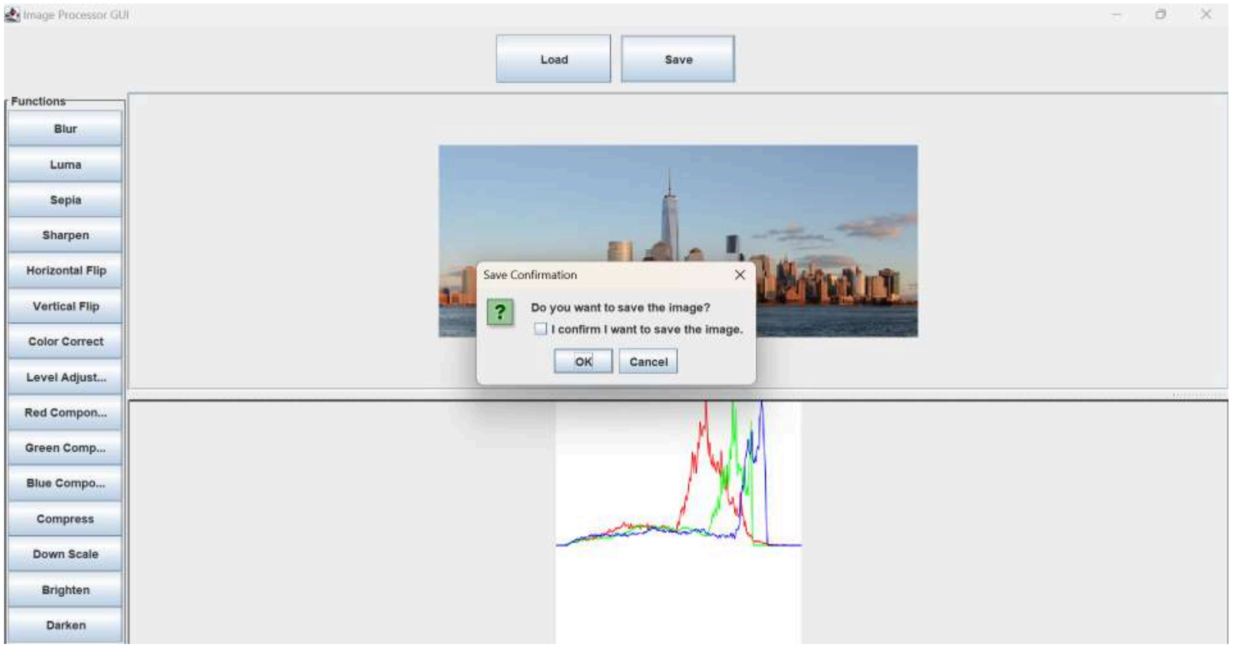
When the user clicks ok.



Darken performed successfully.



17. **Save Button** When the save button is clicked the user is prompted for a image name and the user file path that can be chosen from the file explorer. When saved the user image is saved to any desired location. When cancelled the operation is called off. To save the image, user must check the checkbox and then press save.

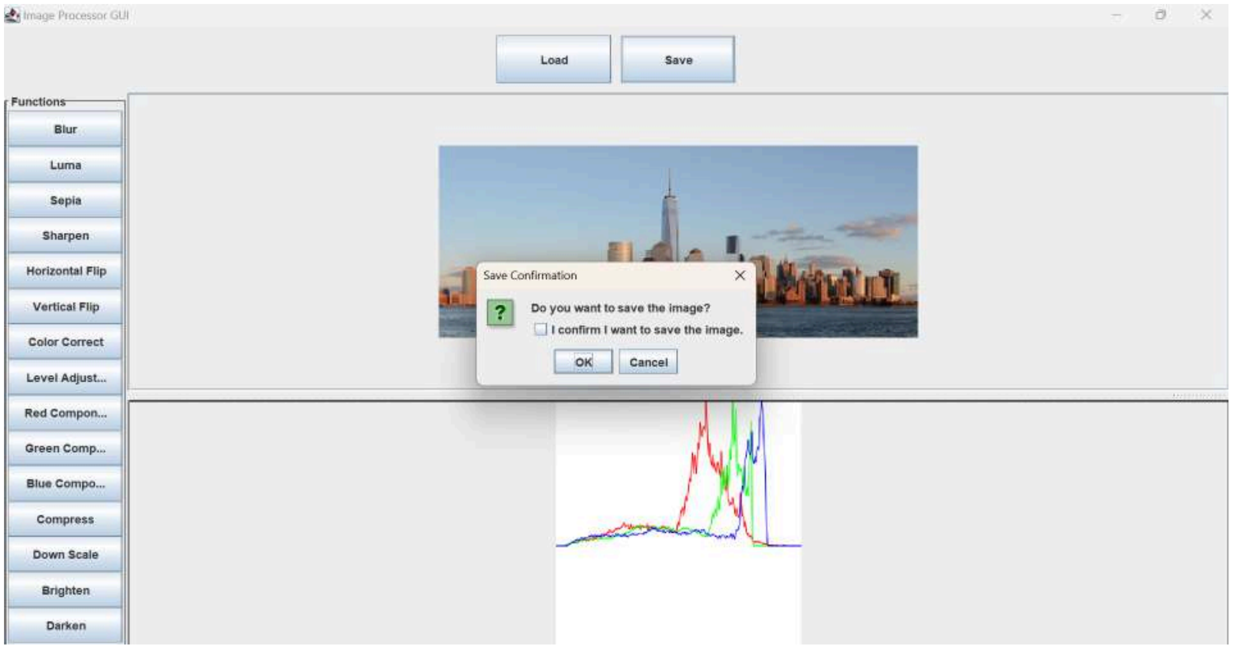


Save Confirmation

Do you want to save the image?

I confirm I want to save the image.

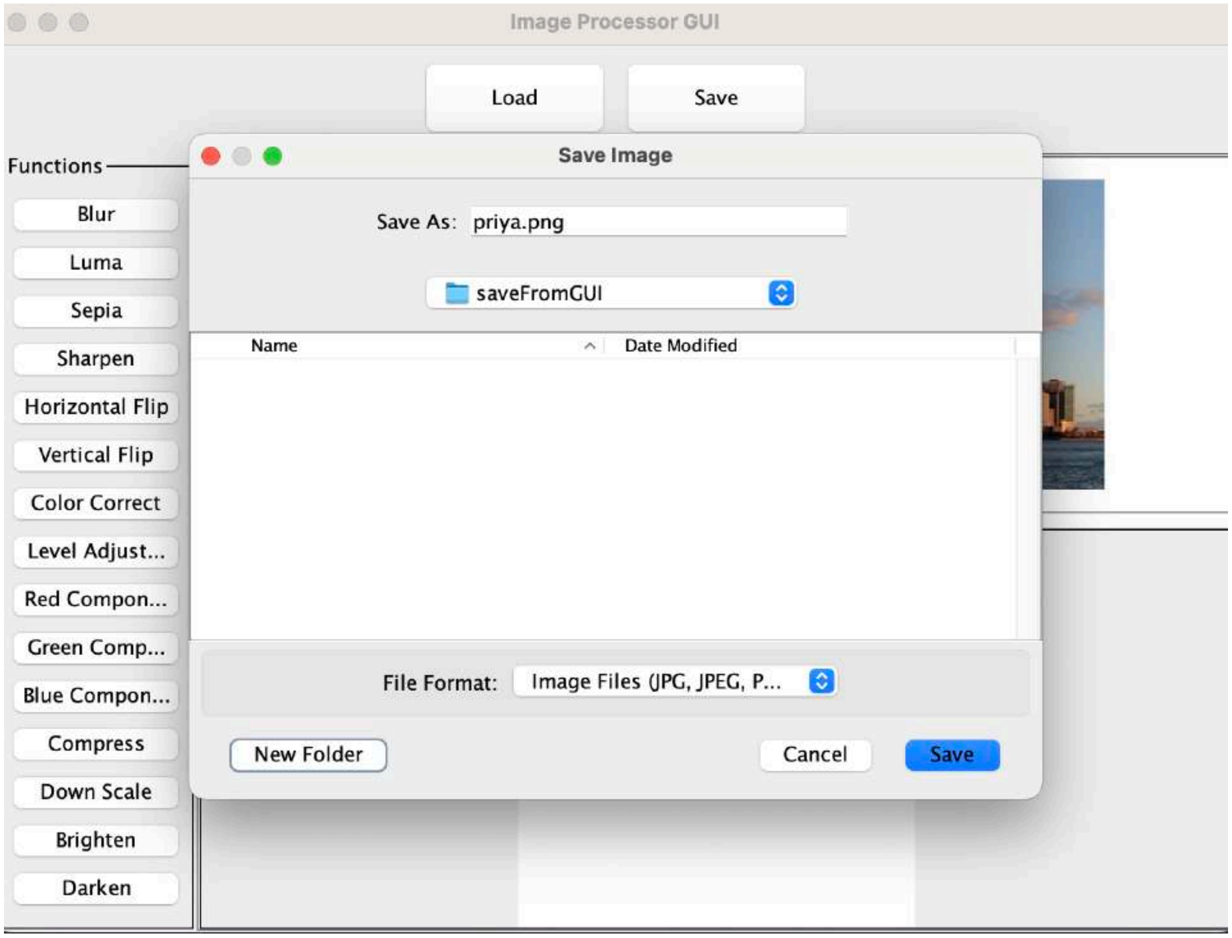
If checkbox is not clicked then error is shown.



Save Confirmation

Do you want to save the image?

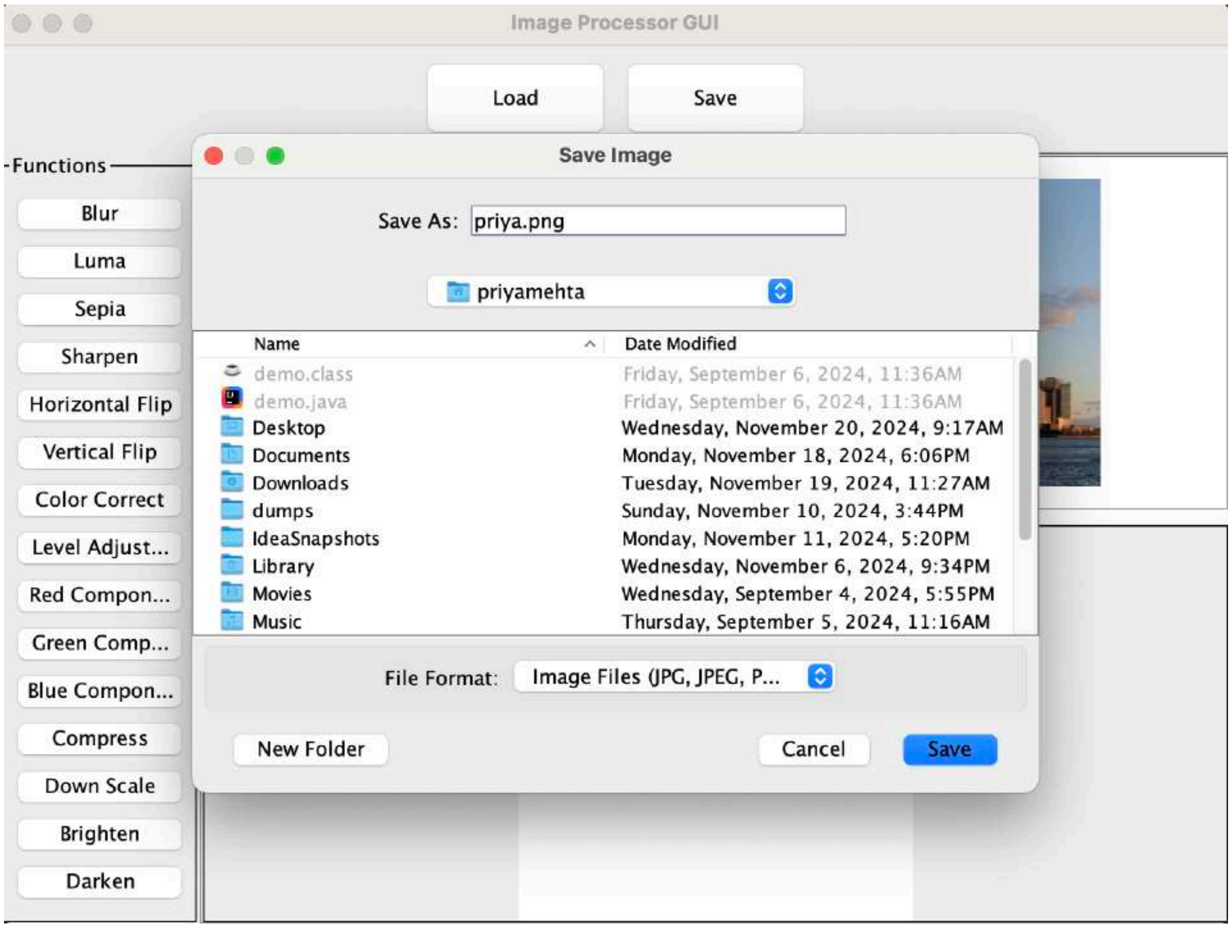
I confirm I want to save the image.



Save As: priya.png

saveFromGUI

File Format: Image Files (JPG, JPEG, P...

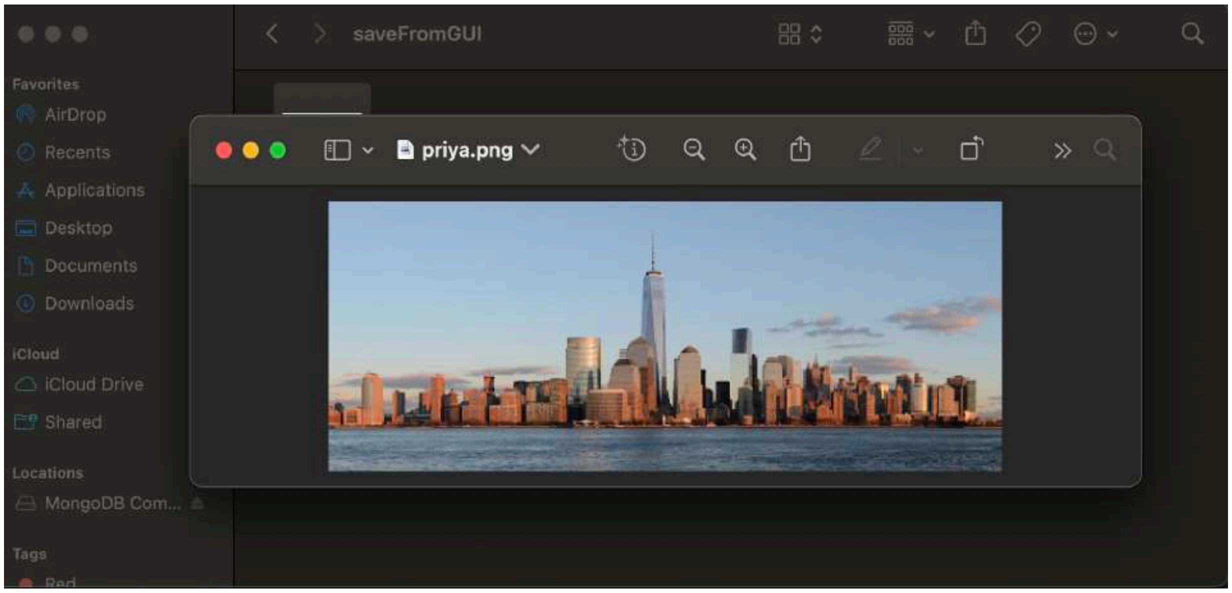


Save As: priya.png

priyamehta

Name	Date Modified
demo.class	Friday, September 6, 2024, 11:36AM
demo.java	Friday, September 6, 2024, 11:36AM
Desktop	Wednesday, November 20, 2024, 9:17AM
Documents	Monday, November 18, 2024, 6:06PM
Downloads	Tuesday, November 19, 2024, 11:27AM
dumps	Sunday, November 10, 2024, 3:44PM
IdeaSnapshots	Monday, November 11, 2024, 5:20PM
Library	Wednesday, November 6, 2024, 9:34PM
Movies	Wednesday, September 4, 2024, 5:55PM
Music	Thursday, September 5, 2024, 11:16AM

File Format: Image Files (JPG, JPEG, P...



18. **Exit Button** On clicking exit button, it simply closes the popup window.

Command Syntax and Structure

Each command has a specific format in which we call it - It has command type and a list of arguments. Also, first we need to `load` an image before applying any functionality, and the `save` command should be used to save any processed image.

Commands

Every command must follow the given syntax precisely. If a command is incorrect or if the image is not located in the specified file path, the application will raise an exception. Moreover, for any operation to be performed on an image, it must already be loaded into the application's memory under the specified name.

1. Load and Save Commands

1. **Load**: Loads an image from the provided path and assigns a custom name to use inside app. This image name is necessary for subsequent operations, as it allows the app to identify which image to change. Ensure the image exists in the specified path; otherwise, an exception will be thrown.

```
load relative-path-from-src image-name
```

Example:

```
load res/koala.jpg sample
```

2. **Save**: Saves the processed image to a specified path with the provided name. Ensure the image to be saved has been processed and exists in the app's memory.

```
save relative-path-from-src image-name
```

Example:

```
save /results/sample-output.png sample
```

2. Color Component Commands

The commands isolate individual color channels - RGB components or grayscale components - Intensity, Value, Luma of an image.

1. **Red/Green/Blue Component**: Isolates a specific color component (red, green, or blue) from the image and saves the output with a new name. The resulting image will display only the selected color channel, while the other channels are set to zero.

```
 red-component image-name dest-image-name
 green-component image-name dest-image-name
 blue-component image-name dest-image-name
```

Example:

```
red-component sample sample-red
```

2. **Intensity/Value/Luma Component Commands**: Generates a grayscale image using various brightness calculation methods: *Intensity* calculates the average of the RGB values, *Value* uses the highest RGB value, and *Luma* applies weighted RGB values based on human visual perception.

```
 intensity-component image-name dest-image-name
 value-component image-name dest-image-name
 luma-component image-name dest-image-name
```

Example:

```
intensity-component sample sample-intensity
```

3. Flip Commands

1. **Horizontal Flip**: Flips the image horizontally.

```
horizontal-flip image-name dest-image-name
```

Example:

```
horizontal-flip sample sample-flipped
```

2. **Vertical Flip**: Flips the image vertically.

```
vertical-flip image-name dest-image-name
```

4. Brightness/Darkness Adjustment Command

1. **Brighten** or **Darken**: Modifies an image's brightness by a specified value, where a positive value increases brightness and a negative value reduces it. This adjustment affects all color channels uniformly. The user must enter an integer for the brightness increment, with negative values darkening the image.

```
 brighten intensity-value image-name dest-image-name
```

Example:

```
brighten 20 sample sample-brightened
```

5. RGB Split and RGB Combine Commands

1. **RGB Split**: Separates the image into 3 separate images for each color channel (red, green, blue).

```
rgb-split image-name dest-image-name-red dest-image-name-green dest-image-name-blue
```

2. **RGB Combine**: Combines 3 images, each containing one of the red, green, blue color channels, into a single image.

```
rgb-combine image-name red-image green-image blue-image
```

6. Blur and Sharpen Commands

1. **Blur**: Applies a blurring effect to soften the image, reducing details and smoothing the transitions.

```
blur image-name dest-image-name
```

2. **Sharpen**: Enhances the edges in an image.

```
sharpen image-name dest-image-name
```

7. Sepia Tone Command

Applies a sepia filter to give the image a warm look.

```
sepia image-name dest-image-name
```

8. Compression Command

Decreases an image's file size by a specified percentage, preserving resolution but possibly impacting quality. The percentage parameter, which can be a decimal, should be within the range of 0 to 100.

```
 compress percentage image-name dest-image-name
```

9. Histogram Command

Generates a histogram of the image’s color distribution.

```
histogram image-name dest-image-name
```

10. Color Correct Command:

Automatically adjusts colors in the image to improve tone and balance.

```
color-correct image-name dest-image-name
```

11. Levels Adjust Command

Modifies the black, mid, and white points to enhance contrast and brightness. The values for black, mid, and white should each be within the 0-255 range and must be arranged in ascending order.

```
 levels-adjust b m w image-name dest-image-name
```

12. Split Preview Mode

Some commands offer a Split Preview mode, allowing only part of the image to be altered while the rest remains untouched. To activate split preview, add `split p` to the command, where `p` is a user-defined parameter between 0 and 100, which can be a decimal. Commands supporting it are: blur, sharpen, sepia, red-component, green-component, blue-component, value-component, luma-component, intensity-component, color-correct, level-adjust. Commands in Split Preview mode must adhere precisely to this syntax to avoid errors.

Example:

```
blur sample sample-blur split p
```

13. Quit Application Command

Exits the app.

```
quit
```

14. Run Script Command

Executes a list of commands from a script file. All the commands in script are in the given syntax only.

```
run absolute/path/to/script.txt
```

15. DownScale Command

This command takes in mainly 2 inputs from the user, new width and new height and downscales the image

```
downscale src-Image-name dest-Image-name new-height new-width
```

14. Mask Script Command

Masking of the image assumes that the same size of the image which acts as the mask is already present in the internal memory of the program, i.e. it is previously loaded. Masking of Image is done on the following :

1. Blur
2. Sharpen
3. Sepia
4. Red-Component
5. Green-Component
6. Blue-Component
7. Value Command
8. Intensity Command

Here "command" maps to any of the above from the list.

```
command src-imagename mask-imagename destination-image-name
```

Example Workflow

1. **Load** an image:

   ```
   load res/manhattan-small.png mh
   ```

2. **Apply operations**:

```
blur mh mh-blur
```

3. **Save** the modified image:

```
save res/mh-blur.png mh-blur
```

4. **Quit** the application:

```
quit
```

This `USEME.md` file offers a detailed guide covering all commands, syntax, and examples. Carefully follow each step and condition to ensure a seamless experience with the application.