



SOUTHERN
UNIVERSITY COLLEGE
南方大學學院

BTPR 2053
FUNDAMENTALS OF IMAGE PROCESSING

Project: Photoshop-like System

Tan Zhang En (B180036A)

Lecturer: Pang Yee Yong

16th March 2020

Department of Computer Science
Faculty of Engineering and Information Technology

Contents

1. Introduction	1
2. Functions	1
3. Examples	3
3.1 Resize Image	4
3.2 Blue Image	5
3.3 Invest Image Color	6
3.4 Flip Image	7
3.4.1 Horizontal Flip Image	7
3.4.2 Vertical Flip Image	8
3.4.3 Horizontal and Vertical Flip Image	9
3.5 Contrast Image	10
3.6 Change Image Opacity	11
3.7 Scale Image Size	12
3.8 Rotate Image	13
3.9 Print Text	14
3.10 Sepia Image Color	16
3.11 Crop Image	17
3.12 Convolution	18
3.12.1 Low-Pass Filter	18
3.12.2 High-Pass Filter	19
3.12.3 Directional Filter	20
3.12.4 Laplacian Filter	21
4 Installation	22
5 Conclusion	23

1. Introduction

In this project, I have developed a Photoshop-like system using NodeJS and Angular programming language. For the image processing part, I used Jimp which is an image processing library for Node written entirely in JavaScript, with zero native dependencies. The system separated with two parts which are Client-side with Angular for select image, render image and image process type, and Server-side with NodeJS for processing image from http request.

2. Functions

In this system is support png, jpeg, and tiff image types, and user able to save the output with png format. For available image processing functions is cover:

1. Resize Image
2. Blur Image
3. Invert Image Color
4. Flip Image Direction (horizontal or vertical, or both)
5. Contrast Image
6. Greysclae Image
7. Change Image Opacity
8. Scale Image Size
9. Rotate Image
10. Print Text
11. Sepia Image Color
12. Crop Image Size
13. Convolution (Low-Pass Filter, High-Pass Filter, Directional Filter and Laplacian Filter)

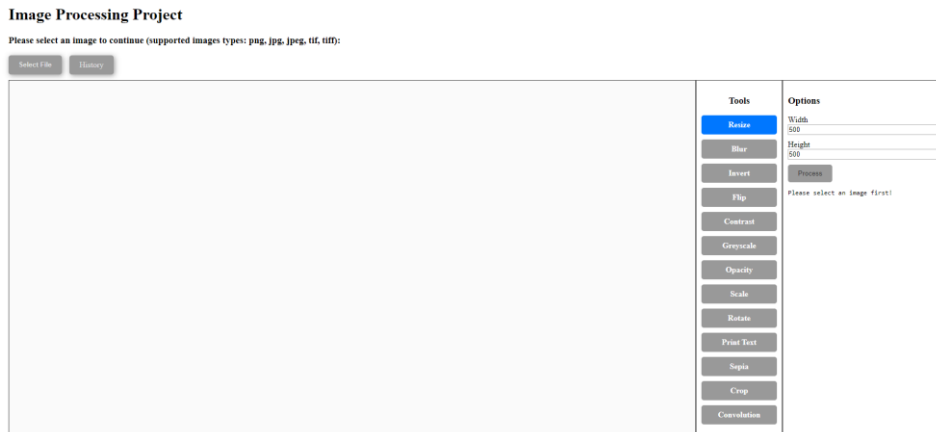


Figure 2.1 – System Screenshot without Image

From figure 2.1, user must select an image to start process the image, otherwise user cannot proceed any image process. User can click “Select File” button to select image with following supported types which are png, jpg, jpeg, tif, and tiff. After User can select any tool and entry different value in the options box, and click “Process” to process the image, after image processed image will update image.

Besides that, this system also can revert to any version of history in which the image data is processed. For example, the user selects an image file which is the lena.tiff, and he processes resize, scale, rotate effects. Suddenly he wants to revert to the original image which hasn’t process any effect, he can hover the “History” button, there will popup a dropdown list, he can click any one history version to revert to a specific version image. But he selects any one of version and tries to process an effect, the history after the selected version will be deprecated.

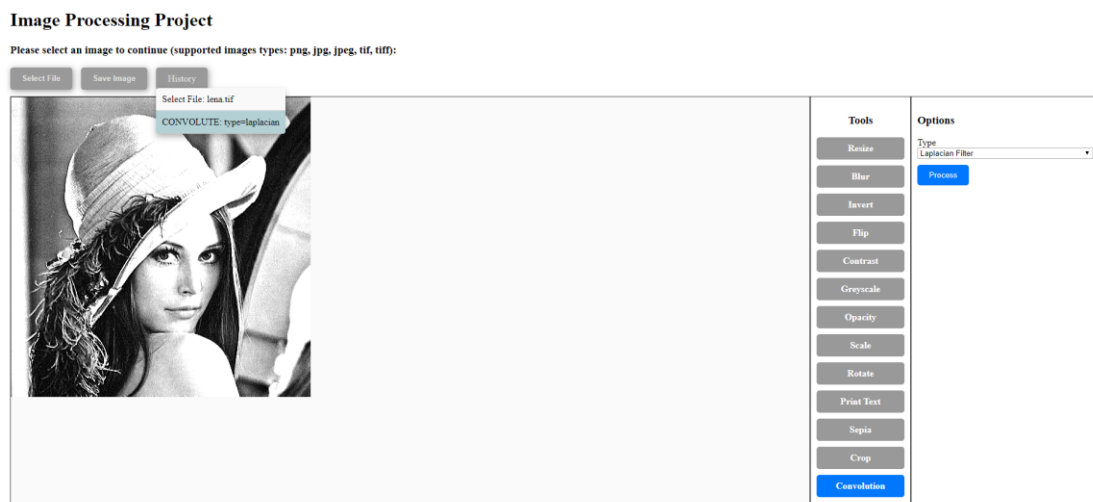


Figure 2.2 – Popup History List while hover the “History” button

3. Examples

For original image I will used figure 3.1 to test several image processes functions and compare the difference before and after, figure 3.2 shown that the figure 3.1 original image rendered in the system.



Figure 3.1 – lena.tiff (512x512)

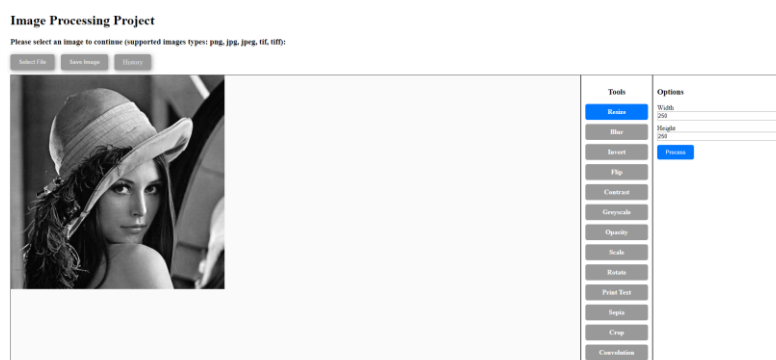


Figure 3.2 – lena.tiff show in system

3.1 Resize Image

Options: width=250, height=250

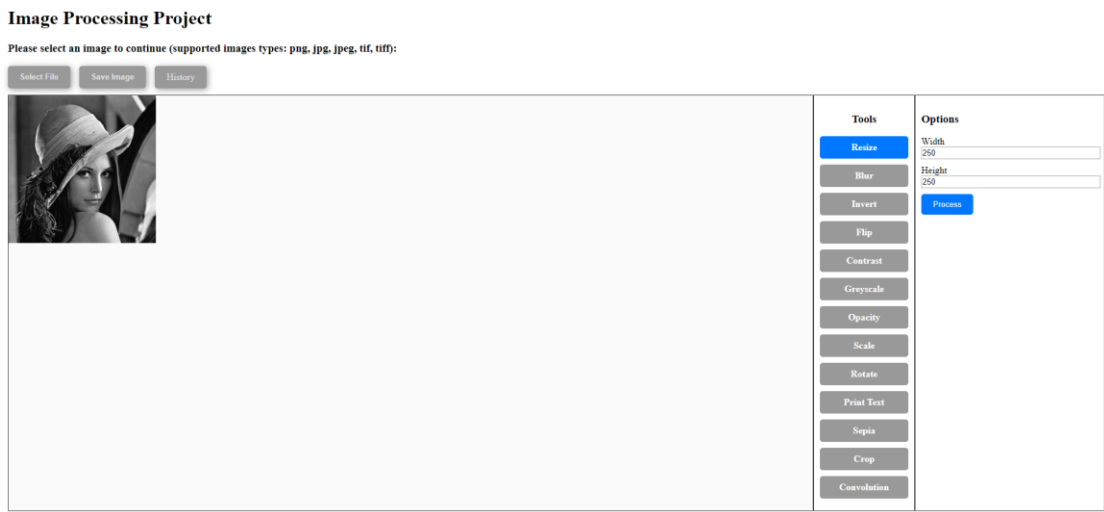


Figure 3.3 – lena.tiff applied Resize Function (250x250)

3.2 Blue Image



Figure 3.4 – lena.tiff applied Blur Function (blur level = 1)

3.3 Invest Image Color



Figure 3.5 – lena.tiff applied Invest Function

3.4 Flip Image

3.4.1 *Horizontal Flip Image*



Figure 3.6 – lena.tiff applied Horizontal Flip Function

3.4.2 *Vertical Flip Image*



Figure 3.6 – lena.tiff applied Vertical Flip Function

3.4.3 Horizontal and Vertical Flip Image



Figure 3.7 – lena.tiff applied Horizontal and Vertical Flip Function

3.5 Contrast Image



Figure 3.8 – lena.tiff applied Contrast Effect

3.6 Change Image Opacity



Figure 3.9 – lena.tiff applied Opacity Effect (opacity level = 0.5)

3.7 Scale Image Size

Image Processing Project

Please select an image to continue (supported images types: png, jpg, jpeg, tif, tiff):

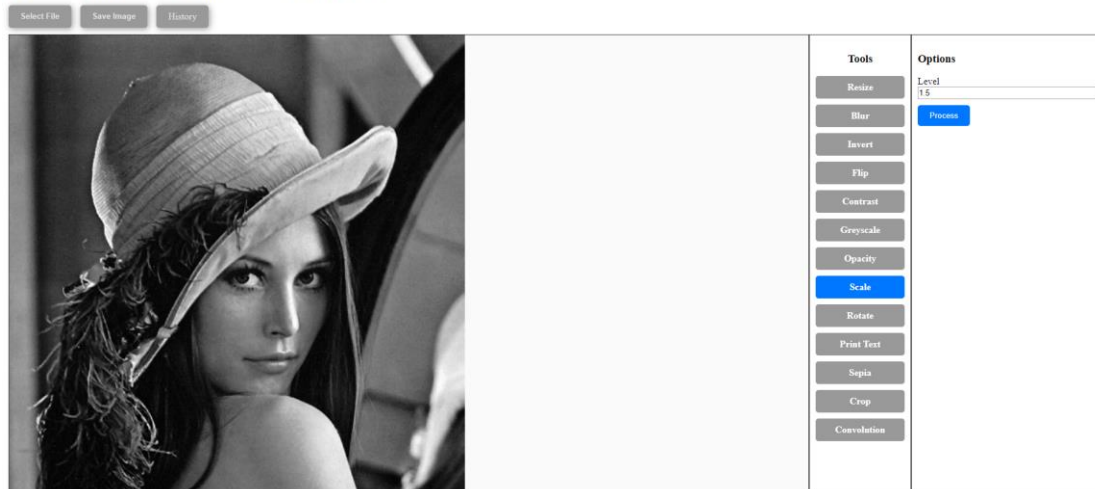


Figure 3.10 – lena.tiff applied Scale Image Size Effect (scale level = 1.5, size=768x768)

3.8 Rotate Image



Figure 3.11 – lena.tiff applied Rotate Effect (rotate degree 90)

3.9 Print Text

In this function, figure 3.12 shown another way to set text position.

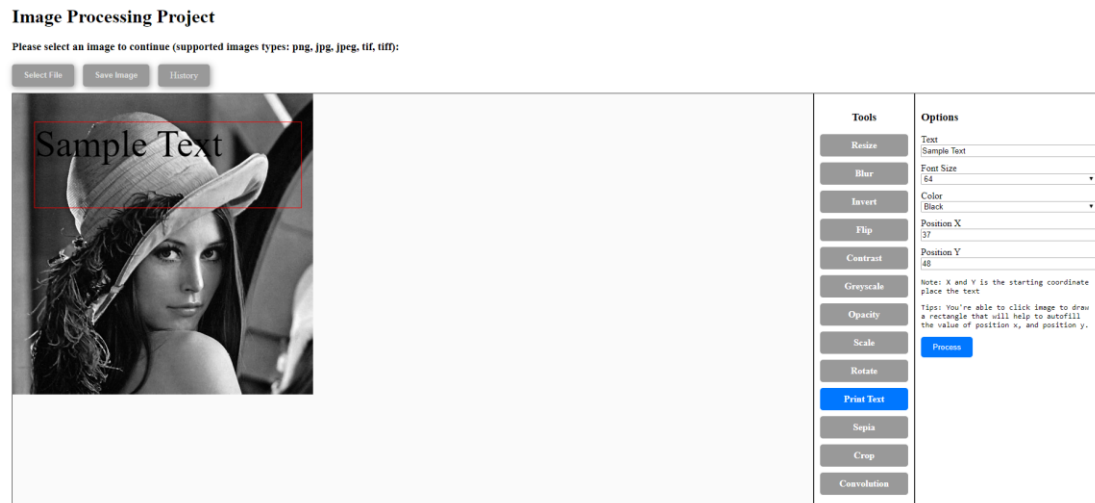


Figure 3.12 – Popup Drawing Box to Autofill Position Value

User can follow the tips in the options box to click the image once to start drawing box and click again to end the drawing box. After the drawing box is closed, system will autofill the position in the box. But the print text function cannot set the width and height directly so it will be no wrap text print over the image with the position x and y.



Figure 3.13 – lena.tiff applied Print Text Effect (Text="Sample Text", Font Size=64, Color=Black, Position X=296, Position Y=110)

3.10 Sepia Image Color



Figure 3.14 – lena.tiff applied Sepia Image Color Effect

3.11 Crop Image

In this function that is same with 3.9 function, there are able to click the image to drawing a box with the red color border, move the mouse and click again to close the box that will autofill the width, height, position x and y values in the options area.



Figure 3.15 – lena.tiff applied Crop Effect (width=378, height=402, x=66, y=25)

3.12 Convolution

3.12.1 Low-Pass Filter



Figure 3.16 – lena.tiff applied Low-Pass Filter

3.12.2 High-Pass Filter



Figure 3.17 – lena.tiff applied High-Pass Filter

3.12.3 Directional Filter



Figure 3.18 – lena.tiff applied Directional Filter

3.12.4 Laplacian Filter



Figure 3.19 – lena.tiff applied Laplacian Filter

4 Installation

Since, this project source code I already uploaded to GitHub which is offered the distributed version control and source code management (SCM) functionality of Git, thus anyone can download and clone the project's source code with following links:

- GitHub Link: <https://github.com/zhangen69/image-processing-student-project>
- Git Checkout: <https://github.com/zhangen69/image-processing-student-project.git>

Before you start to run the project, please make sure your laptop or computer is installed nodejs from <https://nodejs.org/>.

After cloned the project's source code, open terminal/power shell and change directory to the project folder and run following commands:

- Change directory to project folder: `cd your_project_folder_path`
- Install node_modules package: `npm install`
- Run the application: `node app.js`
- Open the application in browser: <http://localhost:3000>

After you open the application, there will be like figure 4.1, and please note that you need to change directory to specific drive if you're place the project folder not in C drive, so you need run the command:

- Change to D drive: `D:`

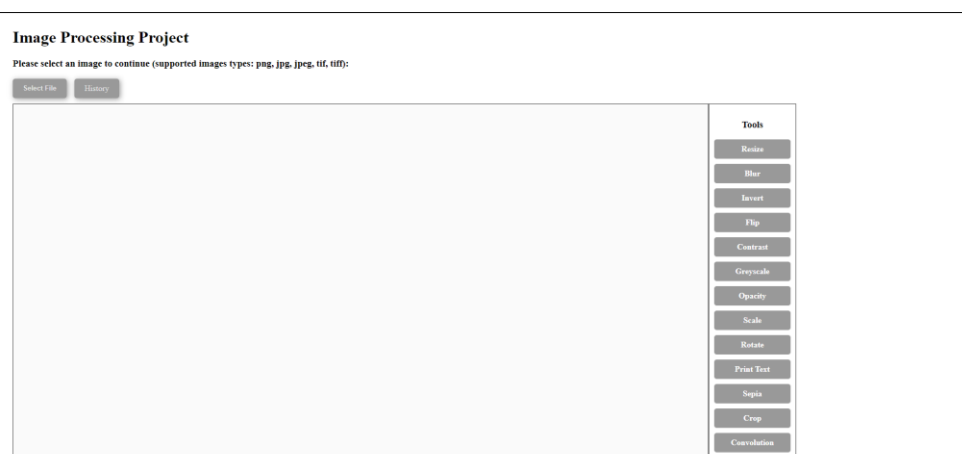


Figure 4.1 – index.html

5 Conclusion

In this project, I learned to apply the theory of image processing into practical and handling image data. In this system, I used to convert the image file to Base64 Data URL before process the image and pass it into the JSON type format send to the server-side through HTTP requests. In the beginning, I'm stuck in how the server-side can receive the image file and convert, finally, I found Base64 Data URL is a string type image data but the client-side and server-side can readable to processing and rendering. But this creates another problem which is request size is larger while the image data has been processed several times with different effects and solved it with set larger request limit.