

Matrix Convolution Image Processing

Multimedia Information System: Final Project Report

Sahli Yacine
E10715005

2018-2019

National Taiwan University of Science and Technology(NTUST)

2018-12-18

Abstract

This *implementation report* is handed for the course **MI5300701** "Multimedia Information Systems", given by the professor *Yáng Chuán Kǎi* during the academic year *2018-2019* .

The goal of this report is to explain how my final project about Convolution Matrix Image Processing work.

I will start by explaining what is the goal of this project and the concept of convolution matrix image processing.

Then I will explain each functionality of my program and i will finish by enumerating what is yet to improve.

1	Goals	3
2	What is a convolution Matrix(Kernel) ?	4
2.1	Edge handling	4
2.2	Normalization	4
3	How to use the program ?	5
4	What can be done ?	6
5	To be improved	7

1 Goals

List of the goals I fixed myself with a grade on 5 according to the accomplishment of each one of them:

- The program should be able to apply Convolution Matrix to images and save the modification in a new image. 5/5
- The user interface should be easy to use and simple. 4/5
- Anyone should be able to use this application. 3/5
- A set of comprehensible kernels should be available. 5/5
- At least 2 different ways to apply the kernel on the border's pixel should be given to the user. 5/5
- Possibility to transform the image in a gray-scale. 5/5
- It should work on Windows, MacOS and Linux. 5/5

2 What is a convolution Matrix(Kernel) ?

In image processing, a convolution matrix or convolution kernel is a small matrix, usually 5x5 or 3x3, used for image transformation such as embossing, blurring, sharpening, etc...

It works by applying a convolution between the kernel and an image.

Convolution is the process of adding each element of the image to its local neighbors, weighted by the kernel.

2.1 Edge Handling

- **Extend**

The value of the closest pixel is extended beyond the border.

- **Wrap**

The value is taken from the opposite border.

- **Mirror**

The value is taken from inside the picture as in a mirror, for example, if the pixel is 3 pixel out of the picture, we will take the value of the pixel which is 3 pixel inside the picture from the border.

2.2 Normalization

Normalization consist in the division of each element in the kernel by the sum of all kernel elements, so that the sum of the elements of a normalized kernel is one. We do this so that the average pixel in the modified image is as bright as the average pixel in the original image.

3 How to use the program ?

First, you should choose an image on your computer by clicking "Select an image" button, then you can select a convolution matrix preset in the "Presets" option bar at the top of the application's window.

If you want, you can input your own convolution matrix manually in the "MATRIX" frame and input the according divisor necessary in the "Divisor" input box

Then, you can check the "Grayscale" check-box if you would like your output image to be gray-scale.

Choose how pixels at the border of the image should be handled in the "Border" frame by clicking on the according radio-box.

You can now press preview to get a small, but very quickly computed, thumbnail of the output image.

Before saving and starting the computation, you should choose whether you want the image to be scaled down to 1024 on its largest dimension so that the computation is not too long.

The aspect ratio will be guaranteed !

Then you can save the image and choose where you want to save it, the calculation will be done in a separate thread so that the graphical user interface stays responsive.

4 What can be done ?

A lot of things can be done with convolution Matrix, such as :

- Blurring
- Sharpening
- Unsharpening
- Embossing
- Edge detection
- Edge enhancing
- Denoising
- It can help us understand convolution in general and apply it to other domain such as Machine Learning

The user is free to experiment his own Matrixes in the program.

5 To be improved

- Use numpy module to make the process faster as using python's list on large matrix is not very efficient.
- Rewrite from scratch in a lower level programming language like C the computing part for faster calculation by using python extension.
- Propose the user to apply the convolution kernel in a set of specifics color channels.
- Put the preview image directly into the application window instead of opening a new window.
- Let the user resize the image in his desired way.
- Make the user interface more user friendly and eye catching.
- Enabling the possibility to process multiple images at the same time in different threads.
- Add a cropping functionality
- Make the program compute the divisor by itself instead of asking for it as an input. (but choosing the divisor can also be useful for the user)
- Implement a live preview, maybe with a refresh rate of 1 or 2 times every 5 seconds.
- Rewrite the user interface so that it looks better when resizing the window.