Image Processing HW1

Shang-Ying,He
communication engineering
National Tsing Hua University
Hsinchu,Taiwan. R.O.C.
sunnyhe1122@gmail.com

I. Introduction

Program about the RAW image file's loading, saving, showing and some image processing features without using any library contain image reading function just like OpenCV, PIL, skimage by python. And you can see the result what you do with current image from interface instantly.

II. Ease of Use

A. Reading RAW Image File Into Type of Array

Use function 'open' to load the file into type of bytes and unpack the structure into tuple with appropriate parameter. Then, translate tuple into array. Array's elements would have numbers of pixel in RAW image file from 0 to 255 correctly. Then reshape this array to suitable shape and loading is complete at the same time.

B. Write Image Array Into Bitmap Format

Bitmap format contain 54 byte header and bitmap array. So, first we should design header relate to the image. Follow Table I. to design the header. Then pack the header and bitmap array into structure, and bitmap data is complete. Next step we just to write the structure into .bmp of deputy file name.

TABLE I.

	Shift	***Name***	***Size (bytes)***
Bitmap File Header	0000h	Identifier(ID)	2
	0002h	File Size	4
	0006h	Reserved	4
	000Ah	Bitmap Data Offset	4
Bitmap Info Header	000Eh	Bitmap Header Size	4
	0012h	Width	4
	0016h	Height	4
	001Ah	Planes	2
	001Ch	Bits Per Pixel	2
	001Eh	Compression	4
	0022h	Bitmap Data Size	4
	0026h	Horizontal Resolution	4
	002Ah	Vertical Resolution	4
	002Eh	Used Colors	4
	0032h	Important Colors	4
Bitmap Array	-	Bitmap Data	-

C. Sobel Mask

Sobel mask is first-order derivative to do edge detection. Convolution source image with the kernels and combine the filtered image which detect with x axis and y axis independently. (1) is the kernel detect x axis and (2) is y axis.

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{2}$$

D. Laplacian Mask

Laplacian mask is a function, using convolution to detect sharpening spatial of image. (3) is the laplacian kernel.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{3}$$

E. Averaging Mask

Just as the name implies, averaging mask is a function to average nearest pixels in spatial of image. This task can denoise the image. (4) is a kernel of average mask so we can understand what it means. Replace each pixel with an average of its neighborhood.

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{4}$$

F. Gaussian Mask

It's a function to do convolution for source image with Gaussian kernel. When you transform Gaussian and average kernels into frequency domain, you can observe that they are very similar, so it would reach similar result as average kernel. So, it's also a task to denoise the image. And Gaussian filter would be more smooth than average filter.

G. Add Gaussian Noise

It's a task to add random Gaussian noise in image. First, I create an array as the same shape of source image and set normal distribution with elements, and then combine these two array to get the result which have the noise in image.

H. Average of 100 Gaussian Noise

Create numbers of 100 added Gaussian noise image randomly, and average them to denoise. This way is to get enough sample and when samples are large enough, that sample space will be a standard Gaussian distribution, call Central Limit Theorem(CLT). And when we average these 100 samples, we will get deviation approach to zero. Just like no noise in image.

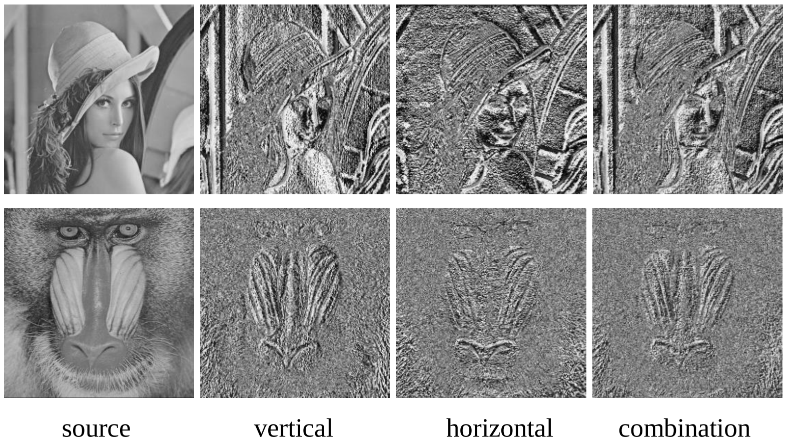

Figure 1. sobel mask

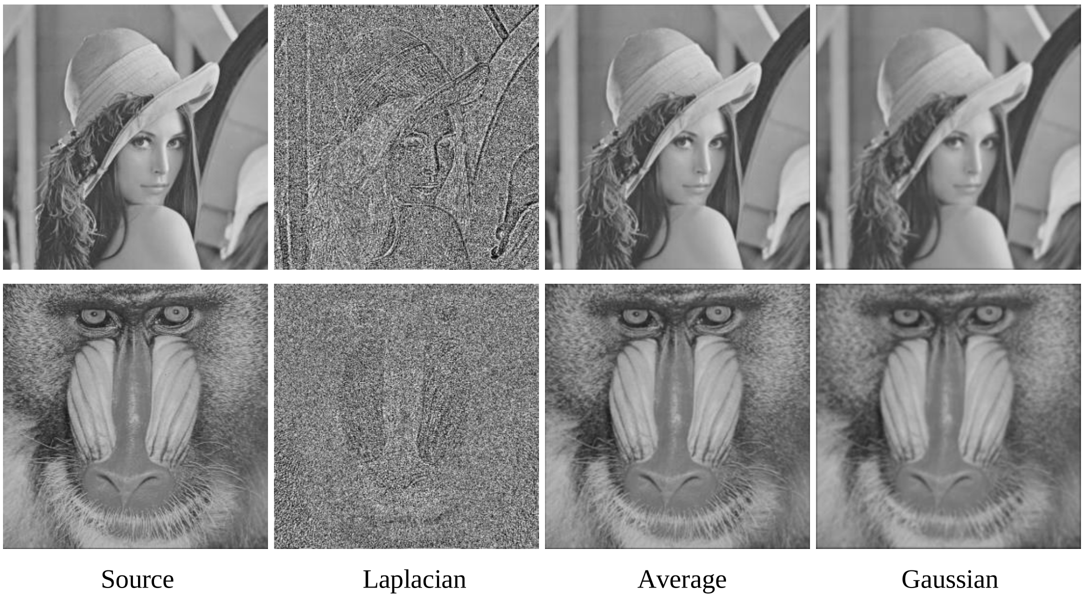

Figure 2. Laplacian, average and Gaussian mask

III. EXPERIMENTS

A. Spatial Filtering

Result of sobel mask in figure 1. we can observe the results in vertical and horizontal kernel obviously detect their feature. Combination of two features in Lena might be more easy than baboon to observe.

Result of Laplacian mask in figure 2. we can observe that kernel detected sharpening features in source image. And the other thing we can do is to combine sharpening feature and source image, see the result of this task in figure 3. is what I done for this task. But we can observe that these image may not be a good choice for this task.

Result of aversge and Gaussian mask in figure 2. If we zoom in the image can observe that Gaussian filtered image is more smooth than average filtered image.

B. Denoise

The result of denoise in figure 4. are very obvious to observe Gaussian mask seems to work better than average mask a little. But when deviation of Gaussian noise is too large, denoise with these two way are not enough to revert image.

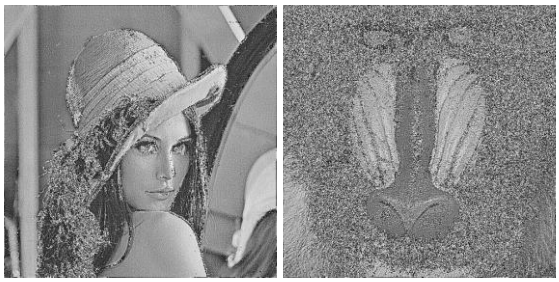

Figure 3. source combination with Laplacian filtered image

So the other way to denoise is use CLT, result of this way in figure 5. ,when deviation equal to 10, it’s much better than average mask and Gaussian mask.

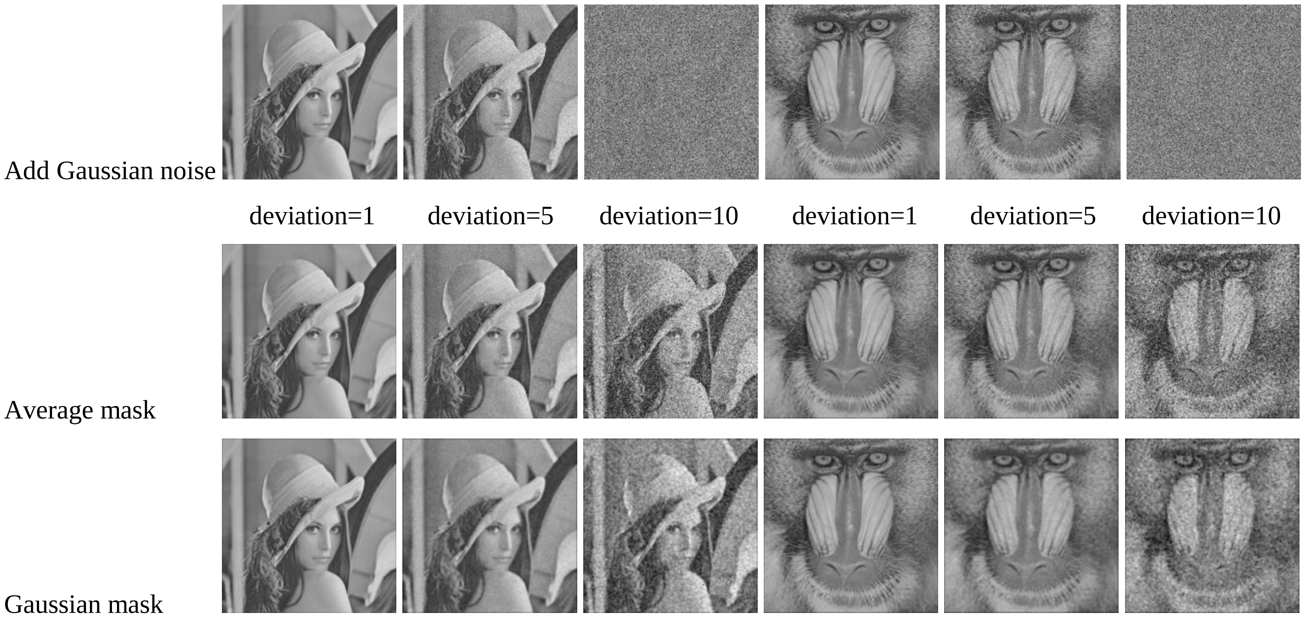

Figure 4. denoise by average mask and Gaussian mask

Figure 5. denoise by average of 100 Gaussian noise

IV. CONCLUSION

This project let me know more about what teacher is teaching in this course and how to realize in the program. I cost most of time to understand how to read image as an array and how to write image array into bitmap format and how to show image instantly without using limited library. But when I done this and program is working well, I feel worth it. It enhance my compile ability and learn more. Not only learn more about data processing but also learn more about image processing. All tasks in this project would help my research in the future.