Point Process-Ordered Dithering using the Classical-4 & Bayer-5 Dither Array

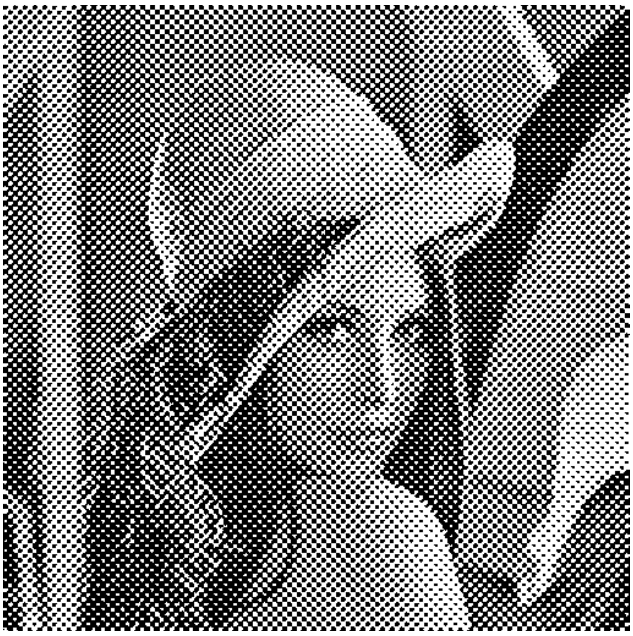
- **Code**

Code please look the path "../Code/Problem_1/Point Process-Ordered Dithering.py".

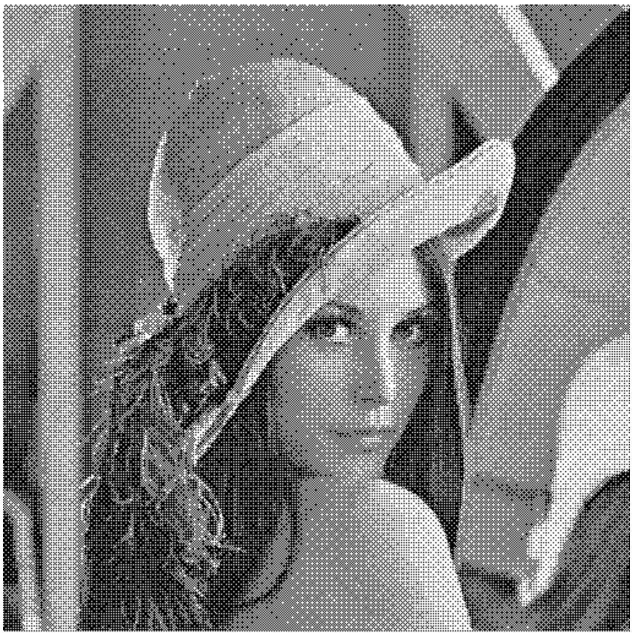
- **Result**

The three result images are shown on the down side. The source image please look the "transformed_by_classical_four.bmp" and "transformed_by_bayer_five.bmp" form the path "../Code/Problem_1/".

Original Lena



Classical-4



Bayer-5

Neighborhood Process – Error Diffusion

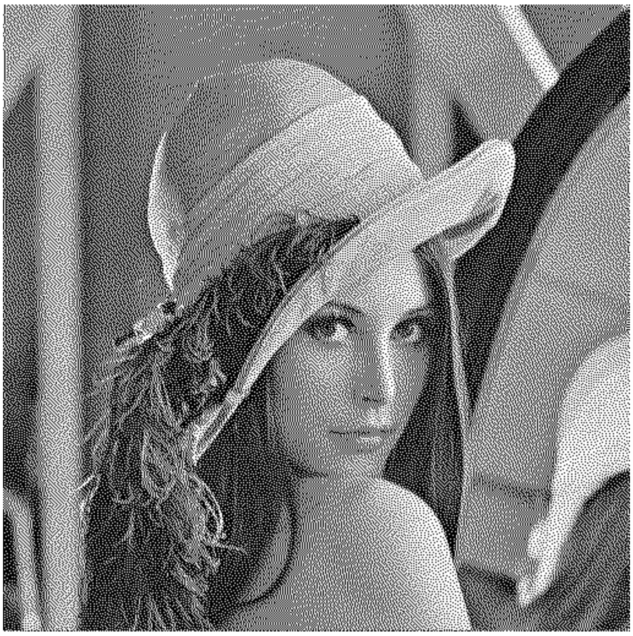
Code

Code please look the path "../Code/Problem_2/ Error_Diffusion.py".

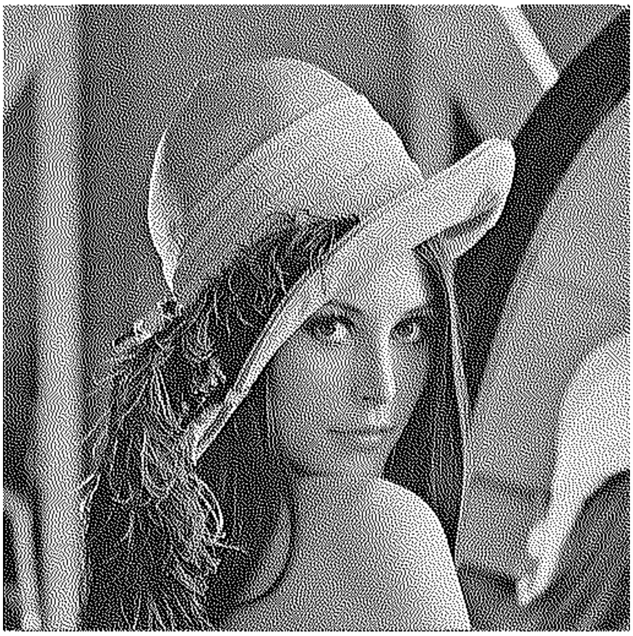
Result

The three result images are shown on the down side. The source image please look the "Error_Diffusion_Floyd_Steinberg.bmp", "Error_Diffusion_Jarvis.bmp" and "Error_Diffusion_Stucki.bmp".

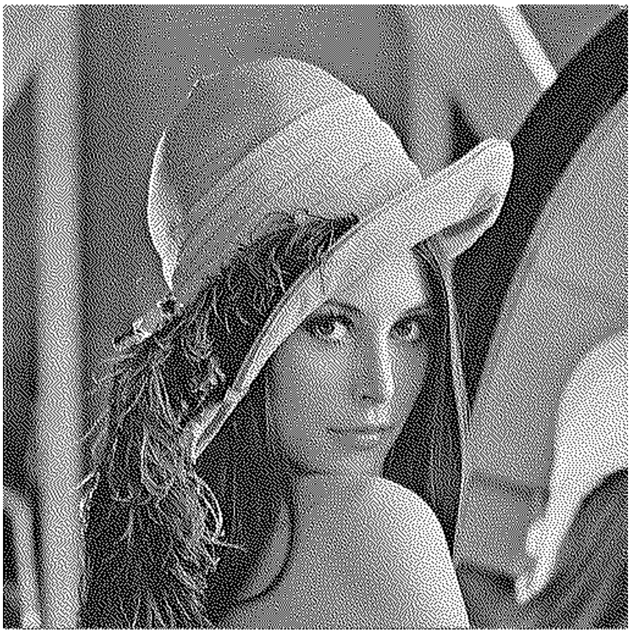
Original Lena



Floyd_Steinberg



Jarvis



Stucki

Discussion

I don’t encounter the problem during doing the first home work point process-ordered dithering. I’m surprised just change the number in dither array and algorithm does not change the image quality will be improved very obviously.

I explain the reasons for several error image output that I encountered when writing the program in the second homework. First, I found the output picture color black and white are reversed when because I take the pixel value that is changed to 0 or 255 to doing calculation instead of using original pixel value. Second, I write each error diffusion method as function but when I calling functions and executing them one by one I found the 3 output picture will be same. Note I’m sure that individual execution the function is normal. Later I noticed that I had to use another numpy array to store data from cv2 read image pixel value when using python language. Third, the numpy array data type must be floating-point number. I use numpy array with double type to store data and calculations because when I use int type, the image has a little mistake in visual.