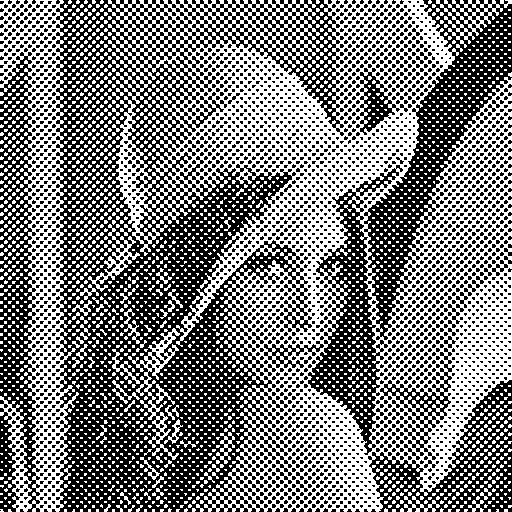
**Point Process-Ordered Dithering using the Classical-4 & Bayer-5 Dither Array**

* **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**

**Bayer-5**

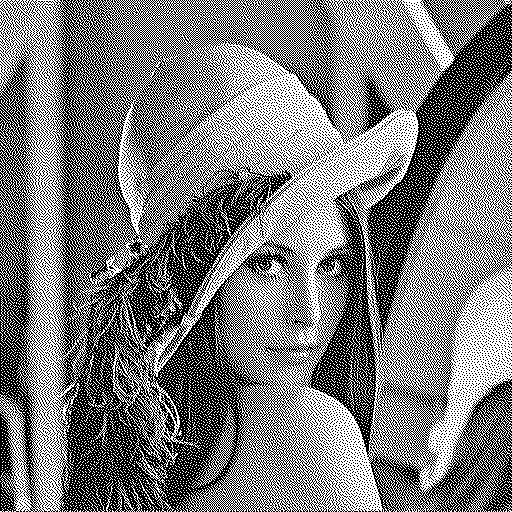
**Classical-4**

**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**

**Stucki**

**Jarvis**

**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.