HW8 report

姓名:徐有慶　學號:R05922162

1. 作業要求:

   Noise removal

   程式語言: Java

   執行環境: Eclipse

2. 程式設計:

   - Box filter(3*3, 5*5): Output value of neighborhood outlier removal

$$Z_{outlier\_removal} = \begin{cases} y & if\,|y-\hat{\mu}|<\theta \\ \hat{\mu} & otherwise \end{cases}$$

   - Median filter(3*3, 5*5): Median

$$Z_{median} = [x_{(\frac{N+1}{2})}]$$

   - Opening then closing

3. 主要程式:

   Generate additive white Gaussian noise (./Surce Code/src/hw8/GaussianNoise.java):

```java
public class GaussianNoise {
    public static int[][] addNoise(int[][] img, float amp){
        int height = img.length;
        int width = img[0].length;
        int[][] noiseImg = new int[height][width];
        Random random = new Random();
        for(int i = 0; i < height; i++){
            for(int j = 0; j < width; j++){
                noiseImg[i][j] = img[i][j] + Math.round((float)(amp * random.nextGaussian()));
            }
        }
        return noiseImg;
    }
}
```

Generate salt-and-pepper noise (./Surce Code/src/hw8/SaltAndPepperNoise.java):

```java
public class SaltAndPepperNoise {
    public static int[][] addNoise(int[][] img, float fraction) {
        int height = img.length;
        int width = img[0].length;
        int[][] noiseImg = new int[height][width];
        Random random = new Random();
        for (int i = 0; i < height; i++) {
            for (int j = 0; j < width; j++) {
                float ranNum = random.nextFloat();
                if (ranNum < fraction / 2) {
                    noiseImg[i][j] = 0;
                } else if (ranNum > 1 - fraction / 2) {
                    noiseImg[i][j] = 255;
                } else {
                    noiseImg[i][j] = img[i][j];
                }
            }
        }
        return noiseImg;
    }
}
```

Box filter (./Surce Code/src/hw8/OutlierRemovalClean.java):

```java
public int[][] clearNoise() {
    int height = inputImg.length;
    int width = inputImg[0].length;
    int edgeX = (int)Math.floor(windowHeight / 2);
    int edgeY = (int)Math.floor(windowWidth / 2);
    int[][] outPutImg = inputImg;
    int[] window = new int[windowHeight * windowWidth];

    for (int i = edgeX; i < height - windowHeight; i++) {
        for (int j = edgeY; j < width - windowWidth; j++) {
            int m = 0;
            for(int fx = 0; fx < windowHeight; fx++){
                for(int fy = 0; fy < windowWidth; fy++){
                    window[m] = inputImg[i + fx - edgeX][j + fy - edgeY];
                    m += 1;
                }
            }

            float meanValue = mean(window);
            if (Math.abs(outPutImg[i][j] - meanValue) >= theata) {
                outPutImg[i][j] = Math.round(meanValue);
            }
        }
    }
    return outPutImg;
}
```

Median filter (./Surce Code/src/hw8/RunningMedianClean.java):

```
public int[][] clearNoise() {
    int height = inputImg.length;
    int width = inputImg[0].length;
    int edgeX = (int)Math.floor(windowHeight / 2);
    int edgeY = (int)Math.floor(windowWidth / 2);
    int[][] outPutImg = inputImg;
    int[] window = new int[windowHeight * windowWidth];
    for (int i = edgeX; i < height - windowHeight; i++) {
        for (int j = edgeY; j < width - windowWidth; j++) {
            int m = 0;
            for(int fx = 0; fx < windowHeight; fx++){
                for(int fy = 0; fy < windowWidth; fy++){
                    window[m] = inputImg[i + fx - edgeX][j + fy - edgeY];
                    m += 1;
                }
            }
            Arrays.sort(window);
            int N = window.length;
            int median = window[((int) Math.floor((N + 1) / 2)) - 1];
            outPutImg[i][j] = median;
        }
    }
    return outPutImg;
}
```

Opening then closing:

同 hw5

4. 執行結果：

Noise image (Gaussian: 10, Gaussian: 30, S&P: 5%, S&P: 10%)

SNR

Gaussian (10): 13.615819497134826

Gaussian (30): 4.057803893690471

S&P (5%): 3.955295398799479

S&P (10%): 0.9288405501400461

3 * 3 Box filer (Gaussian: 10, Gaussian: 30, S&P: 5%, S&P: 10%)

SNR

Gaussian (10): 13.747598588876668

Gaussian (30): 5.126332946909795

S&P (5%): 4.688920816042634

S&P (10%): 1.590767698550887

5 * 5 Box filter (Gaussian: 10, Gaussian: 30, S&P: 5%, S&P: 10%)

SNR

Gaussian (10): 11.290456149935391

Gaussian (30): 3.9033410161802298

S&P (5%): 3.590943008379444

S&P (10%): 0.5269786575230377

3 * 3 Median filter (Gaussian: 10, Gaussian: 30, S&P: 5%, S&P: 10%)

SNR

Gaussian (10): 12.729684460347787

Gaussian (30): 4.111376657582408

S&P (5%): 3.901137313572179

S&P (10%): 1.0197480616148296

5 * 5 Median filter (Gaussian: 10, Gaussian: 30, S&P: 5%, S&P: 10%)

SNR

Gaussian (10): 10.453647077612569

Gaussian (30): 2.4531150255384913

S&P (5%): 3.58323683578921

S&P (10%): 0.8207925720264584

Opening then closing (Gaussian: 10, Gaussian: 30, S&P: 5%, S&P: 10%)

SNR

Gaussian (10): 10.877038463697975

Gaussian (30): 3.5653693172636327

S&P (5%): 3.594966324012277

S&P (10%): 0.8142031263356835

5. 如何執行

執行./source code/ hw8.jar 即可產生作業要求的兩種圖片(lena.bmp 需要和 hw8.jar 在同一個資料夾底下)

程式進入點為 ./Surce Code/src/hw8/DemoNoiseRemoval.java