

高等影像處理

Homework 3: Generation of Gaussian noise and salt-and-pepper noise

學號： 61247051S

姓名： 陳建壹

一、本作業所用之程式語言及編譯器

Language: C#

Compiler: Roslyn

二、程式功能

1. HW1, HW2 之功能
2. 對輸入影像產生 Gaussian Noise
3. 對輸入影像產生 Salt&Pepper Noise

三、程式流程或演算法

```
Image SaltPepperNoise(input, noiseValue){  
  
    output = new(input.width, input.height);  
  
    for( y: 0 -> input.width) {  
        for( x:0 -> input.height) {  
  
            rnd = Random(0 ~ 100);  
  
            if (rnd <= noiseValue)  
                output.SetPixel(x,y, Color.Black);  
  
            elif(rnd >= 100 - noiseValue)  
                output.SetPixel(x,y, Color.White);  
        }  
    }  
    return output;  
}
```

```

Image GaussianNoise(RawImage input, out RawImage noise, float sigma) {

    int width = input.Width;
    int height = input.Height;

    Image output = new(width, height);

    for (y : 0 -> height) {
        for (x : 1 -> width; x+=2) {
            varphi = Random (0~1);
            gamma = Random (0~1);


$$z_1 = \sigma \cos(2\pi\varphi) \sqrt{-2 \ln \gamma}$$


$$z_2 = \sigma \sin(2\pi\varphi) \sqrt{-2 \ln \gamma}$$

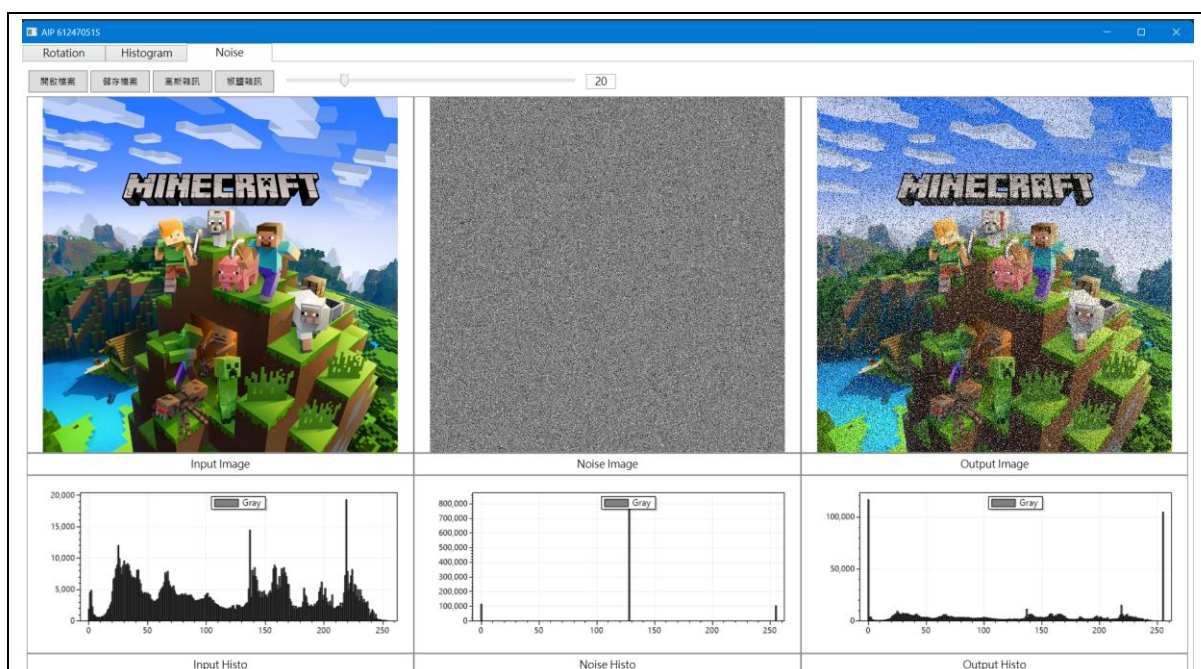

            pixel1 = input.GetPixel(x - 1, y);
            pixel2 = input.GetPixel(x , y);

            noised1 = Clamp(pixel1[B] + z1, 0, 255);
            noised2 = Clamp(pixel1[G] + z2, 0, 255);

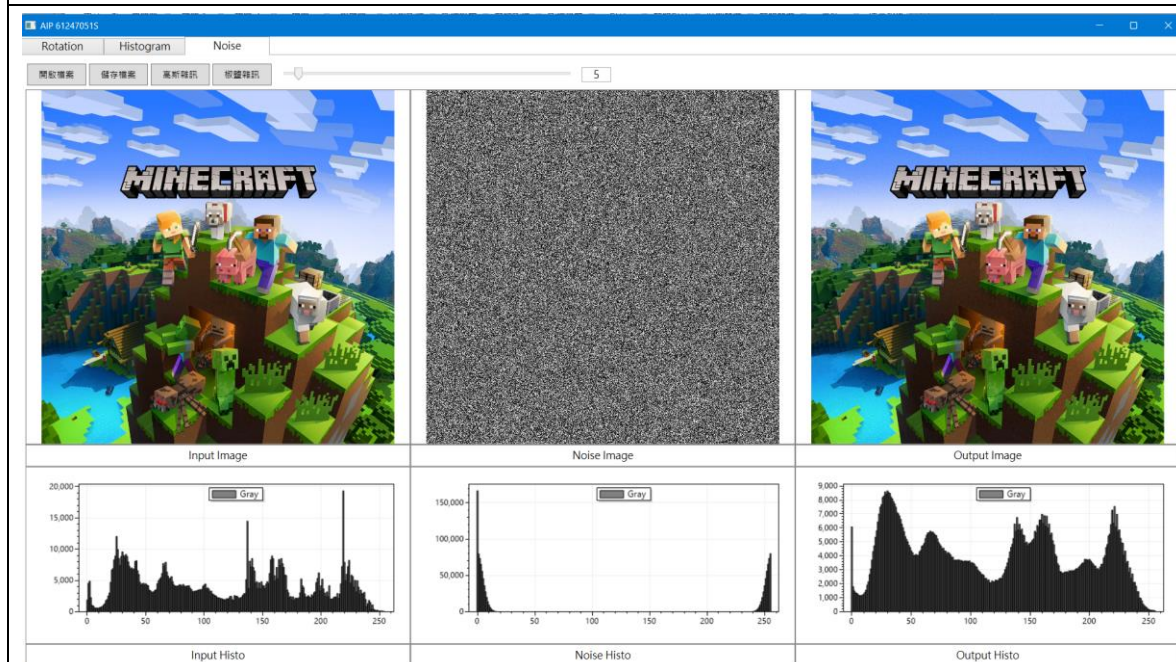
            output.SetPixel(x - 1, y, noised1);
            output.SetPixel(x, y, noised2);
        }
    }
    return output;
}

```

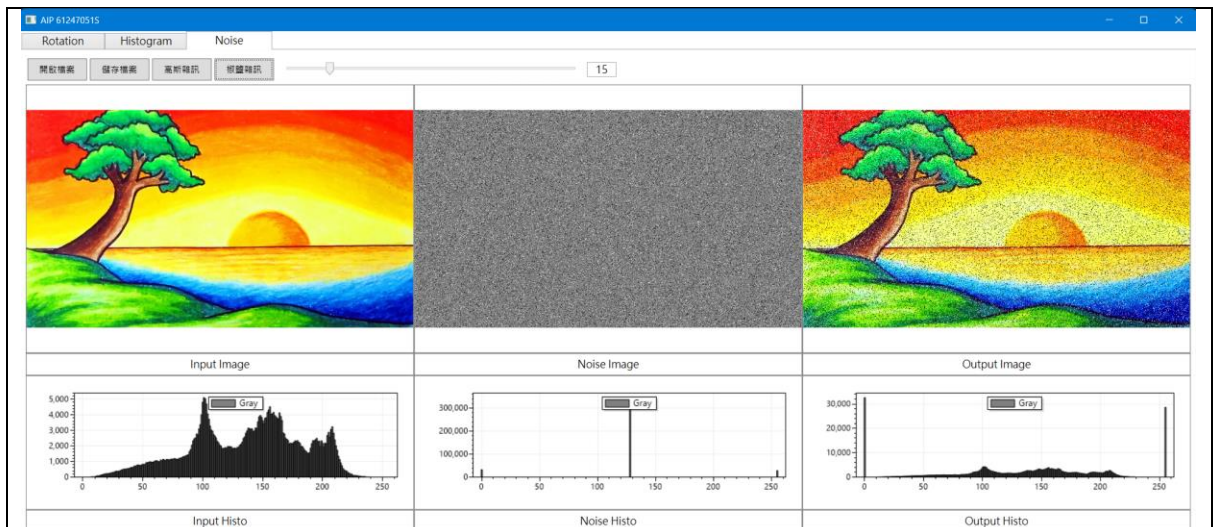
四、測試結果



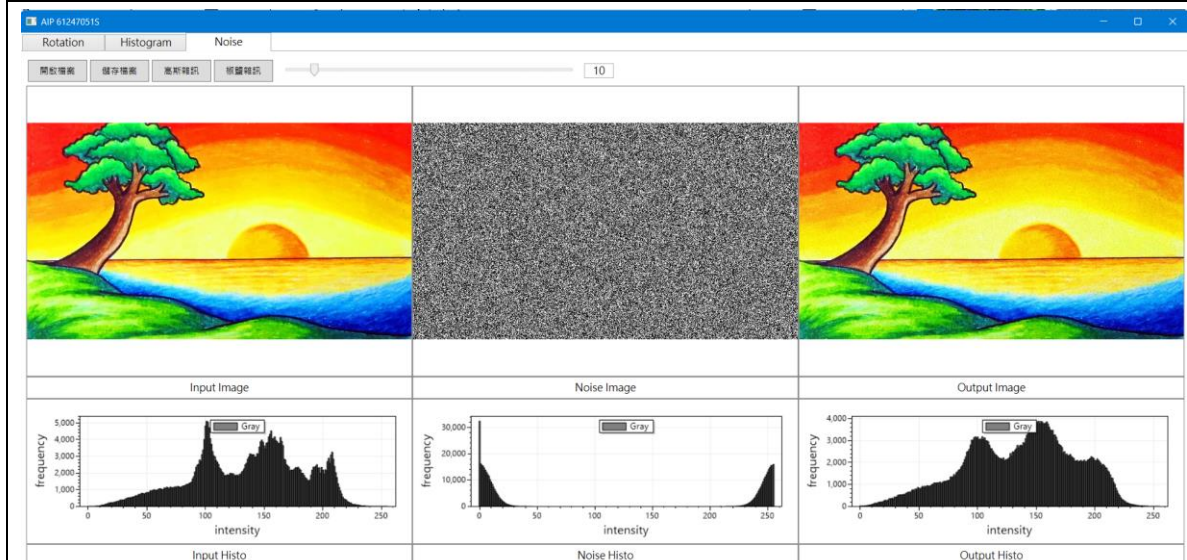
範例一為[椒鹽雜訊 20%]之測試結果。左圖為 input，中間圖為 noise。由右圖 output 可觀察到圖中散佈黑色與白色顆粒。



範例二為[高斯雜訊 $\sigma = 5$]之測試結果。左圖為 input，中間圖為 noise。由右圖 output 可以觀察到其 histogram 被平滑化



範例一為[椒鹽雜訊 20%]之測試結果。左圖為 input，中間圖為 noise。由右圖 output 可觀察到圖中散佈黑色與白色顆粒。



範例三為[高斯雜訊 $\sigma = 10$]兩次之測試結果。左圖為 input，右圖為 output。由右圖 output 可以觀察到其 histogram 被平滑化

五、程式撰寫心得

這次的雜訊產生的計算量比較大，但還是可以在 2 秒內將 noise, output 及其 histogram 都繪製出來。之後的作業考慮進一步的改進。

處理高斯雜訊時，學到了如何使用數學生成符合高斯分佈的隨機數，並將其應用到影像上，加強數學運用在 coding 上熟練度。