Homework No.7

Student ID: 41047902S Name: 鄭淮薰

Problem Statement

1. Create an image g(x,y) whose pixels all have the same gray value of 100. Show the image g(x,y).
2. Generate Gaussian noise n(x,y), with $\mu = 0, \sigma^2 = 25$, using the algorithm shown in the next page. Show the noisy image f(x,y) = g(x,y) + n(x,y).
3. Display the histogram h(i) of f(x,y).
4. Comment on your results

Results

Test Case1：$\sigma = 5$

Fig1.1 為一灰階值均為 100 的灰色圖片，作為 input image $g(x, y)$，而 Fig1.2 為此 input 的 Histogram。此圖以 $\sigma = 5$ 加上 Gaussiam noise 後如 Fig1.3 所示，可見圖中出現些許雜訊，而從此 Noisy Image 的 Histogram Fig1.4 中也可見各 pixel value 的機率分佈以 `100` 為中心向外擴展了一點。

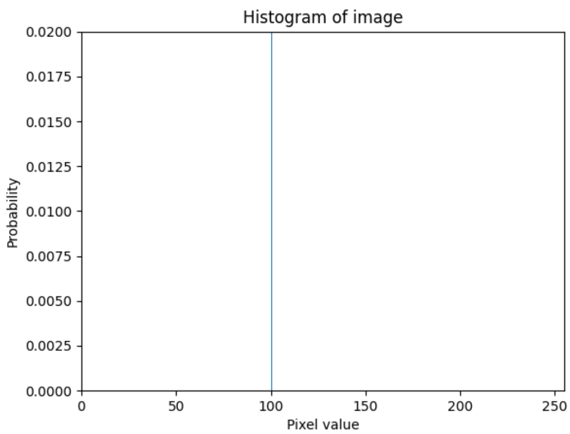
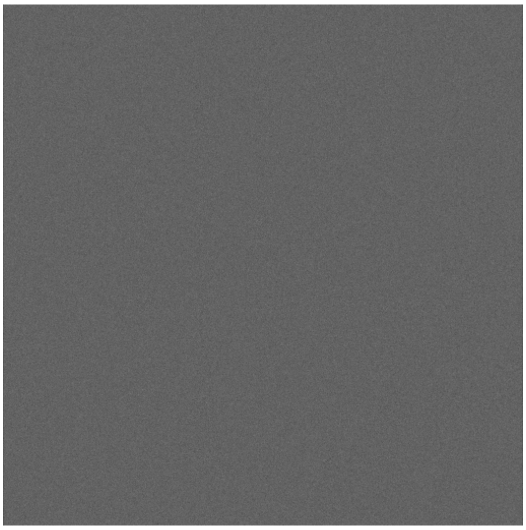
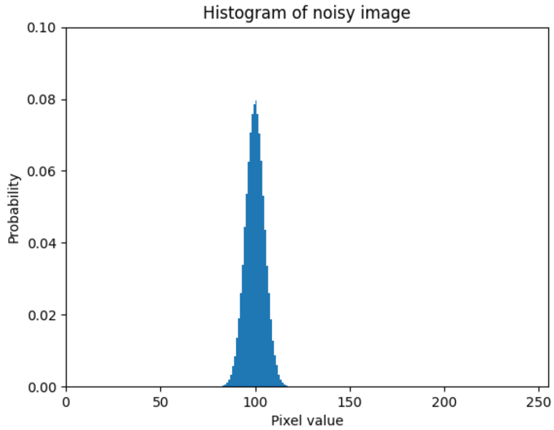
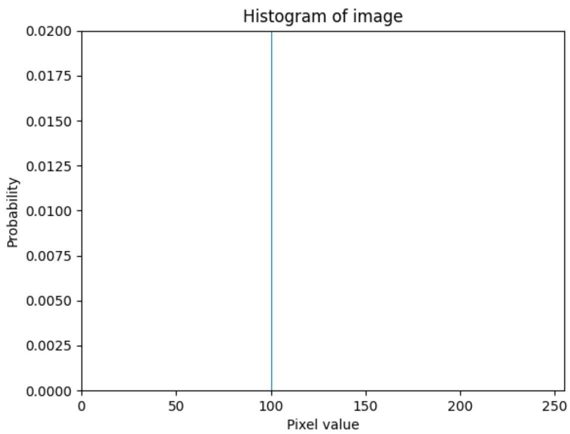
Fig1.1 Input Image g(x,y)	Fig1.2 Histogram of g(x,y)
	

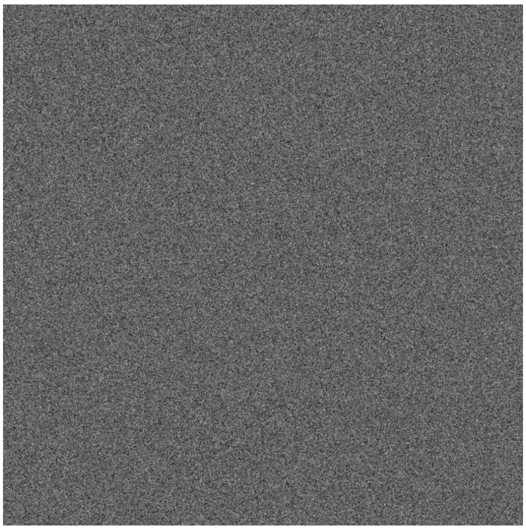
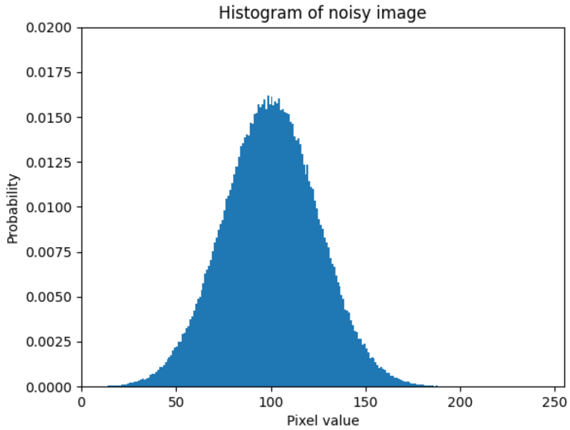
Fig1.3 Noisy Image f(x,y)	Fig1.4 Histogram of f(x,y)
	

Test Case 2：$\sigma = 25$

Fig2.1 為一灰階值均為 100 的灰色圖片，作為 input image $g(x, y)$，而 Fig2.2 為此 input 的 Histogram。此圖以 $\sigma = 25$ 加上 Gaussiam noise 後如 Fig2.3 所示，可見圖中出現明顯雜訊，而從此 Noisy Image 的 Histogram Fig2.4 中也可見各 pixel value 的機率分佈圖與 normal distribution 極為相似。

(此處為觀察方便，Fig2.4 Histogram 之 y 軸顯示範圍改為 0 ~ 0.02)

Fig2.1 Input Image g(x,y)	Fig2.2 Histogram of g(x,y)
	

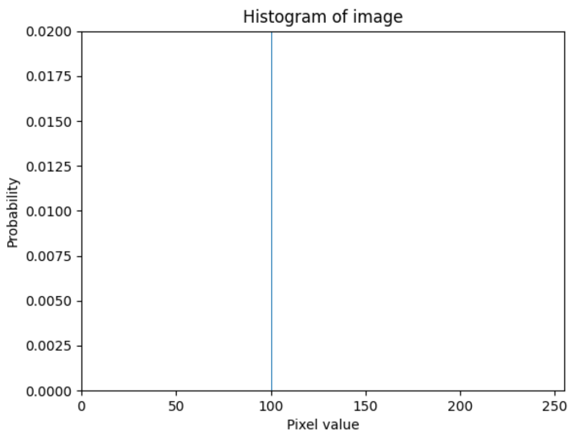
<table>
<tr><th>Fig2.3 Noisy Image f(x,y)</th><th>Fig2.4 Histogram of f(x,y)</th></tr>
<tr><td>

</td><td>


</td></tr>
</table>

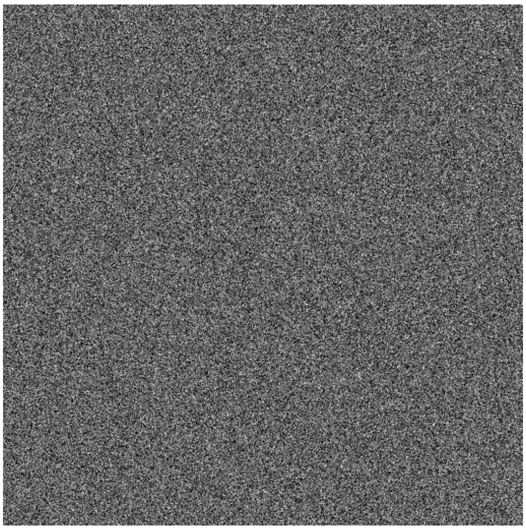
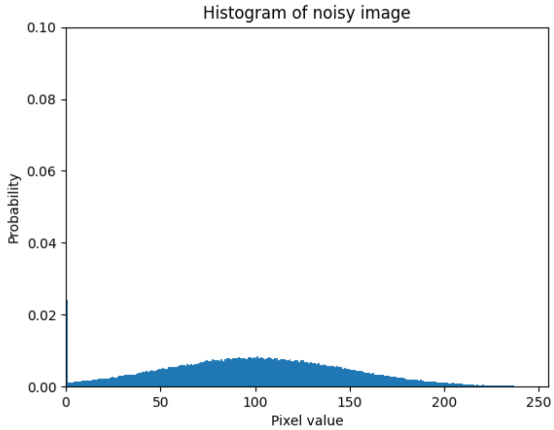
Test Case 3：$\sigma = 50$

Fig3.1 為一灰階值均為 100 的灰色圖片，作為 input image $g(x, y)$，而 Fig3.2 為此 input 的 Histogram。此圖以 $\sigma = 50$ 加上 Gaussiam noise 後如 Fig3.3 所示，可見圖中出現極多雜訊，而從此 Noisy Image 的 Histogram Fig3.4 中也可見各 pixel value 的機率分佈圖趨於平緩，峰值偏低。

<table>
<tr><th>Fig3.1 Input Image g(x,y)</th><th>Fig3.2 Histogram of g(x,y)</th></tr>
<tr><td>

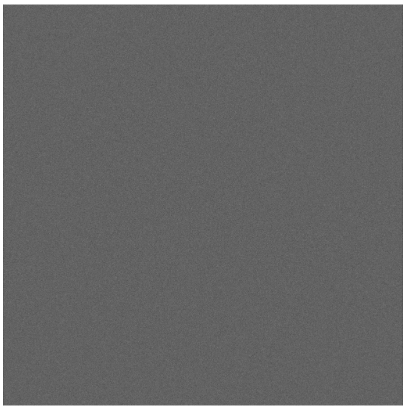
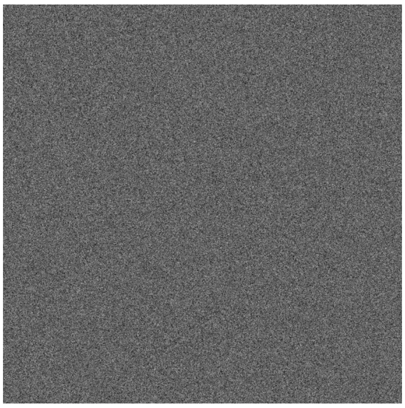
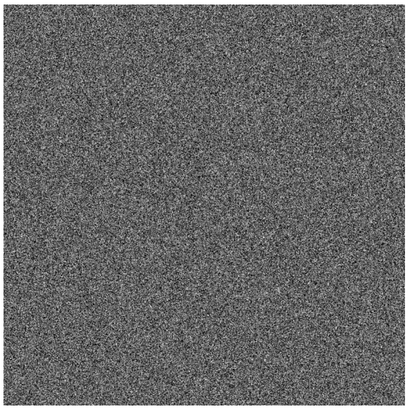
</td><td>


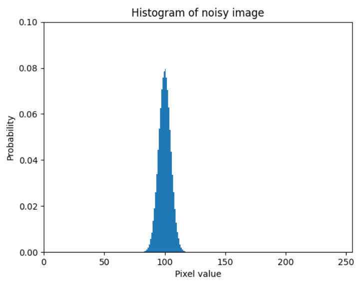
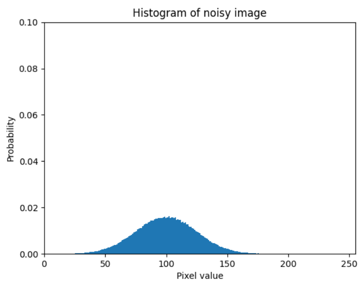
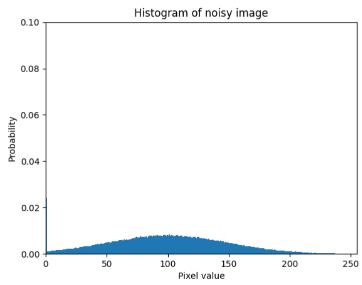
</td></tr>
</table>

Fig3.3 Noisy Image f(x,y)	Fig3.4 Histogram of f(x,y)
	

Comment

在這項作業中，我先使用 cv2 模組來繪製一大小為 `512 x 512`、值均為 `100` 的灰階圖片作為 Input Image，依照題序中步驟產生 Gaussian noise 加於原圖中來生成 Noisy Image，其後我使用 `plt.hist()` 函式來計算並繪製 Input Image 及 Noisy Image 的 Histogram，以利觀察與比較。其後我嘗試調整 σ 值，發現其值越大圖片的雜訊程度越高，從下列 Histogram 也可觀察到，隨著 σ 值增加，Noisy Image 的 Histogram 變得更加分散且峰值下降。

$\sigma = 5$	$\sigma = 25$	$\sigma = 50$
		

$\sigma = 5$	$\sigma = 25$	$\sigma = 50$
		

Source Code

```python
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Create an image g(x,y) whose pixels all have the same gray value of 100.
rows = 512
cols = 512
img = np.zeros((rows, cols), np.uint8)
img.fill(100)
noise = img.copy()

# Generate Gaussian noise n(x,y), with mean 0 and variance 25
mu = 0
sigma = 5

# for each pixel (x,y), (x,y+1) generate a pair of unipform random numbers r and phi in the range [0,1]
for i in range(0, rows):
    for j in range(0, cols-1, 2):
        r = np.random.uniform(0,1)
        phi = np.random.uniform(0,1)

        # compute z1 and z2 using the formula
        z1 = sigma * np.sqrt(-2 * np.log(r)) * np.cos(2 * np.pi * phi)
        z2 = sigma * np.sqrt(-2 * np.log(r)) * np.sin(2 * np.pi * phi)

        # add z1 to the pixel (x,y) and z2 to the pixel (x,y+1)
        noise[i,j] = np.clip(noise[i,j] + z1, 0, 255)
        noise[i,j+1] = np.clip(noise[i,j+1] + z2, 0, 255)
```

```python
# save the image and noise
cv2.imwrite('image.png', img)
cv2.imwrite('noisy_image.png', noise)

# Compute the histogram of the image g(x,y) and the histogram of the noisy
image f(x,y).

plt.figure()
plt.title('Histogram of image')
plt.xlabel('Pixel value')
plt.ylabel('Probability')
plt.hist(img.ravel(), bins=256, range=(0, 256), density=True)
plt.xlim(0, 255)
plt.ylim(0, 0.1)
plt.savefig('image_histogram.png')

plt.figure()
plt.title('Histogram of noisy image')
plt.xlabel('Pixel value')
plt.ylabel('Probability')
plt.hist(noise.ravel(), bins=256, range=(0, 256), density=True)
plt.xlim(0, 255)
plt.ylim(0, 0.1)
plt.savefig('noisy_histogram.png')
```