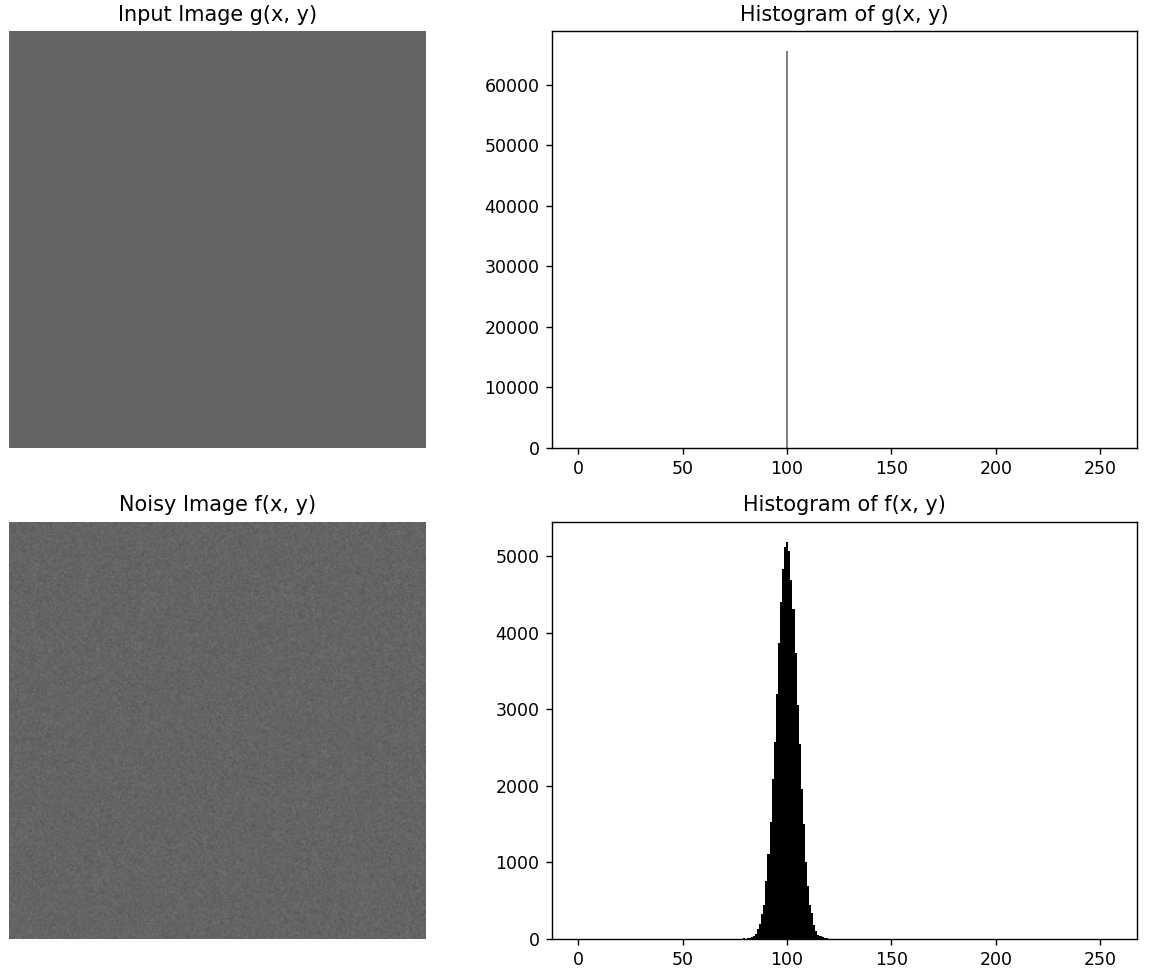
**Comment**

利用 Box-Muller 方法產生高斯雜訊，原本只有一種灰階值的圖加上雜訊後變得比較接近真實影像的樣子。透過 np.clip 把超過範圍的像素限制在 0 到 255，也確保影像不會崩壞。實作後更能理解隨機雜訊在影像處理中的應用。

**Photo**



**Program**

import numpy as np

import matplotlib.pyplot as plt

# Step 1: 建立影像 g(x, y)

height, width = 256, 256

g = np.full((height, width), 100, dtype=np.float32)

# Step 2: 產生高斯雜訊並加入

sigma = 5

G = 256  # 灰階範圍 0~255

# 使用 Box-Muller 方法產生高斯雜訊

f = g.copy()

for x in range(height):

    for y in range(0, width - 1, 2):

        r = np.random.rand()

        phi = np.random.rand()

        z1 = sigma \* np.cos(2 \* np.pi \* phi) \* np.sqrt(-2 \* np.log(r))

        z2 = sigma \* np.sin(2 \* np.pi \* phi) \* np.sqrt(-2 \* np.log(r))

        f[x, y] += z1

        f[x, y + 1] += z2

# 限制像素值在合法範圍 [0, G-1]

f = np.clip(f, 0, G - 1)

# Step 3: 顯示影像與直方圖

fig, axs = plt.subplots(2, 2, figsize=(10, 8))

axs[0, 0].imshow(g, cmap='gray', vmin=0, vmax=255)

axs[0, 0].set\_title('Input Image g(x, y)')

axs[0, 0].axis('off')

axs[0, 1].hist(g.ravel(), bins=256, range=(0, 255), color='gray')

axs[0, 1].set\_title('Histogram of g(x, y)')

axs[1, 0].imshow(f, cmap='gray', vmin=0, vmax=255)

axs[1, 0].set\_title('Noisy Image f(x, y)')

axs[1, 0].axis('off')

axs[1, 1].hist(f.ravel(), bins=256, range=(0, 255), color='black')

axs[1, 1].set\_title('Histogram of f(x, y)')

plt.tight\_layout()

plt.show()

plt.imsave("original\_image.png", g, cmap='gray', vmin=0, vmax=255)

plt.imsave("noisy\_image.png", f, cmap='gray', vmin=0, vmax=255)