40923129L 湯可伊

Homework 7 (Due:)

(1) Create an image $g(x,y)$ whose pixels all have the same gray value of 100. Show the image $g(x,y)$.

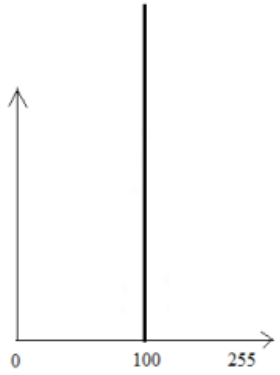
(2) Generate Gaussian noise $n(x,y)$, with $\mu = 0$, $\sigma^2 = 25$, using the algorithm shown in the next page.
Show the noisy image $f(x,y) = g(x,y) + n(x,y)$.

(3) Display the histogram $h(i)$ of $f(x,y)$.

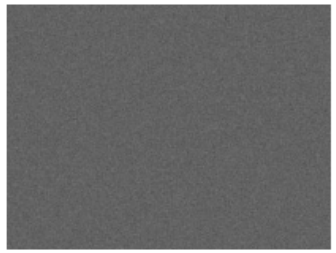
(4) Comment on your results.

Example:

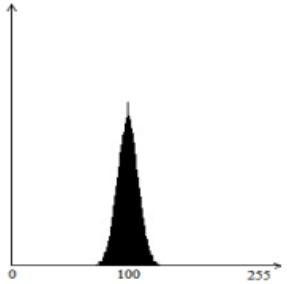
Input image $g(x,y)$ of gray values of 100



Histogram of $g(x,y)$

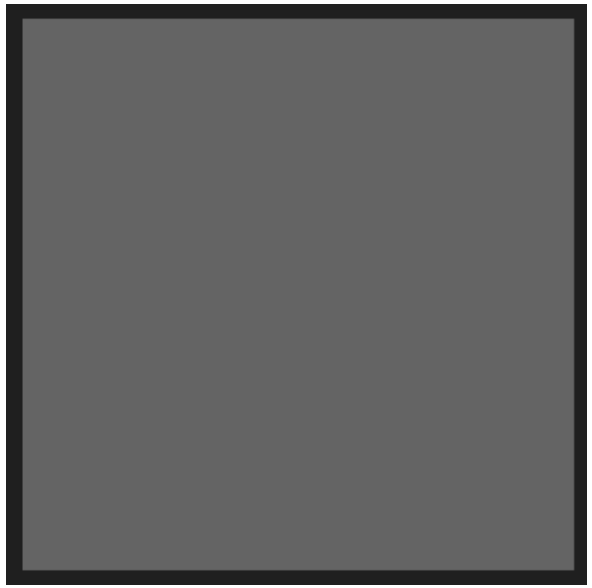


Noisy image $f(x,y)$

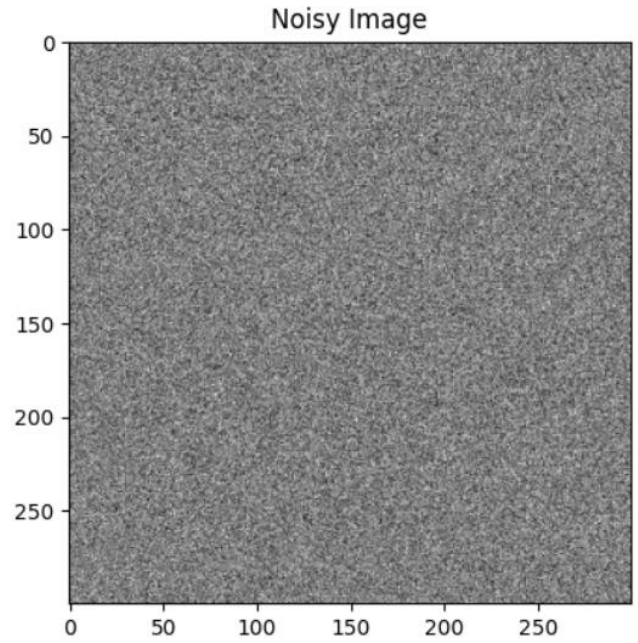


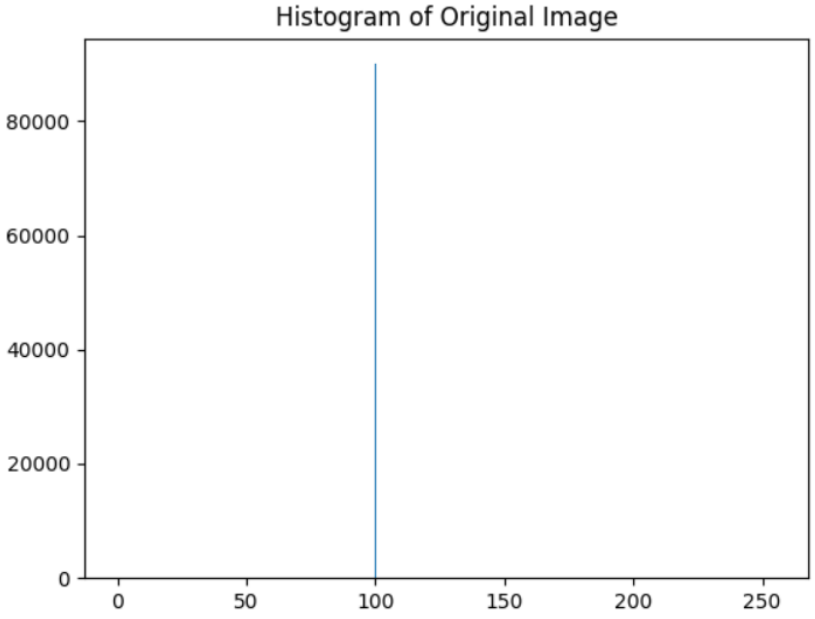
Histogram of $f(x,y)$

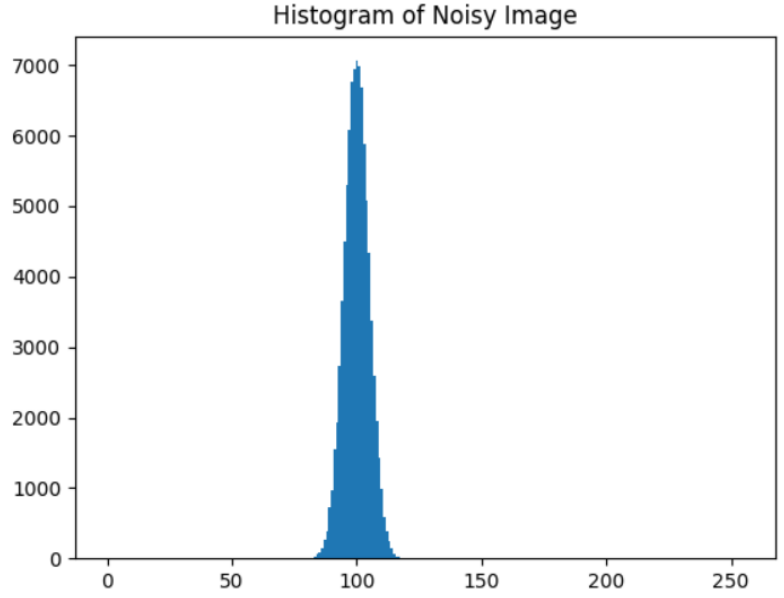
input



output







Source code

```
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt

# Create the input image g(x,y) with a gray value of 100
g = Image.new('L', (300, 300), color=100)
g.show()

# Display the histogram
plt.hist(np.array(g).ravel(), bins=256, range=(0, 255))
plt.title('Histogram of Original Image')
plt.show()

# Generate the Gaussian noise n(x,y) with mu=0 and sigma^2=25
mu, sigma = 0, 5
noise = np.random.normal(mu, sigma, size=(300, 300))

# Add the noise to the input image to create the noisy image f(x,y)
f = np.array(g) + noise

# Display the noisy image f(x,y)
plt.imshow(f, cmap='gray')
plt.title('Noisy Image')
plt.show()

# Display the histogram
plt.hist(f.ravel(), bins=256, range=(0, 255))
plt.title('Histogram of Noisy Image')
plt.show()
```

Comments

Because the noise we added is Gaussian noise, values of the output image is not too far from 100. We can see that the color of these two images still looks similar.

After adding the noise into the original image, the shape of the histogram of the image changes. The histogram of the original image is a line. However, the histogram of the noisy image is separate, like a triangle.

The histogram means the value of the pixel in the image, so the change of the histogram represents the change of the pixel in the image.