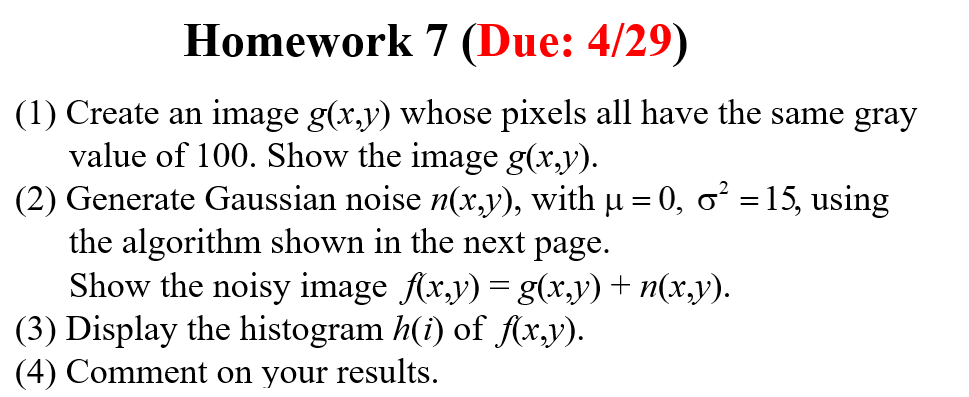
**Problem Statement：**

****

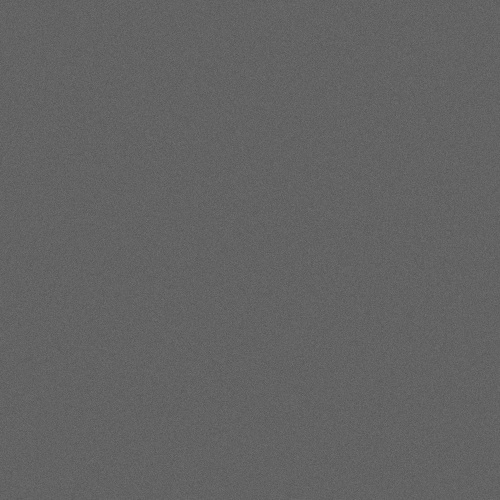
**Input:**

image g(x,y):

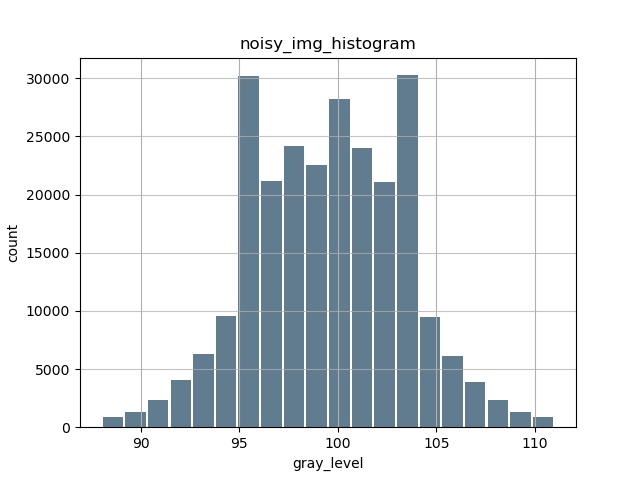


**Output：**

image f(x,y):



**Histogram:**



**Source Code：**

**import cv2**

**import math**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import random**

**def add\_gaussian\_noise(g,sigma=math.sqrt(15)):**

**f=np.array([[100]\*500 for i in range(500)],dtype=np.uint8)**

**f1=np.array([[100]\*500 for i in range(500)],dtype=np.uint8)**

**for x in range(500):**

**for y in range(0,500,2):**

**r=0**

**p=0**

**while(r==0):**

**r=round(random.uniform(0,1),2)**

**while(p==0):**

**p=round(random.uniform(0,1),2)**

**z1=sigma\*math.cos(2\*math.pi\*p)\*math.sqrt(-2\*math.log(r))**

**z2=sigma\*math.sin(2\*math.pi\*p)\*math.sqrt(-2\*math.log(r))**

**f1[x,y]=g[x,y]+z1**

**f1[x,y+1]=g[x,y+1]+z2**

**if f1[x,y]<0:**

**f[x,y]=0**

**elif f1[x,y]>255:**

**f[x,y]=255**

**else:**

**f[x,y]=f1[x,y]**

**if f1[x,y+1]<0:**

**f[x,y+1]=0**

**elif f1[x,y+1]>255:**

**f[x,y+1]=255**

**else:**

**f[x,y+1]=f1[x,y+1]**

**plot\_img('noisy\_img.jpg',np.around(f))**

**plot\_histogram(np.around(f))**

**def plot\_histogram(noisy\_img):**

**noisy\_img=pd.Series(noisy\_img.flatten())**

**noisy\_img.plot.hist(grid=True, bins=20, rwidth=0.9,**

**color='#607c8e')**

**plt.title('noisy\_img\_histogram')**

**plt.xlabel('gray\_level')**

**plt.ylabel('count')**

**plt.grid(axis='y', alpha=0.75)**

**plt.savefig("noisy\_img\_histogram.jpg")**

**plt.show()**

**def plot\_img(filename,img):**

**cv2.imshow(filename,img)**

**cv2.waitKey(0)**

**cv2.imwrite(filename,img)**

**img=np.array([[100]\*500 for i in range(500)],dtype=np.uint8)**

**plot\_img("original.jpg",img)**

**add\_gaussian\_noise(img)**

**Comments：**

**由上圖質方圖之結果可見,加入gaussian noise後之Histogram呈現常態分佈,原因是gaussian noise本身就呈現常態分佈,但不太明白為什麼在與100約距離一個標準差的位置其個數特別多,希望能請教助教。這次的作業主要挑戰在於要理解如何產生gaussian noise 並用程式碼實現,由於python的random函式都會包括0,但log內不能是0,所以之前一直報錯但不曉得問題在哪裡,後來弄了很久才發現這個問題,整體而言並不算太難。**