

▼ Aditi Rao 118A2088 - EXTC B

Q2

```
import matplotlib.pyplot as plt
import numpy as np
import cv2
from google.colab import files
uploaded = files.upload()
```

Choose Files Lc.jpeg
• **Lc.jpeg**(image/jpeg) - 15212 bytes, last modified: 6/25/2021 - 100% done
Saving Lc.jpeg to Lc.jpeg

```
img = cv2.imread("Lc.jpeg")
from google.colab.patches import cv2_imshow
cv2_imshow(img)
```

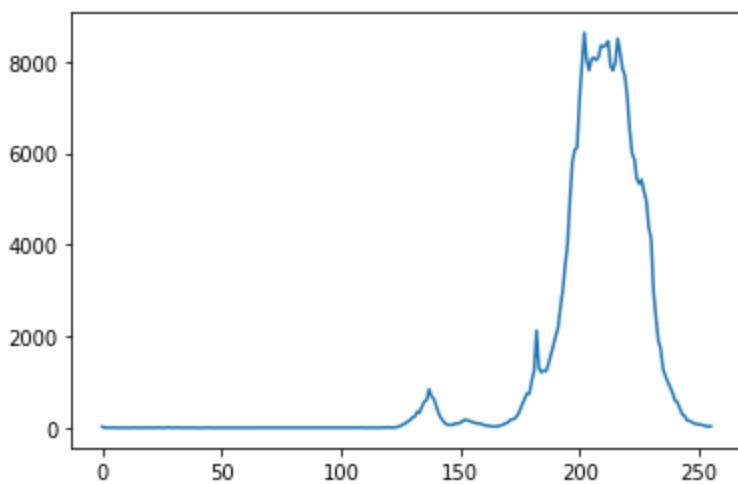
□



```
height = img.shape[0]
width = img.shape[1]
max_intensity = 255
img_gs = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2_imshow(img_gs)
plt.hist(img_gs.ravel(), 256, [0, 255])
plt.show()
```



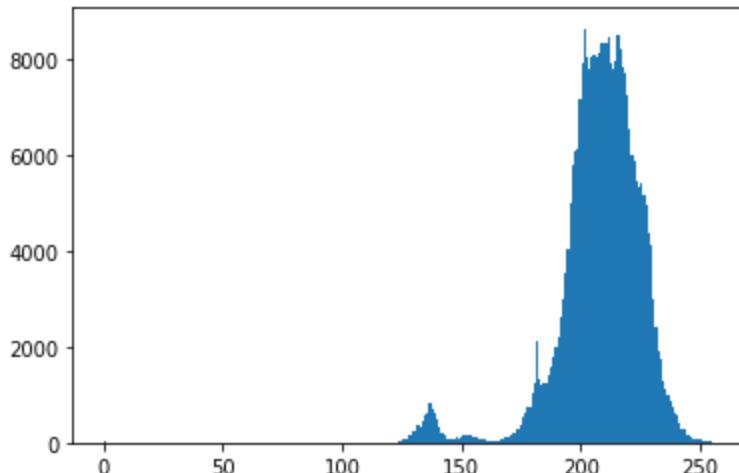
```
histg = cv2.calcHist([img_gs],[0],None,[256],[0,256])
cum_hist =histg.copy()
plt.plot(histg)
plt.show()
```



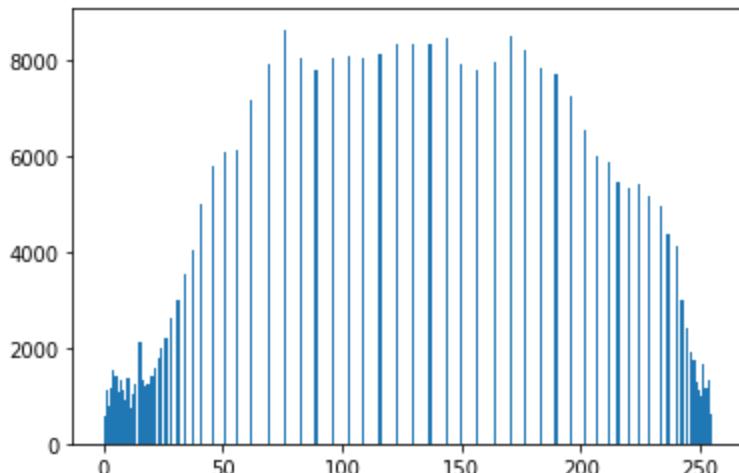
```
equ =cv2.equalizeHist(img_gs)
res = np.hstack((img_gs,equ))
res1 =np.hstack((res,img_gs))
cv2_imshow(res)
cv2_imshow(res1)
```



```
plt.hist(img_gs.ravel(),256,[0,255])
plt.show()
plt.hist(equ.ravel(),256,[0,255])
plt.show
```



<function matplotlib.pyplot.show>



Q3

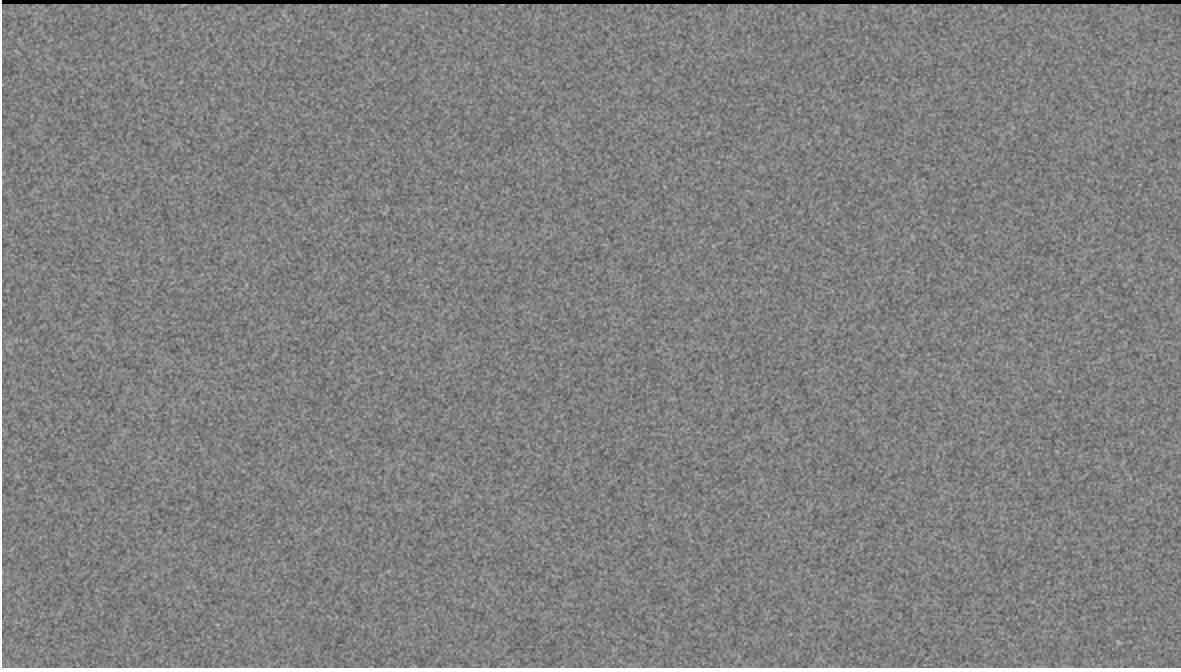
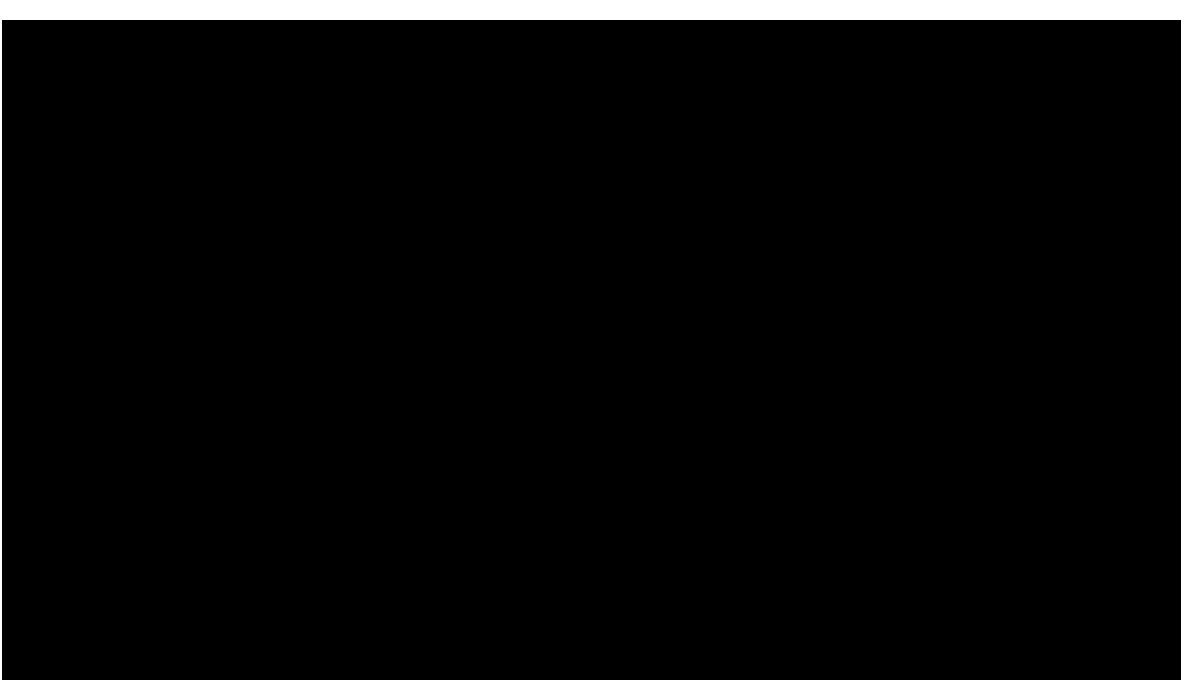
```
from google.colab import files
uploaded = files.upload()
```

Choose Files Color.jpeg
 • **Color.jpeg**(image/jpeg) - 28287 bytes, last modified: 6/25/2021 - 100% done
 Saving Color.jpeg to Color.jpeg

```
image = cv2.imread('Color.jpeg',cv2.IMREAD_GRAYSCALE)
from google.colab.patches import cv2_imshow
cv2_imshow(image)
```

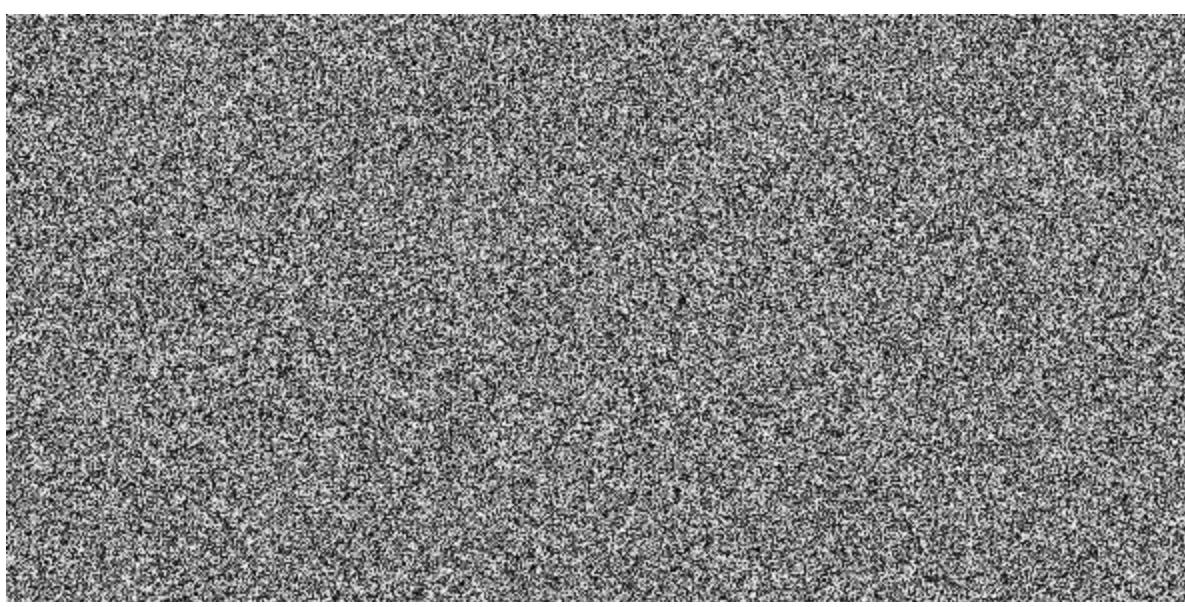


```
gaussian_noise = np.zeros((image.shape[0],image.shape[1]),dtype=np.uint8)
cv2_imshow(gaussian_noise)
cv2.randn(gaussian_noise,128,20)
cv2_imshow(gaussian_noise)
cv2.imwrite("Gaussian random noise.PNG",gaussian_noise)
```



True

```
uniform_noise =np.zeros((image.shape[0],image.shape[1]),dtype=np.uint8)
cv2.randu(uniform_noise,0,255)
cv2_imshow(uniform_noise)
cv2.imwrite("Uniform random noise.PNG",uniform_noise)
```

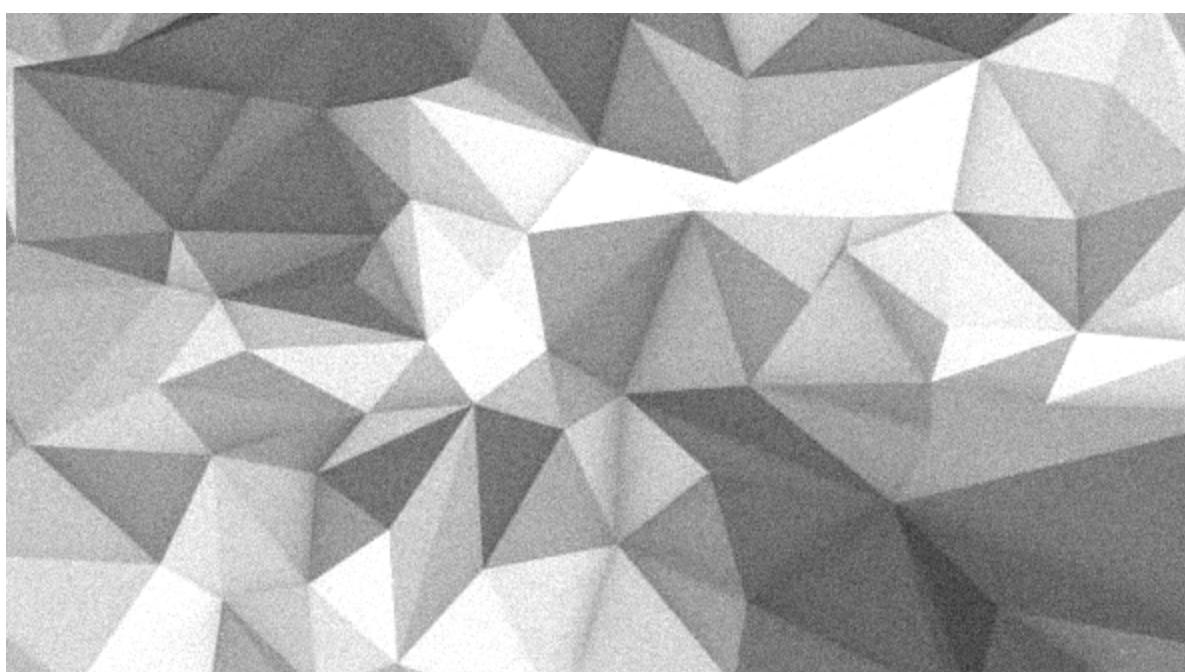


```
impulse_noise = uniform_noise.copy()
ret,impulse_noise = cv2.threshold(uniform_noise,250,255,cv2.THRESH_BINARY)
cv2.imshow(impulse_noise)
cv2.imwrite("Impulse noise.PNG",impulse_noise)
```



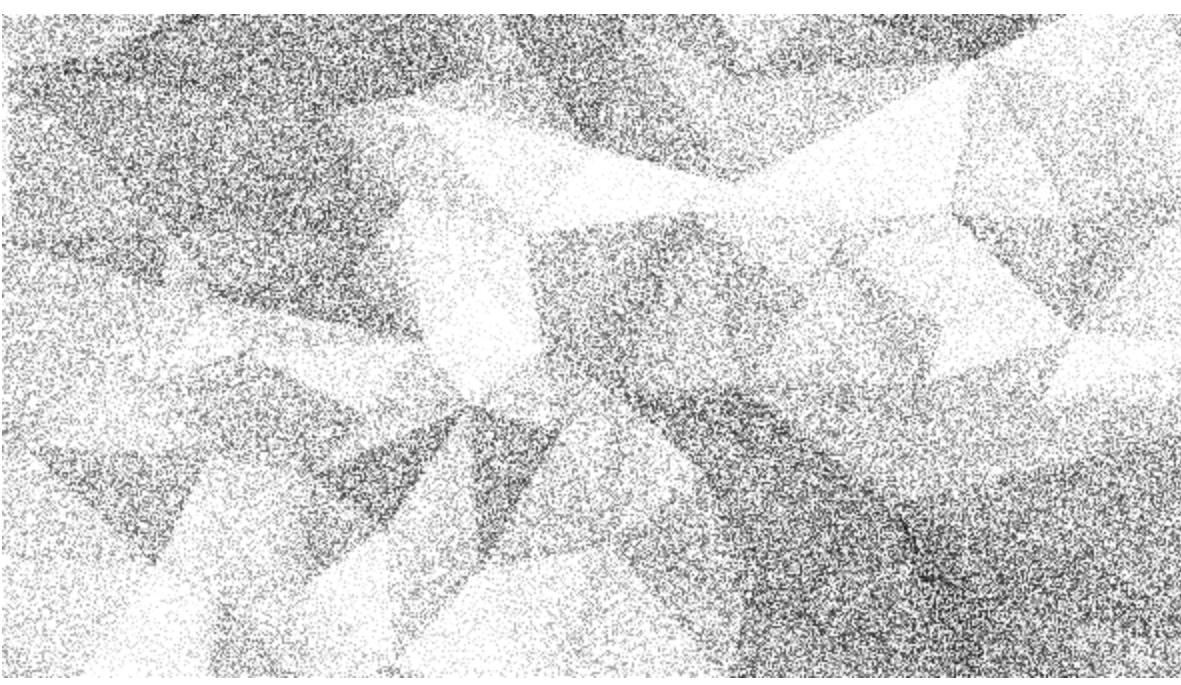
True

```
gaussian_noise = (gaussian_noise*0.5).astype(np.uint8)
noisy_image1 =cv2.add(image,gaussian_noise)
cv2.imshow(noisy_image1)
cv2.imwrite("noisy image 1.PNG",noisy_image1)
```



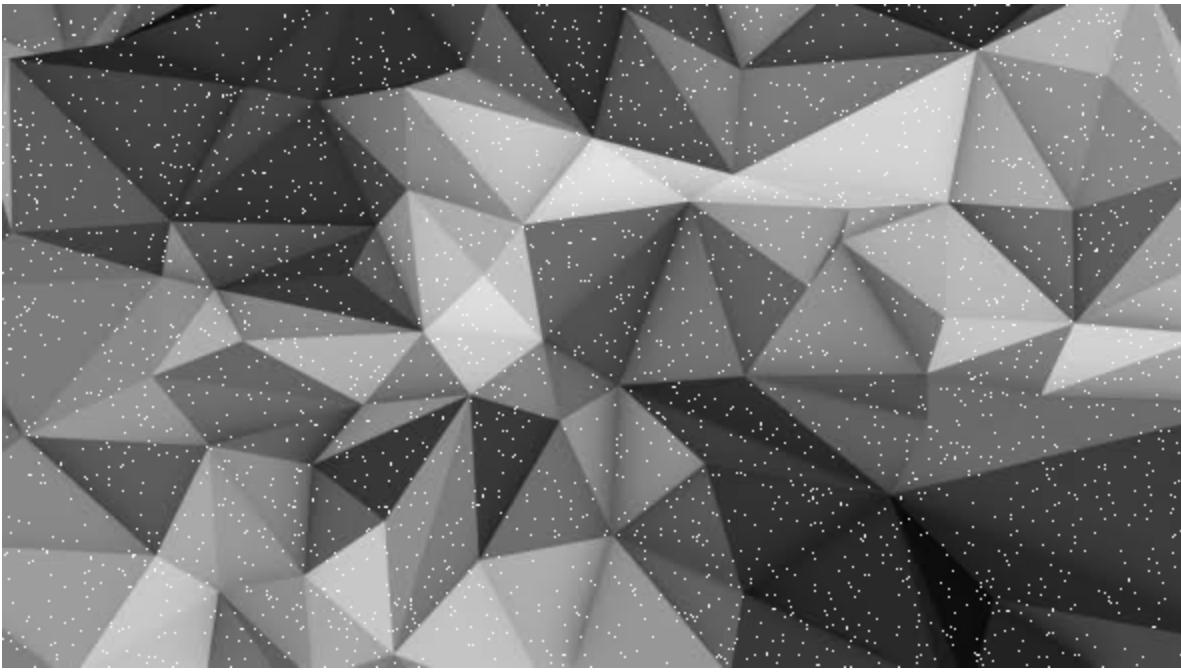
True

```
noisy_image2 =cv2.add(image,uniform_noise)
cv2.imshow(noisy_image2)
cv2.imwrite("noisy image 2.PNG",noisy_image2)
```



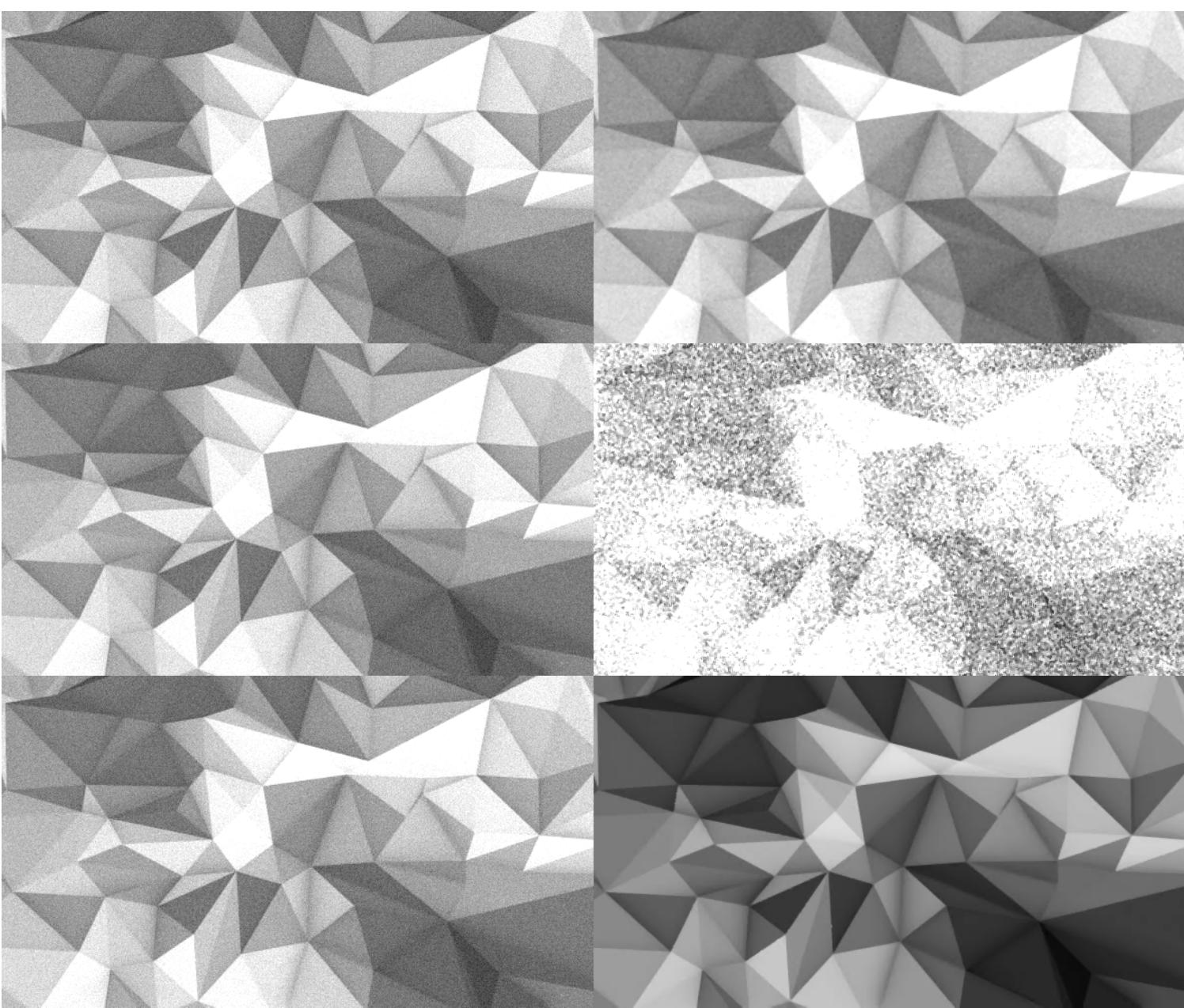
True

```
impulse_noise = (impulse_noise).astype(np.uint8)
noisy_image3 =cv2.add(image,impulse_noise)
cv2_imshow(noisy_image3)
cv2.imwrite("noisy image 3.PNG",noisy_image3)
```



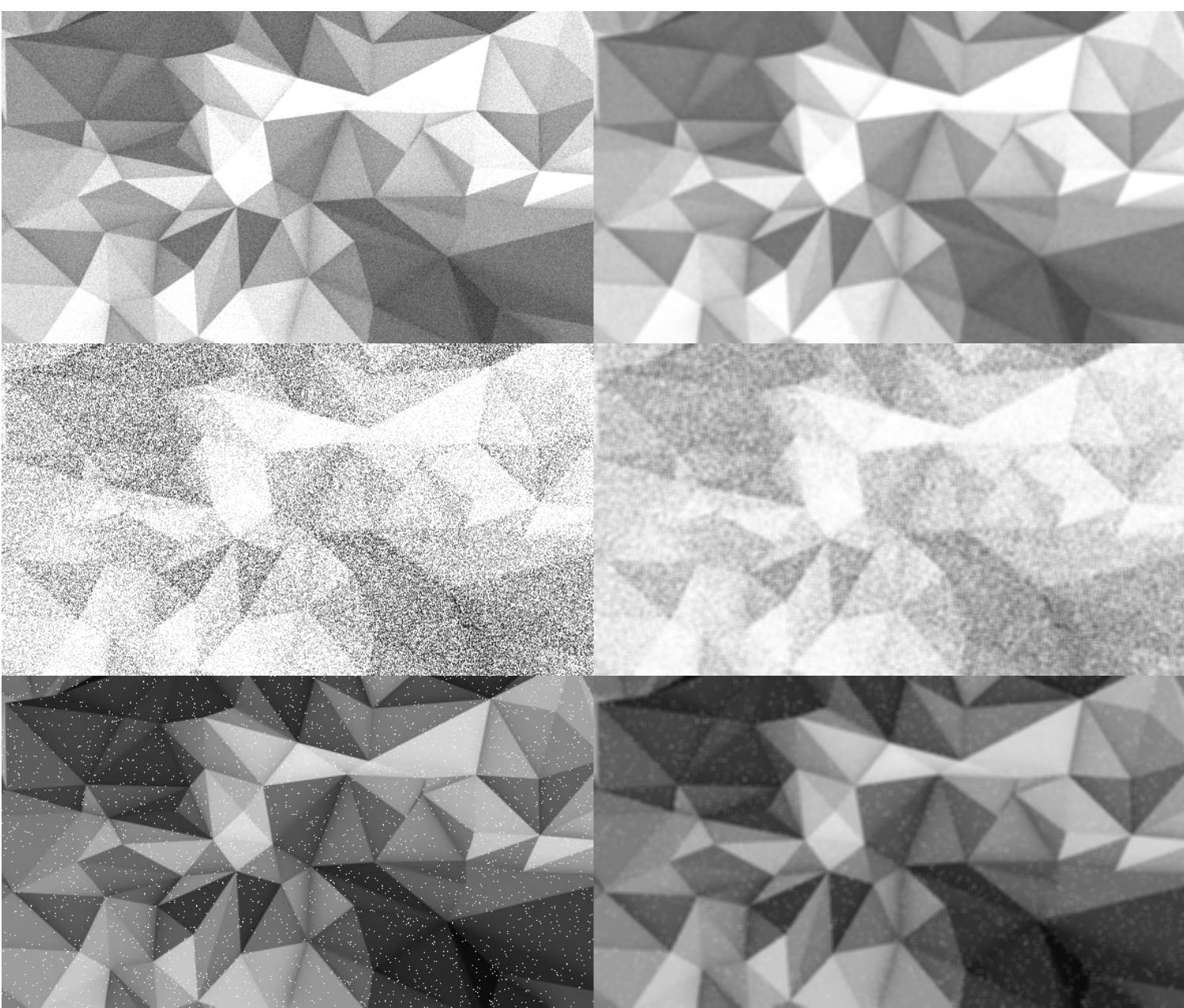
True

```
blurred1 = cv2.medianBlur(noisy_image1,3)
st1=np.hstack((noisy_image1,blurred1))
cv2_imshow(st1)
cv2.imwrite("median filter -Gaussian noise.PNG",blurred1)
blurred2 = cv2.medianBlur(noisy_image2,3)
st2=np.hstack((noisy_image1,blurred2))
cv2_imshow(st2)
cv2.imwrite("median filter -uniform noise.PNG",blurred2)
blurred3 = cv2.medianBlur(noisy_image3,3)
st3=np.hstack((noisy_image1,blurred3))
cv2_imshow(st3)
cv2.imwrite("median filter -impulse noise.PNG",blurred3)
```

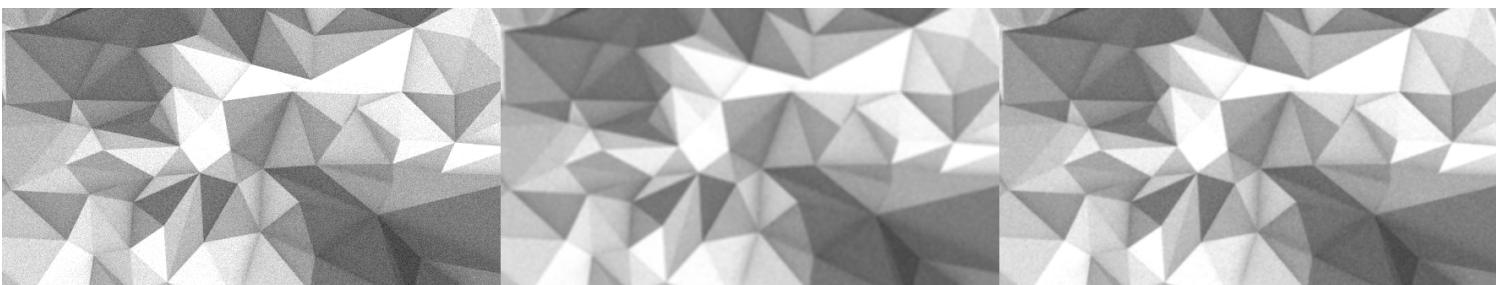


True

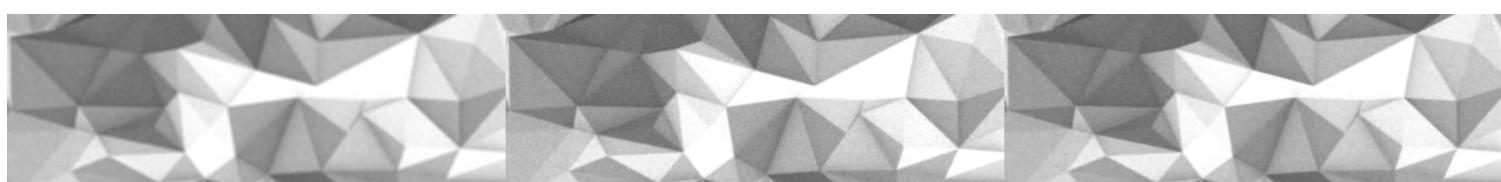
```
processed_image1 = cv2.blur(noisy_image1,(5,5))
st1 = np.hstack((noisy_image1,processed_image1))
cv2_imshow(st1)
processed_image2 = cv2.blur(noisy_image2,(5,5))
st2 = np.hstack((noisy_image2,processed_image2))
cv2_imshow(st2)
processed_image3 = cv2.blur(noisy_image3,(5,5))
st3 = np.hstack((noisy_image3,processed_image3))
cv2_imshow(st3)
```



```
image_gaussian_processed = cv2.GaussianBlur(noisy_image1,(5,5),1)
st1=np.hstack((noisy_image1,processed_image1,image_gaussian_processed))
cv2_imshow(st1)
```

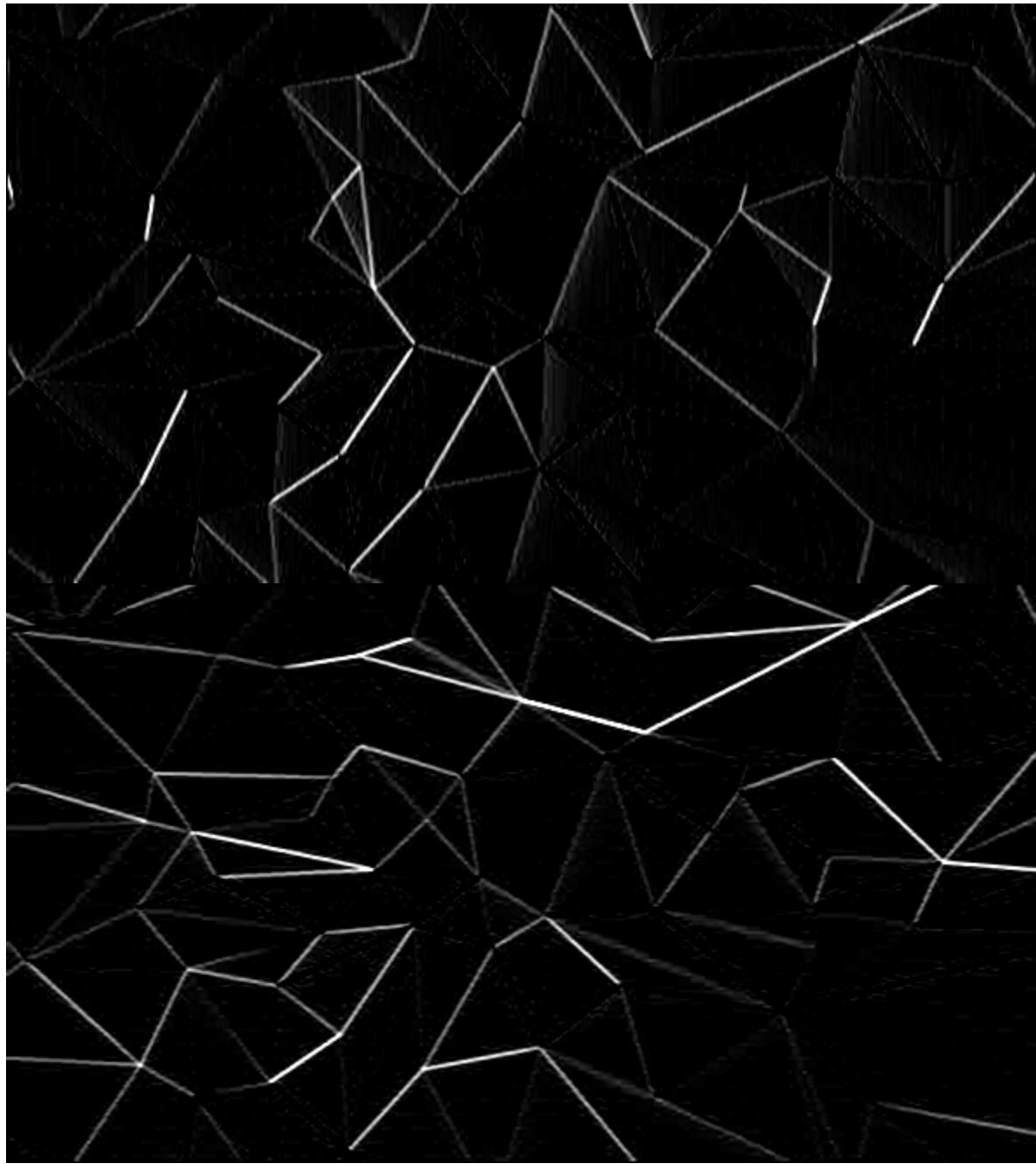


```
bilateral = cv2.bilateralFilter(noisy_image1,5,75,75)
st1 = np.hstack((processed_image1,image_gaussian_processed,bilateral))
cv2_imshow(st1)
cv2.imwrite('Color.jpeg',bilateral)
```



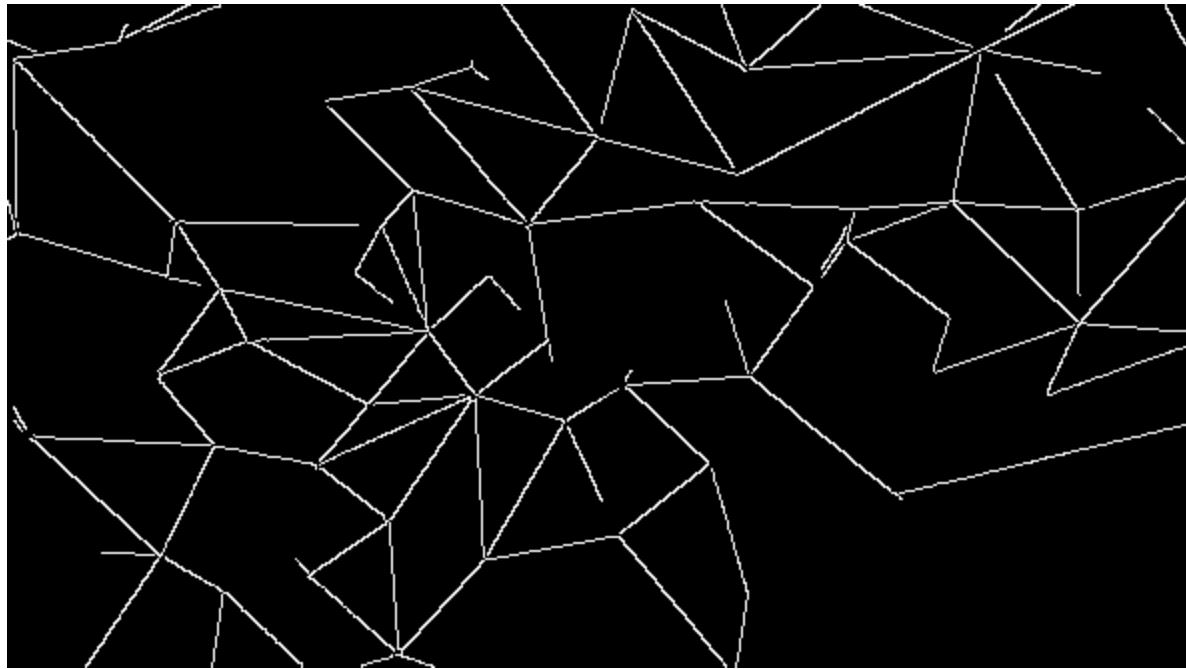
Q4

```
sobelx =cv2.Sobel(image,cv2.CV_64F,1,0)
cv2_imshow(sobelx)
sobely = cv2.Sobel(image,cv2.CV_64F,0,1)
cv2_imshow(sobely)
```



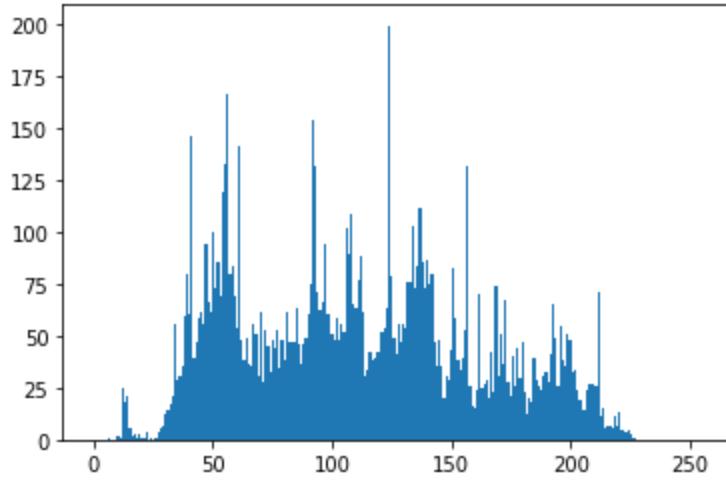
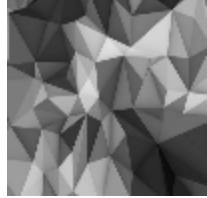
```
laplacian = cv2.Laplacian(image,cv2.CV_64F,ksize=5)
cv2_imshow(laplacian)
```

```
canny = cv2.Canny(image,120,170)
cv2_imshow(canny)
```



Q1

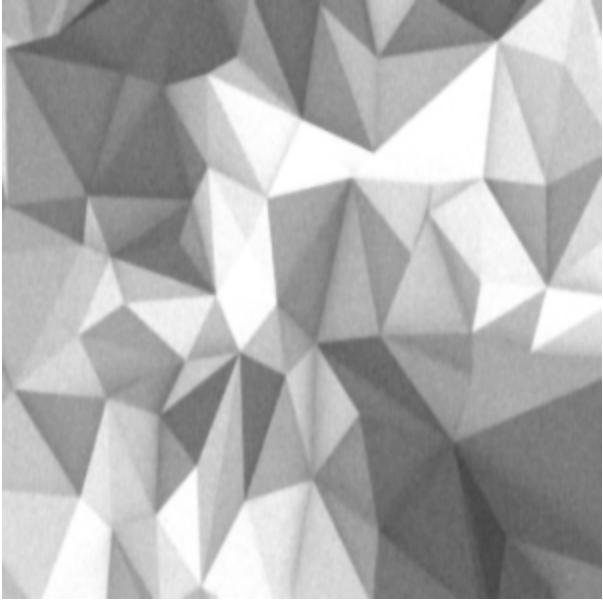
```
height = image.shape[0]
width = image.shape[1]
max_intensity=255
img_re=cv2.resize(image,(100,100))
cv2_imshow(img_re)
plt.hist(img_re.ravel(),256,[0,255])
plt.show()
```



Q5

```
image = cv2.imread ("Color.jpeg",0)
kernel = np.ones((5,5),np.uint8)
max_intensity =255
output =image.copy()
for i in np.arange(height):
    for j in np.arange(width):
        a=output.item(i,j)
        b=max_intensity -a
        output.itemset((i,j),b)

r,imgbw = cv2.threshold(output,24,255,cv2.THRESH_BINARY)
imgbw=cv2.resize(image,(300,300))
cv2_imshow(imgbw)
```



```
imgE = cv2.erode (imgbw,kernel,iterations=2)
imgD = cv2.dilate(imgE,kernel,iterations=2)
from google.colab.patches import cv2_imshow
cv2_imshow(imgbw)
cv2_imshow(imgE)
cv2_imshow(imgD)
```



Q6



```
imgD = cv2.dilate(imgbw,kernel,iterations=7)
from google.colab.patches import cv2_imshow
cv2_imshow(imgbw)
cv2_imshow(imgD)
```

