Code:

```
# %%
import cv2
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from matplotlib import pyplot as plt
from matplotlib.gridspec import GridSpec
import numpy as np

filepath = 'jawad.jpg'

# load image
imageObj = cv2.imread(filepath)
img=cv2.cvtColor(imageObj,cv2.COLOR_BGR2RGB)
img=cv2.cvtColor(imageObj,cv2.COLOR_RGB2GRAY)

hist = cv2.calcHist([img], [0], None, [256], [0,256])
img_eq = cv2.equalizeHist(img)
hist_eq = cv2.calcHist([img_eq], [0], None, [256], [0,256])


row = 3
col =3


#plotting image with the histogram
fig = plt.figure(figsize=(30,30))
gs = GridSpec(row, col)
fig.add_subplot(gs[0,0])
plt.imshow(img, cmap='gray')
fig.add_subplot(gs[0,1])
plt.plot(hist)


# %%
#plotting EQUALIZED images with the Equalized Histogram
fig = plt.figure(figsize=(30,30))
gs = GridSpec(row, col)
fig.add_subplot(gs[0,0])
plt.imshow(img_eq, cmap='gray')
fig.add_subplot(gs[0,1])
plt.plot(hist_eq)

# %%
#adaptive image eqalization with adaptive histogram
cl = cv2.createCLAHE(2.0, (8,8))
```

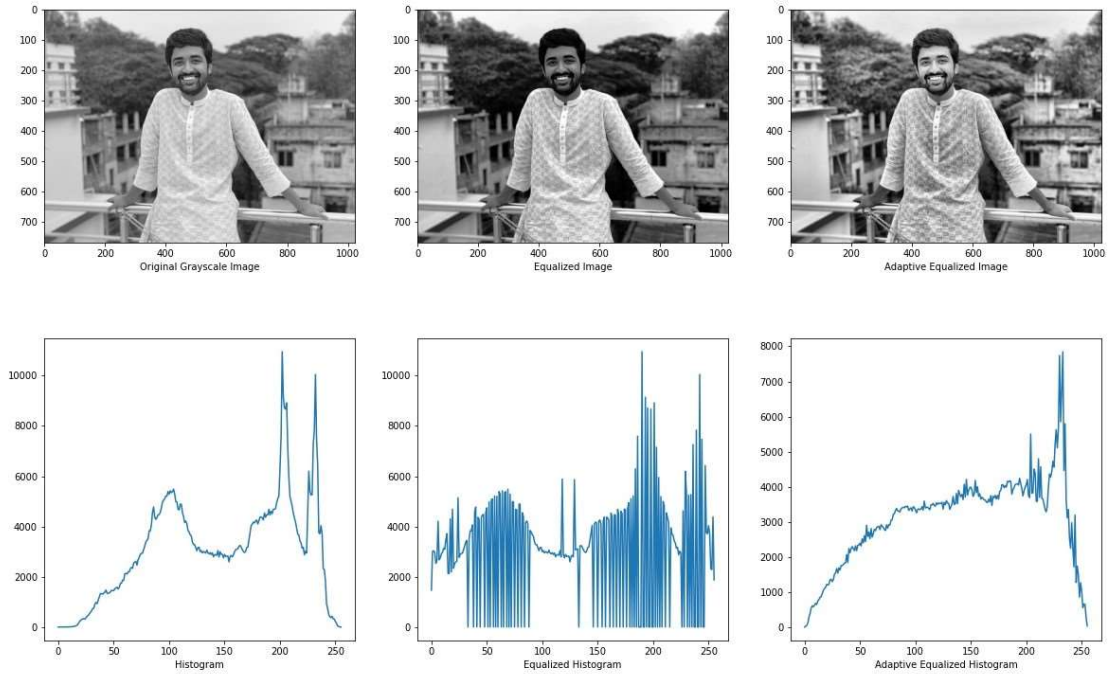
```
img_ad_eq = cl.apply(img)
hist_ad_eq = cv2.calcHist([img_ad_eq], [0], None, [256], [0,256])

fig = plt.figure(figsize=(30,30))
gs = GridSpec(row, col)
fig.add_subplot(gs[0,0])
plt.imshow(img_eq, cmap='gray')
fig.add_subplot(gs[0,1])
plt.plot(hist_ad_eq)

# %%
fig = plt.figure(figsize=(20,20))
gs = GridSpec(row, col)

fig.add_subplot(gs[0,0])
plt.xlabel('Original Grayscale Image')
plt.imshow(img, cmap='gray')

fig.add_subplot(gs[0,1])
plt.xlabel('Equalized Image')
plt.imshow(img_eq, cmap='gray')

fig.add_subplot(gs[0,2])
plt.xlabel('Adaptive Equalized Image')
plt.imshow(img_ad_eq, cmap='gray')

fig.add_subplot(gs[1,0])
plt.xlabel('Histogram')
plt.plot(hist)

fig.add_subplot(gs[1,1])
plt.xlabel('Equalized Histogram')
plt.plot(hist_eq)

fig.add_subplot(gs[1,2])
plt.xlabel('Adaptive Equalized Histogram')
plt.plot(hist_ad_eq)

plt.savefig('1.jpg')

# %%
#gaussian noise
#salt-pepper noise
#random/uniform noise


#gaussian noise simulation

gu_n = np.zeros((img.shape[0], img.shape[1]), dtype = np.uint8)
```

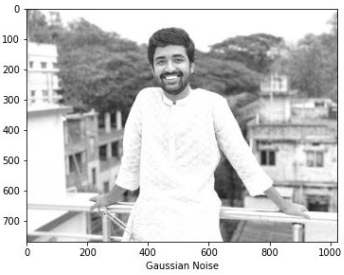
```
cv2.randn(gu_n, 128, 20)
gu_n = (gu_n*0.5).astype(np.uint8)
gu_img= cv2.add(img, gu_n)


fig = plt.figure(figsize=(40,40))
gs = GridSpec(row, col)
fig.add_subplot(gs[0,0])
plt.xlabel('Noise Matrix')
plt.imshow(gu_n, cmap='gray')

fig.add_subplot(gs[0,1])
plt.xlabel('Image with gaussian Noise')
plt.imshow(gu_img, cmap='gray')

# %%
#uniform noise simulation

rand_n = np.zeros((img.shape[0], img.shape[1]), dtype = np.uint8)
cv2.randu(rand_n, 0, 255)
print(rand_n)

rand_n = (rand_n*0.30).astype(np.uint8)
print(rand_n)

rand_n_img= cv2.add(img, rand_n)
print(rand_n_img)

fig = plt.figure(figsize=(40,40))
gs = GridSpec(row, col)
fig.add_subplot(gs[0,0])
plt.xlabel('Random Noise Matrix')
plt.imshow(rand_n, cmap='gray')

fig.add_subplot(gs[0,1])
plt.xlabel('Image with Random Noise')
plt.imshow(rand_n_img, cmap='gray')

# %%
#salt pepper noise simulation

im_n = rand_n.copy()
ret, im_n = cv2.threshold(rand_n, 10, 100, cv2.THRESH_BINARY)
im_n = (im_n*0.8).astype(np.uint8)
im_img = cv2.add(img, im_n)

fig = plt.figure(figsize=(40,40))
gs = GridSpec(row, col)
fig.add_subplot(gs[0,0])
```

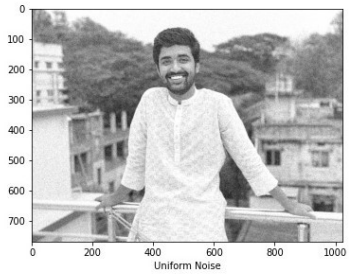
```
plt.xlabel('Salt-Pepper Noise Matrix')
plt.imshow(im_n, cmap='gray')

fig.add_subplot(gs[0,1])
plt.xlabel('Image with Salt-Pepper Noise')
plt.imshow(im_img, cmap='gray')

# %%
fig = plt.figure(figsize=(20,20))
gs = GridSpec(row, col)

fig.add_subplot(gs[0,0])
plt.xlabel('Gaussian Noise')
plt.imshow(gu_img, cmap='gray')

fig.add_subplot(gs[0,1])
plt.xlabel('Uniform Noise')
plt.imshow(rand_n_img, cmap='gray')

fig.add_subplot(gs[0,2])
plt.xlabel('Salt-Pepper Noise')
plt.imshow(im_img, cmap='gray')

fig.add_subplot(gs[1,0])
plt.xlabel('Gaussian Noise Matrix')
plt.imshow(gu_n, cmap='gray')

fig.add_subplot(gs[1,1])
plt.xlabel('Uniform/Random Noise Matrix')
plt.imshow(rand_n, cmap='gray')

fig.add_subplot(gs[1,2])
plt.xlabel('Salt-Pepper Noise Matrix')
plt.imshow(im_n, cmap='gray')

plt.savefig('2.jpg')

# %%
#IMAGE SMOOTHING WITH GAUSSIAN BLUR AND MEDIAN BLUR

gu_sm = cv2.GaussianBlur(gu_img, (3,3), 5)
im_sm = cv2.medianBlur(im_img,5)

fig = plt.figure(figsize=(20,20))
gs = GridSpec(row, col)

fig.add_subplot(gs[0,0])
plt.xlabel('Gaussian noise added image')
```

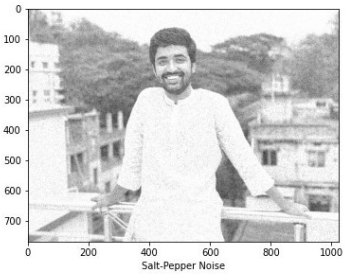
```
plt.imshow(gu_img, cmap='gray')

fig.add_subplot(gs[0,1])
plt.xlabel('Salt-pepper noise added image')
plt.imshow(im_img, cmap='gray')

fig.add_subplot(gs[1,0])
plt.xlabel('Gaussian blurred image')
plt.imshow(gu_sm, cmap='gray')

fig.add_subplot(gs[1,1])
plt.xlabel('Median blurred image')
plt.imshow(im_sm, cmap='gray')

plt.savefig('3.jpg')
```

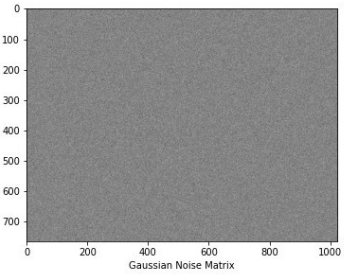
Output:

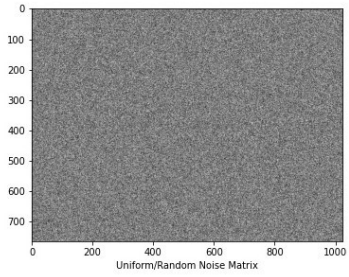


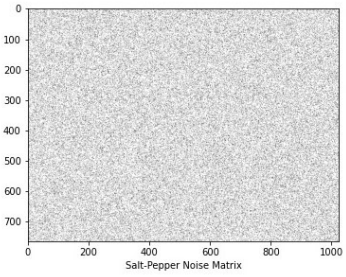

Gaussian Noise

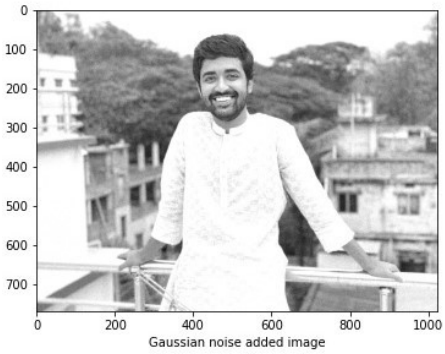

Uniform Noise

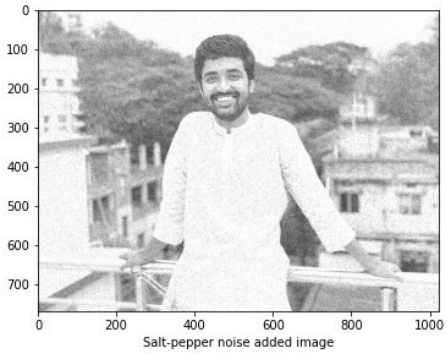

Salt-Pepper Noise

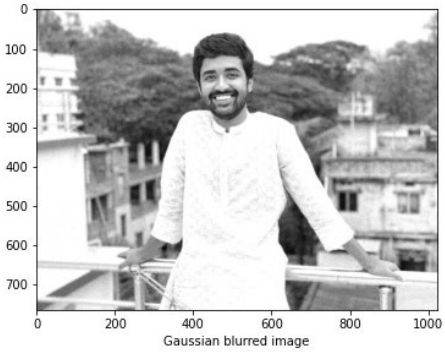

Gaussian Noise Matrix

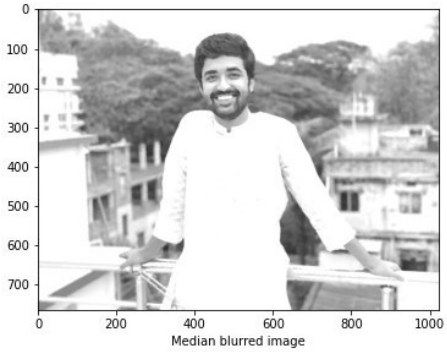

Uniform/Random Noise Matrix


Salt-Pepper Noise Matrix


Gaussian noise added image


Salt-pepper noise added image


Gaussian blurred image


Median blurred image