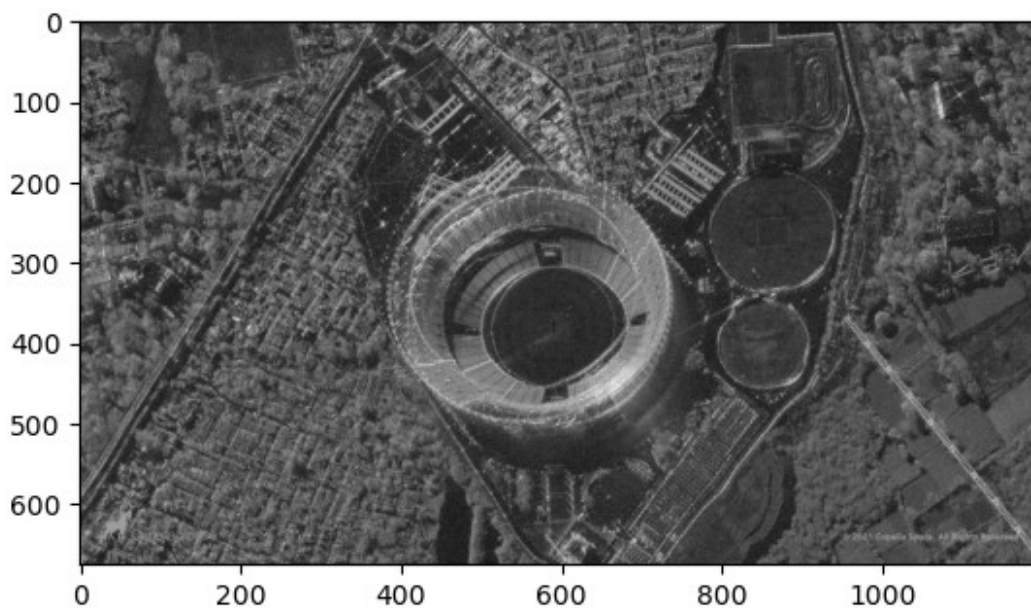


```
# Зашумить изображение при помощи шума гаусса, постоянного шума.
```

```
import numpy as np
import cv2
import matplotlib.pyplot as plt

image1 = cv2.imread('sar_1.jpg')
image1_gray = cv2.cvtColor(image1, cv2.COLOR_BGR2GRAY)
plt.imshow(image1_gray, cmap = "gray")
<matplotlib.image.AxesImage at 0x797acc8>
```



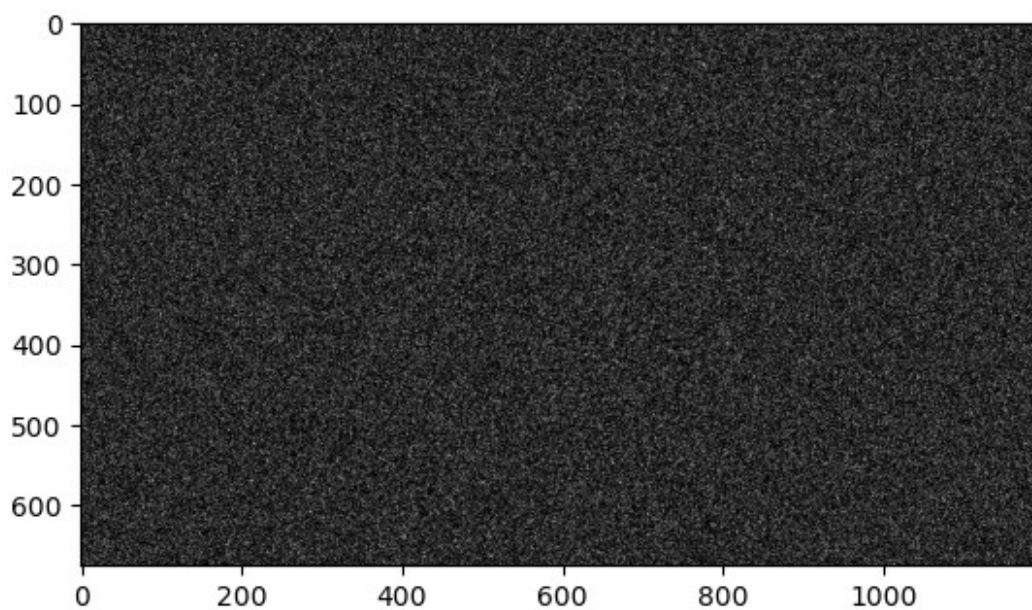
```
# Шум Гаусса
```

```
mean = 0
stdev = 110
noise_gauss = np.zeros(image1_gray.shape, np.uint8)
cv2.randn(noise_gauss, mean, stdev)

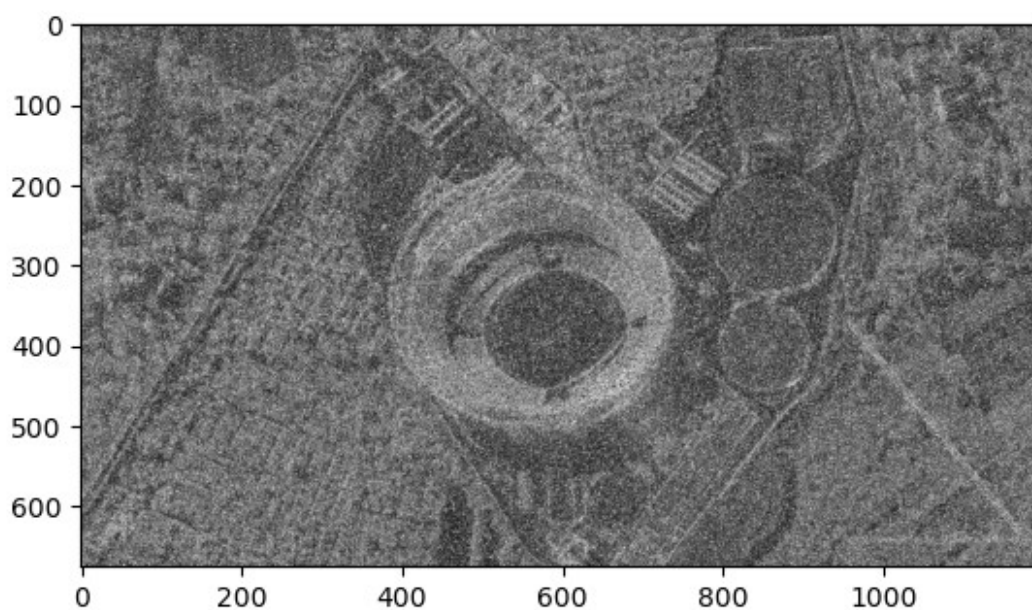
array([[ 0, 17,  0, ..., 122,  0, 88],
       [ 1, 114, 75, ...,  0, 114,  0],
       [ 0,  0, 57, ...,  0,  0, 160],
       ...,
       [ 0,  0, 43, ..., 255,  0, 16],
       [ 0, 37,  0, ...,  0,  6, 148],
       [176, 224,  0, ..., 23,  0, 120]], dtype=uint8)

plt.imshow(noise_gauss, cmap="gray")
```

```
<matplotlib.image.AxesImage at 0x7b0c308>
```



```
image_noise_gauss = image1_gray + noise_gauss  
plt.imshow(image_noise_gauss, cmap="gray")  
<matplotlib.image.AxesImage at 0x76969a0>
```



```
# Постоянный шум
```

```

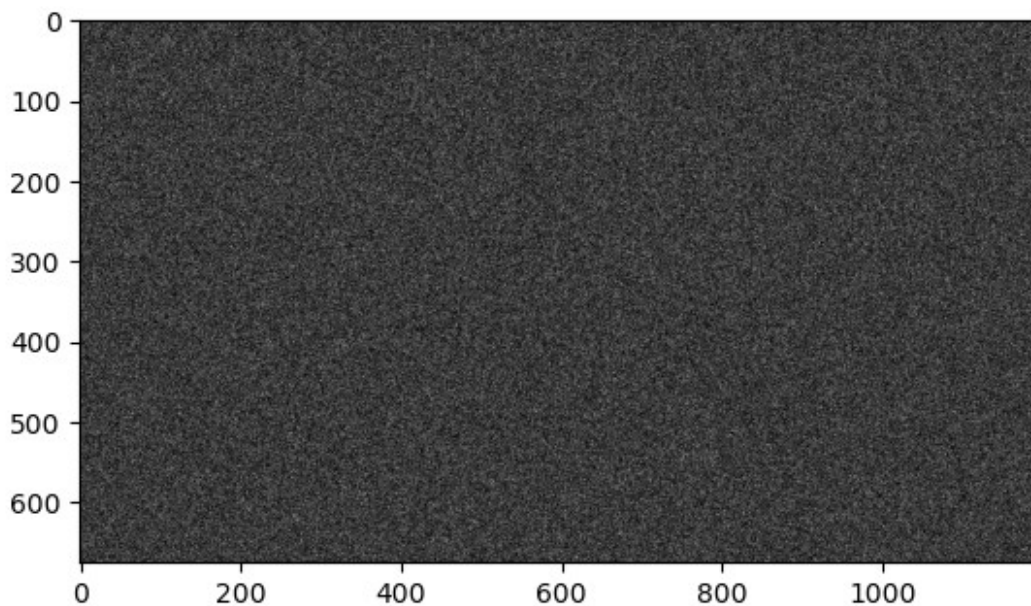
a = 110
b = 0
mid = (a+b)/2

mean1 = mid
stdev1 = a-mid
noise_constant = np.zeros(image1_gray.shape, np.uint8)
cv2.randn(noise_constant, mean1, stdev1)

array([[ 29, 101,  98, ...,  0, 202, 127],
       [ 89,  70,  67, ..., 113,  59, 116],
       [ 70,  33,  86, ..., 125, 135,  44],
       ...,
       [  0,  38, 130, ...,  83, 119, 128],
       [ 78,  64,  73, ...,  80,  53, 100],
       [134, 142,  15, ...,  69,  32, 148]], dtype=uint8)

plt.imshow(noise_constant, cmap="gray")
<matplotlib.image.AxesImage at 0x789d230>

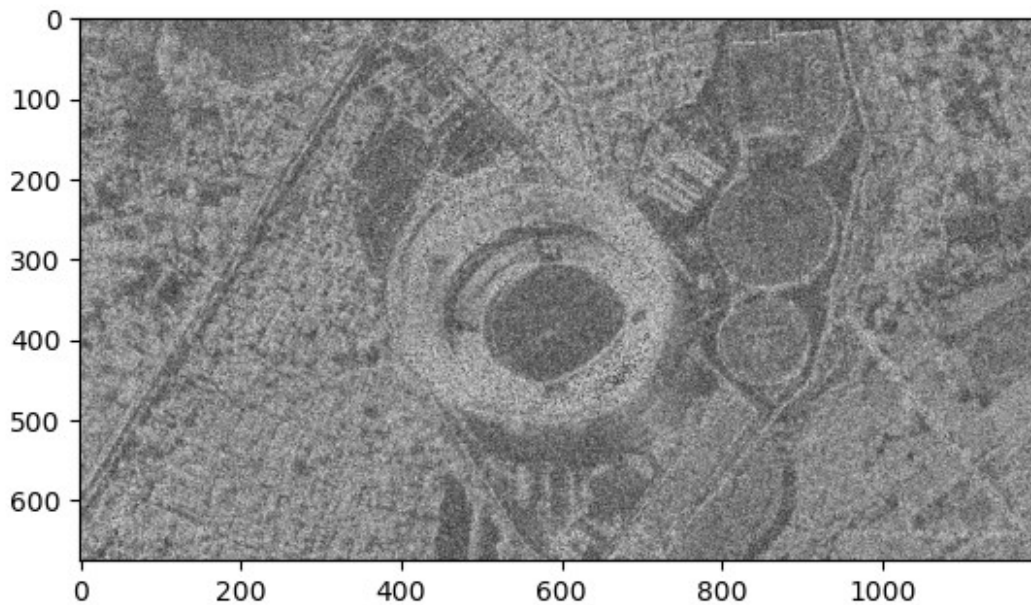
```



```

image_noise_constant = image1_gray + noise_constant
plt.imshow(image_noise_constant, cmap="gray")
<matplotlib.image.AxesImage at 0x7d884b8>

```



```
# Шум соль-перец
```

```
noise_sp = np.random.randint(0, 110, size = (image1_gray.shape[0],  
image1_gray.shape[1]), dtype=int)
```

```
zeros_pixel = np.where(noise_sp == 0)
```

```
ones_pixel = np.where(noise_sp == 100)
```

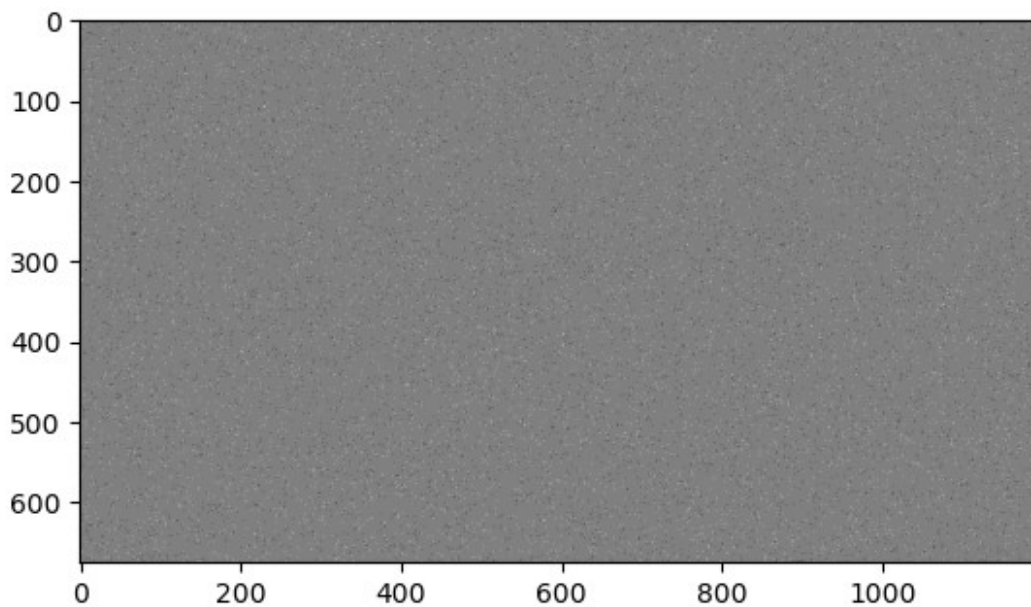
```
bg_image = np.ones(image1_gray.shape, np.uint8) * 128
```

```
bg_image[zeros_pixel] = 0
```

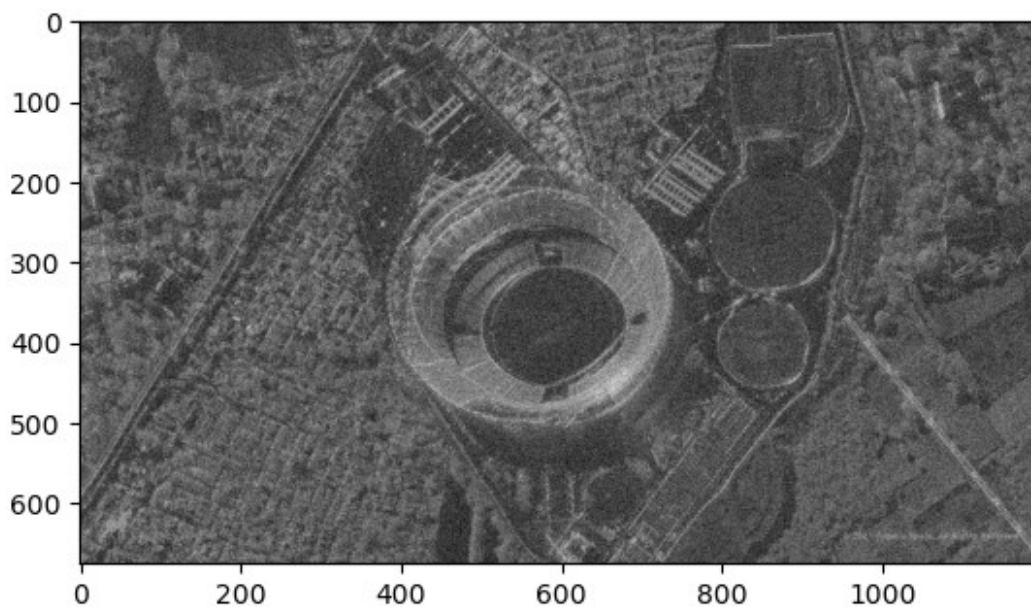
```
bg_image[ones_pixel] = 255
```

```
plt.imshow(bg_image, cmap="gray")
```

```
<matplotlib.image.AxesImage at 0x79d4308>
```

```
image_noise_sp = image1_gray + noise_sp  
plt.imshow(image_noise_sp, cmap="gray")  
<matplotlib.image.AxesImage at 0x7520a88>
```



Протестировать медианный фильтр, фильтр гаусса, билатериальный фильтр, фильтр нелокальных средних с различными параметрами.

```

# Медианный фильтр

from skimage.metrics import structural_similarity, mean_squared_error

# Сравнение: зашумлённое (шум Гаусса) и исходное изображение
mse_gauss = mean_squared_error(image1_gray, image_noise_gauss)
(ssim_gauss, diff) = structural_similarity(image1_gray,
image_noise_gauss, full=True)
print(mse_gauss, ssim_gauss)

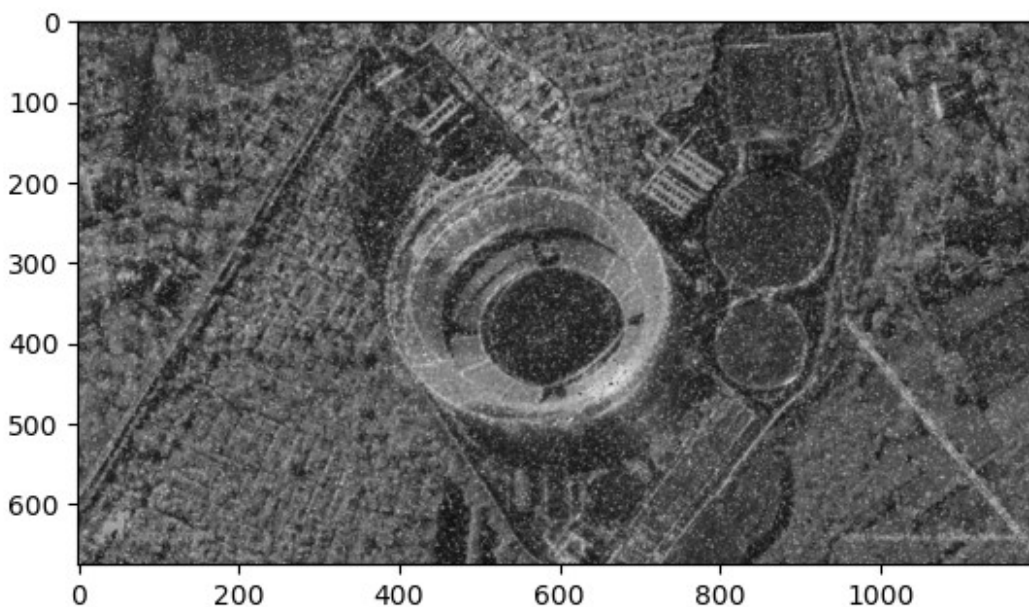
3563.1211555555556 0.15545442994021116

# Сравнение: зашумлённое (постоянный шум) и исходное изображение
mse_constant = mean_squared_error(image1_gray, image_noise_constant)
(ssim_constant, diff) = structural_similarity(image1_gray,
image_noise_constant, full=True)
print(mse_constant, ssim_constant)

5406.52117654321 0.13729035695073863

image_gauss_median = cv2.medianBlur(image_noise_gauss, 3)
image_constant_median = cv2.medianBlur(image_noise_constant, 3)
plt.imshow(image_gauss_median, cmap="gray")
<matplotlib.image.AxesImage at 0xadaeca8>

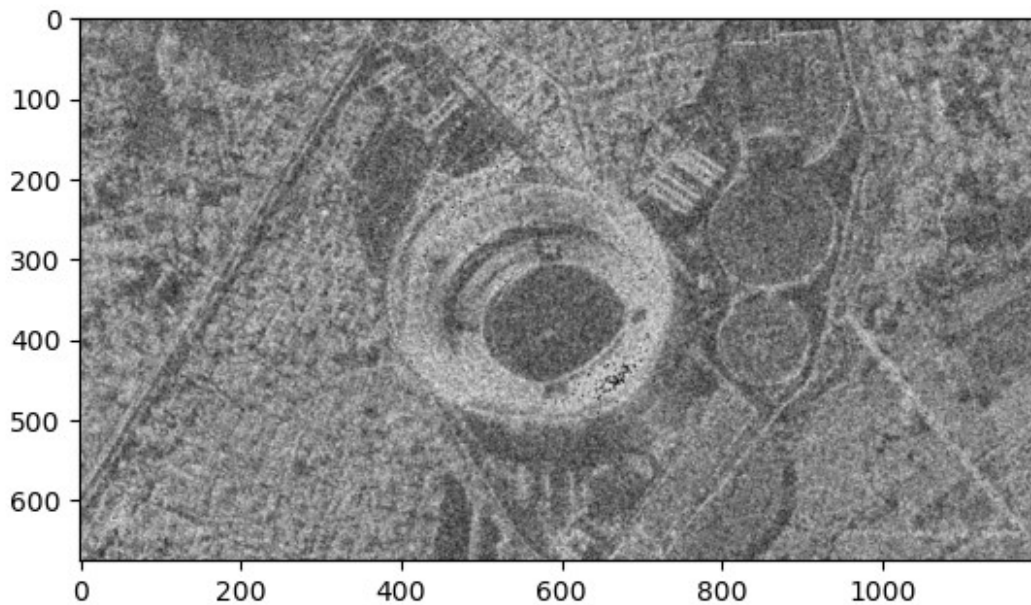
```



```

plt.imshow(image_constant_median, cmap="gray")
<matplotlib.image.AxesImage at 0xbf0c308>

```



```
# Сравнение: исходное и отфильтрованное (гауссовский шум, медианный
фильтр) изображение
```

```
mse_gauss_median_cleared = mean_squared_error(image1_gray,
image_gauss_median)
(ssim_gauss_median_cleared, diff) = structural_similarity(image1_gray,
image_gauss_median, full=True)
print(mse_gauss_median_cleared, ssim_gauss_median_cleared)
```

```
722.475149382716 0.4538273698436858
```

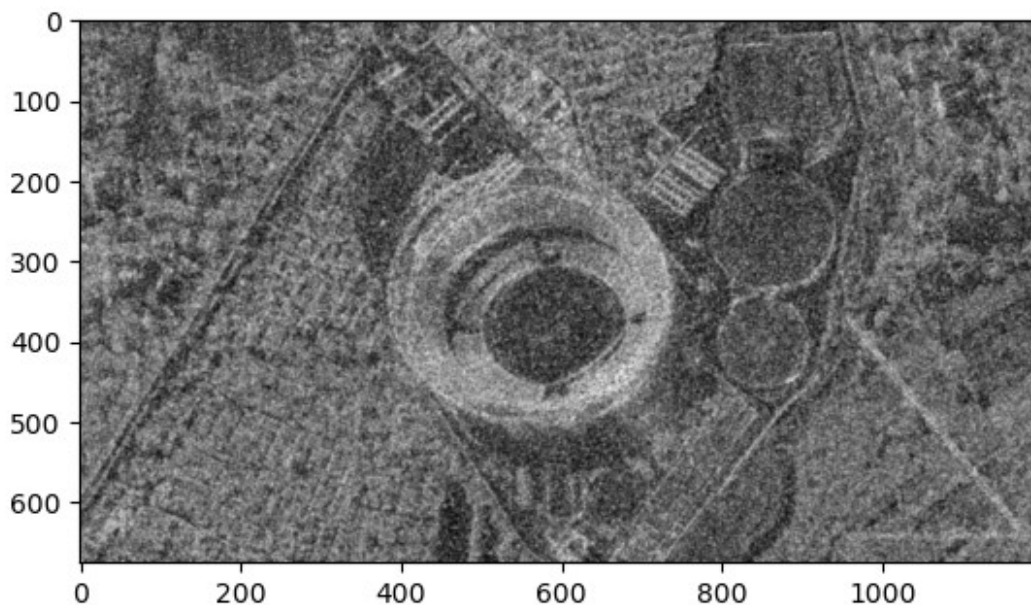
```
# Сравнение: исходное и отфильтрованное (постоянный шум, медианный
фильтр) изображение
```

```
mse_constant_median_cleared = mean_squared_error(image1_gray,
image_constant_median)
(ssim_constant_median_cleared, diff) =
structural_similarity(image1_gray, image_constant_median, full=True)
print(mse_constant_median_cleared, ssim_constant_median_cleared)
```

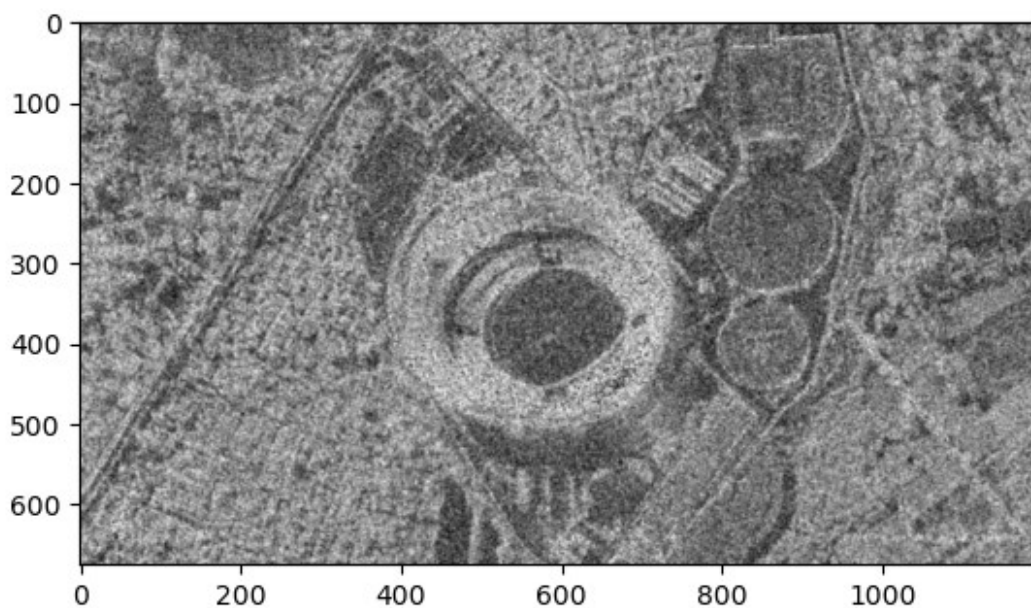
```
3274.839475308642 0.3116543001738666
```

```
# Фильтр Гаусса
```

```
image_gauss_gauss = cv2.GaussianBlur(image_noise_gauss,(5,5),0)
image_constant_gauss = cv2.GaussianBlur(image_noise_constant,(5,5),0)
plt.imshow(image_gauss_gauss, cmap="gray")
<matplotlib.image.AxesImage at 0xc5bda50>
```

```
plt.imshow(image_constant_gauss, cmap="gray")
<matplotlib.image.AxesImage at 0xc405510>
```



```
# Сравнение: исходное и отфильтрованное (гауссовский шум, фильтр Гаусса) изображение
mse_gauss_gauss_cleared = mean_squared_error(image1_gray,
image_gauss_gauss)
(ssim_gauss_gauss_cleared, diff) = structural_similarity(image1_gray,
```



```

image_gauss_gauss, full=True)
print(mse_gauss_gauss_cleared, ssim_gauss_gauss_cleared)

1216.6419407407407 0.4325970255709457

# Сравнение: исходное и отфильтрованное (постоянный шум, фильтр Гаусса) изображение
mse_constant_gauss_cleared = mean_squared_error(image1_gray,
image_constant_gauss)
(ssim_constant_gauss_cleared, diff) =
structural_similarity(image1_gray, image_constant_gauss, full=True)
print(mse_constant_gauss_cleared, ssim_constant_gauss_cleared)

3186.9474641975307 0.3925272164310304

# Билатеральный фильтр (на примере гауссовского шума)

image_gauss_bilat_1 =
cv2.bilateralFilter(image_noise_gauss, 49, 150, 150)

image_gauss_bilat_2 =
cv2.bilateralFilter(image_noise_gauss, 25, 150, 150)

image_gauss_bilat_3 = cv2.bilateralFilter(image_noise_gauss, 9, 150, 150)
image_gauss_bilat_4 = cv2.bilateralFilter(image_noise_gauss, 9, 75, 75)

# Сравнение: исходное и отфильтрованное (гауссовский шум, фильтр Гаусса) изображение
mse_gauss_bilat_cleared_1 = mean_squared_error(image1_gray,
image_gauss_bilat_1)
(ssim_gauss_bilat_cleared_1, diff) =
structural_similarity(image1_gray, image_gauss_bilat_1, full=True)
print(mse_gauss_bilat_cleared_1, ssim_gauss_bilat_cleared_1)

1277.5124740740741 0.2821560176737664

# Сравнение: исходное и отфильтрованное (гауссовский шум, фильтр Гаусса) изображение
mse_gauss_bilat_cleared_2 = mean_squared_error(image1_gray,
image_gauss_bilat_2)
(ssim_gauss_bilat_cleared_2, diff) =
structural_similarity(image1_gray, image_gauss_bilat_2, full=True)
print(mse_gauss_bilat_cleared_2, ssim_gauss_bilat_cleared_2)

1196.2125222222223 0.29476651286369254

# Сравнение: исходное и отфильтрованное (гауссовский шум, фильтр Гаусса) изображение
mse_gauss_bilat_cleared_3 = mean_squared_error(image1_gray,
image_gauss_bilat_3)
(ssim_gauss_bilat_cleared_3, diff) =

```

```
structural_similarity(image1_gray, image_gauss_bilat_3, full=True)
print(mse_gauss_bilat_cleared_3, ssim_gauss_bilat_cleared_3)
```

```
1075.0680185185186 0.40168140982978584
```

```
# Сравнение: исходное и отфильтрованное (гауссовский шум, фильтр Гаусса) изображение
```

```
mse_gauss_bilat_cleared_4 = mean_squared_error(image1_gray,
image_gauss_bilat_4)
(ssim_gauss_bilat_cleared_4, diff) =
structural_similarity(image1_gray, image_gauss_bilat_4, full=True)
print(mse_gauss_bilat_cleared_4, ssim_gauss_bilat_cleared_4)
```

```
1296.8504765432099 0.298701417709512
```

```
# Фильтр нелокальных средних (на примере гауссовского шума)
```

```
image_gauss_nlm_1 = cv2.fastNlMeansDenoising(image_noise_gauss, h =
20)
```

```
image_gauss_nlm_2 = cv2.fastNlMeansDenoising(image_noise_gauss, h =
120)
```

```
image_gauss_nlm_3 = cv2.fastNlMeansDenoising(image_noise_gauss, h = 2)
```

```
# Сравнение: исходное и отфильтрованное (гауссовский шум, фильтр Гаусса) изображение
```

```
mse_gauss_nlm_cleared_1 = mean_squared_error(image1_gray,
image_gauss_nlm_1)
(ssim_gauss_nlm_cleared_1, diff) = structural_similarity(image1_gray,
image_gauss_nlm_1, full=True)
print(mse_gauss_nlm_cleared_1, ssim_gauss_nlm_cleared_1)
```

```
3557.8316777777777 0.15565741070147426
```

```
# Сравнение: исходное и отфильтрованное (гауссовский шум, фильтр Гаусса) изображение
```

```
mse_gauss_nlm_cleared_2 = mean_squared_error(image1_gray,
image_gauss_nlm_2)
(ssim_gauss_nlm_cleared_2, diff) = structural_similarity(image1_gray,
image_gauss_nlm_2, full=True)
print(mse_gauss_nlm_cleared_2, ssim_gauss_nlm_cleared_2)
```

```
1338.0881543209875 0.24829356626555155
```

```
# Сравнение: исходное и отфильтрованное (гауссовский шум, фильтр Гаусса) изображение
```

```
mse_gauss_nlm_cleared_3 = mean_squared_error(image1_gray,
image_gauss_nlm_3)
(ssim_gauss_nlm_cleared_3, diff) = structural_similarity(image1_gray,
image_gauss_nlm_3, full=True)
print(mse_gauss_nlm_cleared_3, ssim_gauss_nlm_cleared_3)
```

```
3563.1211555555556 0.15545442994021116
```

```
# Выяснить, какой фильтр показал лучший результат фильтрации шума.
```

```
# На примере шума Гаусса
```

```
# image_gauss_median (медианный) - лучший
```

```
print(mse_gauss_median_cleared, ssim_gauss_median_cleared)
```

```
722.475149382716 0.4538273698436858
```

```
# image_gauss_gauss (гауссовский)
```

```
print(mse_gauss_gauss_cleared, ssim_gauss_gauss_cleared)
```

```
1216.6419407407407 0.4325970255709457
```

```
# image_gauss_bilat_3 (билатериальный)
```

```
print(mse_gauss_bilat_cleared_3, ssim_gauss_bilat_cleared_3)
```

```
1075.0680185185186 0.40168140982978584
```

```
# image_gauss_nlm_2 (нелокальных средних)
```

```
print(mse_gauss_nlm_cleared_2, ssim_gauss_nlm_cleared_2)
```

```
1338.0881543209875 0.24829356626555155
```