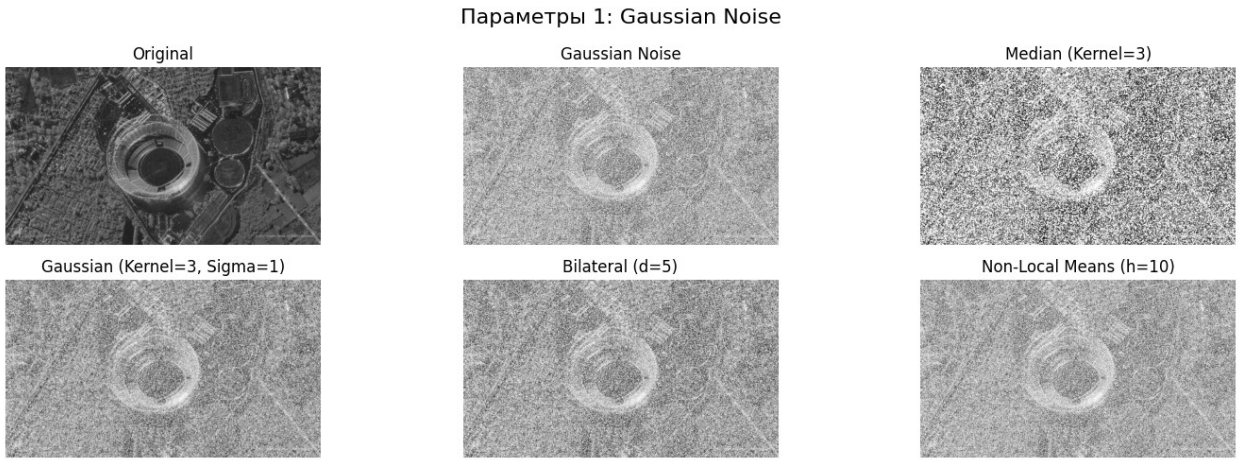
```
import cv2
import numpy as np
import matplotlib.pyplot as plt

def psnr(img1, img2):
    mse = np.mean((img1 - img2) ** 2)
    if mse == 0:
        return float('inf')
    max_pixel = 255.0
    psnr = 20 * np.log10(max_pixel / np.sqrt(mse))
    return psnr

img = cv2.imread('sar_1.jpg', cv2.IMREAD_GRAYSCALE)

gn = np.random.normal(0, 25, img.shape).astype(np.uint8)
img_gn = cv2.add(img, gn)

un = np.random.uniform(-50, 50, img.shape).astype(np.uint8)
img_un = cv2.add(img, un)

kernel_sizes = [3, 5]
sigma_values = [1, 2]
d_values = [5, 9]
h_values = [10, 20]

for i in range(2):

    kernel_size = kernel_sizes[i]
    sigma = sigma_values[i]
    d = d_values[i]
    h = h_values[i]

    mf_gn = cv2.medianBlur(img_gn, kernel_size)
    gf_gn = cv2.GaussianBlur(img_gn, (kernel_size, kernel_size),
sigma)
    bf_gn = cv2.bilateralFilter(img_gn, d, 75, 75)
    nlm_gn = cv2.fastNlMeansDenoising(img_gn, None, h, 7, 21)

    mf_un = cv2.medianBlur(img_un, kernel_size)
    gf_un = cv2.GaussianBlur(img_un, (kernel_size, kernel_size),
sigma)
    bf_un = cv2.bilateralFilter(img_un, d, 75, 75)
    nlm_un = cv2.fastNlMeansDenoising(img_un, None, h, 7, 21)

    titles_gn = ['Original', 'Gaussian Noise',
                 f'Median (Kernel={kernel_size})', f'Gaussian
(Kernel={kernel_size}, Sigma={sigma})',
                 f'Bilateral (d={d})', f'Non-Local Means (h={h})']
    images_gn = [img, img_gn, mf_gn, gf_gn, bf_gn, nlm_gn]
```

```
    plt.figure(figsize=(15, 5))
    plt.suptitle(f'Параметры {i + 1}: Gaussian Noise', fontsize=16)
    for j in range(len(images_gn)):
        plt.subplot(2, 3, j + 1)
        plt.imshow(images_gn[j], cmap='gray')
        plt.title(titles_gn[j])
        plt.axis('off')

    plt.tight_layout()
    plt.show()

    print(f"PSNR for Median Filter (Gaussian, Kernel={kernel_size}):",
psnr(img, mf_gn))
    print(f"PSNR for Gaussian Filter (Gaussian, Kernel={kernel_size},
Sigma={sigma}):", psnr(img, gf_gn))
    print(f"PSNR for Bilateral Filter (Gaussian, d={d}):", psnr(img,
bf_gn))
    print(f"PSNR for Non-Local Means (Gaussian, h={h}):", psnr(img,
nlm_gn))

    titles_un = ['Original', 'Uniform Noise',
                 f'Median (Kernel={kernel_size})', f'Gaussian
(Kernel={kernel_size}, Sigma={sigma})',
                 f'Bilateral (d={d})', f'Non-Local Means (h={h})']
    images_un = [img, img_un, mf_un, gf_un, bf_un, nlm_un]

    plt.figure(figsize=(15, 5))
    plt.suptitle(f'Параметры {i + 1}: Uniform noise', fontsize=16)
    for j in range(len(images_un)):
        plt.subplot(2, 3, j + 1)
        plt.imshow(images_un[j], cmap='gray')
        plt.title(titles_un[j])
        plt.axis('off')

    plt.tight_layout()
    plt.show()

    print(f"PSNR for Median Filter (Uniform, Kernel={kernel_size}):",
psnr(img, mf_un))
    print(f"PSNR for Gaussian Filter (Uniform, Kernel={kernel_size},
Sigma={sigma}):", psnr(img, gf_un))
    print(f"PSNR for Bilateral Filter (Uniform, d={d}):", psnr(img,
bf_un))
    print(f"PSNR for Non-Local Means (Uniform, h={h}):", psnr(img,
nlm_un))

img2 = cv2.imread('cells_2.jpg', cv2.IMREAD_GRAYSCALE)


frequencies = [0.1, 0.25, 0.5]
```

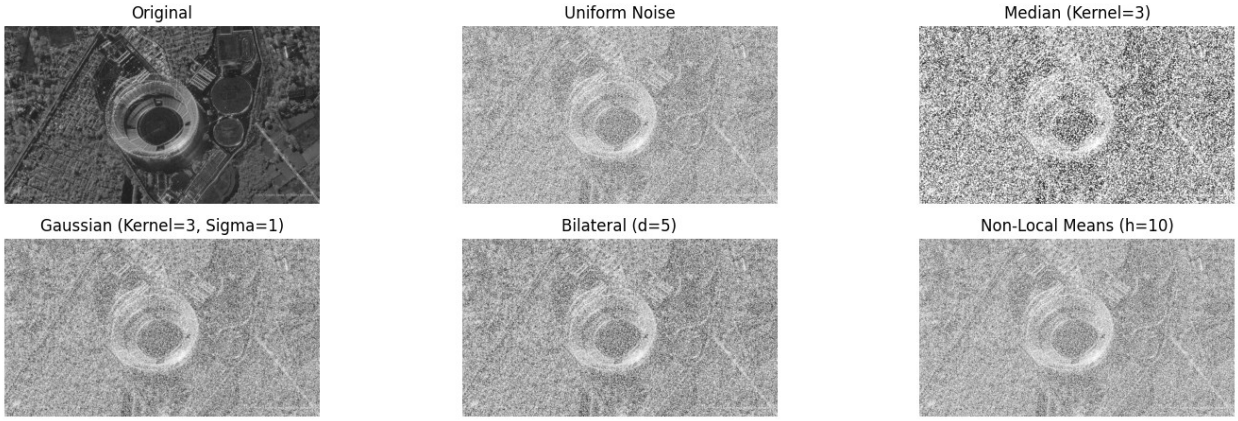
```
kernel = np.ones((3, 3), np.uint8)

plt.figure(figsize=(15, 5))
for i, freq in enumerate(frequencies):
    spn = np.random.choice([0, 255], size=img2.shape, p=[1 - freq,
freq])
    spn = spn.astype(np.uint8)
    img2_sp = cv2.add(img2, spn)

    opening = cv2.morphologyEx(img2_sp, cv2.MORPH_OPEN, kernel)
    closing = cv2.morphologyEx(img2_sp, cv2.MORPH_CLOSE, kernel)




    plt.subplot(3, 3, i * 3 + 1)
    plt.imshow(img2_sp, cmap='gray')
    plt.title(f'Шум (частота={freq})')
    plt.axis('off')

    plt.subplot(3, 3, i * 3 + 2)
    plt.imshow(opening, cmap='gray')
    plt.title('Открытие')
    plt.axis('off')

    plt.subplot(3, 3, i * 3 + 3)
    plt.imshow(closing, cmap='gray')
    plt.title('Закрытие')
    plt.axis('off')

plt.tight_layout()
plt.show()
```




```
PSNR for Median Filter (Gaussian, Kernel=3): 27.567132452729915
PSNR for Gaussian Filter (Gaussian, Kernel=3, Sigma=1):
```

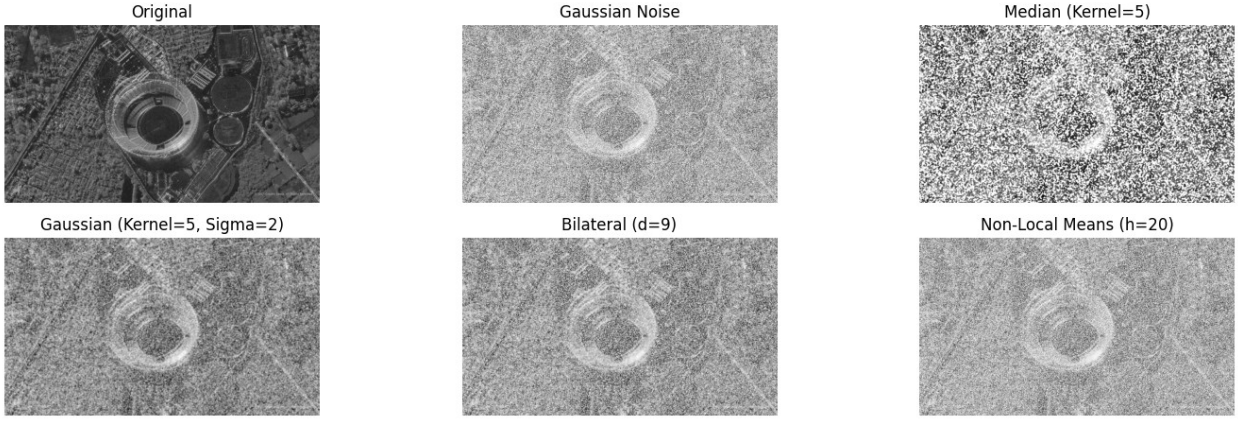
```
27.855304695310092
PSNR for Bilateral Filter (Gaussian, d=5): 27.63121672734773
PSNR for Non-Local Means (Gaussian, h=10): 28.104473442306354
```

Параметры 1: Uniform noise



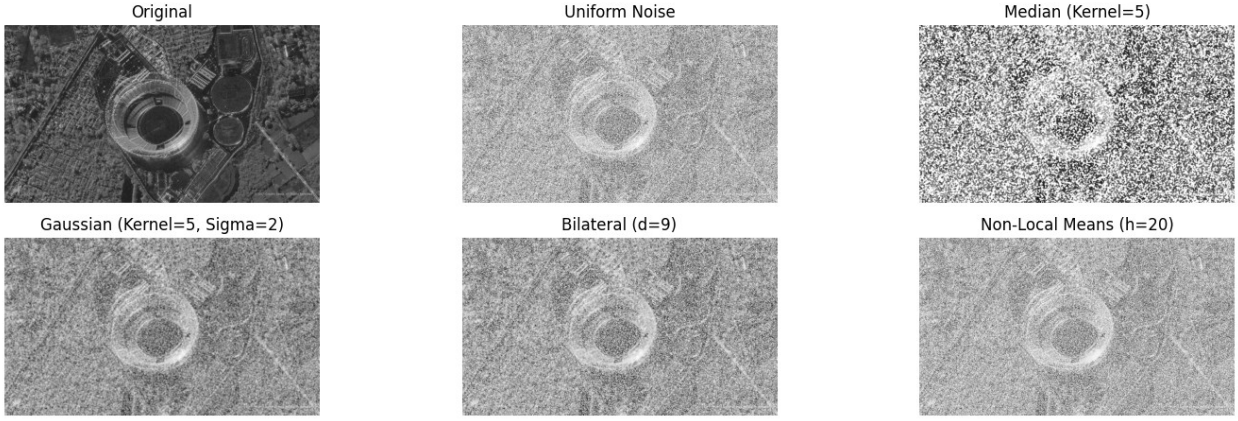

```
PSNR for Median Filter (Uniform, Kernel=3): 27.498526399335802
PSNR for Gaussian Filter (Uniform, Kernel=3, Sigma=1):
27.898193277445348
PSNR for Bilateral Filter (Uniform, d=5): 27.603317230047114
PSNR for Non-Local Means (Uniform, h=10): 27.872277987502653
```

Параметры 2: Gaussian Noise




```
PSNR for Median Filter (Gaussian, Kernel=5): 27.53949751519993
PSNR for Gaussian Filter (Gaussian, Kernel=5, Sigma=2):
27.852269386149388
PSNR for Bilateral Filter (Gaussian, d=9): 27.61975938751175
PSNR for Non-Local Means (Gaussian, h=20): 28.10164367879551
```

Параметры 2: Uniform noise




```
PSNR for Median Filter (Uniform, Kernel=5): 27.5143045745763
PSNR for Gaussian Filter (Uniform, Kernel=5, Sigma=2):
27.93751103114172
PSNR for Bilateral Filter (Uniform, d=9): 27.591291621932932
PSNR for Non-Local Means (Uniform, h=20): 27.87117966153369
```