

## Задание 2. Типы шума на изображении, моделирование аддитивного шума, алгоритмы фильтрации изображений.



1) Реализовать моделирование аддитивного шума (на выбор).

Реализация шума была выполнена методом Монте-Карло с использованием метода обратных функций.

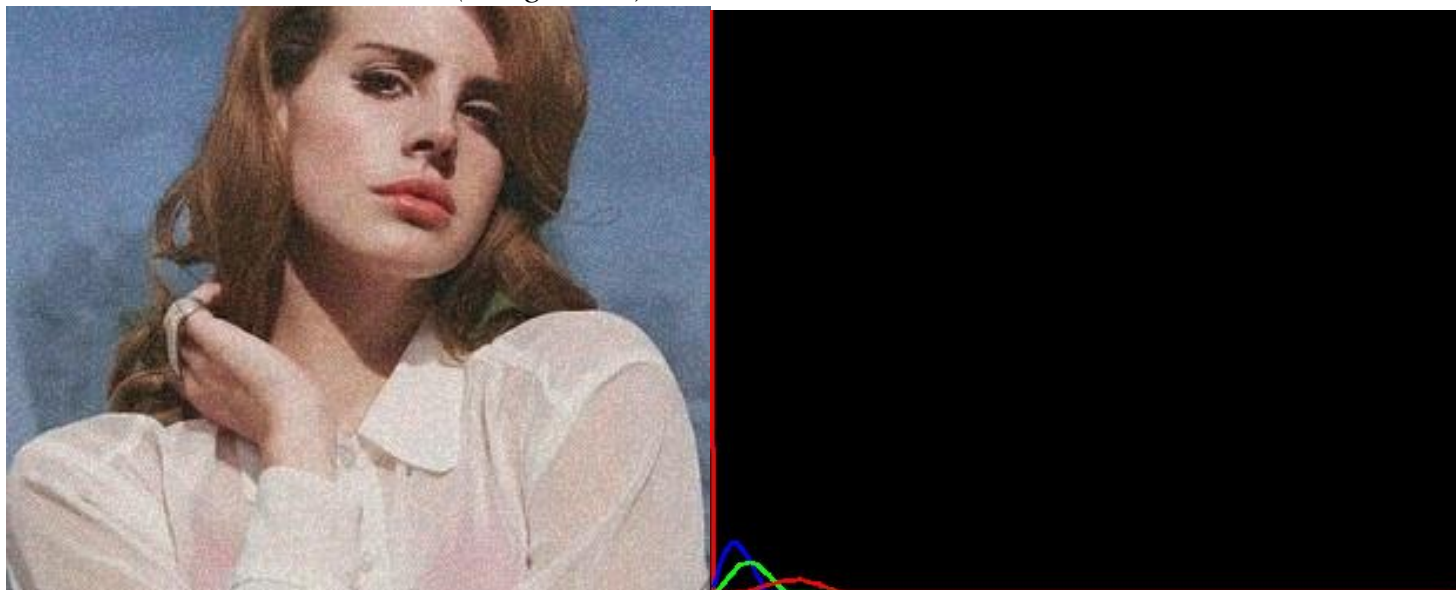
Гистограммы можно увеличить путем увеличения процента шумовых пикселей

1. GammaNoise(50%)

*vector<float> a = {4.5, 4.5, 4.5};*

*vector<int> b = {4, 5, 8};*

*GammaNoise(image, a, b);*



## 2. GaussianNoise(10%)

```
vector<int> m = {140, 120, 100};
```

```
vector<float> dis = {2, 1.5, 1};
```

```
GaussianNoise(image, m, dis);
```



## 3. ExponentialNoise(50%)

```
vector<float> a = {0.05, 0.1, 1};
```

```
ExponentialNoise(image, a);
```

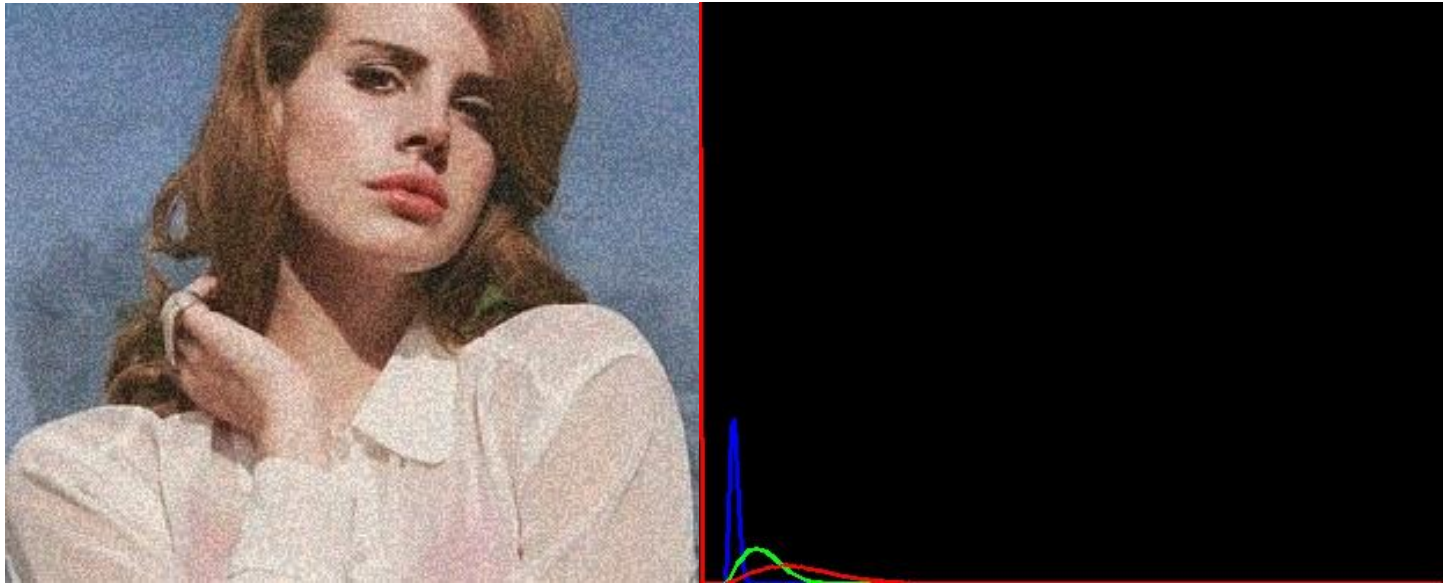


## 4. RayleighNoise(50%)

```
vector<float> a = {10, 10, 10};
```

```
vector<float> disp = {2, 10, 20};
```

```
RayleighNoise(image, a, disp);
```



5. ConstantNoise(50%)

```
vector<int> a = {30, 50, 80};  
vector<int> b = {35, 60, 100};  
ConstantNoise(image, a, b);
```



6. SaltPepperNoise(50%)

```
vector<int> min = {10, 20, 30};  
vector<int> max = {210, 220, 230};  
SaltPepperNoise(image, min, max);
```





2) Удалить шум с использованием стандартных функций библиотеки OpenCV: гаусс, медианный, билатериальный, фильтра нелокальных средних.(на выбор)

1.Gaussian Noise + Gaussian Blur



2. Gamma noise + Median Filter



3) Реализовать алгоритм удаления шума (на выбор из таблицы).

1. Gamma Noise + Median Filter



2. Uniform Noise + Mean Filter



3. Gaussian Noise + Gaussian Filter



4. Rayleigh Noise + Opening



5. Exponential Noise + Closing



6. Salt&Pepper Noise + Midpoint Filter



- 4) Исследовать качество фильтрации алгоритма. (создать собственную базу, или найти открытые тестовые базы в Интернете)
- 5) Сравните качество работы алгоритмов фильтрации или качество фильтрации для различных типов шумов. Метрика для сравнения: SSIM. Какой вывод можно сделать, глядя на результаты?

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)},$$

Приведённая формула применима только для яркости изображения, по которой и происходит оценка качества. Полученный SSIM-индекс лежит в пределах от  $-1$  до  $+1$ . Значение  $+1$  достигается только при полной аутентичности образцов. Как правило, метрика рассчитана на окно размером  $8 \times 8$  пикселей. Окно может смещаться через пиксель, но специалисты рекомендуют использовать группы окон для уменьшения сложности вычислений.

изация фильтров: a / openCV	ssian Blur	lian Filter
ssian Filter	8181	
lian Filter		8136

Сравнение качества фильтрации разных типов шумов с помощью различных фильтров. Метрика SSIM, сравнение с идеальным (незашумленным) изображением.

Фильтрация шума на полноцветных изображениях:

r\noise	hama	form	ss	p
lian	844	900	562	086
n	468	095	7259	938
ss	152	508	605	298
point	125	871	737	446

На бинарных изображениях:

r\noise	leigh	ponential
ning	355	346
sing	267	257

Качество работы алгоритмов фильтрации и алгоритмов фильтрации OpenCV:

r\noise	ssBlur	lianFilter
ss	750	-----
lian	-----	175