

Статистика, прикладной поток

Продвинутое практическое задание 1

Иван Генрихович завидует крутости преподавателей прикладного потока. Зайдя в очередной раз на страницу курса прикладного потока, он не выдержал и зашумил фотографии всех преподавателей. Теперь преподаватели грустят. Помогите им восстановить исходные изображения. Тому, кто восстановит изображение одного из преподавателей точнее всего, соответствующий преподаватель подарит некоторый подарок.

Зашумленные изображения: Никита (https://mipt-stats.gitlab.io/files/applied_stats_18/practice/super_1/Nikita.jpg), Оля (https://mipt-stats.gitlab.io/files/applied_stats_18/practice/super_1/Olya.jpg), Лена (https://mipt-stats.gitlab.io/files/applied_stats_18/practice/super_1/Lena.jpg), Настя (https://mipt-stats.gitlab.io/files/applied_stats_18/practice/super_1/Nastya.jpg).

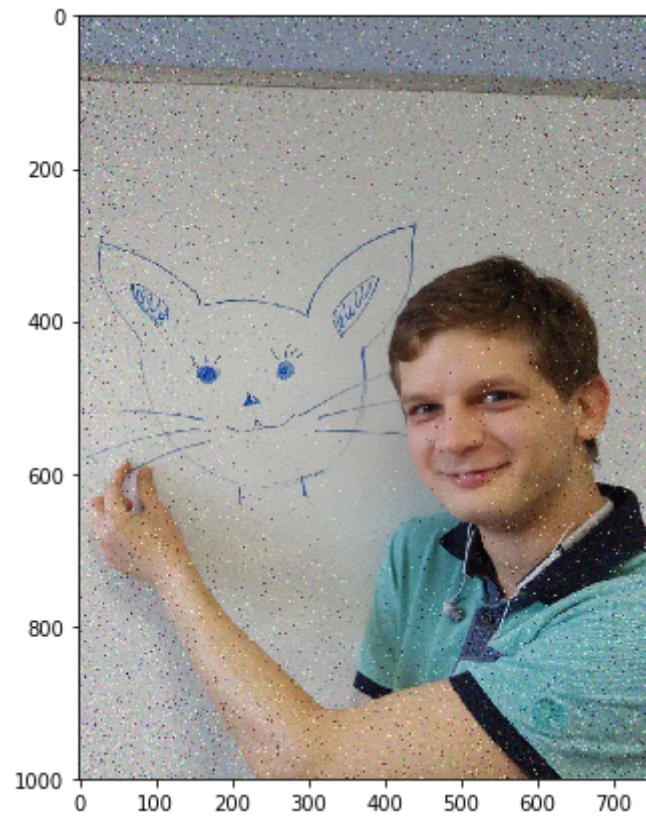
Разберемся с технической частью работы с изображениями в Питоне. Загрузите изображение в `numpy.array` и посмотрите на размер полученного массива чисел.

```
In [11]: import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

images = [plt.imread('Nikita.jpg'), plt.imread('Olya.jpg'), plt.imread('Lena.jpg'), plt.imread('Nastya.jpg')]
```

Изображение представляет собой трехмерный массив чисел с размерностями, соответствующим высоте и длине изображения, а также его цветовым слоям RGB или RGBA (а -- прозрачность). В нашем случае все значения являются целыми числами от 0 до 255. Теперь напечатайте изображение.

```
In [12]: plt.figure(figsize=(14, 7))  
plt.imshow(images[0]);
```



Как видим, изображение достаточно зашумленное. Ваша задача -- избавиться от шума на изображении. В данной задаче гарантируется, что шум устроен следующим образом. Пусть $X = (x_{ijk}, i = 1...H, j = 1...W, k \in \{1, 2, 3\})$ -- исходное изображение, а $\varepsilon = (\varepsilon_{ijk}, i = 1...H, j = 1...W, k \in \{1, 2, 3\})$ -- шум, являющийся набором независимых случайных величин. Тогда зашумленное изображение $Y = (y_{ijk}, i = 1...H, j = 1...W, k \in \{1, 2, 3\})$ определяется как

$$y_{ijk} = \begin{cases} \lfloor x_{ijk} + \varepsilon_{ijk} \rfloor, & \text{если } 0 < x_{ijk} + \varepsilon_{ijk} < 255; \\ 0, & \text{если } x_{ijk} + \varepsilon_{ijk} \leq 0; \\ 255, & \text{если } x_{ijk} + \varepsilon_{ijk} \geq 255. \end{cases}$$

Более того, $P(\varepsilon_{ijk} = 0) > 0$, то есть зашумляется не каждый пиксель.

Вам выдано зашумленное изображение Y , а исходное изображение X и шум ε вам неизвестны. Задача -- на основе изображения Y построить новое изображение $Z = (z_{ijk}, i = 1...H, j = 1...W, k \in \{1, 2, 3\})$ минимизируя среднеквадратичное отклонение по цветам пикселей, определяемое как

$$MSE = \frac{1}{3HW} \sum_{i=1}^H \sum_{j=1}^W \sum_{k=1}^3 (x_{ijk} - z_{ijk})^2$$

Ограничение на решения -- денойзинг изображения должен быть произведен исключительно статистическими методами, то есть в основе вашего решения должна быть некоторая статистическая модель. Каждый пиксель восстановленного изображения должен быть функцией от пикселей зашумленного изображения по окрестности необходимого размера. Например, восстановление каждого пикселя может быть произведено на основе следующих статистик, посчитанных по окрестности каждого пикселя одного цветового канала:

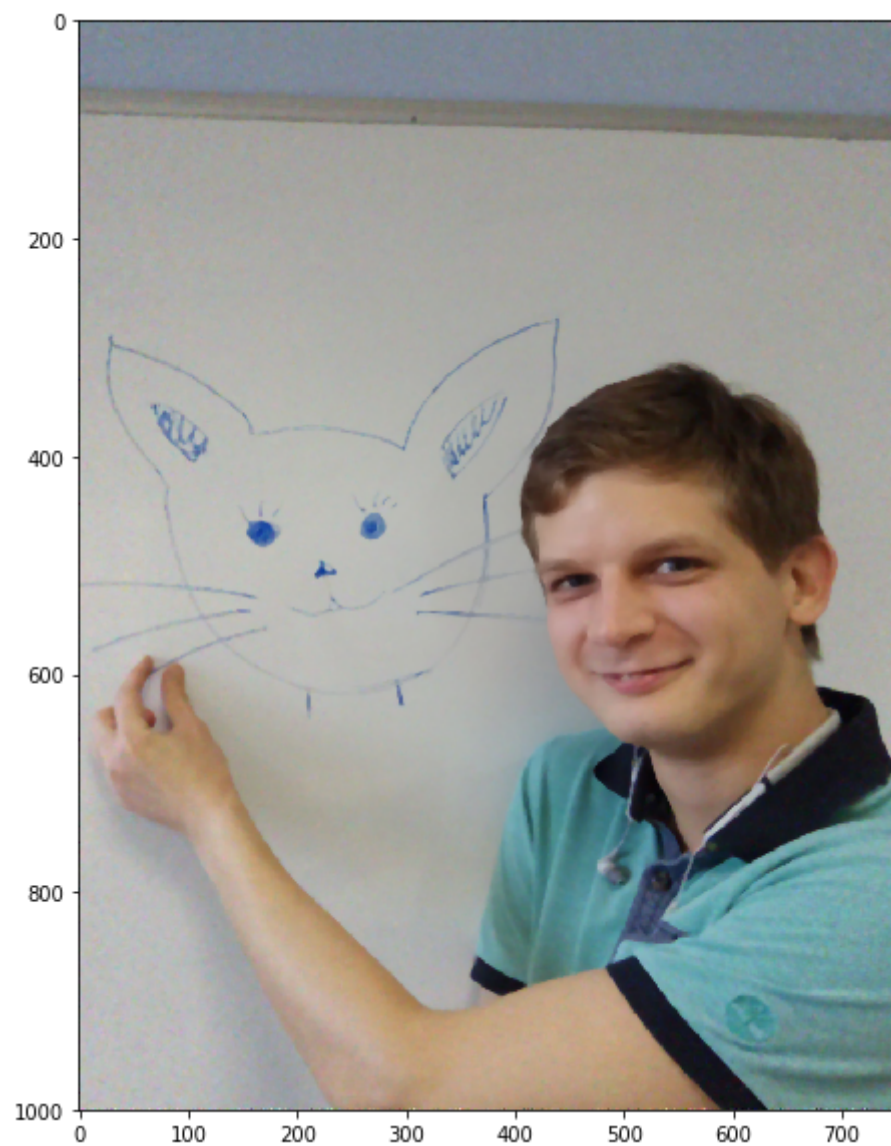
- среднее;
- усеченное среднее;
- медиана;
- медиана средних Уолша;
- оценка Питмена;
- взвешенное среднее;
- другие матричные фильтры (см. <https://habr.com/post/142818/> (<https://habr.com/post/142818/>)).

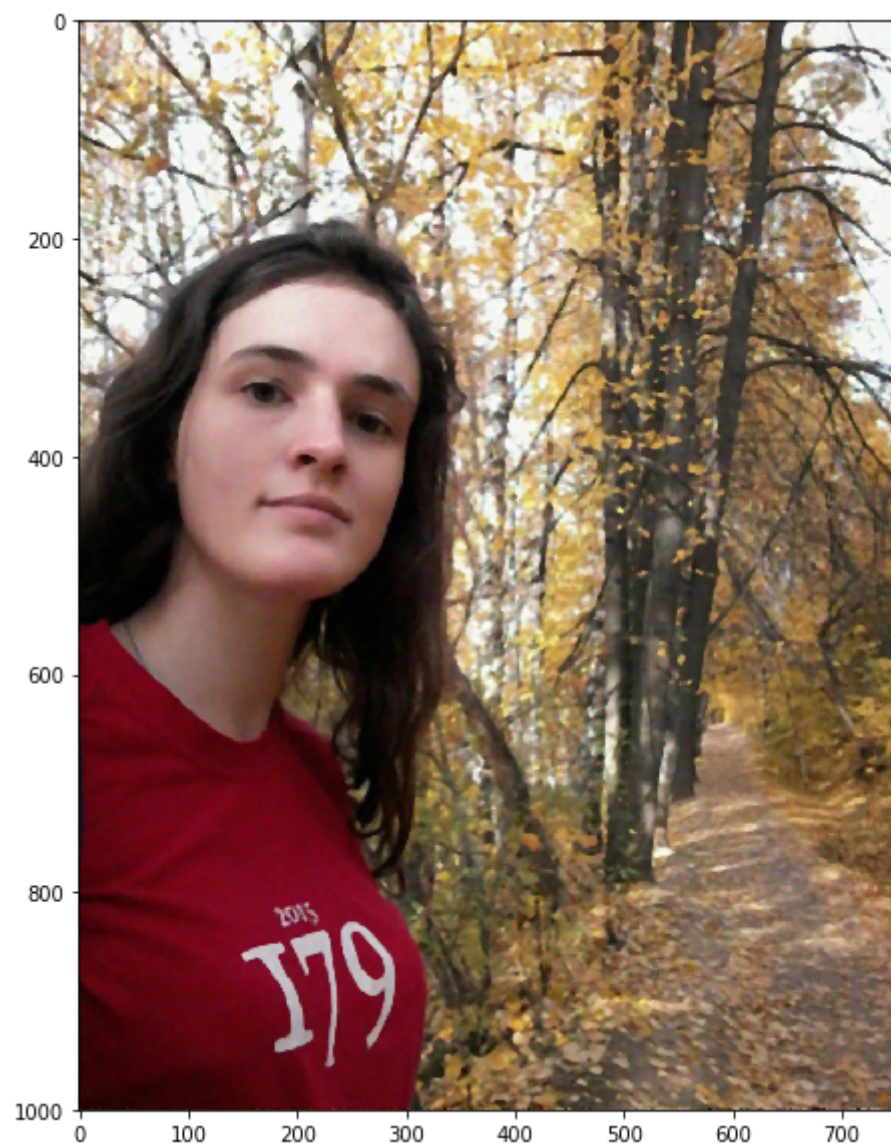
В решении приветствуется также статистический и иной *первичный* анализ изображения.

```
In [14]: def median_filter(image, half_window=2):  
        """  
        For every pixel calculates median and returns this image.  
        """  
        result = image.copy()  
        h, w, colors = image.shape  
        window_size = (2*half_window+1)**2  
  
        for i in range(half_window, h-half_window):  
            for j in range(half_window, w-half_window):  
                for col in range(colors):  
                    window = image[i-half_window:i+half_window+1, j-half_window:j+half_window+1, col]  
                    result[i, j, col] = np.median(window)  
  
        return result
```

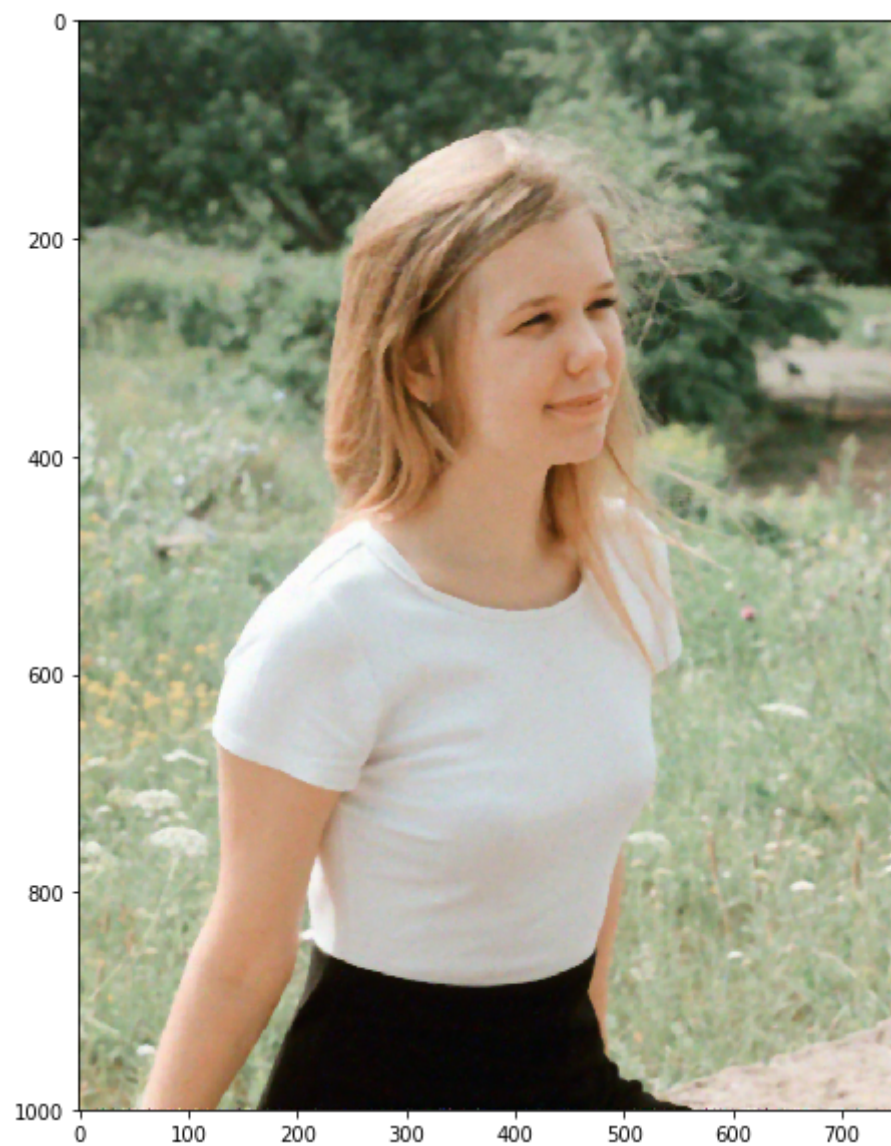
```
In [15]: results = [median_filter(image, half_window=2) for image in images]
```

```
In [16]: for image in results:
          plt.figure(figsize=(15,10))
          plt.imshow(image)
```









Kaggle

Для сравнения решений мы используем систему Kaggle. Для фоток семинаристов проводятся отдельные соревнования. *Участвовать в каждом из них могут все слушатели прикладного потока.*

- Никита: [ссылка \(https://www.kaggle.com/c/mipt-stats18-nikita\)](https://www.kaggle.com/c/mipt-stats18-nikita), инвайт (<https://www.kaggle.com/t/1ca7f53610ba47b7b82f1f79b16d9e3c>)
- Лена: [ссылка \(https://www.kaggle.com/c/mipt-stats18-lena\)](https://www.kaggle.com/c/mipt-stats18-lena), инвайт (<https://www.kaggle.com/t/330e6b00a32a4c38b6b98c55a70d5a16>)
- Оля: [ссылка \(https://www.kaggle.com/c/mipt-stats18-olya\)](https://www.kaggle.com/c/mipt-stats18-olya), инвайт (<https://www.kaggle.com/t/52e0a6d528b14be8bd3e9a9dc565913b>)
- Настя: [ссылка \(https://www.kaggle.com/c/mipt-stats18-nastya\)](https://www.kaggle.com/c/mipt-stats18-nastya), инвайт (<https://www.kaggle.com/t/06c1e496b8ed47df9762b0194f6897ea>)

Правила:

- В Kaggle можно отправлять не более 5 решений в день (6-ю система не позволит);
- Решения индивидуальные;
- Качество считается по метрике MSE (см. выше);
- До окончания соревнования доступны значения качества, посчитанные только на случайных 30% пикселей. Значения отображаются в Public Leaderboard;
- После окончания соревнования становится доступным Private Leaderboard, в котором значения качества посчитаны на оставшихся 70% пикселей;
- Баллы и подарки выдаются согласно Private Leaderboard;
- Для включения в Private Leaderboard можно выбрать две посылки;
- В Leaderboard должны отображаться ваши реальные имя и фамилия. В противном случае решение может быть не зачтено;
- Для каждого соревнования полное решение с подробным его описанием, соответствующим наилучшей попытке, должно быть **в течении суток после окончания соревнования** выслано на почту `mipt.stats@yandex.ru`, указав тему письма "[applied] Фамилия Имя - Денойзинг изображений". Квадратные скобки обязательны. Если письмо дошло, придет ответ от автоответчика.
- Решения, размещенные на каких-либо интернет-ресурсах не принимаются. Кроме того, публикация решения в открытом доступе может быть приравнена к предоставлению возможности списать.
- Код решений-победителей будет запущен для проверки. Код других решений также может быть запущен.

Для отправки в Kaggle из восстановленного изображения нужно получить csv-файл пикселей, который можно получить с помощью следующего кода.

```
In [17]: import pandas as pd
```

```
In [18]: for i in range(4):  
        pixels = pd.DataFrame({'Pixel': results[i].ravel()})  
        pixels.index.name = 'Id'  
        pixels.to_csv('./answer%d.csv' % i)
```