**Лабораторная работа №2**

**Пространственная обработка изображений**

**Содержание**

1. Яркостные преобразования. Гистограмма…………………...…………..……..2
2. Пространственная фильтрация………………………..…………………...……6

Примечание – все задания выполнять на языке программирования Python!

1. **Яркостные преобразования. Гистограмма.**

**Перед выполнением данной части лабораторной работы ознакомьтесь с приложенными теоретическими сведениями из папки – “Яркостные преобразования”, а также информацию по ссылке:**

**https://russianblogs.com/article/83571365081/**

**Задание 1.**

[**http://opencv.jp/opencv-2.1\_org/cpp/histograms.html**](http://opencv.jp/opencv-2.1_org/cpp/histograms.html)

[**https://matplotlib.org/stable/api/\_as\_gen/matplotlib.pyplot.hist.html**](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html)

Изучить функцию calcHist() библиотеки OpenCV и функцию matplot.pyplot.hist() библиотеки matplotlib по ссылкам выше. Написать программу, которая:

- загружает произвольное цветное изображение

- вычисляет и выводит на экран **!нормированную!(далее по тексту речь идет о нормированной гистограмме, даже если слово пропущено)** гистограмму каждого из трех каналов, следующими способами:

1. Используя функцию matplot.pyplot.hist()(для расчета гистограммы) библиотеки matplotlib и функцию show()(для вывода на экран), той же библиотеки;
2. Используя функцию calcHist()(для расчета гистограммы) библиотеки OpenCV и средства библиотеки matplotlib для вывода ее на экран(функции plot() и show());
3. Самостоятельно рассчитать **нормированную** гистограмму не использую сторонние библиотеки и вывести ее на экран использую средства рисования библиотеки OpenCV(например с помощью функции polylines()).

- сохраняет полученные гистограммы на диск.

Сравните, полученные разными способами, гистограммы.

***ПРИМЕЧАНИЕ 1!!!***

Для дальнейших заданий разрешается использовать любой способ расчета и отображения гистограмм!

***ПРИМЕЧАНИЕ 2!!!***

Яркостные преобразования позволяют повышать качество монохромных(одноканальных) изображений. Для применения таких преобразований к цветным изображениям необходимо перевести изображение из цветового пространства RGB в такое цветовое пространство, где яркость представлена отдельным каналом (YCbCr, Lab, Luv, HLS, YUV, HSV) и производить яркостные преобразования только с каналом яркости, т.к. применение процедуры яркостных преобразовании к каждому каналу, может привести к искажению цвета на изображении.

Для дальнейшего выполнения 1 части лабораторной работы необходимо найти в свободном доступе 4 цветных изображения:

1. Излишне светлое (переэкспонированное) изображение (найти, используя поиск по картинкам в поисковых системах Яндекс или Google по запросу “пересвеченные изображения” или “ переэкспонированные изображения”)

Например:



1. Излишне темное (недоэкспонированное) изображение (найти, используя поиск по картинкам в поисковых системах Яндекс или Google по запросу “недоэкспонированные изображения”)

Например:



1. Изображение с низкой контрастностью. (Взять обычное изображение (без дефектов) и понизить контрастность искусственно (например, с помощью любого онлайн сервиса по изменению контрастности или графического редактора)), либо аналогично найти в сети интернет.

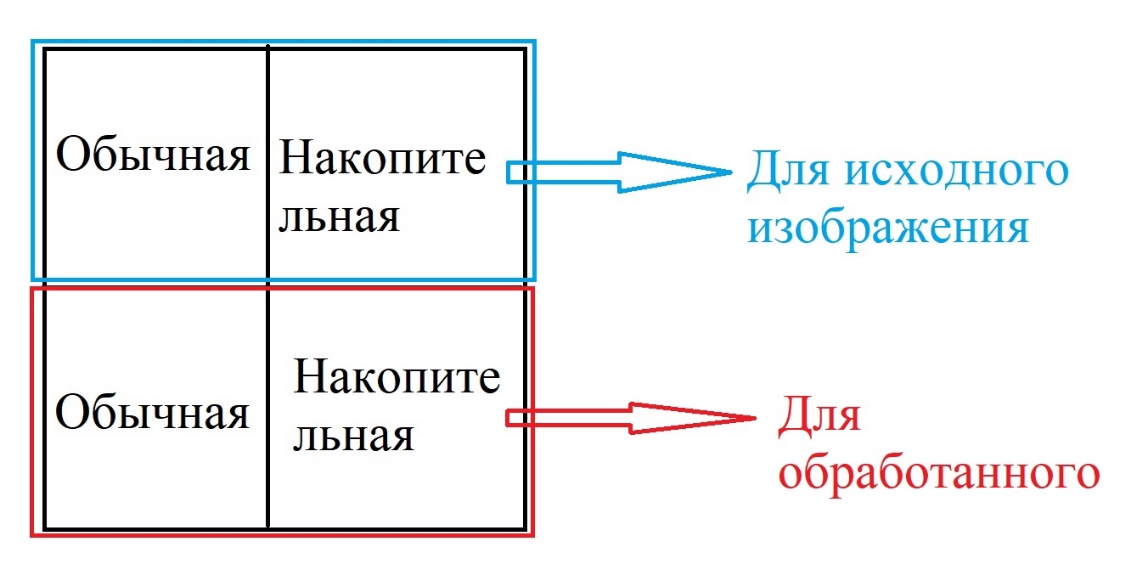
Например:



1. Обычное изображение (без значительных дефектов яркости и контрастности).

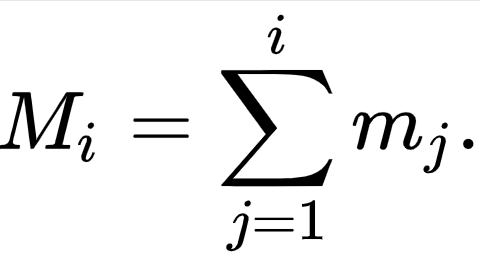
**ПРИМЕЧАНИЕ 3!!!**

Для заданий 2,3,4 приводить изображения накопительной и обычной гистограмм в отчете по следующему шаблону (для удобства сравнения):



Также рядом приводить свое исходное изображение и преобразованное!

**Для выполнения заданий 2,3,4 необходимо научится строить средствами языка python** кумулятивную гистограмму (Накопительная гистограмма) - это отображение, которое подсчитывает совокупное количество наблюдений во всех интервалах до указанного интервала. То есть кумулятивная гистограмма M по обычной гистограмме m определяется как:



**!Пример! построения такой гистограммы по ссылке:**

(<https://russianblogs.com/article/7880503771/>)

**Задание 2. Линейное преобразование**

При линейном контрастировании используется линейное поэлементное преобразование вида:

g(n,m) = af(n, m) + b

Параметры преобразования a и b определяются желаемыми значениями

минимальной gmin и максимальной gmax выходной яркости. Решив

систему уравнений:

gmin= afmin+ b

gmax= afmax+ b

относительно параметров преобразования a и b, получим

a=(gmax-gmin)/(fmax-fmin)

a=(gminfmax-gmaxfmin)/(fmax-fmin)

Можно привести к виду:

g=((f-fmin)/(fmax-fmin))(gmax-gmin)+gmin

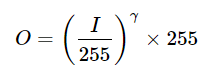
Для всех 4-х изображений:

1. Выполнить Линейное преобразование (Другие названия - “Линейное контрастирование”, “Линейная коррекция яркости”, “Коррекция контрастности изображения линейной растяжкой гистограммы”), для равномерного растягивания на весь максимальный интервал уровней яркости от 0 до 255, согласно информации, приведенной в лекционном и, приложенном, теоретическом материале (не используя специализированные функции и библиотеки);
2. Вывести исходные и преобразованные изображения;
3. Построить и вывести нормированную и кумулятивную (построенную по нормированной) гистограммы исходных и преобразованных изображений.
4. Оценить полученные результаты. Сделать выводы.

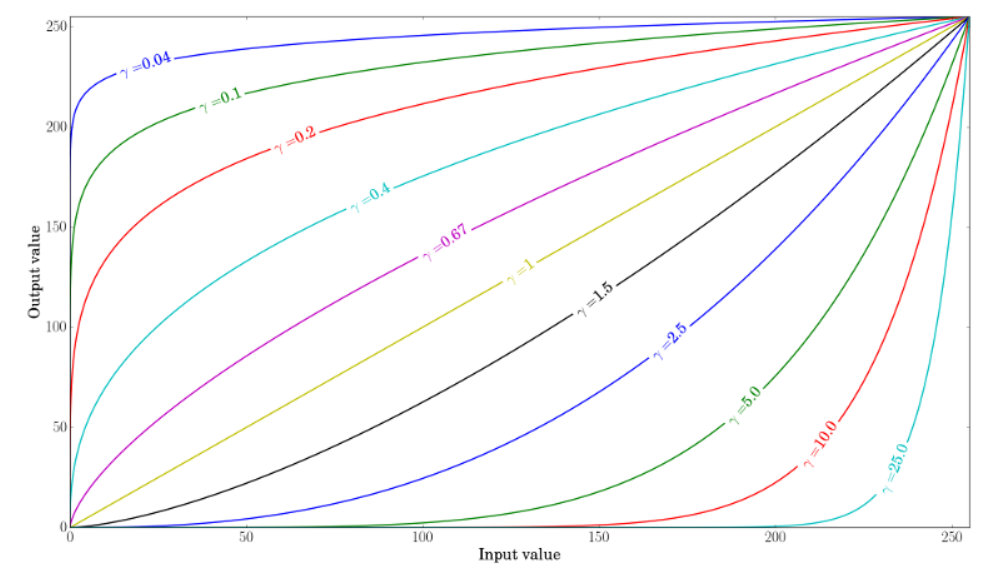
**Задание 3. Степенные преобразования.** К степенным преобразованиям яркости относятся преобразования вида:

g(n, m) = c(f(n, m) + f0)γ.

Одним из наиболее часто применяемых в обработке изображений степенных преобразований является преобразование, называемое гамма- коррекцией. Гамма коррекция может использоваться для коррекции яркости изображения с помощью нелинейного преобразования между входными значениями и отображенными выходными значениями. Поскольку это соотношение нелинейно, эффект не будет одинаковым для всех пикселей и будет зависеть от их исходного значения.



Когда γ<1, исходные темные области будут ярче, а гистограмма будет сдвинута вправо, в то время как с γ>1 наоборот.



Для всех 4-х изображений:

1. Выполнить Гамма-коррекцию, согласно информации, приведенной в лекционном и, приложенном, теоретическом материале (не используя специализированные функции и библиотеки), подобрав эмпирическим путем одно наилучшее (с точки зрения визуального качества изображения) значение гамма для каждого из изображений;
2. Вывести исходные и преобразованные изображения;
3. Построить и вывести нормированную и кумулятивную (построенную по нормированной)) гистограммы исходных и преобразованных изображений.
4. Оценить полученные результаты. Сделать выводы.

**Задание 4. Выравнивание гистограммы**

Для всех 4-х изображений:

*!!!Все еще не забываем про примечание 2!!!*

1. Выполнить операцию выравнивания гистограммы, согласно информации, приведенной в лекционном и, приложенном, теоретическом материале (не используя специализированные функции и библиотеки);
2. Выполнить пункт 1, но используя функцию equalizeHist() библиотеки OpenCV. Сравнить с результатами выполнения первого пункта;
3. Произвести адаптивное выравнивание гистограммы c помощью библиотеки OpenCV(понадобятся функции createCLAHE(), clahe.apply()). Ознакомиться с методом, который используется;
4. Вывести исходные и преобразованные изображения;
5. Построить и вывести нормированную и кумулятивную (построенную по нормированной) гистограммы исходных и для всех преобразованных изображений.
6. Оценить полученные результаты. Сделать выводы.
7. **Пространственная фильтрация**

**Перед выполнением данной части лабораторной работы ознакомьтесь с приложенными теоретическими сведениями из папки – “Пространственная фильтрация” и по ссылкам:**

* + - 1. https://spec-zone.ru/RU/iOS/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html
      2. https://robocraft.ru/blog/computervision/427.html
      3. https://en.wikipedia.org/wiki/Kernel\_(image\_processing)
      4. https://russianblogs.com/article/140722179/
      5. https://russianblogs.com/article/77801672445/
      6. https://russianblogs.com/article/6194867611/
      7. https://desktop.arcgis.com/ru/arcmap/10.3/manage-data/raster-and-images/convolution-function.htm
      8. https://annimon.com/article/3663
      9. <https://habr.com/ru/post/142818/>
      10. https://setosa.io/ev/image-kernels/

***ПРИМЕЧАНИЕ 4!!!***

Для выполнения заданий части 2, понадобится одно обычное цветное изображение.

**Задание 5. Пространственная фильтрация**

1. Написать функцию реализующую двумерную свертку изображения с ядром фильтра. Функция должна принимать исходное изображение, ядро фильтра, дополнительную информацию (при необходимости) и возвращать обработанное изображение.
2. Обработать изображение, приведенными ниже в таблице 1, фильтрами, используя написанную функцию. В каждой группе фильтров (**I-IV**) выбрать фильтр по номеру в группе/подгруппе). Контроль правильности реализации свертки осуществлять путем сравнения результатов с результатами функции cvFilter2D() библиотеки OpenCV(они должны полностью совпадать). Для некоторых фильтров в библиотеке OpenCV имеются отдельные функции (третий столбец таблицы 1). Для таких действий, также сравнить результат с результатом выполнения этих функций. Для фильтров обнаружения краев (Группа III в таблице 1), изображение преобразовать в одноканальное в оттенках серого и применить фильтр к нему. Для остальных применять фильтр отдельно для каждого из трех каналов цветного изображения.

**Обязательно отфильтрованные значения заносить в новое пустое изображение, а не изменять исходное!!!**

1. Вывести на экран исходные и обработанные изображения. Для обработанного изображения в отчет для каждого фильтра вставлять только одно изображение(преобразованное собственноручно написанной функцией)
2. Оценить полученные результаты. Сделать выводы.

Таблица1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Группа фильтров | Название | № | Ядро | Функция OpenCV |
| **I**  **Размытие** | *Усредненное размытие* | 1 | 3x3 | Blur() |
| 2 | 5x5 |
| *Размытие по Гауссу* | 3 | 3x3 | GaussianBlur() |
| 4 | 5x5 |
| **II Резкость** | *Резкость* | 5 |  |  |
| 6 |  |
| 7 |  |
| **III**  **Выделение границ** | *Оператор Собеля* | 8 | Обнаружение края в направлении X | Sobel() |
| 9 | Обнаружение края в направлении Y: |
| *Оператор Лапласа* | 10 | Фильтр 3x3 | Laplacian() |
| 11 |  |
| *Оператор Прюитт* | 12 | Обнаружение края в направлении X |  |
| 13 | Обнаружение края в направлении Y: |
| **IV** | *Размытие в движении*  *(! В этом ядре не хватает множителя перед матрицей! Исправить самостоятельно!)* | 14 |  |  |

**Задание 6. Пространственная фильтрация. Медианный фильтр**

1. Написать функцию реализующую медианную фильтрацию изображения (документ 2.pdf, 6.1.2.2 Медианный фильтр, cтр 7(68), в приложенных теоретических сведениях). Функция должна, как минимум работать с окнами 3x3, 5x5 или быть универсальной.
2. Обработать изображение с помощью написанной функции. Контроль правильности реализации свертки осуществить путем сравнения результатов с результатами функции medianBlur() библиотеки OpenCV(они должны полностью совпадать). **Обязательно отфильтрованные значения заносить в новое пустое изображение, а не изменять исходное!!!**
3. Вывести на экран исходное и обработанное изображения.
4. Оценить полученные результаты. Сделать выводы.

Оформите отчет по проделанной работе!