Практическая работа №2

Фильтрация изображений с использованием свертки в Python

Техническая документация: https://pythonexamples-org.cdn.ampproject.org/c/s/pythonexamples.org/python-opencv-image-filter-convolution-cv2-filter2d.amp/

В этой практической работе мы рассмотрим способы фильтрации изображений с помощью 2D-свертки, представленной в виде функции cv2.filter2D() в библиотеке OpenCV.

Свертка происходит между исходным изображением и ядром. Ядро – это еще один массив, который обычно меньше исходного изображения и определяет действие фильтрации. Ядро может быть высокочастотным, низкочастотным или настраиваемым, которое может обнаруживать определенные функции в изображении.

Фильтр низких частот больше похож на процесс усреднения, но с весами и размахом усреднения в зависимости от формы и содержимого ядра.

Фильтр высоких частот похож на детектор границ. Он дает высокий уровень при значительном изменении значений соседних пикселей.

Кроме того, вы можете использовать настраиваемый фильтр для обнаружения кругов, квадратов или некоторых нестандартных форм, которые вы хотите обнаружить на изображении.

Матрица свёртки – это матрица коэффициентов, которая «умножается» на значение пикселей изображения для получения требуемого результата.

Установка библиотек

Открываем среду разработки PyCharm, создаём новый проект (в качестве основного интерпретатора в проекте выбираем Python 3.7, как в предыдущей работе), затем открываем терминал и запускаем в нём последовательно следующий код для установки необходимых библиотек:

1.

```
pip install numpy
```

2.

```
pip install opencv-python
```

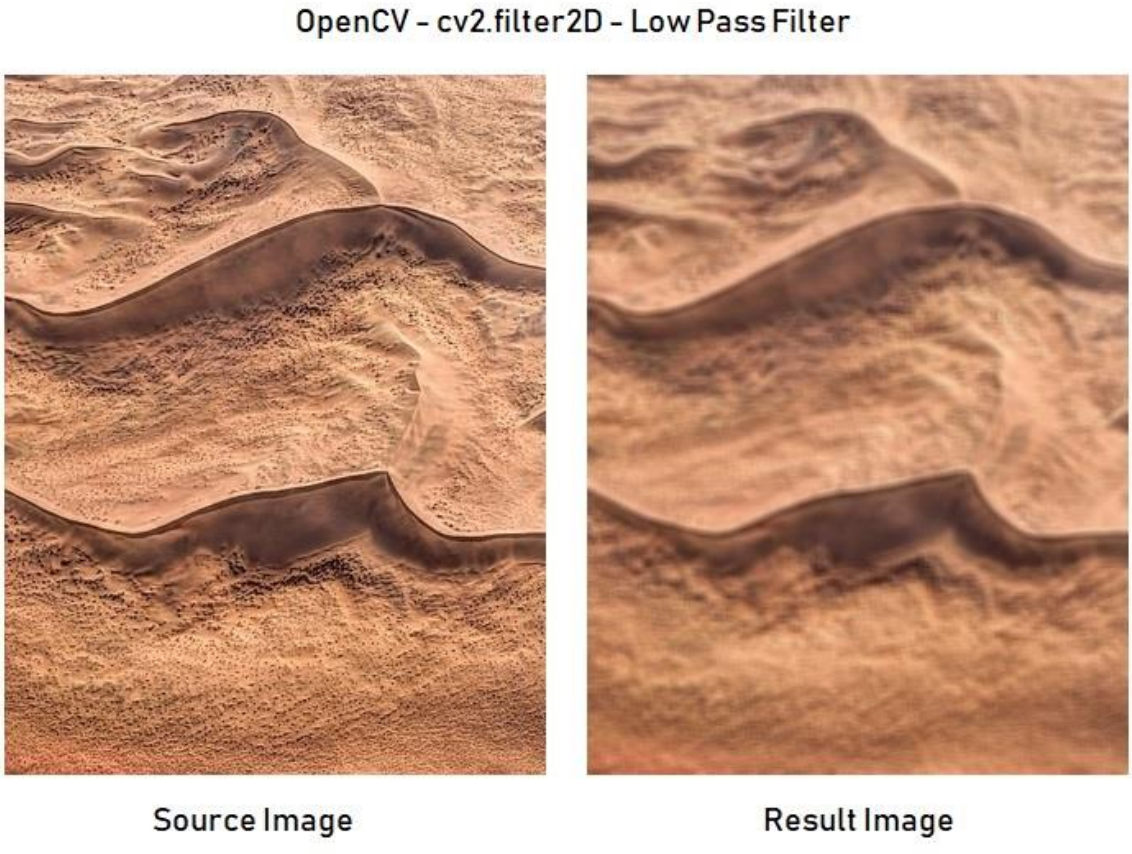
Затем, переносим растры в папку с проектом.

Применение фильтра нижних частот

В данном примере наш фильтр нижних частот представляет собой массив 5 × 5 со всеми единицами и усреднением.

Примените свертку между исходным изображением и ядром с помощью функции cv2.filter2D().

```
import numpy as np
import cv2

# чтение изображения
img_src = cv2.imread('sample.jpg') # вместо 'sample.jpg' введите имя вашего
растра

# подготовка фильтра в виде матрицы 5x5
kernel = np.array([[1, 1, 1, 1, 1],
                   [1, 1, 1, 1, 1],
                   [1, 1, 1, 1, 1],
                   [1, 1, 1, 1, 1],
                   [1, 1, 1, 1, 1]])
kernel = kernel/(np.sum(kernel) if np.sum(kernel)!=0 else 1) # нормализация
ядра

#фильтрация исходного изображения
img_rst = cv2.filter2D(img_src,-1,kernel)

#сохранение конечного изображения
cv2.imwrite('result.jpg',img_rst)
```

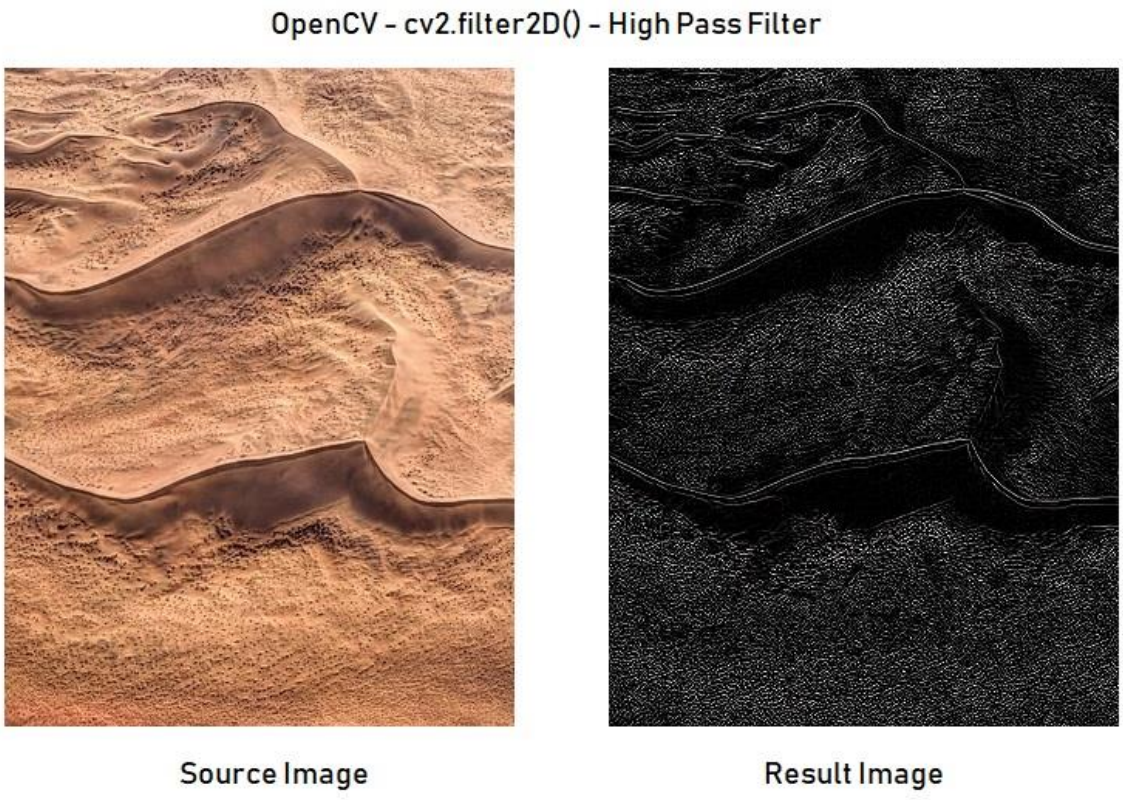


Применение фильтра высоких частот

В этом примере фильтр верхних частот представляет собой массив 3 × 3, который является переменной ядра в приведенной ниже программе.

Примените свертку между исходным изображением и ядром с помощью функции cv2.filter2D().

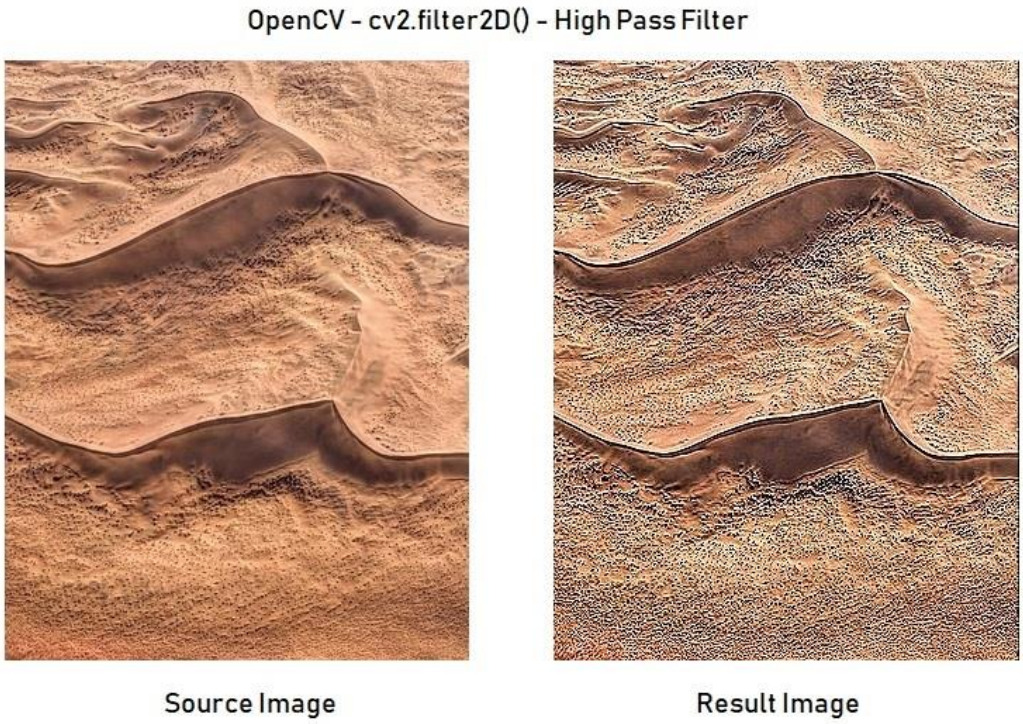
```
import numpy as np
import cv2

# чтение изображения
img_src = cv2.imread('sample.jpg')

# подготовка фильтра в виде матрицы 3x3
kernel = np.array([[0.0, -1.0, 0.0],
                   [-1.0, 4.0, -1.0],
                   [0.0, -1.0, 0.0]])

kernel = kernel/(np.sum(kernel) if np.sum(kernel)!=0 else 1)

#фильтрация исходного изображения
img_rst = cv2.filter2D(img_src,-1,kernel)

#сохранение конечного изображения
cv2.imwrite('result.jpg',img_rst)
```



Выходное изображение выглядит так, как будто вся зернистая информация сохранена, а остальная часть исчезла.

Если вы измените массив ядра на следующий, информация о цвете будет сохранена с выделенными высокочастотными пиксельными областями:

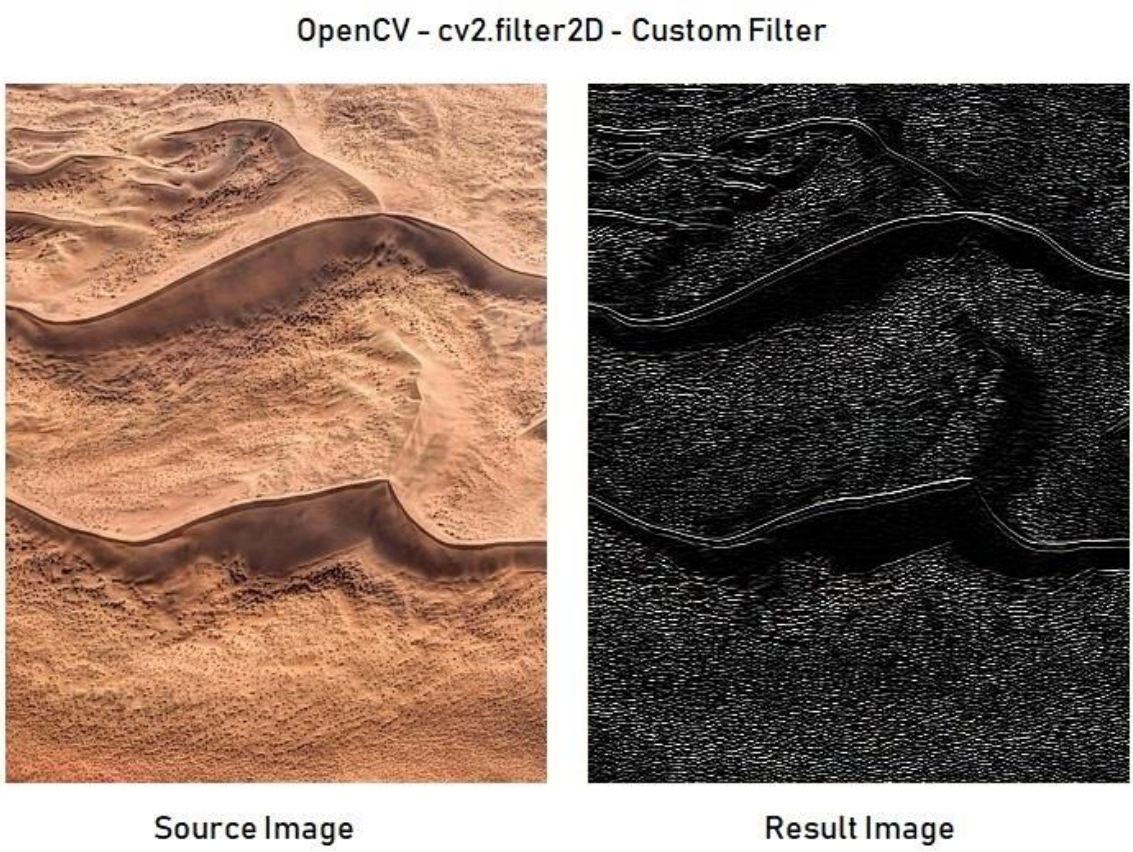
```
kernel = np.array([[0.0, -1.0, 0.0],
                   [-1.0, 5.0, -1.0],
                   [0.0, -1.0, 0.0]])
```



Применение настраиваемого фильтра

В этом примере мы определим собственное ядро для более чёткого выявления горизонтальных линий на растре.

```
import numpy as np
import cv2

# чтение изображения
img_src = cv2.imread('sample.jpg')

# подготовка фильтра в виде матрицы 3x2
kernel = np.array([[-1.0, -1.0],
                   [2.0, 2.0],
                   [-1.0, -1.0]])

kernel = kernel/(np.sum(kernel) if np.sum(kernel)!=0 else 1)

#фильтрация исходного изображения
img_rst = cv2.filter2D(img_src,-1,kernel)

#сохранение конечного изображения
cv2.imwrite('result.jpg',img_rst)
```



Задачи работы:

1. Применить ко всем растрам из папки «Изображения» фильтры высоких и низких частот. Сохранить обработанные изображения

2. Сравнить полученные изображения с исходными и составить описание (до-после). Указать, какие классы объектов стали наиболее ярко выражены. Сделать общий вывод о том, для каких целей лучше всего использовать фильтры высоких и низких частот

3. Определить свои параметры ядра (задать матрицу со своими параметрами) и указать, для каких целей его можно использовать. Привести примеры. Изображения, для обработки по Вашим параметрам можно выбирать любые, по желанию.

4. Оформить всё вышеописанное в виде отчета и отправить на проверку по электронной почте на адрес: a_streltsov@edu.miigaik.ru