**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

Лабораторная работа №5

**Случайные процессы в обработке изображений**

Вариант 2

**Выполнили:**

Крученков Евгений Андреевич

Кендысь Алексей Максимович,

студенты 4 курса, 7 группы,

специальность

“прикладная математика”

**Преподаватель:**

Старший преподаватель

кафедры ТВиМС ФПМИ,

Л.А. Хаткевич

Минск, 2023

**Выполнение задания**

**Подключение библиотек и задание параметров.**

import cv2

import numpy as np

import matplotlib.pyplot as plt

Вывод изображений

image1 = cv2.imread('image1.png')

image2 = cv2.imread('image2.png')

image\_show1 = cv2.cvtColor(image1, cv2.COLOR\_BGR2RGB)

image\_show2 = cv2.cvtColor(image2, cv2.COLOR\_BGR2RGB)

fig, ax = plt.subplots(figsize=(6, 6), layout="tight")

show\_image(ax, image\_show1, title="Image 1", cmap="gray")

fig, ax = plt.subplots(figsize=(6, 6), layout="tight")

show\_image(ax, image\_show2, title="Image 2", cmap="gray")





Удаление шума

Функция fastNlMeansDenoisingColored(). Использует алгоритм нелокального шумоподавления для удаления шума на изображении.

h: третий параметр, определяющий силу фильтра. Более высокое значение h лучше удаляет шум, но также удаляет детали изображения.

noiseless\_image1 = cv2.fastNlMeansDenoisingColored(image1,None,5)

noiseless\_image2 = cv2.fastNlMeansDenoisingColored(image2,None,4)

image\_show1 = cv2.cvtColor(noiseless\_image1, cv2.COLOR\_BGR2RGB)

image\_show2 = cv2.cvtColor(noiseless\_image2, cv2.COLOR\_BGR2RGB)

fig, ax = plt.subplots(figsize=(6, 6), layout="tight")

show\_image(ax, image\_show1, title="Image 1 without noise", cmap="gray")

fig, ax = plt.subplots(figsize=(6, 6), layout="tight")

show\_image(ax, image\_show2, title="Image 2 without noise", cmap="gray")





Применение фильтра (повышение резкости)

Функция filter2D() делает свертку изображения с ядром, чтобы применить фильтр к изображению. Второй параметр отвечает за глубину изображения, в случае -1 результатирующее изображение не меняет свою глубину. Третий параметр - ядро, которое требуется применить. В качестве ядра используем лапласиан для повышения резкости.

kernel = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])

sharpened\_image1 = cv2.filter2D(noiseless\_image1, -1, kernel)

sharpened\_image2 = cv2.filter2D(noiseless\_image2, -1, kernel)

image\_show1 = cv2.cvtColor(sharpened\_image1, cv2.COLOR\_BGR2RGB)

image\_show2 = cv2.cvtColor(sharpened\_image2, cv2.COLOR\_BGR2RGB)

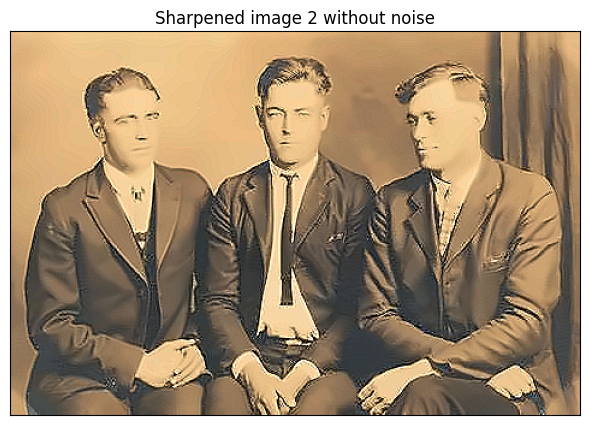
fig, ax = plt.subplots(figsize=(6, 6), layout="tight")

show\_image(ax, image\_show1, title="Sharpened image 1 without noise", cmap="gray")

fig, ax = plt.subplots(figsize=(6, 6), layout="tight")

show\_image(ax, image\_show2, title="Sharpened image 2 without noise", cmap="gray")





Изменение яркости и контрастности изображения

Функция convertScaleAbs() позволяет изменить контрастность и яркость изображения. Третий параметр - коэффициент изменения контраста. Четвертый параметр – на сколько изменить яркость.

brightness = 35

contrast = 1.05

brightened\_image1 = cv2.convertScaleAbs(sharpened\_image1, None, contrast, brightness)

brightened\_image2 = cv2.convertScaleAbs(sharpened\_image2, None, contrast, brightness)

image\_show1 = cv2.cvtColor(brightened\_image1, cv2.COLOR\_BGR2RGB)

image\_show2 = cv2.cvtColor(brightened\_image2, cv2.COLOR\_BGR2RGB)

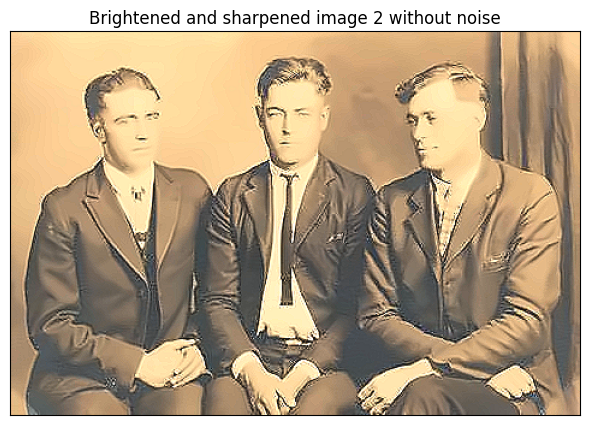
fig, ax = plt.subplots(figsize=(6, 6), layout="tight")

show\_image(ax, image\_show1, title="Brightened and sharpened image 1 without noise", cmap="gray")

fig, ax = plt.subplots(figsize=(6, 6), layout="tight")

show\_image(ax, image\_show2, title="Brightened and sharpened image 2 without noise", cmap="gray")

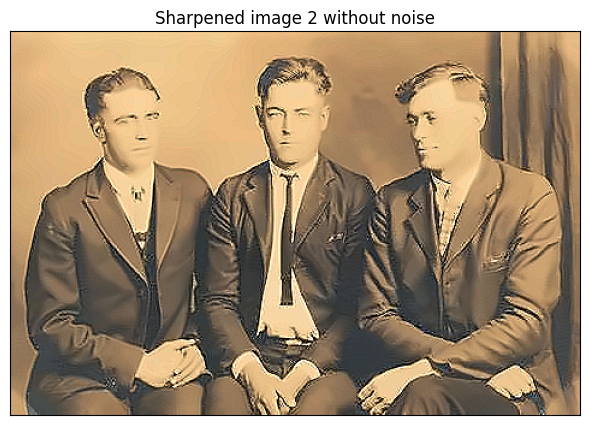


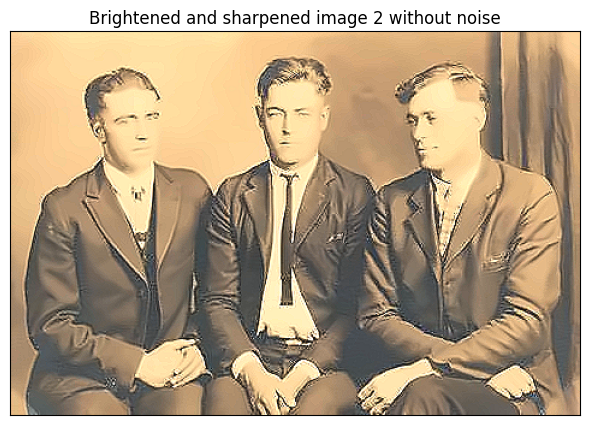


**Итоговые результаты**

****





****

**Выводы**

У нас успешно получилось понизить шум изображения, увеличить резкость изображения и изменить яркость и контраст. Для первого изображения удаление шума сработало очень хорошо, а для второго чуть хуже, т.к. немного удалились детали изображения. Резкость тоже успешно повысилась, но опять же для второго изображения функция сработала хуже, повысилась зернистость изображения. Это может быть связано с тем, что изначальное изображение было худшего качества, чем первое. Яркость и контраст хорошо сработали для обоих изображений.