**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САУ**

отчет

**по лабораторной работе №3**

**по дисциплине «Техническое зрение»**

**Тема: «Введение в OpenCV»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6491 |  | Бабурин А. Д. |
| Преподаватель |  | Моклева К.А. |

Санкт-Петербург

2020 г.

**Цель работы**: научиться выполнять простые операции над изображениями.

**Задание**:

1. Скачайте любое полноцветное изображение. Откройте его в окне с

названием “RGB”. Откройте его же так, чтобы оно было представлено в оттенках серого. Откройте его в окне с названием “GRAYSCALE”. Сохраните рядом с исходным изображением то же изображение в оттенках серого.

1. Откройте изображение в оттенках серого. Для каждого пикселя

инвертируйте его значение (Если возможные значения - диапазон от

0 до 255, то противоположным значением для 0 является 255, для 1 -

254, для 2 - 253, и так далее).

1. Откройте полноцветное изображение. Поменяйте местами значения

красного и зеленого каналов.

**Код программы:**

**import** cv2.cv2 **as** cv2

**import** numpy

*#Чтение изображения в BGR*

img = cv2.imread('D:/LAB\_PYTHON/LAB\_3/test.jpeg', cv2.IMREAD\_COLOR)

*#Чтение изображения в оттенках серого*

img\_gr = cv2.imread('D:/LAB\_PYTHON/LAB\_3/test.jpeg', cv2.IMREAD\_GRAYSCALE)

*#Открытие изображений в окнах и запись изображения в оттенках серого*

cv2.imshow('RGB', img)

cv2.imshow('GRAYSCALE', img\_gr)

cv2.imwrite('D:/LAB\_PYTHON/LAB\_3/test\_gr.jpeg', img\_gr)

*#Инвертирование изображения в оттенках серого, вывод его на экран, и запись*

img\_gr2 = numpy.copy(img\_gr)

img\_gr2 = 255-img\_gr2

cv2.imshow('INVERT\_GRAYSCALE', img\_gr2)

cv2.imwrite('D:/LAB\_PYTHON/LAB\_3/test\_gr\_inv.jpeg',img\_gr2)

'''

Чтение изображения,

Замена местами значений красного и зеленого каналов,

Вывод результата на экран и его запись

'''

img2 = cv2.imread('D:/LAB\_PYTHON/LAB\_3/test\_RGB.jpg', cv2.IMREAD\_COLOR)

**for** i **in** range(len(img2)):

**for** j **in** range(len(img2[i])):

a = img2[i][j][1]

img2[i][j][1] = img2[i][j][2]

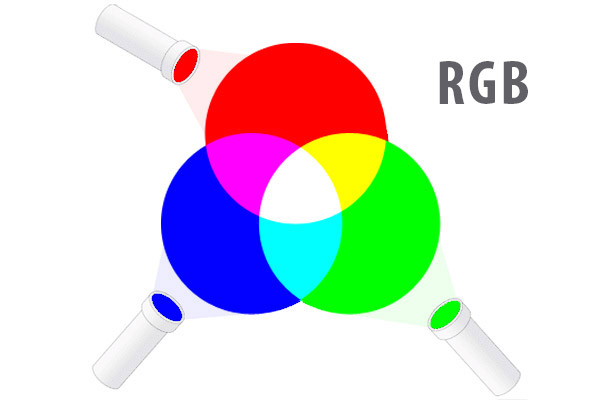
img2[i][j][2] = a

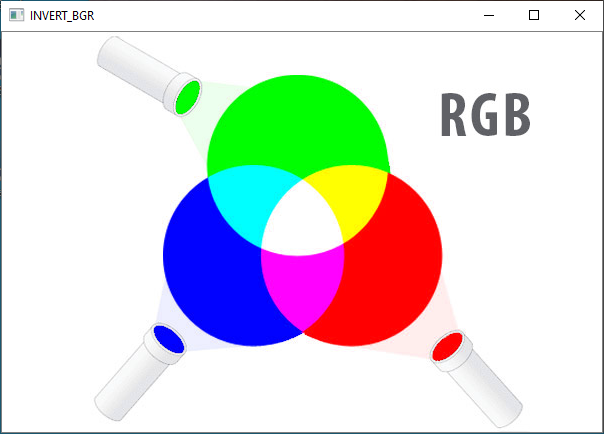
cv2.imshow('INVERT\_BGR', img2)

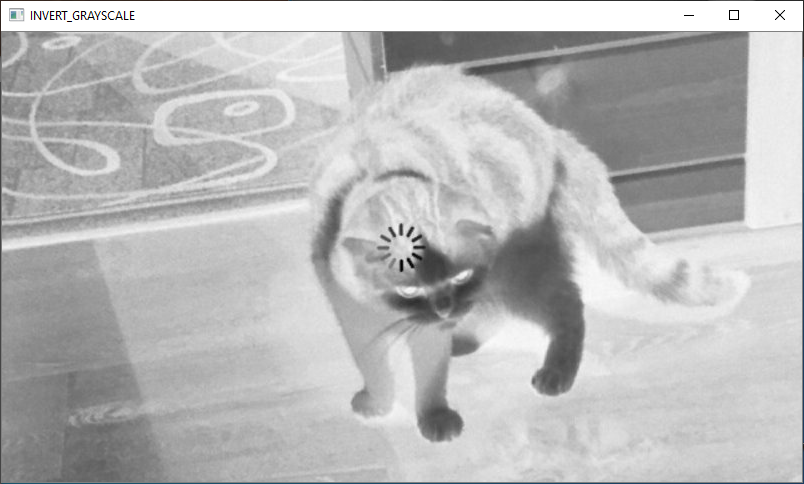
cv2.imwrite('D:/LAB\_PYTHON/LAB\_3/test\_BGR2BRG.jpg',img2)

cv2.waitKey(-1)

cv2.destroyAllWindows()

**Исходные изображения:**

**Результат выполнения программы:**



**Дополнительные задания:**

1. Создайте пустое полноцветное изображение. Нарисуйте флаг

России.

1. Создайте пустое полноцветное изображение. Проверьте, в каком

порядке записываются в ndarray красный, зеленый и синий каналы.

Проведите эксперимент: вычислите, какое сочетание значений

каналов дает различные цвета. Приведите примеры для как минимум

7 цветов.

**Код программы для дополнительных заданий:**

**import** cv2.cv2 **as** cv2

**import** numpy

*#BGR*

white = (255, 255, 255)

blue = (255, 0, 0)

red = (0, 0, 255)

*#((ч строк, ч столбцов, ч в элем), тип данных)*

IMG = numpy.zeros((450, 800, 3), dtype = numpy.uint8)

'''

Syntax: cv2.rectangle(image, start\_point, end\_point, color, thickness)

start\_point: It is the starting coordinates of rectangle.

end\_point: It is the ending coordinates of rectangle.

color: It is the color of border line of rectangle to be drawn.

thickness(толщина): It is the thickness of the rectangle border line in px.

Thickness of -1 px will fill the rectangle shape by the specified color.

'''

cv2.rectangle(IMG, (0, 0), (800, 150), white, -1 )

cv2.rectangle(IMG, (0, 150), (800, 300), blue, -1 )

cv2.rectangle(IMG, (0, 300), (800, 450), red, -1 )

cv2.imshow('Flag', IMG)

IMG\_C = numpy.zeros((700, 200, 3), dtype = numpy.uint8)

*#Проверка порядка записи каналов (B, G, R)*

cv2.rectangle(IMG\_C, (0, 0), (200, 100), (255, 0, 0), -1 )

cv2.rectangle(IMG\_C, (0, 100), (200, 200), (0, 255, 0), -1 )

cv2.rectangle(IMG\_C, (0, 200), (200, 300), (0, 0, 255), -1 )

*#silver*

cv2.rectangle(IMG\_C, (0, 300), (200, 400), (192, 192, 192), -1 )

*#yellow*

cv2.rectangle(IMG\_C, (0, 400), (200, 500), (0, 255, 255), -1 )

*#orange*

cv2.rectangle(IMG\_C, (0, 500), (200, 600), (0, 165, 255), -1 )

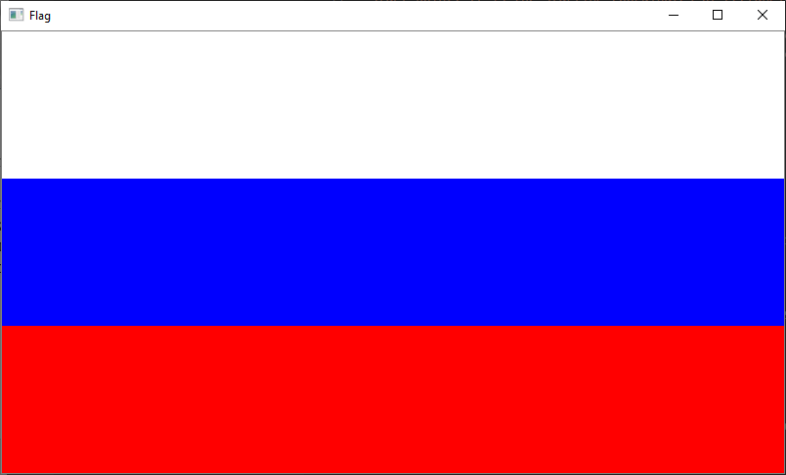
*#chocolate*

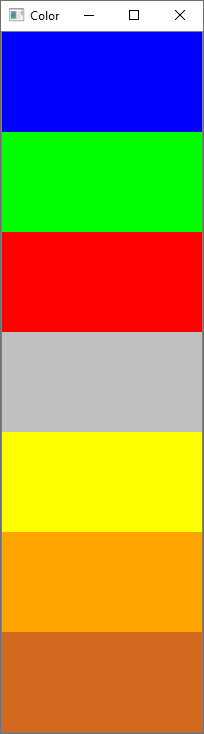
cv2.rectangle(IMG\_C, (0, 600), (200, 700), (30, 105, 210), -1 )

cv2.imshow('Color', IMG\_C)

cv2.waitKey(-1)

cv2.destroyAllWindows()

**Результат выполнения программы:**



**Вывод:** в ходе выполнения данной лабораторной работы были освоены простые операции над изображениями и при помощи документации были изучены методы библиотек NumPy и OpenCV.