

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра САУ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Техническое зрение»**

Студентка гр. 6492

\_\_\_\_\_

Огурецкий Д.В.

\_\_\_\_\_

Мурашко А.

Преподаватель

\_\_\_\_\_

Моклева К.А.

Санкт-Петербург

2019

**Задание:** научиться выполнять простые операции над изображениями

**Ход лабораторной работы:**

Версия OpenCV 3.2.0 была получена с помощью строки:

```
print(cv2.__version__)
```

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.

Исходная картинка:

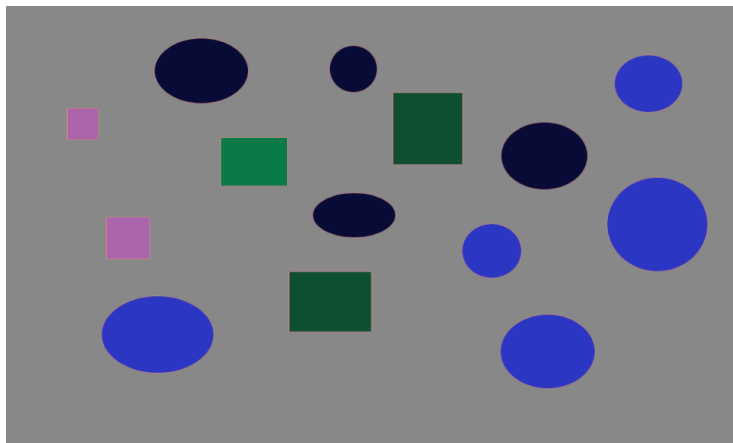


Рис.1

В оттенках серого

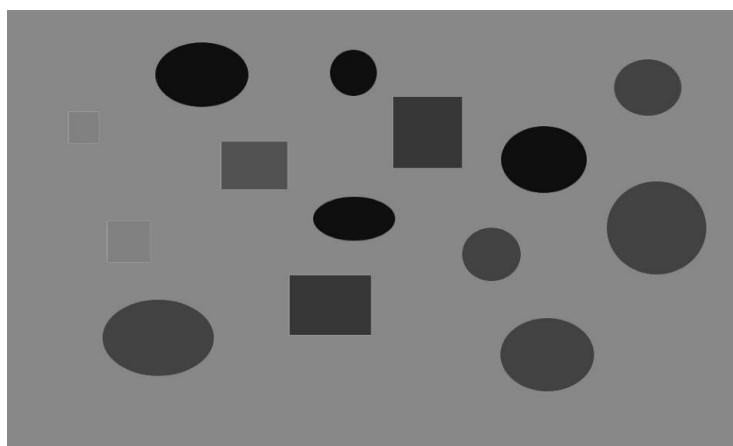


Рис.2

2. 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).

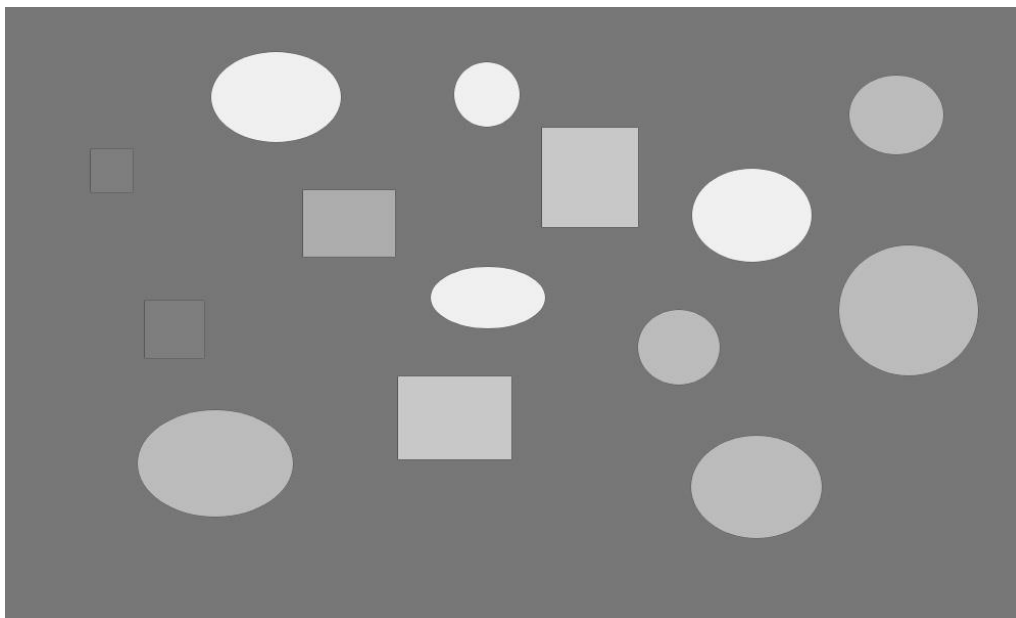


Рис.3

Инвертированная картинка записывается в файл testinv.jpg

3. Откройте полноцветное изображение. Поменяйте местами значения красного и зеленого каналов.

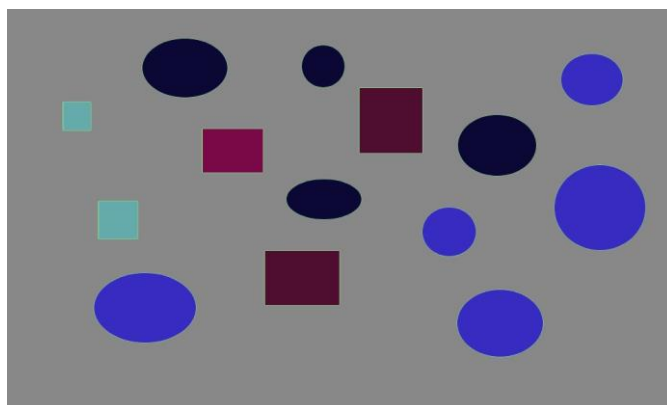


Рис.4

Измененная картинка в файле test3.jpg

Код

```
1 cv2.imshow('GRAYSCALE' , imggrey)import cv2
```

```

2 import numpy as np
3 img = cv2.imread('lab3.png', cv2.IMREAD_COLOR)
4 imggrey = cv2.imread('lab3.png', cv2.IMREAD_GRAYSCALE)
5
6 #1
7 cv2.imshow('RGB' ,img)
8 cv2.imshow('GRAYSCALE' , imggrey)
9 cv2.imwrite('testg.jpg',imggrey)
10
11 #Для каждого пикселя инвертируйте его значение
12 height, width, _ = imggrey.shape
13 for y in range(height):
14     for x in range(width):
15         imggrey[x][y] = 255 - imggrey[x][y]
16 cv2.imwrite('testinv.jpg',imggrey)
17
18 #Поменяйте местами значения красного и зеленого каналов.
19 for y in range(height):
20     for x in range(width):
21         imggrey[x][y][1],imggrey[x][y][2] = imggrey[x][y][2],imggrey[x][y][1]
22 cv2.imwrite('test3.jpg',imggrey)

cv2.waitKey(0)

```

Дополнительно:

1. Создайте пустое полноцветное изображение. Нарисуйте флаг России.

Флаг России имеет три цвета : белый, синий , красный. Если привести это в трем спискам в формате BGR, то это будет: (255,255,255),(255,0,0),(0,0,255).

Полученное изображение.



Рис.5

Код:

```

import cv2
import numpy as np

```

```

#дополнительные
#флаг РФ
RFflag = np.empty((300, 200, 3))
for y in range(300):
    for x in range(200):
        if y<100:
            RFflag[y][x] = (255,255,255)
        elif y<200:
            RFflag[y][x] = (255,0,0)
        else:
            RFflag[y][x] = (0,0,255)
cv2.imshow('Rf flag',RFflag)

```

2. 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.

При использовании функции imread в списке цвета записываются в порядке BGR согласно документации.

Порядок цветов в массиве: BGR

Сочетание значений в списке дает различные цвета. Для ознакомления с возможными цветами можно создать цветовую палитру:

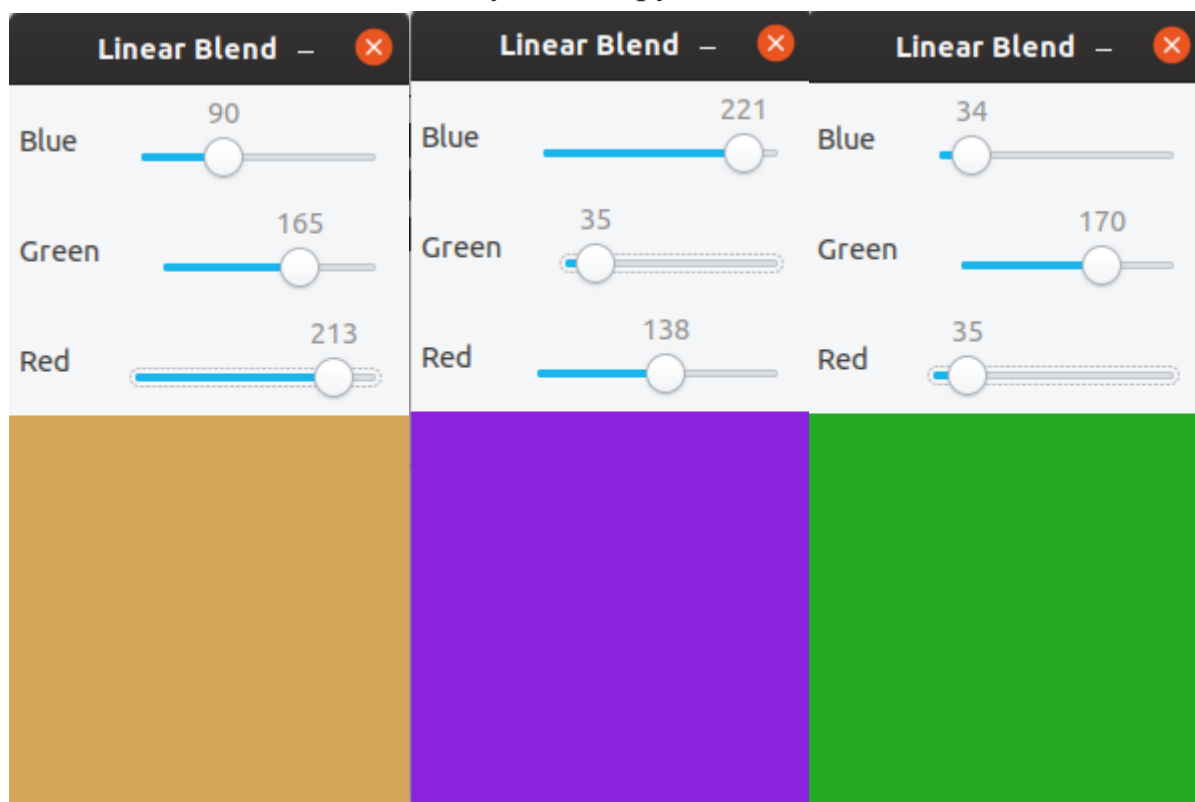


Рисунок 6

Рисунок 7

Рисунок 8

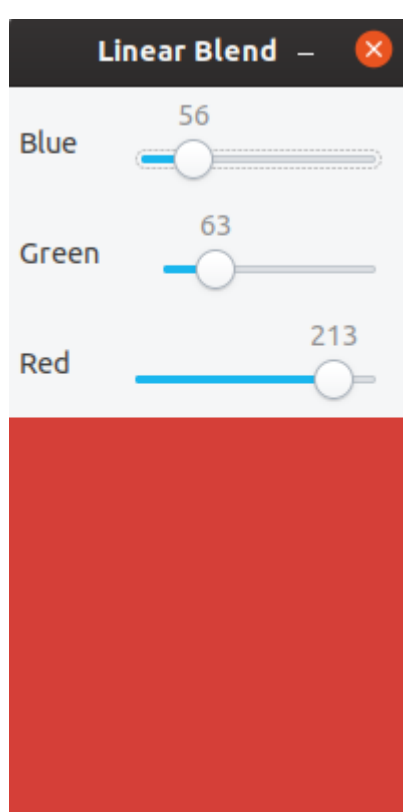


Рисунок 9

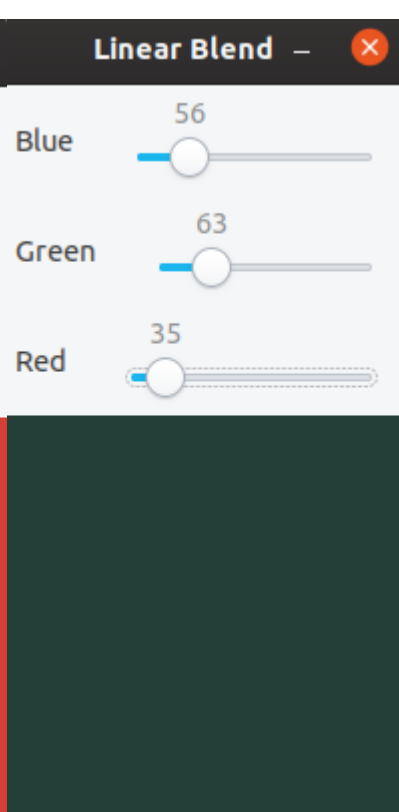


Рисунок 10

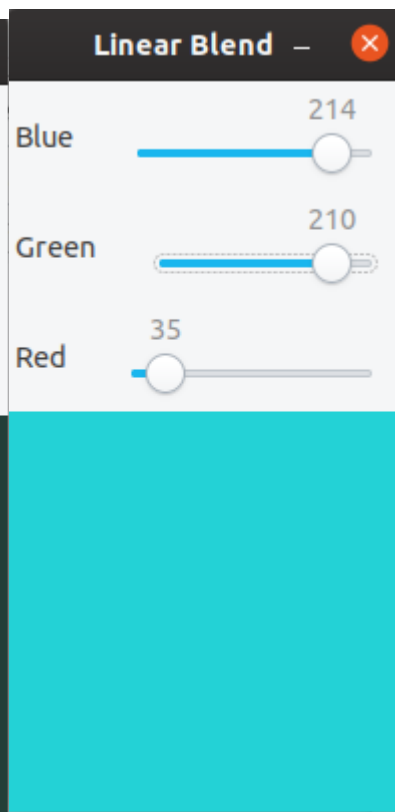


рисунок 11



Рисунок 12

Код

```
import cv2
import numpy as np
```

```

#палитра
slider_max = 255
title_window = 'Linear Blend'
height,width=200,200
img = np.empty((width,height, 3),np.uint8)
blue = 0
green=0
red=0

cv2.namedWindow(title_window, cv2.WINDOW_AUTOSIZE )

def on_trackbar_blue(blue):
    for y in range(height):
        for x in range(width):
            img[x][y][0] = blue
        cv2.imshow(title_window, img)
        trackbar_name = 'Blue'
        cv2.createTrackbar(trackbar_name, title_window, blue , slider_max, on_trackbar_blue )

def on_trackbar_green(green):
    for y in range(height):
        for x in range(width):
            img[x][y][1] = green
        cv2.imshow(title_window, img)
        trackbar_name = 'Green'
        cv2.createTrackbar(trackbar_name, title_window, green , slider_max, on_trackbar_green )

def on_trackbar_red(red):
    for y in range(height):
        for x in range(width):
            img[x][y][2] = red
        cv2.imshow(title_window, img)
        trackbar_name = 'Red'
        cv2.createTrackbar(trackbar_name, title_window, red , slider_max, on_trackbar_red )

while 1:
    if cv2.waitKey(0)==27: #esc
        break

```

**Выводы:** в ходе лабораторной работы мы ознакомились с основами работы с библиотекой `opencv`. Работали с многомерными массивами `ndarray` пакета `numpy` для научных вычислений в `python3`. Научились задавать цвета с помощью RGB модели.