**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Систем автоматического управления**

отчет

**по лабораторной работе №3**

**по дисциплине «Техническое зрение»**

Тема: **Введение в OpenCV**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6491 |  | Калинников Н.С. |
| Преподаватель |  | Моклева К. А. |

Санкт-Петербург

2020 г.

Лабораторная работа 3

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

**Основные задания:**

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.

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).

3) Откройте полноцветное изображение. Поменяйте местами значения

красного и зеленого каналов.

**Дополнительные задания:**

1) Создайте пустое полноцветное изображение. Нарисуйте флаг

России.

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.

**Код заданий:**

import cv2

import numpy as np

#Задание 1

image = cv2.imread("./technical-vision/lab3/image.png", cv2.IMREAD\_COLOR)

cv2.imshow("RGB",image)

gray\_image = cv2.imread("./technical-vision/lab3/image.png", cv2.IMREAD\_GRAYSCALE)

cv2.imshow("GRAYSCALE", gray\_image)

cv2.imwrite("./technical-vision/lab3/image\_gray.png", gray\_image)

#Задание 2

#inv\_gray\_image = cv2.bitwise\_not(gray\_image)

(hight, width, channels) = image.shape

inv\_gray\_image = np.copy(gray\_image)

for i in range(hight):

for j in range(width):

inv\_gray\_image[i,j] = 255 - gray\_image[i,j]

cv2.imshow("INVERTED GRAYSCALE", inv\_gray\_image)

cv2.imwrite("./technical-vision/lab3/inv\_image\_gray.png", inv\_gray\_image)

#Задание 3

grb\_image = np.copy(image)

G, R, B = cv2.split(image)

for i in range(hight):

for j in range(width):

grb\_image[i,j] = [R[i,j], G[i,j], B[i,j]]

cv2.imshow("Reversed Channels", grb\_image)

cv2.imwrite("./technical-vision/lab3/grb\_image.png", grb\_image)

#Доп.задание 1

h = 300

w = 300

flag = np.zeros((h,w, 3), np.uint8)

flag[0:w//3:] = (255,255,255)

flag[w//3:w-w//3:] = (255,0,0)

flag[w-w//3:w:] = (0,0,255)

cv2.imshow("FLAG", flag)

cv2.imwrite("./technical-vision/lab3/flag.png", flag)

cv2.waitKey(0)

cv2.destroyAllWindows()

Полученные изображения:



**Вывод:** в ходе выполнения работы были выполнены все основные и одно дополнительное задания. Получен опыт работы с библиотекой OpenCV.