БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра интеллектуальных информационных технологий

ОТЧЁТ  
по лабораторной работе №3  
по дисциплине

«ОБРАБОТКА ИЗОБРАЖЕНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ»

Выполнил студент гр. 321701

Романов К.В.

Проверил

Сальников Д. А.

Минск 2025

**Тема:** Програмнная реализация выравнивания яркости для двух изображений.

**Задание:**

1. Изучить алгоритм выравнивания яркости изображений.

2. Выполнить программную реализацию алгоритма выравнивания яркости изображений.

3. На вход подать 2 изображения разной яркости

**Листинг кода:**

import cv2

import numpy as np

import matplotlib.pyplot as plt

def match\_brightness(img1, img2):

"""

Выравнивание яркости img2 под img1

"""

# Конвертируем в оттенки серого

gray1 = cv2.cvtColor(img1, cv2.COLOR\_BGR2GRAY)

gray2 = cv2.cvtColor(img2, cv2.COLOR\_BGR2GRAY)

# Вычисляем среднюю яркость

mean1 = np.mean(gray1)

mean2 = np.mean(gray2)

# Коэффициент коррекции яркости

ratio = mean1 / (mean2 + 1e-8)

# Корректируем яркость

img2\_adjusted = cv2.convertScaleAbs(img2, alpha=ratio, beta=0)

return img2\_adjusted

# === Загрузка изображений === 4-2 4-1 2-1 2-3 2-4 1-2

img1 = cv2.imread("/home/kirillromanoff/University/OIIS/3/4.jpg")

img2 = cv2.imread("/home/kirillromanoff/University/OIIS/3/2.jpg")

# Проверка

if img1 is None or img2 is None:

raise FileNotFoundError("Не удалось загрузить одно из изображений.")

# === Выравнивание яркости ===

img2\_aligned = match\_brightness(img1, img2)

# === Сохранение результата ===

cv2.imwrite("image2\_aligned.jpg", img2\_aligned)

# === Конвертация для отображения в matplotlib (BGR → RGB) ===

img1\_rgb = cv2.cvtColor(img1, cv2.COLOR\_BGR2RGB)

img2\_rgb = cv2.cvtColor(img2, cv2.COLOR\_BGR2RGB)

img2\_aligned\_rgb = cv2.cvtColor(img2\_aligned, cv2.COLOR\_BGR2RGB)

# === Построение графиков ===

plt.figure(figsize=(15, 8))

# --- Исходное изображение 1 ---

plt.subplot(2, 3, 1)

plt.imshow(img1\_rgb)

plt.title("Изображение 1 (эталон)")

plt.axis('off')

# --- Исходное изображение 2 ---

plt.subplot(2, 3, 2)

plt.imshow(img2\_rgb)

plt.title("Изображение 2 (до выравнивания)")

plt.axis('off')

# --- После выравнивания ---

plt.subplot(2, 3, 3)

plt.imshow(img2\_aligned\_rgb)

plt.title("Изображение 2 (после выравнивания)")

plt.axis('off')

# --- Гистограммы яркости ---

gray1 = cv2.cvtColor(img1, cv2.COLOR\_BGR2GRAY)

gray2 = cv2.cvtColor(img2, cv2.COLOR\_BGR2GRAY)

gray2\_aligned = cv2.cvtColor(img2\_aligned, cv2.COLOR\_BGR2GRAY)

plt.subplot(2, 3, 4)

plt.hist(gray1.ravel(), bins=50, color='blue', alpha=0.7)

plt.title("Гистограмма яркости — Изображение 1")

plt.subplot(2, 3, 5)

plt.hist(gray2.ravel(), bins=50, color='red', alpha=0.7)

plt.title("Гистограмма яркости — Изображение 2 (до)")

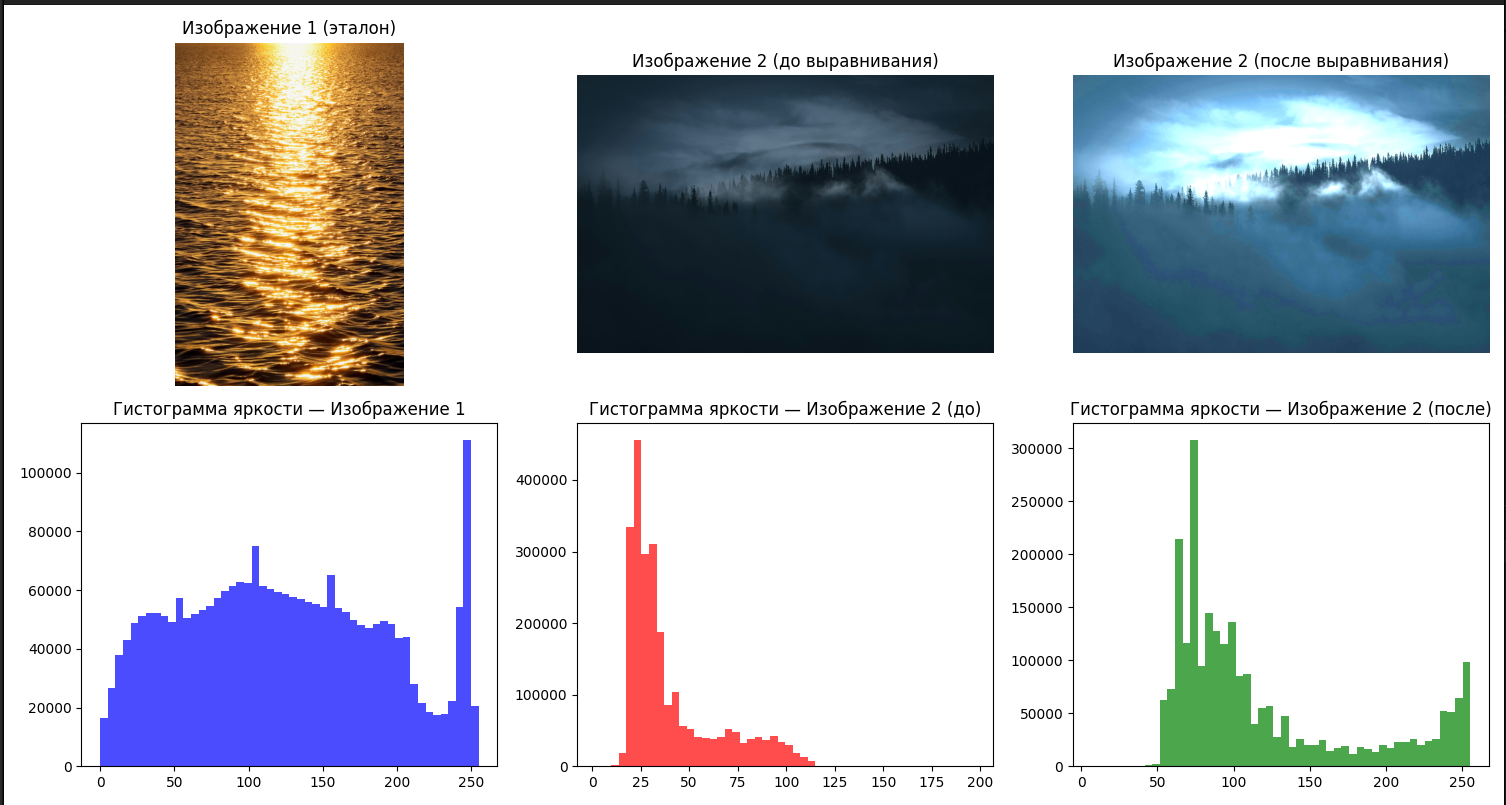
plt.subplot(2, 3, 6)

plt.hist(gray2\_aligned.ravel(), bins=50, color='green', alpha=0.7)

plt.title("Гистограмма яркости — Изображение 2 (после)")

plt.tight\_layout()

plt.show()



**Вывод:** в рамках лабораторной работы был изучен и программно реализован на алгоритм выравнивания яркости для двух изображений, .