Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

     Факультет информационных технологий и управления

     Кафедра интеллектуальных информационных технологий

ОТЧЁТ

по лабораторной работе №3

по дисциплине

ОБРАБОТКА ИЗОБРАЖЕНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

Выполнила: Рублвская Е. А.

Гр. 321702

Проверил: Сальников Д. А.

Минск 2025

**Цель:**

Реализовать программно выравнивание яркости для двух изображений.

**Ход работы:**

1. Загружаем два изображения, которые мы будем корректировать;
2. Для каждого изображения вызываем функцию, которая выполняет выравнивание яркости;
3. Сохраняем полученные результаты в переменные;
4. Выводим оригинальные изображения и измененные, чтобы сравнить результаты.

**Средства разработки:**

* OpenCV — библиотека для загрузки, обработки и преобразования изображений (выравнивание гистограммы, работа с цветовым пространством, конвертация форматов);
* Matplotlib — библиотека для отображения изображений в графическом виде и их визуального сравнения.

**Основные части кода:**

* Загрузка исходных изображений:

image1 = cv2.imread('giraffe.jpg')  
image2 = cv2.imread('night.jpg')

* Применение функции выравнивания яркости:

equalized\_image1 = equalize\_image(image1)  
equalized\_image2 = equalize\_image(image2)

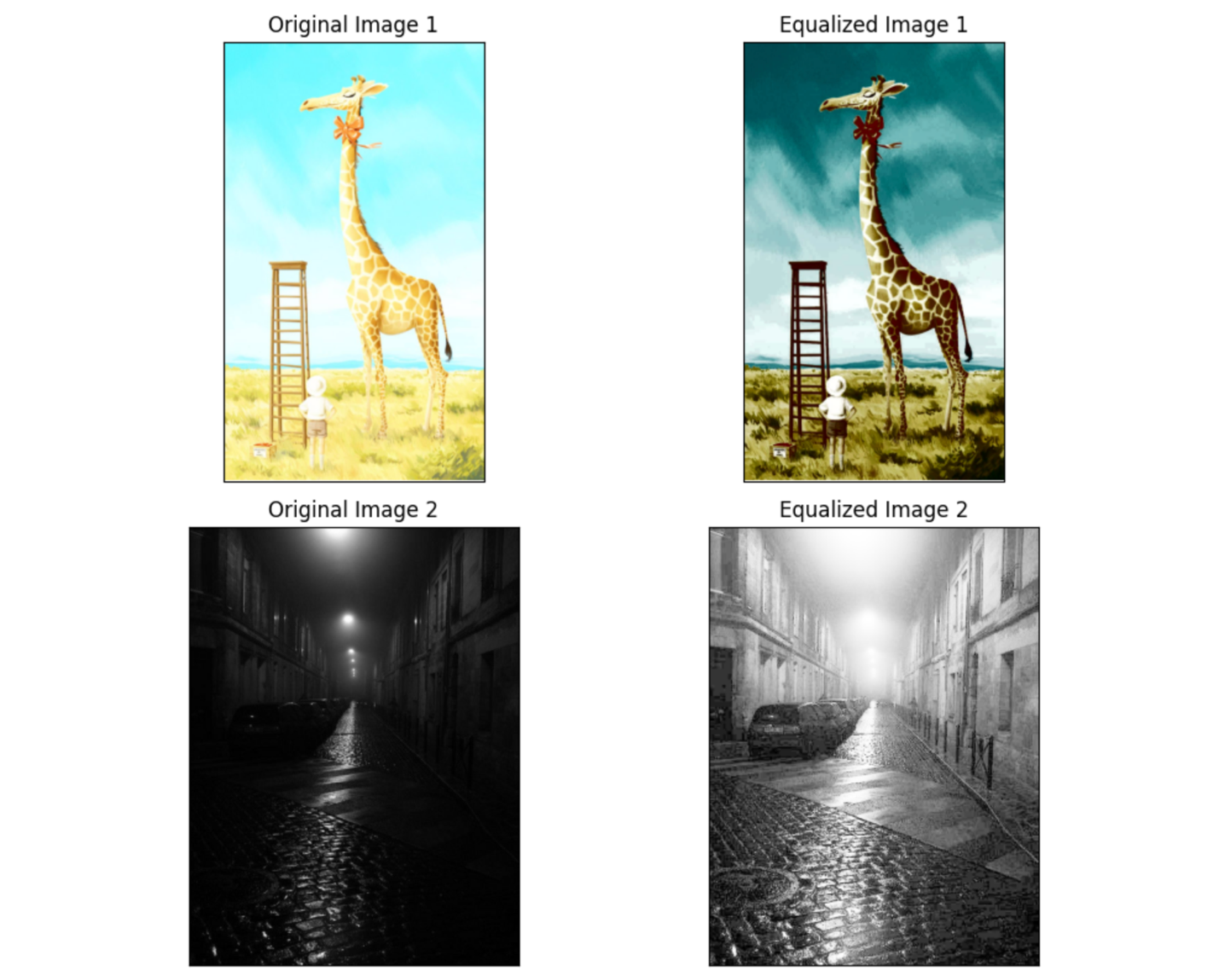
* Функция выравнивания яркости: изначально идет проверка, является ли изображение черно-белым. Если у изображения всего два измерения (ширина и высота), оно черно-белое, тогда применяется функция выравнивания гистограммы напрямую. Если изображение цветное, то оно преобразуется из BGRв формат YUV, т. к. оно состоят из яркостного канала (Y) и двух цветовых каналов (U и V). Затем применяется выравнивание гистограммы только к каналу Y, чтобы сохранить естественные цвета. А после идет преобразование обратно в BGR.

def equalize\_image(image):  
 if len(image.shape) == 2: # Если изображение черно-белое  
 return cv2.equalizeHist(image)  
 else:  
 img\_yuv = cv2.cvtColor(image**,** cv2.COLOR\_BGR2YUV)  
 img\_yuv[:**,** :**,** 0] = cv2.equalizeHist(img\_yuv[:**,** :**,** 0])  
 return cv2.cvtColor(img\_yuv**,** cv2.COLOR\_YUV2BGR)

* Графическое отображение результатов

images = [image1**,** equalized\_image1**,** image2**,** equalized\_image2]  
titles = ['Original Image 1'**,** 'Equalized Image 1'**,** 'Original Image 2'**,** 'Equalized Image 2']  
  
plt.figure(figsize=(10**,** 8))  
  
for i in range(4):  
 plt.subplot(2**,** 2**,** i + 1)  
 plt.imshow(cv2.cvtColor(images[i]**,** cv2.COLOR\_BGR2RGB))  
 plt.title(titles[i])  
 plt.xticks([])**,** plt.yticks([])  
  
plt.tight\_layout()  
plt.show()

**Результат работы программы:**

****

**Вывод:**

В ходе лабораторной работы была реализована программа для выравнивания яркостно-контрастных характеристик изображений с использованием метода выравнивания гистограммы. Этот метод перераспределяет значения яркости пикселей, что позволяет улучшить контраст изображения, особенно в случаях с недостаточной освещенностью или избыточной яркостью.

Были рассмотрены как черно-белые, так и цветные изображения, где в случае цветных изображений выравнивание применялось к каналу яркости, что позволило сохранить естественную цветовую палитру.

В результате выполнения программы мы увидели значительное улучшение визуального восприятия изображений: участки с плохой освещенностью стали более четкими, а изображения в целом — более контрастными и сбалансированными по яркости.

Таким образом, использование выравнивания гистограммы является эффективным методом для улучшения качества изображений и может применяться в различных областях, таких как обработка фотографий, медицинская визуализация, а также в системах компьютерного зрения.