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

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Основы компьютерного зрения»

Тема: Нормализация и операции над гистограммами

Доможиров Д. А _____

Коротков А. В. _____

Кравченко С. В. _____

Студенты гр. 1306 Тряпша Е. Д. _____

Преподаватель Костичев С. В. _____

Санкт-Петербург

2024

Цель работы

Цель работы: изучить применение нормализации над изображениями и базовых функций OpenCV для работы с гистограммами

Постановка задачи

1. Нормализация изображения и построение гистограмм исходного и нормализованного изображений

Выполните:

- загрузку изображения image3.jpg и его отображение
- преобразование изображения в оттенки серого и его отображение
- постройте и отобразите гистограмму серого
- нормализация типа MINMAX серого и его отображение. Выберите значения нижней и верхней границ диапазона яркостей 63 и 255 соответственно
- постройте и отобразите гистограмму нормализованного серого

При реализации используйте следующие функции:

normalize(src, dst[, alpha[, beta[, norm_type[, dtype[, mask]]]]]) -> dst

Параметры:

src - входной массив.

dst - выходной массив того же размера, что и src.

alpha - значение нормы для нормализации или нижняя граница диапазона в случае нормализации диапазона.

beta - верхняя граница диапазона при нормировке диапазона; не используется для нормализации нормы.

normType - тип нормализации (NORM_MINMAX, NORM_INF, NORM_L1, NORM_L2).

dtype - при отрицательном значении выходной массив имеет тот же тип, что и *src*;

mask - необязательная маска операции (может указывать часть входного массива для нормализации).

По дефолту *alpha=1, beta=0, norm_type=NORM_L2*

Обычно используют *norm_type=NORM_MINMAX и задают* нижнюю и верхнюю границы диапазона

calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]]) -> hist

Параметры:

images - список изображений в виде массивов numpy. Все изображения должны быть одного типа и размера.

channels - список каналов, используемых для вычисления гистограммы. Число каналов изменяется от 0 до 2 (0- blue, 1- green, 2- red).

mask - необязательная маска, показывающая, какие пиксели учитывать при вычислении гистограммы.

histSize: размеры гистограммы в каждом измерении.

ranges - массив массивов, описывающих границы интервалов гистограммы по каждому измерению.

hist - результирующая гистограмма

accumulate=false - по умолчанию этот параметр равен false. Он определяет, является ли гистограмма кумулятивной (накопительная идентификация). Позволяет накапливать простую гистограмму по нескольким изображениям или обновлять гистограмму во времени.

3. Вычисление и эквализация гистограммы (на примере простого изображения)

Выполните:

- загрузку изображения color4.jpg (956x279пикс). и его отображение. Изображение содержит 4 квадрата по 66681пикс (956x279:4) с яркостями 0,1,2 и 3.
- преобразование изображения в оттенки серого и его отображение
- постройте и отобразите гистограмму серого
- эквализация серого и отображение полученного изображения
- постройте и отобразите гистограмму изображения с эквализацией

Объясните визуальную разницу серого и результирующего изображений.

При реализации используйте функцию

equalizeHist(src[, dst]) -> dst

Параметры:

src - исходное 8-битное одноканальное изображение.

dst - целевое изображение того же размера и типа, что и src.

3. Вычисление, нормализация и эквализация гистограммы (на реальном изображении)

Выполните:

- загрузку изображения image_gray_63-192.jpg и его отображение (диапазон яркости 63-192).
- преобразование изображения в оттенки серого и его отображение
- постройте и отобразите гистограмму серого
- нормализация типа MINMAX серого и его отображение. Выберите значения нижней и верхней границ диапазона яркостей 0 и 255 соответственно
- постройте и отобразите гистограмму нормализованного изображения
- эквализация серого и отображение полученного изображения
- постройте и отобразите гистограмму изображения с эквализацией

Программное и аппаратное окружение

При выполнении лабораторной работы были использованы:

- Операционная система Windows 10
- Visual Studio Code
- Язык Python версии 3.12.3
- Библиотека opencv-python версии 4.10.0.84
- Библиотека numpy версии 2.0.0
- Библиотека matplotlib версии 3.9.2

Примеры запуска программы

Исходный код программы смотреть в приложении А.

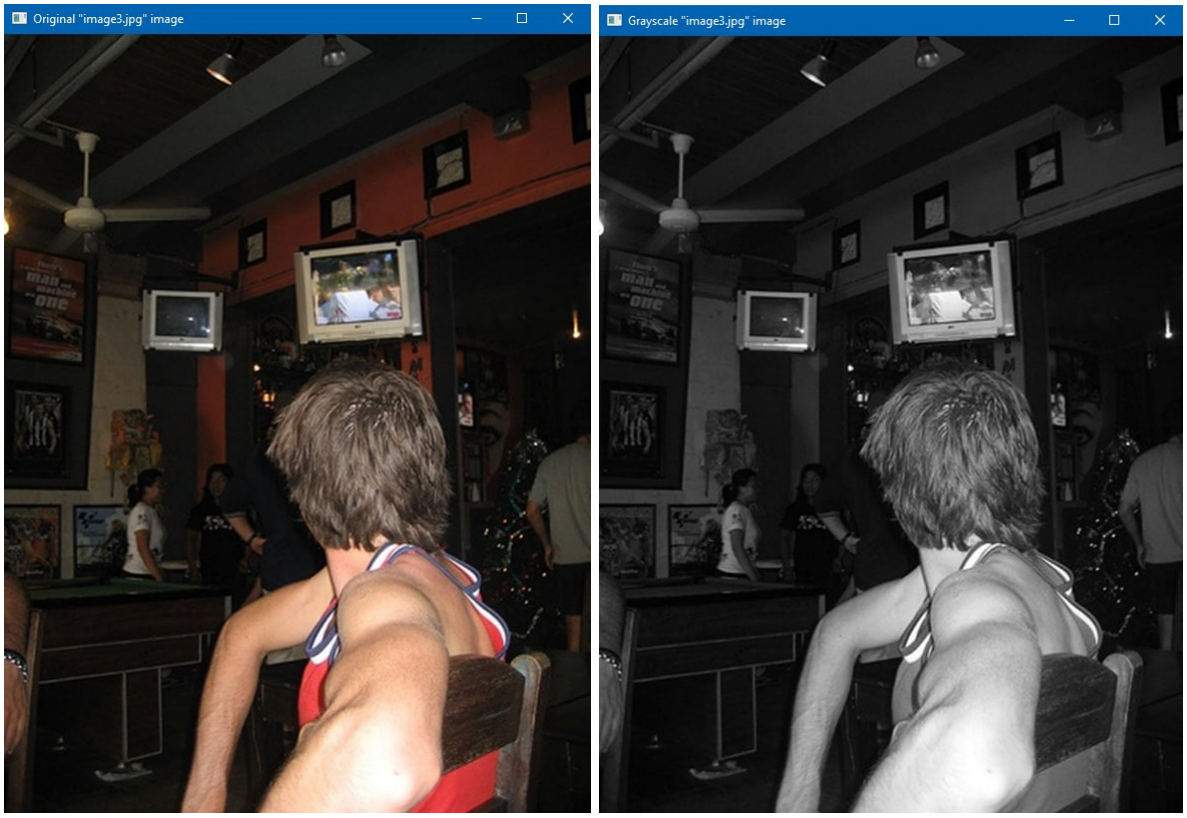

Рисунок 1 – Отображение оригинального изображения и его версия в оттенках серого (п. 1)

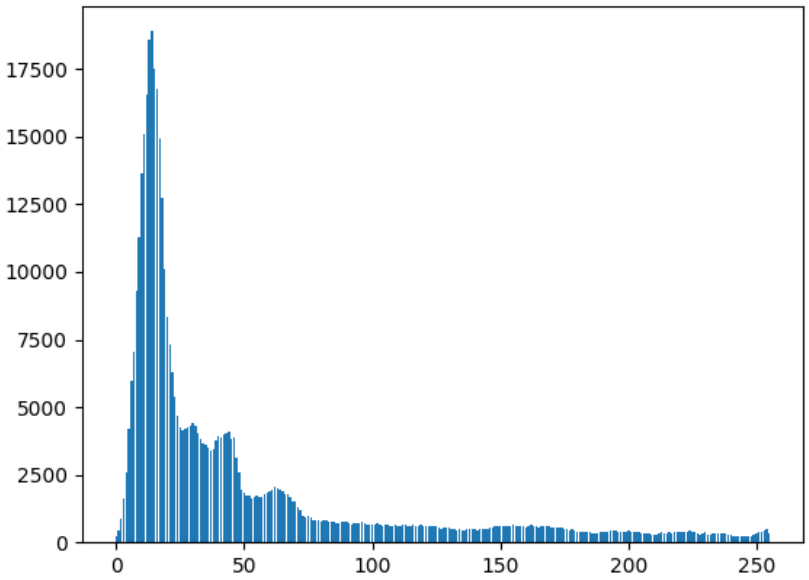

Рисунок 2 – Гистограмма серого оригинального изображения (п. 1)

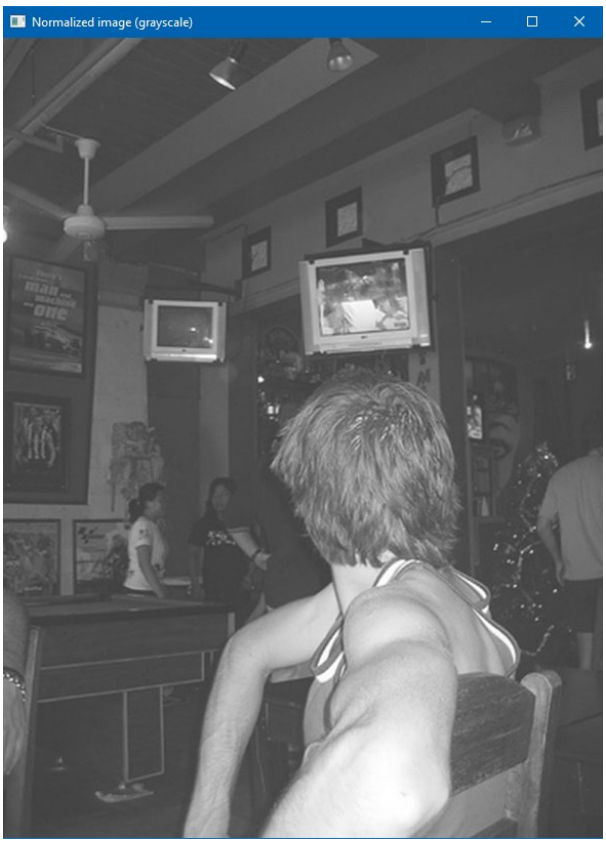

Рисунок 3 – Отображение результата MINMAX-нормализации с границами 63 и 255 (п. 1)

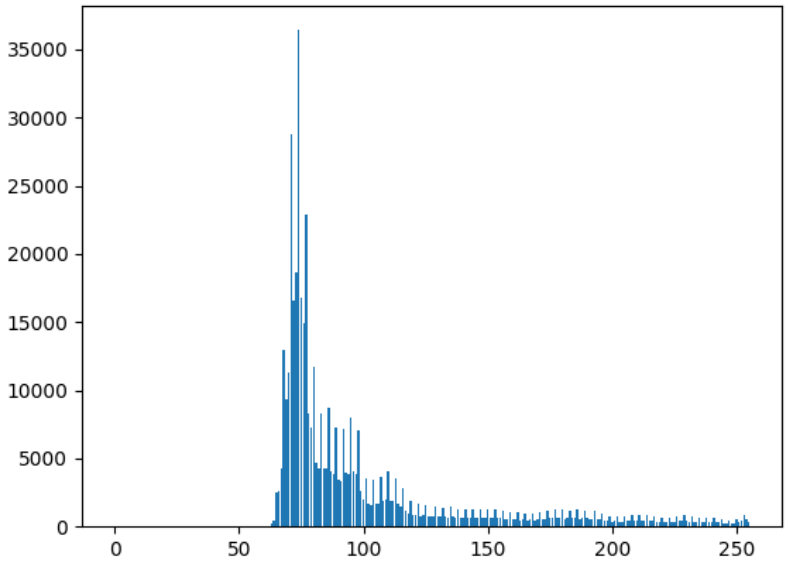

Рисунок 4 – Гистограмма серого изображения после MINMAX-нормализации (п. 1)

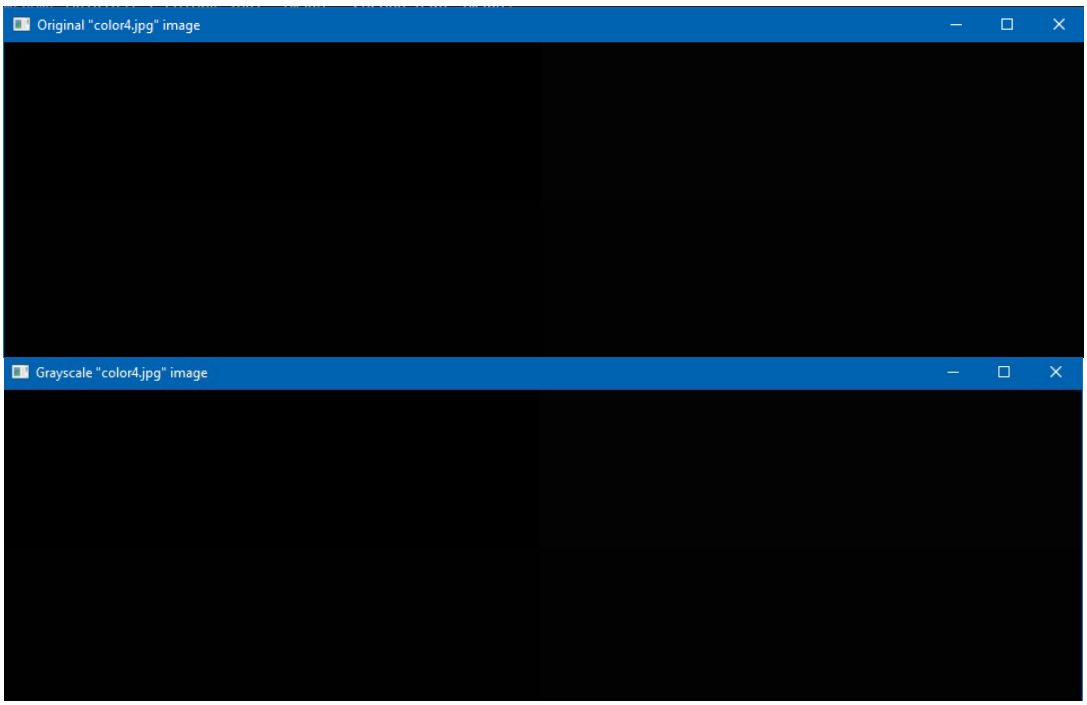

Рисунок 5 – Отображение оригинального изображения и его версия в оттенках серого (п. 2)

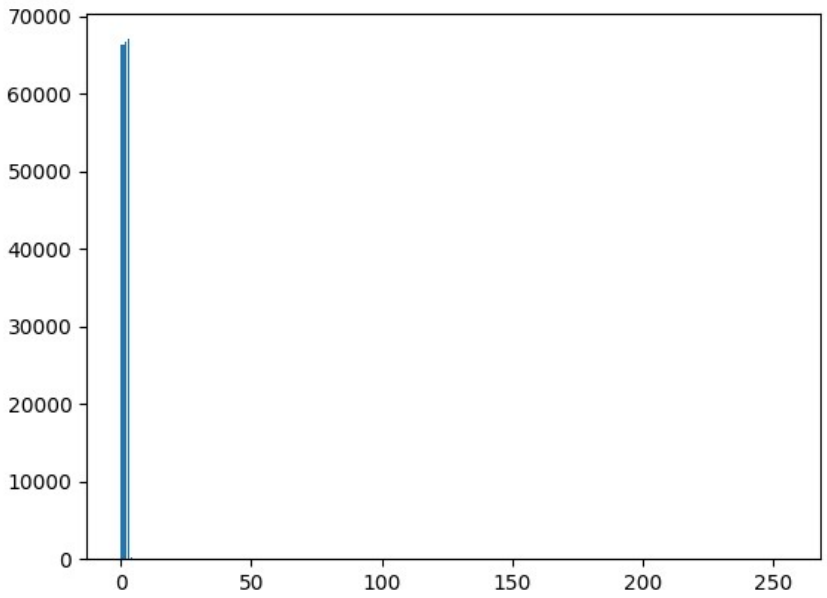

Рисунок 6 – Гистограмма серого оригинального изображения (п. 2)

Рисунок 7 – Отображение результата эквализации изображения (п. 2)

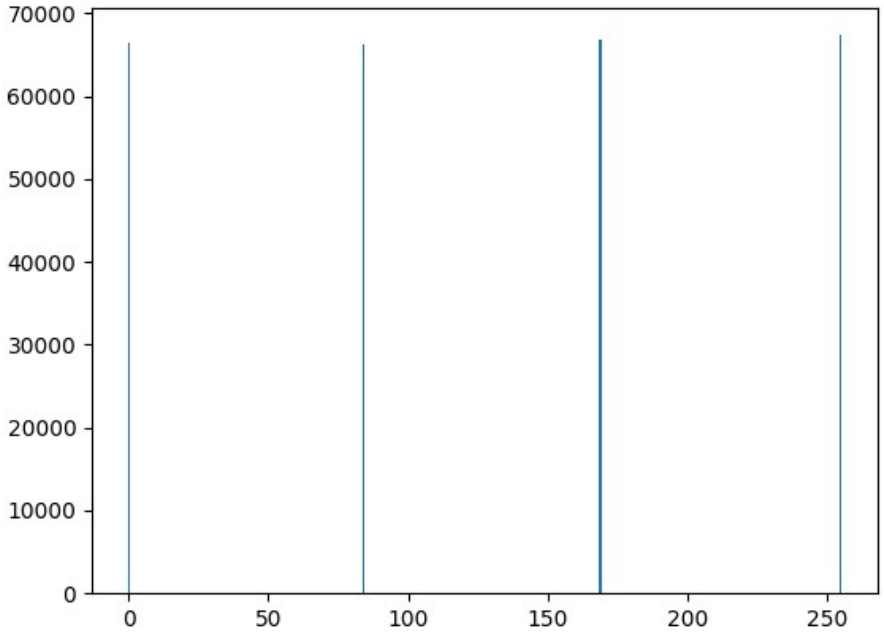

Рисунок 8 – Гистограмма серого изображения после эквализации (п. 2)

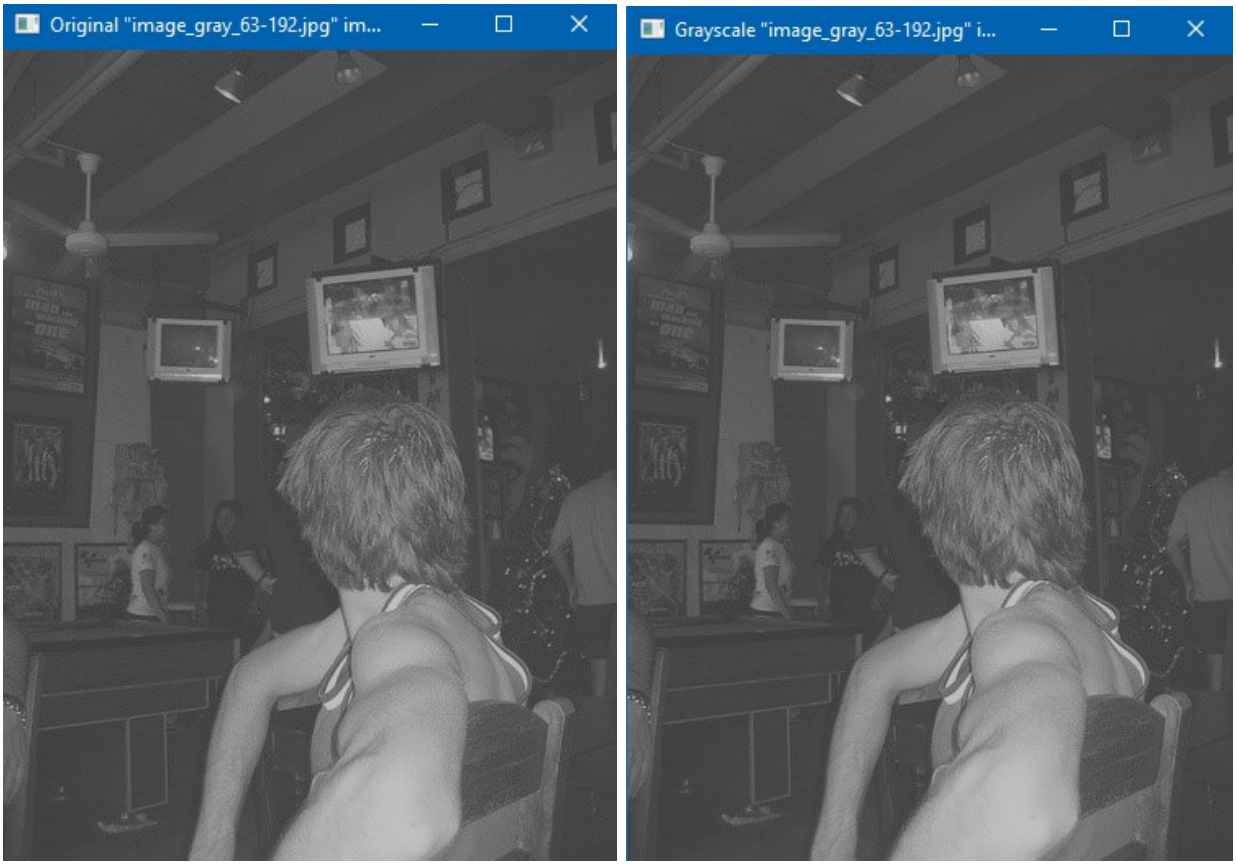

Рисунок 9 – Отображение оригинального изображения и его версия в оттенках серого (п. 3)

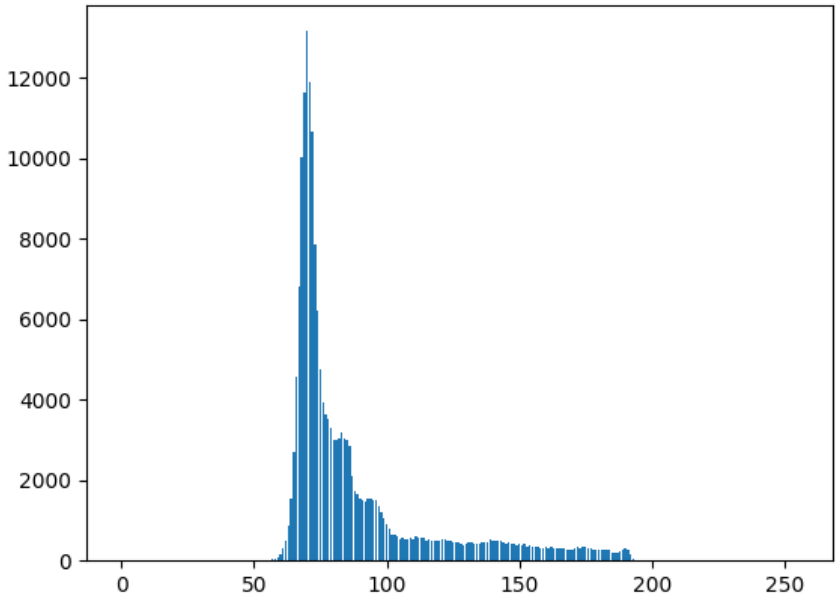

Рисунок 10 – Гистограмма серого оригинального изображения (п. 3)

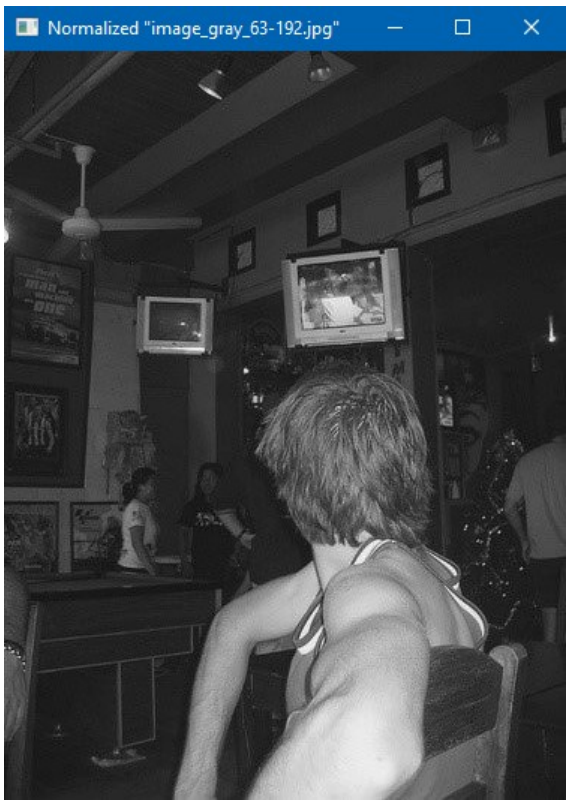

Рисунок 11 – Отображение результата MINMAX-нормализации с границами 63 и 255 (п. 3)

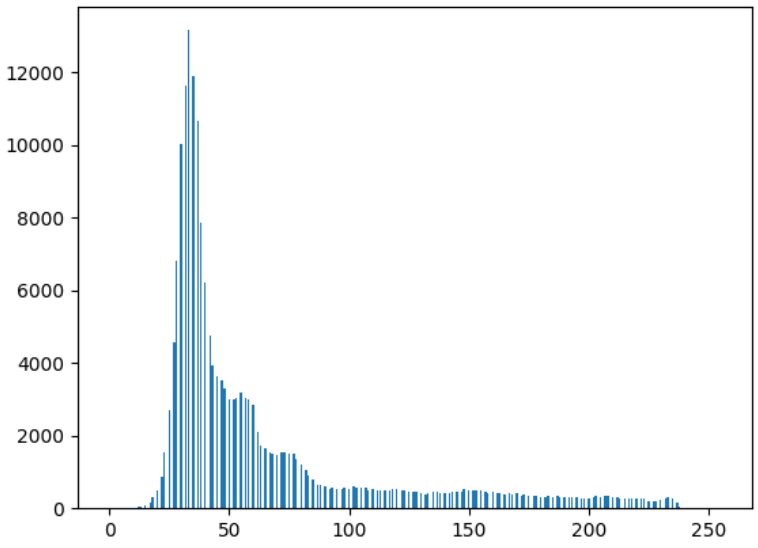

Рисунок 12 – Гистограмма серого изображения после MINMAX-нормализации (п. 3)

Рисунок 13 – Отображение результата эквализации изображения (п. 3)

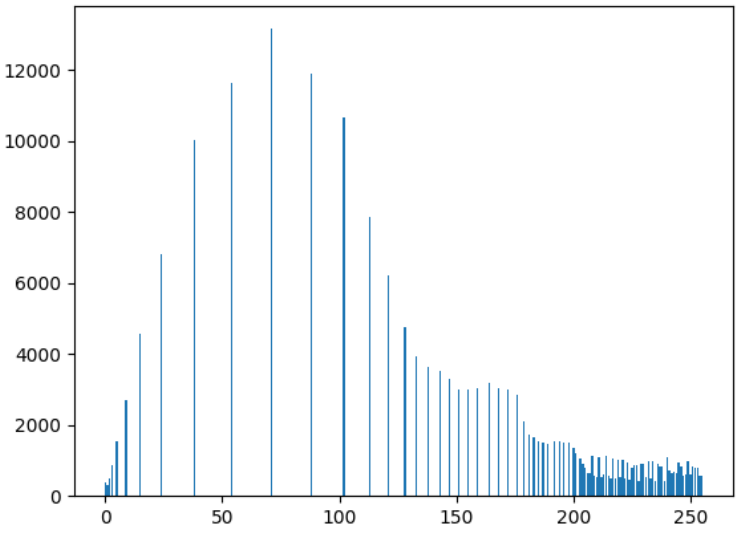

Рисунок 14 – Гистограмма серого изображения после эквализации (п. 3)

Выводы

В ходе выполнения лабораторной работы было изучено применение нормализации над изображениями, а также использование базовых функций библиотеки OpenCV для работы с гистограммами.

ПРИЛОЖЕНИЕ А

Исходный код программы

Название файла: *1306_1_3.py*

```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt

def main():
    # -------- FIRST PART --------
     first_part_image = cv.imread("D:\\7th semester\\Computer Vision\\Lab
3\\image3.jpg")
    cv.imshow("Original \"image3.jpg\" image", first_part_image)
    first_part_image = cv.cvtColor(first_part_image, cv.COLOR_BGR2GRAY)
    cv.imshow("Grayscale \"image3.jpg\" image", first_part_image)

    hist = cv.calcHist([first_part_image], [0], None, [256], [0, 256])
    plt.bar(range(256), hist.flatten())
    plt.show()

    minmax_normalization_image = cv.normalize(first_part_image, None, 63,
255, cv.NORM_MINMAX)
    cv.imshow("Normalized image (grayscale)", minmax_normalization_image)

   hist = cv.calcHist([minmax_normalization_image], [0], None, [256], [0,
256])
    plt.bar(range(256), hist.flatten())
    plt.show()

    # -------- SECOND PART --------
    second_part_image = cv.imread("D:\\7th semester\\Computer Vision\\Lab
3\\color4.jpg")
    cv.imshow("Original \"color4.jpg\" image", second_part_image)
    second_part_image = cv.cvtColor(second_part_image, cv.COLOR_BGR2GRAY)
    cv.imshow("Grayscale \"color4.jpg\" image", second_part_image)

    hist = cv.calcHist([second_part_image], [0], None, [256], [0, 256])
    plt.bar(range(256), hist.flatten())
    plt.show()

    equalization_image = cv.equalizeHist(second_part_image)
    cv.imshow("Equalized \"color4.jpg\"", equalization_image)

    hist = cv.calcHist([equalization_image], [0], None, [256], [0, 256])
    plt.bar(range(256), hist.flatten())
    plt.show()

    cv.waitKey(0)

    # -------- THIRD PART --------
     third_part_image = cv.imread("D:\\7th semester\\Computer Vision\\Lab
3\\image_gray_63-192.jpg")
                cv.imshow("Original    \"image_gray_63-192.jpg\"    image",
third_part_image)

    third_part_image = cv.cvtColor(third_part_image, cv.COLOR_BGR2GRAY)
```

```
            cv.imshow("Grayscale   \"image_gray_63-192.jpg\"   image",
third_part_image)

    hist = cv.calcHist([third_part_image], [0], None, [256], [0, 256])
    plt.bar(range(256), hist.flatten())
    plt.show()

     minmax_normalization_image = cv.normalize(third_part_image, None, 0,
255, cv.NORM_MINMAX)

                    cv.imshow("Normalized     \"image_gray_63-192.jpg\"",
minmax_normalization_image)

    hist = cv.calcHist([minmax_normalization_image], [0], None, [256], [0,
256])
    plt.bar(range(256), hist.flatten())
    plt.show()

    equalization_image = cv.equalizeHist(third_part_image)

    cv.imshow("Equalized \"image_gray_63-192.jpg\"", equalization_image)

    hist = cv.calcHist([equalization_image], [0], None, [256], [0, 256])
    plt.bar(range(256), hist.flatten())
    plt.show()

    cv.waitKey(0)


if __name__ == "__main__":
    main()
```