

ГУАП  
КАФЕДРА № 44

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

доцент, канд. техн. наук

Н. В. Соловьёв

\_\_\_\_\_  
должность, уч. степень,  
звание

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

РАЗРАБОТКА WINDOWS-ПРИЛОЖЕНИЯ

по курсу: ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4843

С. Л. Прокопьева

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2021

## 1. Индивидуальное задание

Локальное увеличение контрастности эквализацией гистограммы яркости.

## 2. Теоретические положения

Локальное увеличение контрастности фрагмента изображения можно получить, используя гистограмму яркости некоторой окрестности преобразуемого пикселя (локальную гистограмму). После получения локальной гистограммы, выполняется требуемое преобразование шкалы яркости, например, эквализация согласно (3.8), и по его результату определяется яркость центрального пикселя окрестности. Далее центр окрестности смещается на следующий пиксель и процедура повторяется. В данном случае на каждом шаге не требуется заполнять массив соответствия яркостей, т.к. необходимо найти только один элемент этого массива, соответствующий яркости преобразуемого пикселя.

Увеличение различимости фрагментов изображения можно получить и путем эквализации или выравнивания гистограммы, т.е. такого преобразования шкалы яркости, при котором гистограмма результирующего изображения будет приближаться по форме к равномерной. Метод основан на предположении, что наибольшая контрастность достигается на изображении, гистограмма которого представляет равномерное распределение пикселей по яркостям на всем диапазоне (0 ... 255). Преобразование шкалы яркости имеет вид

$$z'_i = z_m \sum_{k=0}^i p(z_k),$$

где  $z'_i$  – значение элемента преобразованной шкалы яркости, соответствующее  $i$ ой яркости исходной шкалы,  $p(z_k)$  – нормализованная гистограмма яркости исходного изображения ( $i = 0 \dots 255$ ).

### 3. Процесс выполнения работы

Язык разработки – python, фреймворк – Qt, версия – 6.2.3

Фреймворк Qt позволяет создавать десктоп-приложения, используя готовые модули для отображения объектов в окне программы.

Программа осуществляет поиск по названию среди изображений, доступных для просмотра. По нажатию кнопки система начинает процесс обработки изображения. После всех математических вычислений приложение выводит обработанное изображение в python-окне.

1. Вычисляется гистограмма исходного изображения
2. Гистограмма исходного изображения нормализуется
3. Вычисляется кумулятивная гистограмма
4. Вычисляется новое значение яркости

Был использован специальный модуль opencv для того, чтобы упростить задачу загрузки изображения в память в виде массива, вывода изображения на экран и сохранения получившегося изображения в виде файла формата .jpg

Код программы и результат выполнения представлены ниже.

### 4. Код программы

```
import cv2
from PyQt6.QtWidgets import QApplication, QMainWindow, QPushButton,
QLabel, QLineEdit, QVBoxLayout, QWidget
import matplotlib.pyplot as plt

# Загрузка изображения
image = cv2.imread('roses.jpeg', cv2.IMREAD_GRAYSCALE)
res_image = image.copy()
image_pixels = res_image.shape[0] * res_image.shape[1]

# Создание графика
figure = plt.figure(figsize=(6, 4))
ax = figure.add_subplot()
hist_cumulative = hist = ax.hist(res_image.ravel(), 256)
hist_data = [i for i in hist[0]]
hist_data_len = len(hist_data)

# Нормализация гистограммы
for i in range(hist_data_len):

    hist[0][i] = int(hist_data[i]) / image_pixels

# Вычисление кумулятивной гистограммы
for i in range(1, hist_data_len):
```

```

hist_cumulative[0][i] = hist[0][i - 1] + hist[0][i]

# Обработка изображения
for i in range(res_image.shape[0]):
    for j in range(res_image.shape[1]):

        res_image[i][j] = 255 * hist_cumulative[0][res_image[i][j]]

# cv2.imshow('image', image)
# cv2.waitKey(0)
# print(len(biba[0]))
ax.grid()
plt.show()
cv2.imshow('tiger', res_image)
cv2.waitKey(0)
cv2.imwrite('roses-result.jpeg', res_image)

class MainWindow(QMainWindow):

    def __init__(self):
        super(MainWindow, self).__init__()

        self.button_is_checked = True

        self.setWindowTitle("Matrix 3x3 income/result")

        self.label = QLabel()

        self.input = QLineEdit()
        self.input.setObjectName("host")
        self.input.setText("host")
        self.input.setMaxLength(10)
        self.input.textEdited.connect(self.label.setText)

        self.button3 = QPushButton("3x3")
        self.button3.clicked.connect(self.button_3)

        self.button7 = QPushButton("7x7")
        self.button7.clicked.connect(self.button_7)

        layout = QVBoxLayout()
        layout.addWidget(self.input)
        layout.addWidget(self.label)
        layout.addWidget(self.button3)
        layout.addWidget(self.button7)

        container = QWidget()
        container.setLayout(layout)

        self.button = QPushButton("Start calculating")
        self.button.clicked.connect(self.button_is_checked)

        # Устанавливаем центральный виджет Window
        self.setCentralWidget(container)

    def set_text(self, text):
        self.label.setText(text)

if __name__ == '__main__':
    # Передаём sys.argv, чтобы разрешить аргументы командной строки для
    приложения

```

```
app = QApplication(sys.argv)

# Создаём виджет Qt - окно
window = MainWindow()
window.show()

# Цикл событий
app.exec()
```

## 5. Результаты выполнения программы

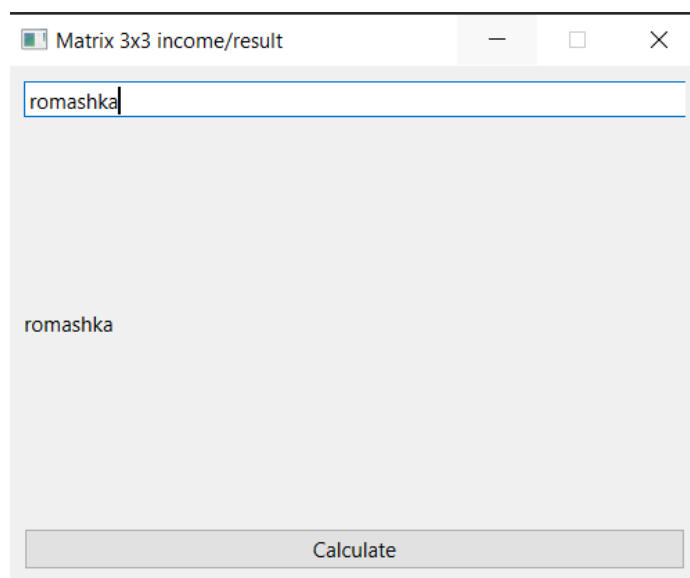


Рисунок 2 – графический интерфейс для работы с приложением



Рисунок 2 – исходное изображение 1

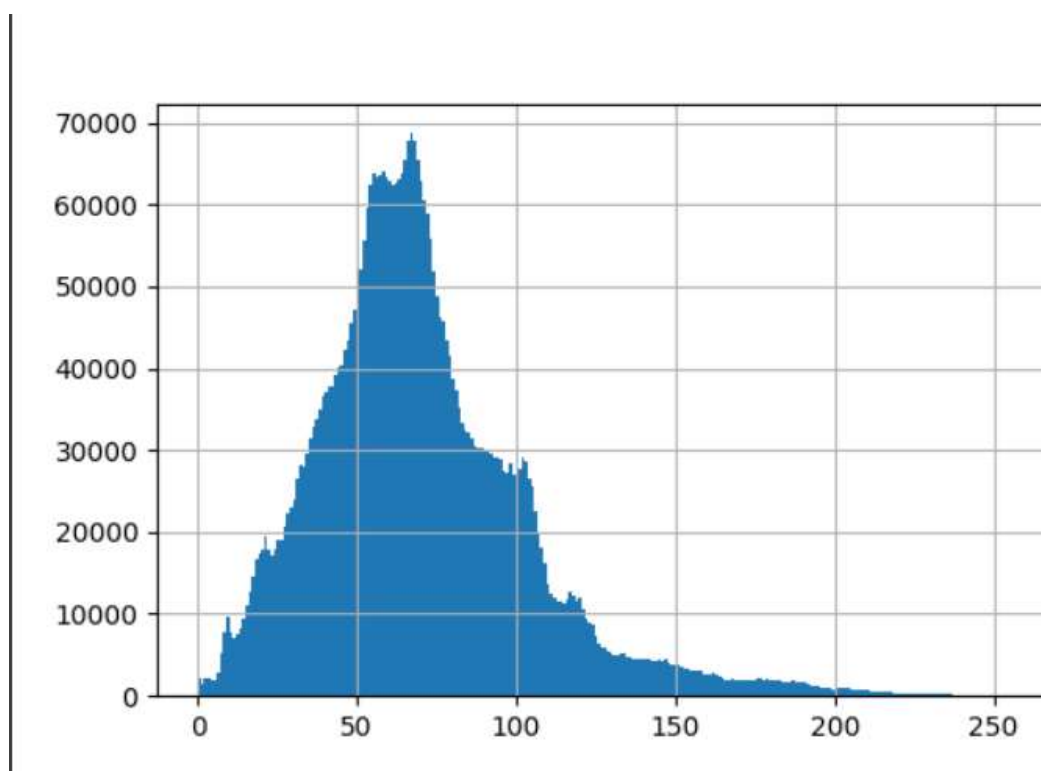


Рисунок 3 – гистограмма исходного изображения 1

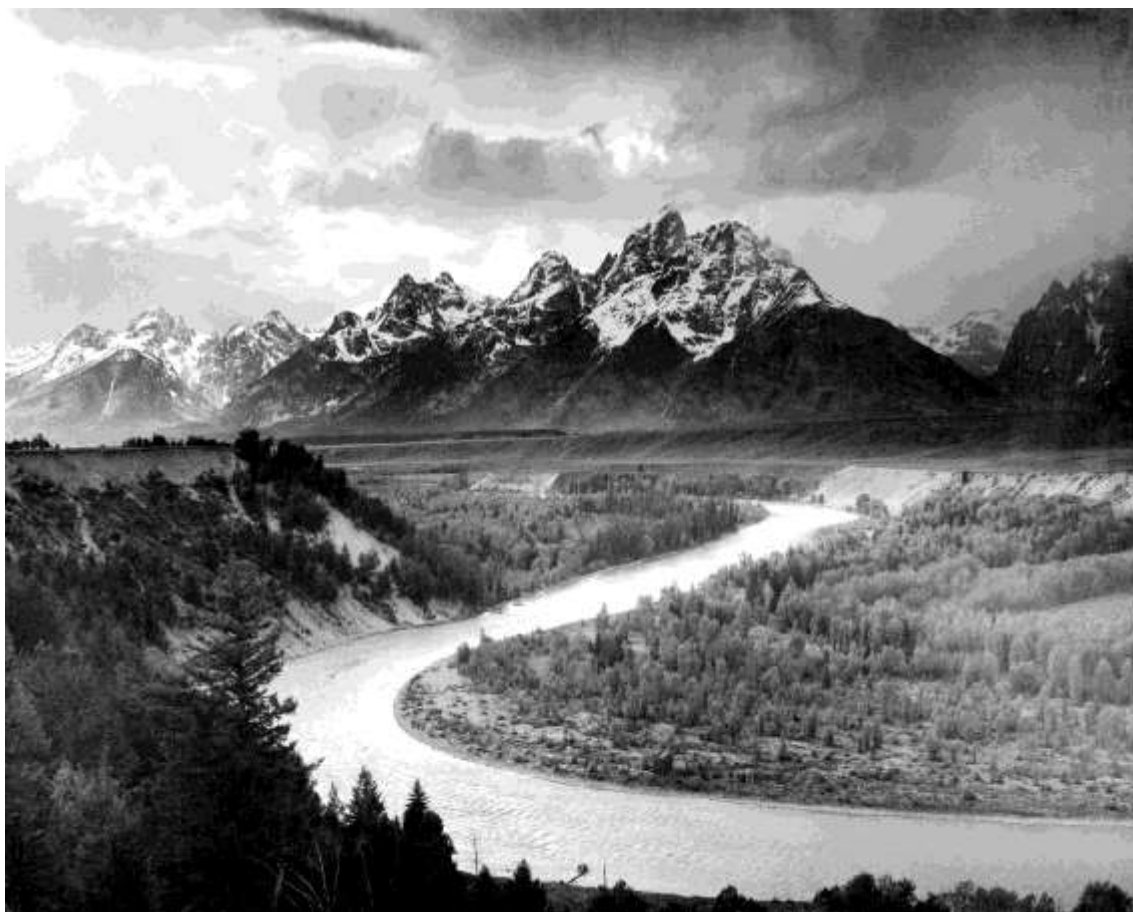


Рисунок 4 – обработанное изображение 1

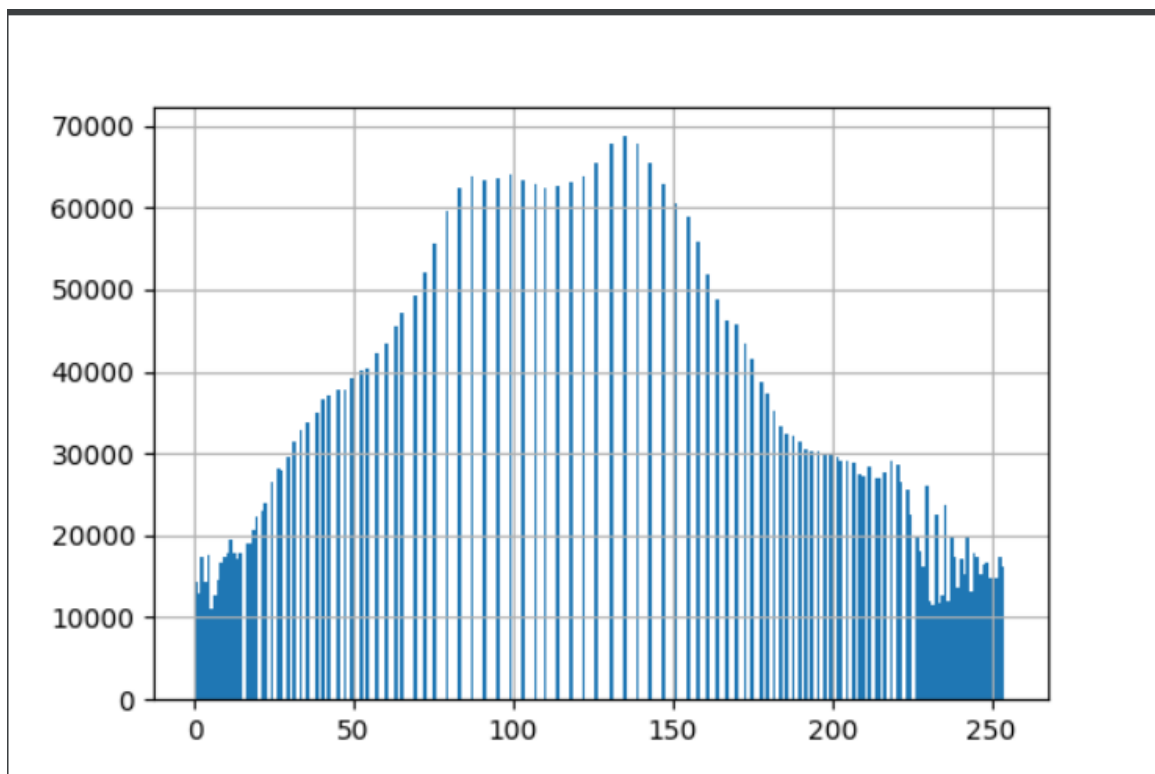


Рисунок 4 – гистограмма обработанного изображения 1





Рисунок 5 –исходное изображение 2

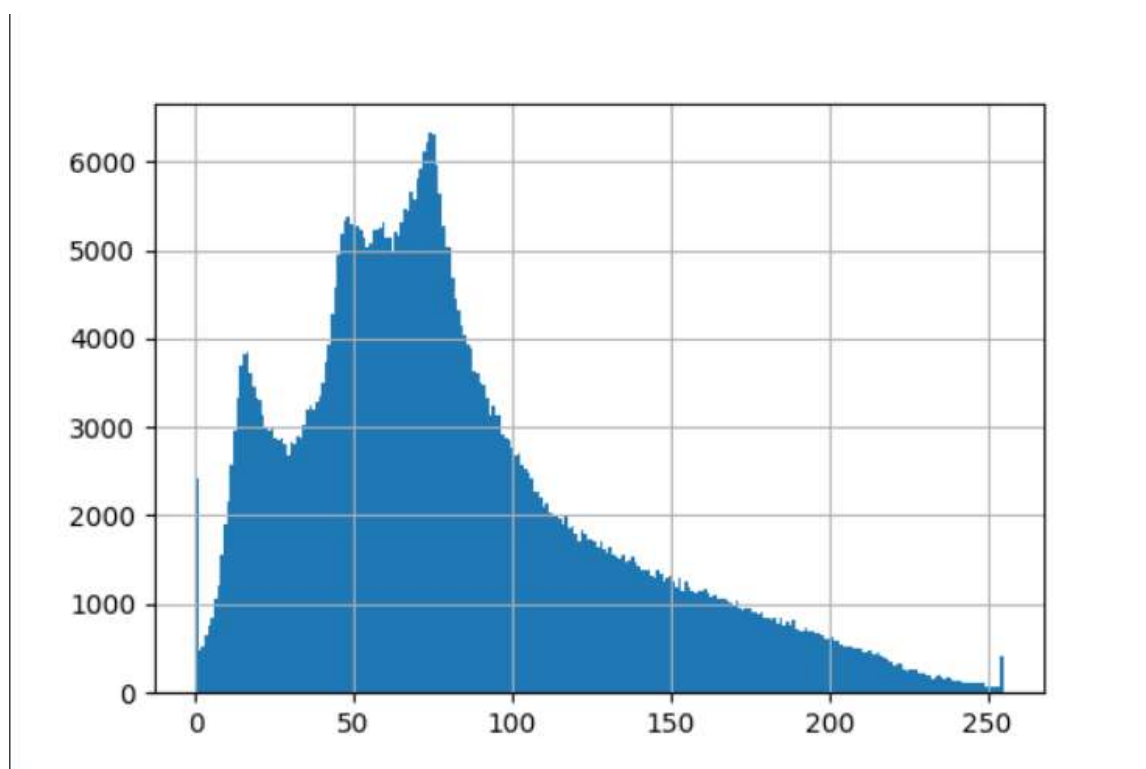


Рисунок 6 – гистограмма исходного изображения 2





Рисунок 7 – обработанное изображение 2

---

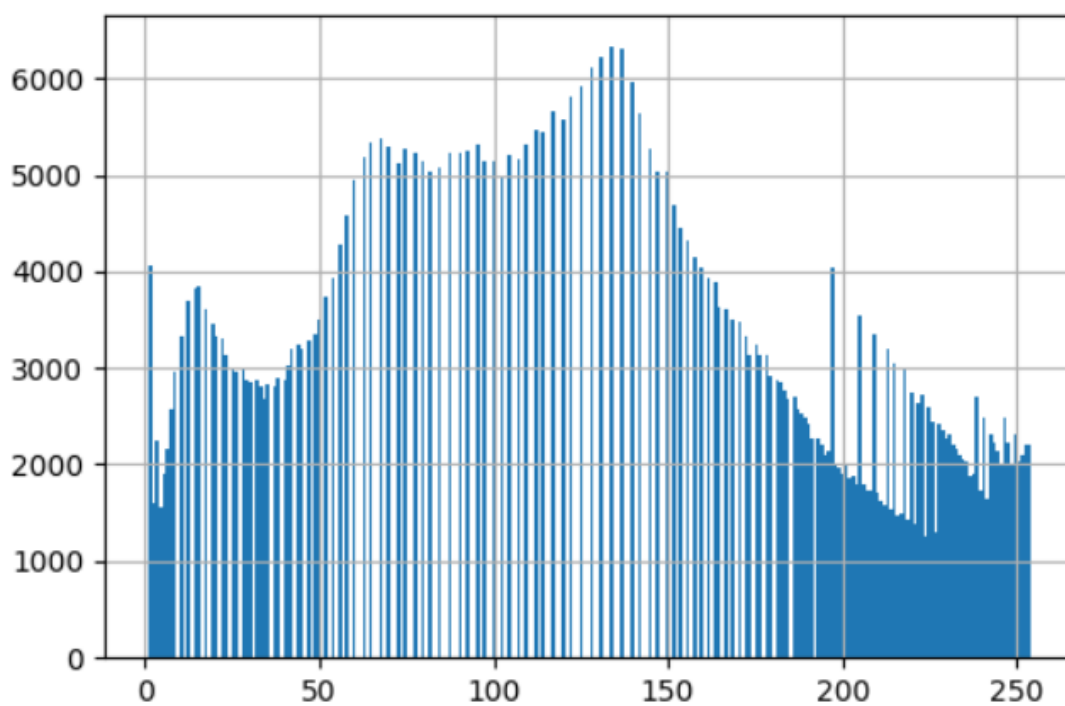


Рисунок 8 – гистограмма обработанного изображения 2

## **6. Вывод**

Выполнена лабораторная работа “Разработка Windows-приложения”, получены навыки локального повышения контрастности эквализацией гистограммы.