

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

«Процессы дискретизации и квантования изображения»

ОТЧЕТ
по лабораторной работе №8
дисциплины
«Технологии распознавания образов»

Выполнила:

Кувшин Ирина Анатольевна
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Цель работы:

Изучение функций, использующихся для моделирования процессов квантования и дискретизации изображения на языке Python.

Ход выполнения:

2.1. Дискретизация изображения

Задание 2.1.

```
2. Процессы дискретизации и квантования изображений

2.1. Дискретизация изображения

Задание 2.1.

Выбрать значение шага дискретизации в пределах от 5 до 15. Продискретизировать с этим шагом дискретизации изображение и вывести его на экран.

In [8]: 1 import cv2
2 import numpy as np
3
4 image = cv2.imread('nature.jpg') # Загрузим и сохраним изображение
5 img = image.copy() # Создадим копию изображения, над которой и выполним дискретизацию
6 K = 10 # Зададим размер шага изображения (количество пикселей в этом изображении будет K*K)
7 s = img.shape # Получаем размер исходного изображения и его тональность.
8 h1, w1 = s[0], s[1] # Запоминаем отдельно высоту и ширину исходного изображения
9 h = (s[0] - s[0] % K) # Делим высоту на шаг с выделением остатка и вычитаем это из начальной высоты
10 # Например: (641 - 641 % 10) = 641 - 1 = 640. Это позволяет за указанное кол-во шагов пройти все изображение точно без остатка
11 w = (s[1] - s[1] % K) # То же самое для ширины
12 img = cv2.resize(img, (w, h)) # Меняем размер изображения на новую высоту и ширину
13 # Выводим на экран изображение на экран (каждый шаг K в ширину и K в высоту)
```

Рисунок 8.1 – Код программы

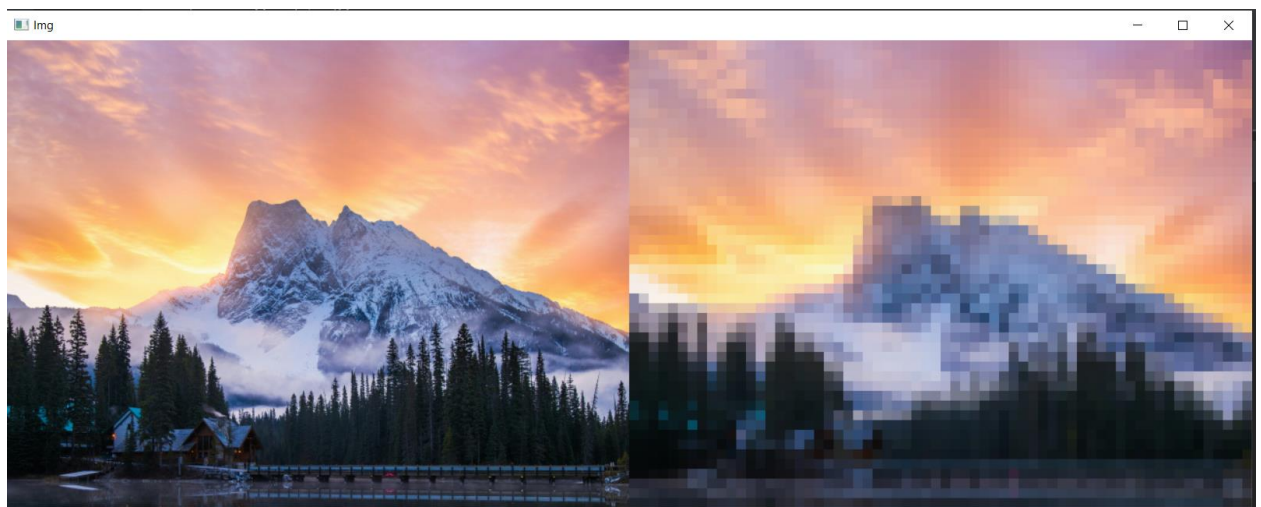


Рисунок 8.2 – Результат выполнения программы

2.2. Квантование изображения

Задание 2.2.

```
File Edit View Navigate Code Refactor Run Tools Git Window Help IRT_8 - Jupyter_Lab8.ipynb
notebooks Jupyter_Lab8.ipynb
Project IRT_8 C:\Users\...
notebooks
  Jupyter_Lab8.ipynb
  README.md
  nature.jpg
  .gitignore
  LICENSE
  README.rst
  External Libraries
  Scratches and Snippets

2.2. Квантование изображения

Задание 2.2.

Проквантовать изображение, сократив число градаций до 4

In 3
1 import cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 plt.subplot(121)
6 img = cv2.imread('nature.jpg')
7 plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB)) # Меняем формат BGR на RGB
8 plt.axis('off')
9 plt.subplot(122)
10 # Изменяем размер массива изображения, где (-1,3) - задаем число строк и столбцов, в которые помещаем прежний массив. В данном случае 3 - означает в 3
11 # столбца, а -1 это исключение, означает что мы даем python самому понять во сколько строк это можно уместить
12 Z = img.reshape(-1, 3) # преобразуем массив к float (числа с плавающей точкой)
13 crt = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)
14 K = 4 # Количество градаций изображения
15 ret, label, center = cv2.kmeans(Z, K, None, crt, 10, cv2.KMEANS_RANDOM_CENTERS) # Функция квантования
16 # Преобразуем изображение к начальному виду:
17 center = np.uint8(center) # Преобразование матрицы к формату uint8 (от 0 до 255)
18 res = center[label.flatten()] # Сворачиваем массив нескольких матриц в одну
```

Рисунок 8.3 – Код программы

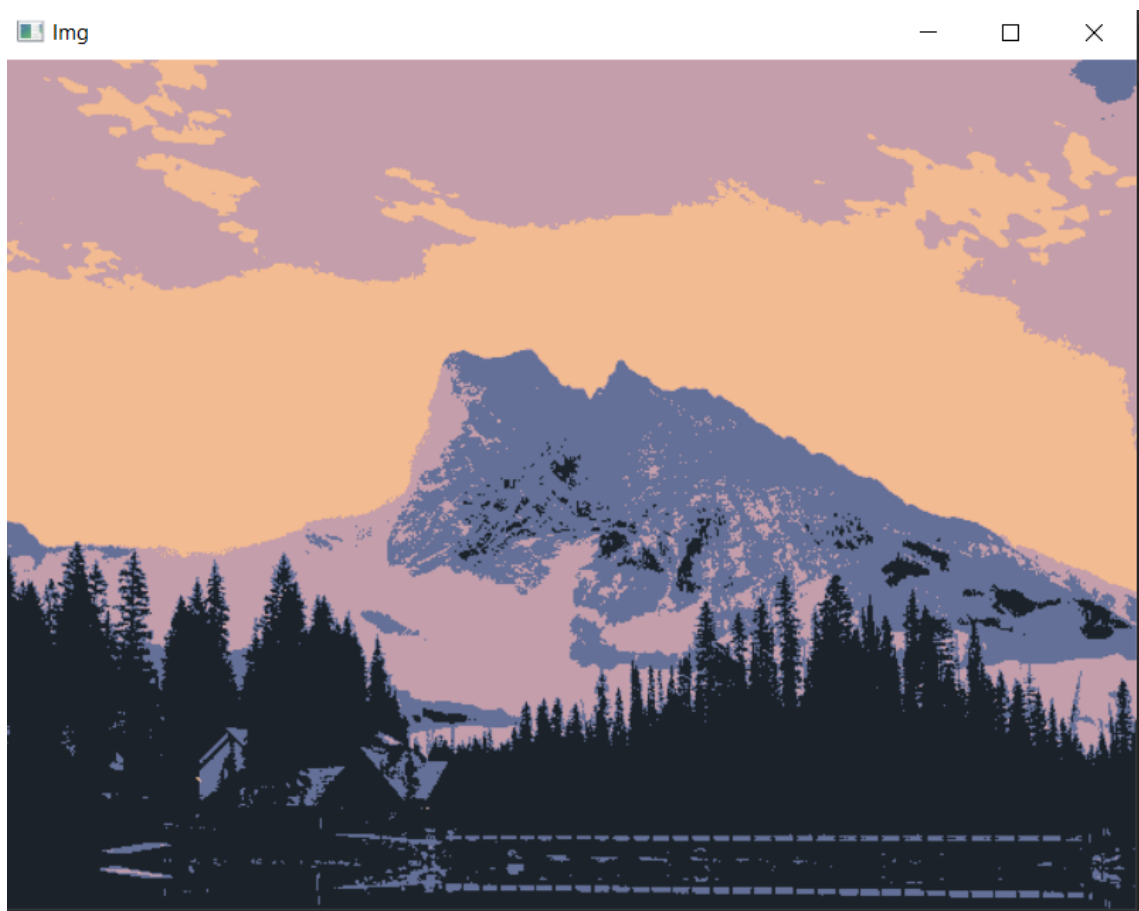


Рисунок 8.4 – Результат выполнения программы

Индивидуальное задание

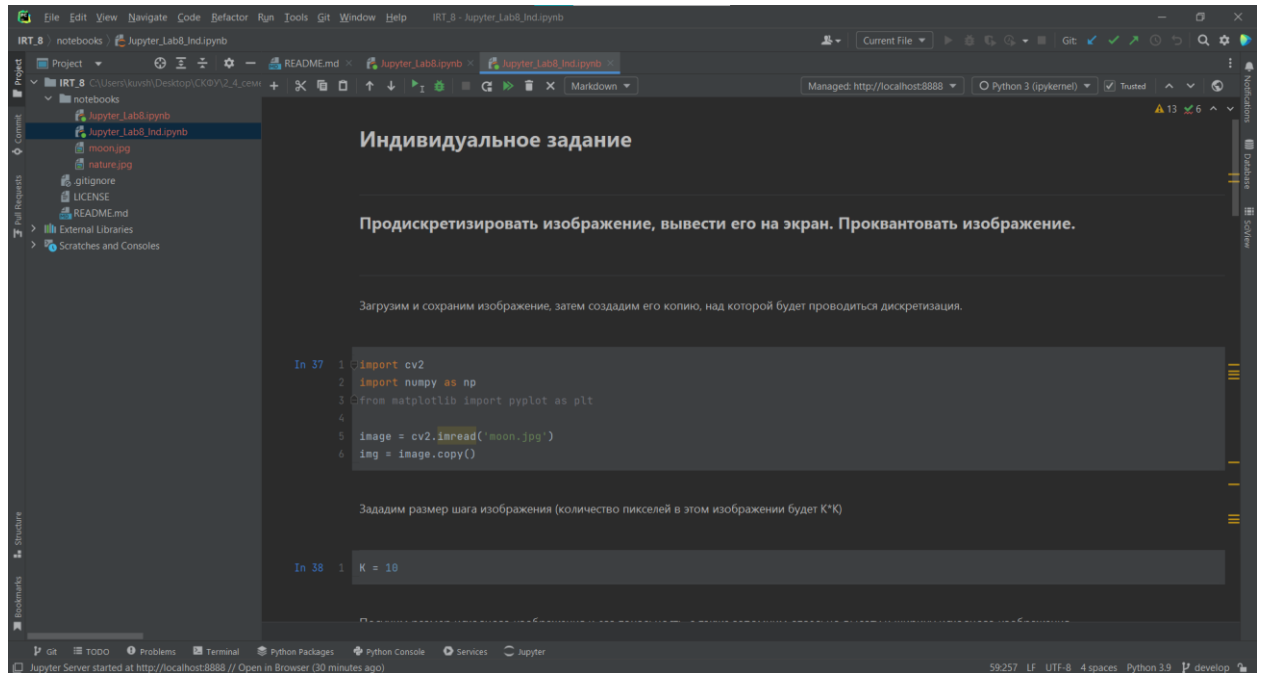


Рисунок 8.5— Код программы



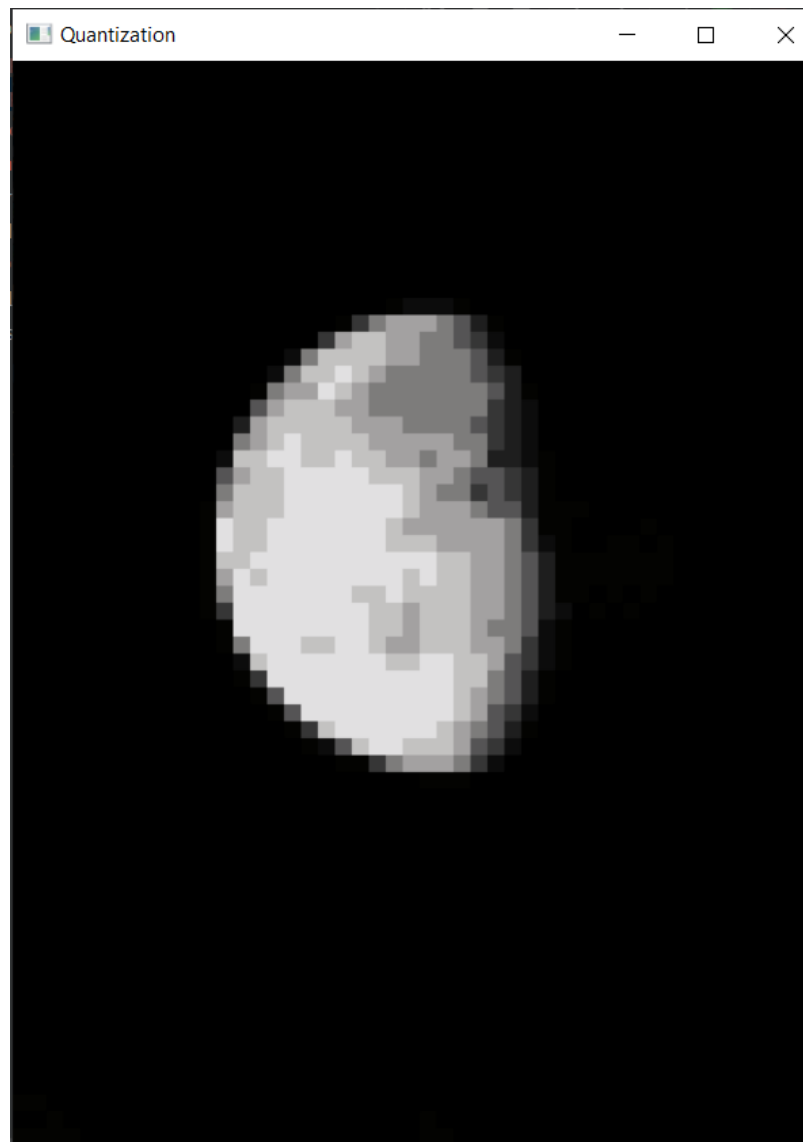


Рисунок 8.6 – Результат выполнения программы