

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

Кафедра инфокоммуникаций

Отчёт по практическому занятию №3.8

«Процессы дискретизации и квантования изображения»

по дисциплине «Теории распознавания образов»

Выполнил студент группы ПИЖ-б-о-21-1
Коновалова В.Н. « » _____ 20__ г.
Подпись студента _____
Работа защищена « » _____ 20__ г.
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Цель работы: изучение функций, использующихся для моделирования процессов квантования и дискретизации изображения на языке Python.

Выполнение работы:

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 [1]: import cv2
import numpy as np
from matplotlib import pyplot as plt

In [2]: image = cv2.imread("cat.jpeg")
img = image.copy();

In [3]: K = 10 # размер шага
s = img.shape

h1, w1 = s[0], s[1]
h = (s[0] - s[0] % K)
w = (s[1] - s[1] % K)

img = cv2.resize(img, (w, h))

In [4]: for y in range(0, h-1, K):
        for x in range(0, w-1, K):
            if len(s) > 2:
                s = np.average(img[y:(y + K), x:(x + K)], axis=0)
                img[y:(y + K), x:(x + K)] = np.average(s, axis=0)
            else:
                s = img[y:(y+K), x:(x+K)]
                img[y:(y+K), x:(x+K)] = np.average(s)

In [5]: img = cv2.resize(img, (w1, h1))
res = np.hstack((image, img))
plt.imshow(img);
```

Необходимо написать программу для дискретизации изображения с заданным пользователем шагом. Программа должна загружать изображение из файла, запрашивать у пользователя шаг дискретизации, производить дискретизацию и сохранять результат в новый файл. Также программа должна отображать исходное и обработанное изображение на экране. ¶

```
In [1]: import cv2
import numpy as np
from matplotlib import pyplot as plt
```

Загрузка изображения

```
In [2]: img = cv2.imread('image.jpeg')
```

Размеры изображения

```
In [3]: height, width = img.shape[:2]
```

Шаг дискретизации

```
In [4]: step = int(input("Введите шаг дискретизации: "))
```

Введите шаг дискретизации: 5

Вычисление новых размеров изображения

```
In [5]: new_height = height - height % step
new_width = width - width % step
```

Масштабирование изображения обратно к исходным размерам

```
In [7]: img = cv2.resize(img, (width, height))
```

Сохранение результата в файл

```
In [8]: cv2.imwrite('discretized_image.jpg', img)
```

```
Out[8]: True
```

Отображение изображения

```
In [9]: plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))  
plt.show();
```



