

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

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

Кафедра инфокоммуникаций

**Отчёт по практическому занятию №3.7
«Основы цифровой обработки изображений в OpenCV»**

по дисциплине «Теории распознавания образов»

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

Ставрополь 2023

Цель работы: изучение типов изображений, способов их формирования. Изучение основных функций OpenCv, применяемых для цифровой обработки изображений.

1. Считывание изображения и вывод его на экран, запись изображения в файл.
2. Вывод свойств изображения и сформированной матрицы на экран.
3. Доступ к изображению для изменения значений цвета пикселей.
4. Создание бинарного изображения и его негатива.
5. Применение библиотеки matplotlib для вывода нескольких изображений в общем окне.
6. Выделение и взятие в рамку определенного региона изображения.
7. Уменьшение размера изображения и вывод матрицы на экран.
8. Знакомство с процессом дискретизации и квантования изображения.
9. Приобретение практических навыков использования этих функций.

Выполнение работы

Задание 1.1.

Считать файл полноцветного изображения cat.jpg, создать для него матрицу изображения, затем вывести сначала полутоновое, затем цветное изображение на экран. Перед выполнением задания получить согласно номеру в списке группы свой файл с изображением.

```
In [2]: import cv2

In [ ]: img = cv2.imread('cat.jpg', 0)
        cv2.imshow('image_1', img)
        cv2.waitKey(0)

In [ ]: img = cv2.imread('cat.jpg', cv2.IMREAD_GRAYSCALE)
        cv2.imshow('image_2', img)
        cv2.waitKey(0)
```

```

n = 28
a = np.ones([28, 28])
for i in range(n):
    a[i][i] = 1

for i in range(n):
    for j in range(0, i):
        a[i][j] = 0

cv2.imwrite('ris.png', a)
img = cv2.imread('ris.png', 0)
cv2.imshow('image_6', img)
print(img)
cv2.waitKey(0)

```

```

[[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]]

```

Задание 1.5.

Вывести свойства матрицы изображения на экран.

```

In [1]: import cv2
import numpy as np

img = cv2.imread('cat.jpg', 0)
cv2.imshow('image', img)

print(type(img))
print(img.shape)
print(img.size)
print(img.dtype)

<class 'numpy.ndarray'>
(457, 685)
313045
uint8

```

Самостоятельная работа

У нас есть изображение, на котором необходимо заменить белый цвет на зеленый

```
In [10]: import cv2
import numpy as np
from matplotlib import pyplot as plt
```

```
In [11]: img = cv2.imread('cat.jpg')
new_image = img.copy()
```

Определяем цвета, которые нужно заменить

в нашем случае-белый

```
In [12]: lower_range = np.array([200, 200, 200])
upper_range = np.array([250, 250, 250])
```

Создаем маску, выбирая пиксели изображения, которые находятся в диапазоне цветов, который мы хотим заменить

Заменяем цвета, находящиеся под маской, на новый цвет (здесь - зеленый)

```
In [13]: mask = cv2.inRange(new_image, lower_range, upper_range)
new_image[mask > 0] = (0, 255, 0)
```

Визуализируем результаты обработки с помощью библиотеки matplotlib.

```
In [14]: plt.subplot(121)
plt.axis("off")
plt.imshow(img)
plt.title('Original')

plt.subplot(122)
plt.axis("off")
plt.imshow(new_image)
plt.title('Modified')

plt.show()
```

Original



Modified



```
In [ ]:
```
