

**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
Коновалова В.Н. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

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

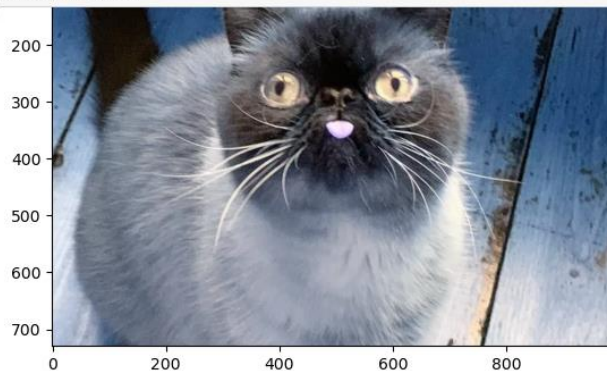
1. Считывание изображения и вывод его на экран, запись изображения в файл.

Задание 1.1

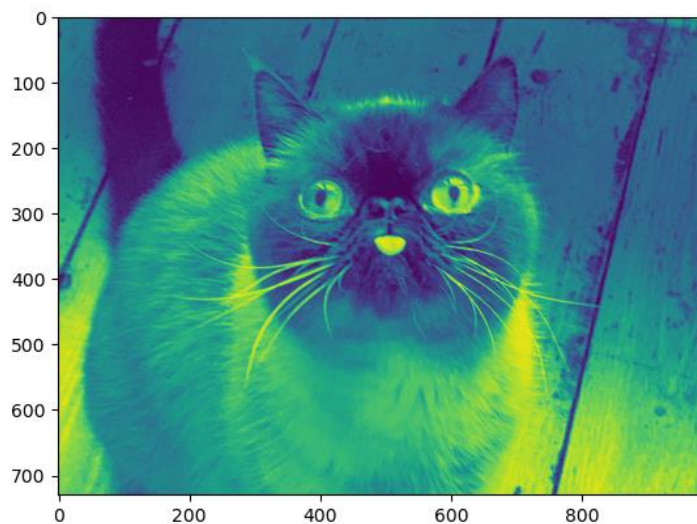
Считать файл полноцветного изображения cat.jpg, создать для него матрицу изображения, затем вывести сначала полутоновое, затем цветное изображение на экран. Перед выполнением задания получить согласно номеру в списке группы свой файл с изображением.

```
: import cv2
import numpy as np
from matplotlib import pyplot as plt
```

```
: img = cv2.imread('cat.jpg', 1)
plt.imshow(img);
```



```
In [3]: img = cv2.imread('cat.jpg', 0)
plt.imshow(img);
```

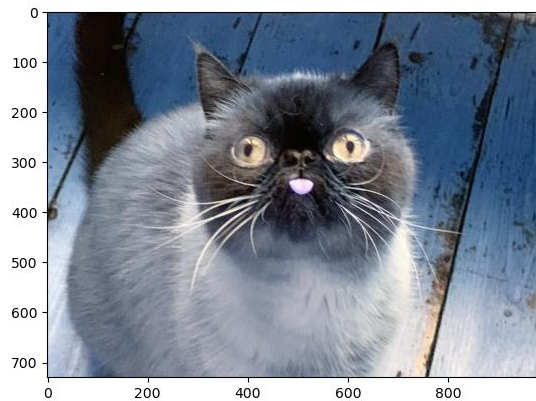


2. Вывод свойств изображения и сформированной матрицы на экран.

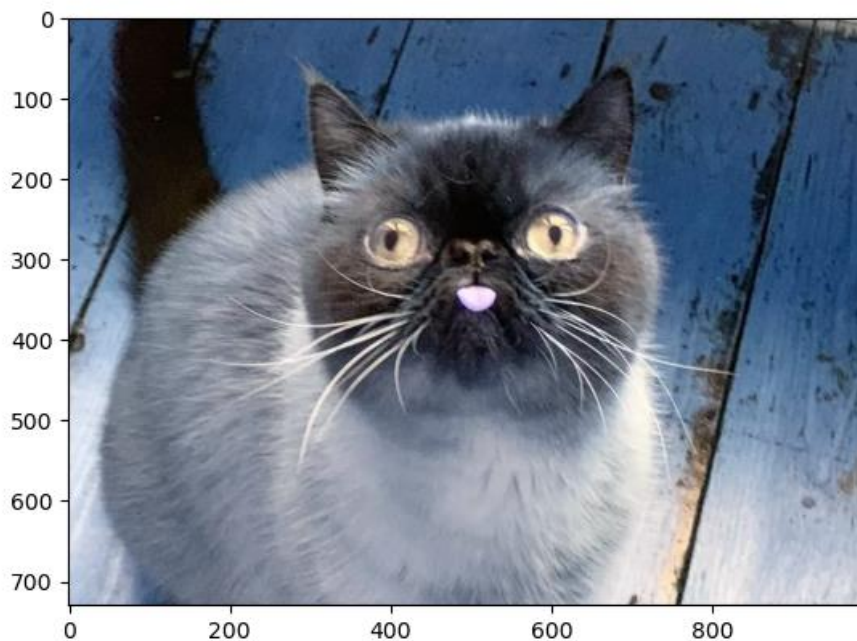
Задание 1.2 ¶

Используя код задания 1.1, в функции `cv2.imread()` присвоить флагу значение 1, затем вывести изображение на экран. Выполнить этот же код, заменив в функции `cv2.imread('cat.jpg', 1)` флаг 1 на флаг `cv2.IMREAD_COLOR`.

```
In [4]: img = cv2.imread('cat.jpg', 1)  
plt.imshow(img);
```



```
5]: img = cv2.imread('cat.jpg', cv2.IMREAD_COLOR)  
plt.imshow(img);
```

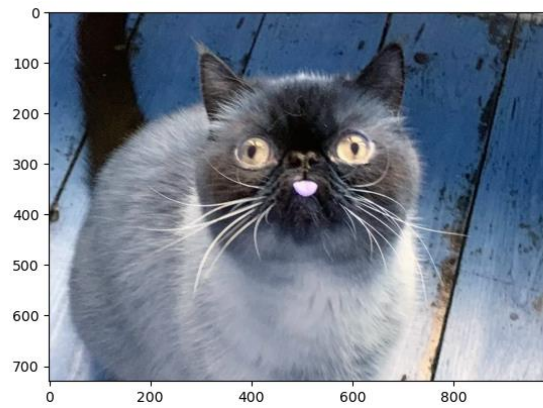


3. Доступ к изображению для изменения значений цвета пикселей.

Задание 1.3

Сформировать матрицу изображения, записать ее в файл с расширением png. Изображение, записанное в этом файле, вывести на экран.

```
j: img = cv2.imread('cat.jpg')  
  
#запись изображения из матрицы в файл  
cv2.imwrite('img.png', img)  
img = cv2.imread('img.png')  
plt.imshow(img);
```



4. Создание бинарного изображения и его негатива.

Сформировать матрицу, у которой выше диагонали единицы, а ниже – нули, записать ее в файл, затем считать файл и вывести на экран.

```

]: # Строим массив 28x28, заполненный нулями
n = 28
a = np.ones([28, 28])

# Заполняем диагональ массива
for i in range(n):
    a[i][i] = 1

# Заполняем массив выше диагонали
for i in range(n):
    for j in range(i, n):
        a[i][j] = 0

cv2.imwrite('z4.png', a) # Записываем изображение в файл
img = cv2.imread('z4.png', 0) # Считываем изображение с файла

print(img)
plt.imshow(img);

[[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 1 1]]

```


6. Выделение и взятие в рамку определенного региона изображения.

Задание 1.6

Определить с помощью функции `print(img.shape)` максимальное число пикселей по ширине и высоте изображения. Вы- брать координаты так, чтобы они не выходили за пределы размеров изображения. Задать координату по горизонтали равной сумме номера по списку группы плюс 70, по вертикали равной сумме номера по списку группы плюс 50.

Изменить значения пикселей, интенсивности B, G, R цветов, взяв интенсивности первого упражнения [95 129 175] и прибавив к ним номер по списку группы.

```
: img = cv2.imread('cat.jpg')
print("Максимальное число пикселей:")
print(img.shape)#определение размера матрицы

# номер по списку 6 [76; 56]
px = img[76, 56] # доступ к пикселю цветного изображения с координатами 76, 56
print("\nЗначение пикселя с координатами 76, 56 (B, G, R):")
print(px)

blue = img[76, 56, 0] # доступ только к синему пикселю с координатами (76, 56)
print("\nСиний пиксель:")
print(blue)

#[56, 62, 85] + 6 = [62, 68, 91]
img[76, 56] = [62, 68, 91]
print("\nИзмененные значения: ")
print(img[76, 56])

# Доступ с помощью функций array.item() и array.itemset()
print("\nКрасный пиксель:")
print(img.item(76, 56, 2))
# изменение значения красного пикселя
img.itemset((76, 56, 2), 85)
print("Измененное значение пикселя:")
print(img.item(76, 56, 2))
```

Максимальное число пикселей:
(730, 991, 3)

Значение пикселя с координатами 76, 56 (B, G, R):
[56 62 85]

Синий пиксель:
56

Измененные значения:
[62 68 91]

Красный пиксель:
91
Измененное значение пикселя:
85

7. Уменьшение размера изображения и вывод матрицы на экран.

Задание 1.7

Считать файл полноцветного изображения cat.jpg, создать для него два места в окне в ширину и два места в высоту. Преобразовать матрицу цветного изображения в полутоновое, из него, используя функцию cv2.threshold, получить бинарное монохромное изображение. Из бинарного монохромного изображения получить его негатив.

```
img = cv2.imread("cat.jpg")
imag = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Оригинальное изображение выводится в первое окно:
plt.subplot(221)
plt.imshow(imag)
plt.axis('off')

# полутоновое ч/б изображение
gray_img = cv2.imread("cat.jpg", 0)

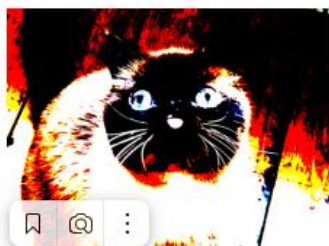
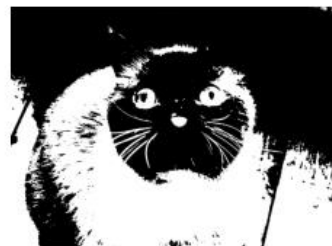
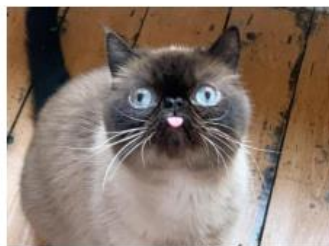
"""Функция для получения бинарного изображения, в скобках gray_img - исходное изображение;
 128 - пороговое значение; 255 - значение, которое придаем пикселю,
если его значение больше порогового;
[1] - ожидаем один ответ (по умолчанию 2)
"""
im_bw = cv2.threshold(gray_img, 128, 255, cv2.THRESH_BINARY)[1]

# выведем бинарное монохромное изображение во второе окно
plt.subplot(222)
plt.imshow(im_bw, 'gray')
plt.axis("off")

# полутоновое цветное изображение
im_bwa = cv2.threshold(imag, 128, 255, cv2.THRESH_BINARY)[1]

# выводим монохромное цветное изображение в третье окно
plt.subplot(223)
plt.imshow(im_bwa)
plt.axis("off")

# инвертирует монохромное изображение
im_bwb = cv2.threshold(gray_img, 128, 255, cv2.THRESH_BINARY_INV)[1]
plt.subplot(224)
plt.imshow(im_bwb, 'gray')
plt.axis("off")
plt.show()
```

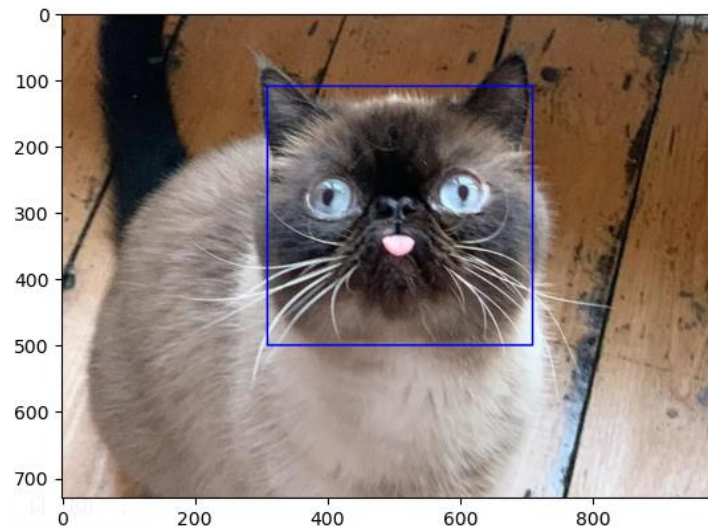


8. Знакомство с процессом дискретизации и квантования изображения.

Задание 1.8

На заданном изображении выделить его характерный участок.

```
# выделим мордочку кота  
img = cv2.imread("cat.jpg", 1)  
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  
image = cv2.rectangle(img, (310, 110), (710, 500), (0, 0, 255), 2)  
plt.imshow(image);
```



9. Приобретение практических навыков использования этих функций.

Задание 1.9

Уменьшить заданное изображение и вывести на печать матрицу уменьшенного изображения.

```
|: img = cv2.imread('cat.jpg')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
final_wide = 200
r = float(final_wide) / img.shape[1]
#уменьшаем изображение до подготовленных размеров
dim = (final_wide, int(img.shape[0]*r))
resized = cv2.resize(img,dim,interpolation=cv2.INTER_AREA)

plt.subplot(221)
plt.imshow(resized)
print(resized.shape)
print(resized)

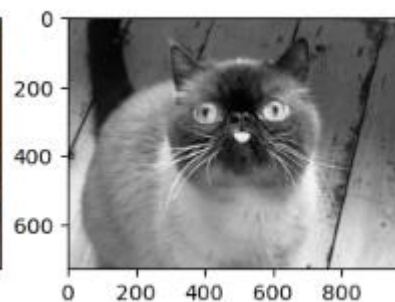
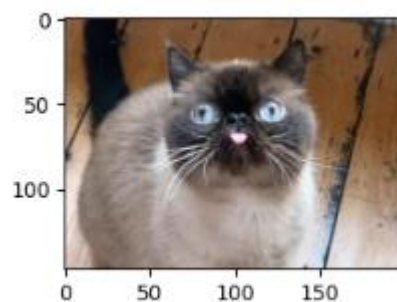
img = cv2.imread('cat.jpg', 0)
plt.subplot(222)
plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
print(img)
plt.show()
```

```
[[222 200 186]
 [224 201 187]
 [220 196 182]
 ...
 [199 189 187]
 [220 211 208]
 [230 220 219]]
```

```
[[220 199 183]
 [221 199 185]
 [217 194 180]
 ...
 [207 195 195]
 [228 218 217]
 [229 222 220]]
```

```
[[219 197 181]
 [219 197 183]
 [214 191 177]
```

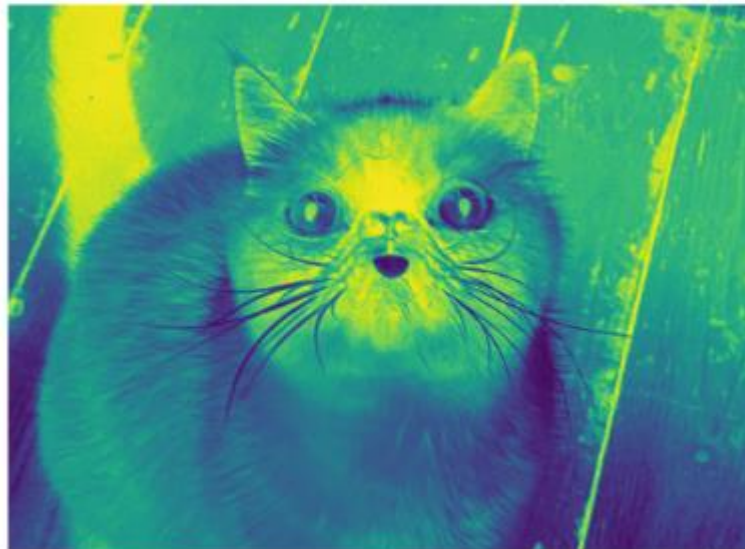
```
[205 205 205 ... 217 216 216]]
```



Задание 1.10

Считать цветное изображение, конвертировать его в полутоновое, затем получить негатив полутонового изображения.

```
img = cv2.imread('cat.jpg', 1)
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
img = cv2.bitwise_not(img)
plt.axis("off")
plt.imshow(img);
```



10. Индивидуальное задание

Программа считывает изображение из файла, затем на изображении выделяется характерный участок и инвертируются его цвета. После обработанное изображение сохраняется в новый файл.

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

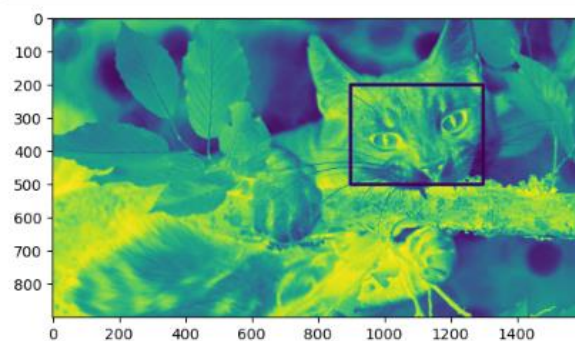
[2]: # Загрузка изображения из файла
img = cv2.imread('img.jpg', 1)
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Инвертировать цвета изображения
inverted_img = cv2.bitwise_not(img)

# Нарисовать прямоугольник на изображении
ind_img=cv2.rectangle(inverted_img, (900, 500), (1300,200), (0, 255, 0), thickness=10)

# Сохранить обработанное изображение в новый файл
cv2.imwrite('ind_img.jpg', ind_img)

# Отобразить обработанное
plt.imshow(ind_img);
```



Вопросы

1. Типы изображений?

- бинарные – изображения, пиксели которого принимают только два значения: 0 и 1, что соответствует черному или белому цвету;
- полутоновые (серые или изображения в градациях серого) – диапазон значений интенсивности пикселей в формате `uint8 [0, 255]` или в формате `double [0,1]` (для языка `python` вещественные числа `float`);
- палитровые – каждому пикселу сопоставляется номер ячейки карты цветов, в карте цветов содержится описание цвета пиксела в некоторой цветовой системе (палитре);
- цветные (RGB) – пиксели непосредственно хранят информацию об интенсивностях цветного изображения, например, об интенсивности красного, зеленого, синего цвета.

2. По способу хранения описания изображения оно может быть:

- векторным, если изображение создается набором графических примитивов (отрезок прямой, угол, многоугольник, окружность, дуга и т.д.), из которых и формируется изображение;
- растровым, если изображение кодируется двумерным массивом, элементами которого являются интенсивности серого цвета, либо одного из цветов (красного, зеленого, синего).

3. Функция чтения изображения из файла?

`imread(,)`.

4. Что делает функция `imwrite`?

Для создания изображения из его матрицы в виде файла используется функция `cv2.imwrite (,)`.

5. Функция `cv2.threshold(gray,128,255,cv2.THRESH_BINARY)`

предназначена для?

Для получения монохромного изображения, в матрице которого записано либо 0, либо 255.

6. Что такое дискретная функция интенсивности?

Интенсивность (яркость) изображения является функцией $f(x,y)$ двух пространственных переменных x и y на ограниченной прямоугольной области. Ее называют непрерывной функцией интенсивности. Перейдем от непрерывной функции интенсивности к дискретной функции интенсивности.

7. Загрузка цветного изображения?

`cv2.imread color`

8. Загрузка изображения в режиме градаций серого?

`cv2.imread grayscale`

9. Загрузка изображения как такового, включая альфа-канал?

`cv2.imread unchanged`

10. Значение по умолчанию флага?

`cv2.imread color`