

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Отчет по лабораторной работе № 3.7
по дисциплине «Технологии распознавания образов»

Выполнил студент группы ПИЖ-б-о-21-1

Трушева В. О. .«__»__ 2023г.

Подпись студента _____

Работа защищена « » 20 г.

Проверила Воронкин Р.А. _____
(подпись)

Ставрополь 2023

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

Методика и порядок выполнения работы

1. Изучить теоретический материал работы.

2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный Вами язык программирования (выбор языка программирования будет доступен после установки флажка Add .gitignore).

Owner * Repository name *

pynikcmd / 3.7_LR_TRO ✓

Great repository names are short and memorable. Need inspiration? How about [probable-carnival?](#)

Description (optional)

☒ Public Anyone on the internet can see this repository. You choose who can commit.

☐ Private You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ Add a README file This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python ▼

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License ▼

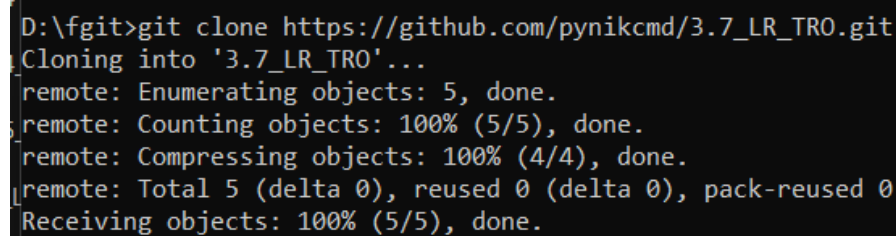
This will set `main` as the default branch. Change the default name in your [settings](#).

① You are creating a public repository in your personal account.

Create repository

Рисунок 1 – Создание репозитория

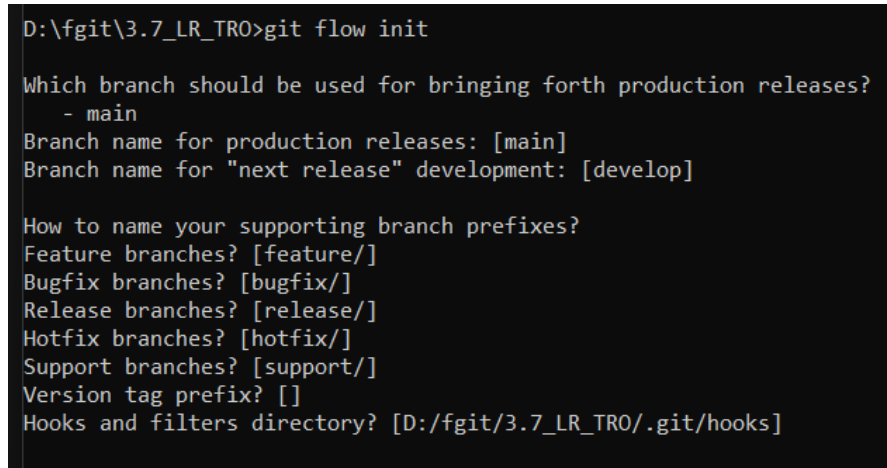
3. Выполните клонирование созданного репозитория на рабочий компьютер.



```
D:\fgit>git clone https://github.com/pynikcmd/3.7_LR_TR0.git
Cloning into '3.7_LR_TR0'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 2 – Клонирование репозитория

4. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.



```
D:\fgit\3.7_LR_TR0>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [D:/fgit/3.7_LR_TR0/.git/hooks]
```

Рисунок 3 – Модель git-flow

5. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.

6. Проработать примеры лабораторной работы в отдельном ноутбуке.

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

In [5]: img = cv2.imread('image.jpg', 1)

        plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
        plt.xticks([])
        plt.yticks([])
        plt.show()
```

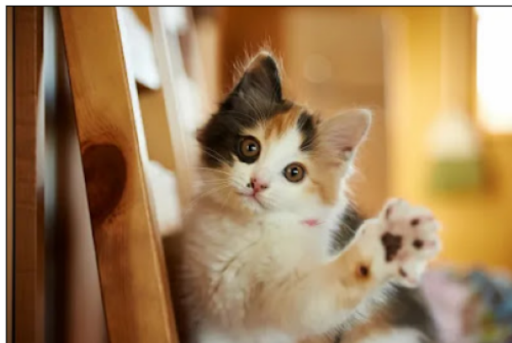


Рисунок 4 – Примеры лабораторной работы

Задание 2

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

```
In [8]: img = cv2.imread('image.jpg', cv2.IMREAD_COLOR)

        plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
        plt.xticks([])
        plt.yticks([])
        plt.show()
```

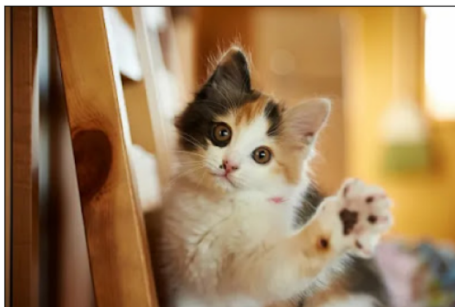


Рисунок 5 – Примеры лабораторной работы

Задание 3

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

```
[11]: img = cv2.imread('image.jpg')  
  
# Запись изображения из матрицы в файл  
cv2.imwrite('img.png', img)  
img = cv2.imread('img.png')  
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))  
plt.xticks([])  
plt.yticks([])  
plt.show()
```



Рисунок 6 – Примеры лабораторной работы

Задание 4

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

```
In [32]: # 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(0, i):  
        a[i][j] = 0  
  
cv2.imwrite('matr.png', a) # Записываем изображение в файл  
img = cv2.imread('matr.png', 0) # считываем изображение с файла
```

```
In [33]: print(img)  
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))  
plt.xticks([])  
plt.yticks([])  
plt.show()
```

```
[[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]  
[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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 0 0]
```

Рисунок 7 – Примеры лабораторной работы

Задание 5

Вывести свойства матрицы изображения на экран.

```
In [30]: img = cv2.imread('image.jpg', 0)

plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.xticks([])
plt.yticks([])
plt.show()
```



```
In [31]: print(type(img))
print(img.shape) # Кортеж числа строк и столбцов (разрешение)
print(img.size) # Общее количество пикселей
print(img.dtype) # Тип данных изображения
```

Рисунок 8 – Примеры лабораторной работы

Задание 6

Определить с помощью функции `print(img.shape)` максимальное число пикселей по ширине и высоте изображения. Выбрать 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.

```
In [17]: img = cv2.imread('image.jpg')

print(img.shape) # Определение размера матрицы

# Доступ к пикселю цветного изображения с координатами 100, 150
px = img[100, 150]
print(px)

# Доступ только к синему пикселю с координатами (100, 150)
blue = img[100, 150, 0]
print(blue)

img[100, 150] = [105, 139, 185]
print(img[100, 150])

img.item(100,150,2)
img.itemset((100,150,2),100)

(267, 400, 3)
[ 99 125 155]
99
[105 139 185]
```

Рисунок 9 – Примеры лабораторной работы

Задание 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.

```
In [19]: img = cv2.imread("image.jpg")
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

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

gray_img = cv2.imread("image.jpg", 0) # полутоновое изображение
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')

im_bwa = cv2.threshold(img, 128, 255, cv2.THRESH_BINARY)[1]

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

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()
cv2.waitKey(0)
cv2.destroyAllWindows()
```



Рисунок 10 – Примеры лабораторной работы

Задание 8

На заданном изображении выделить его характерный участок.

```
In [28]: img = cv2.imread("image.jpg")
image = cv2.rectangle(img, (190,150), (210,130), (0,0,255), 2)

In [29]: plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.xticks([])
plt.yticks([])
plt.show()
```

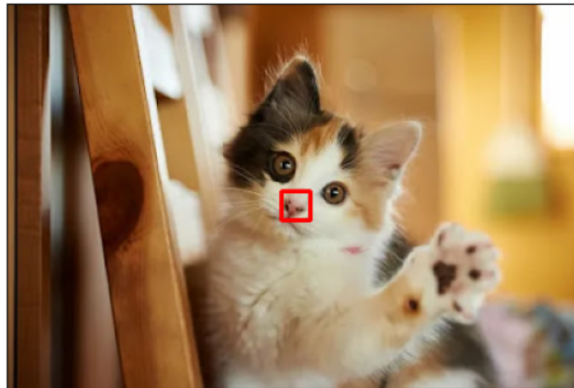


Рисунок 11 – Примеры лабораторной работы

Задание 9

Уменьшить заданное изображение и вывести на печать матрицу уменьшенного изображения. Нам надо сохранить соотношение сторон, чтобы изображение не исказилось при уменьшении.

```
In [26]: img = cv2.imread('image.jpg')
final_wide = 200
r = float(final_wide) / img.shape[1]
dim = (final_wide, int(img.shape[0]*r))
resized = cv2.resize(img,dim,interpolation=cv2.INTER_AREA)
```

```
In [27]: plt.subplot(221)
plt.imshow(cv2.cvtColor(resized,cv2.COLOR_BGR2RGB))
print(resized.shape)
print(resized)
plt.xticks([])
plt.yticks([])
```

```
img = cv2.imread('image.jpg', 0)
plt.subplot(222)
plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
print(img)
plt.xticks([])
plt.yticks([])
plt.show()
```

```
(133, 200, 3)
[[[ 38  71 107]
 [ 33  71 109]
 [ 26  50  76]
 ...
 [254 254 254]
 [254 254 254]
 ...]]]
```

Рисунок 12 – Примеры лабораторной работы

Задание 10

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

```
In [34]: img = cv2.imread("image.jpg", 1)
img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
img = cv2.bitwise_not(img) # функция инвертирования изображения

plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
plt.xticks([])
plt.yticks([])
plt.show()
```



Рисунок 13 – Примеры лабораторной работы

7. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.).

Условие. Считать цветное изображение и выполнить вывод, следующих получившихся изображений: конвертировать цвета

изображения инверсивно, сделать цвета более яркими и насыщенными, убрать красный и синий цвет.

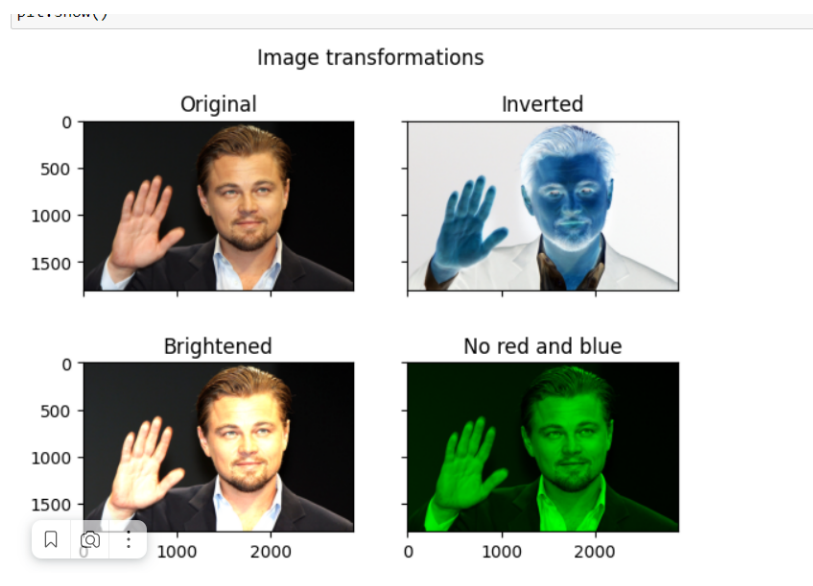


Рисунок 14 – Результат работы программы

8. Зафиксируйте сделанные изменения в репозитории.
9. Выполните слияние ветки для разработки с веткой main (master).
10. Отправьте сделанные изменения на сервер GitHub.