

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО
ОБРАЗОВАНИЯ

**РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ЛИПЕЦКИЙ ФИЛИАЛ

КАФЕДРА ГУМАНИТАРНЫХ И ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН

Лабораторная работа №4

по дисциплине

«Информационные технологии в управлении »

ВАРИАНТ №1

Выполнил: Колесников Михаил

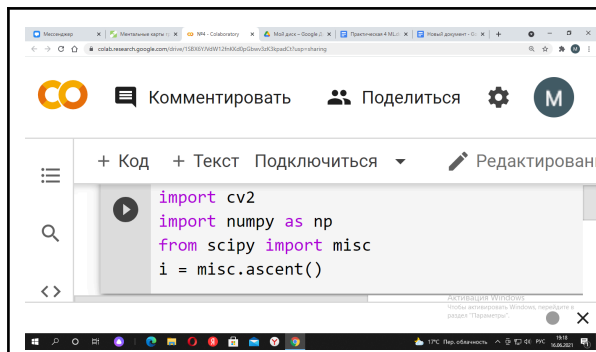
студент 1 курса

группы У20-1

Проверил:

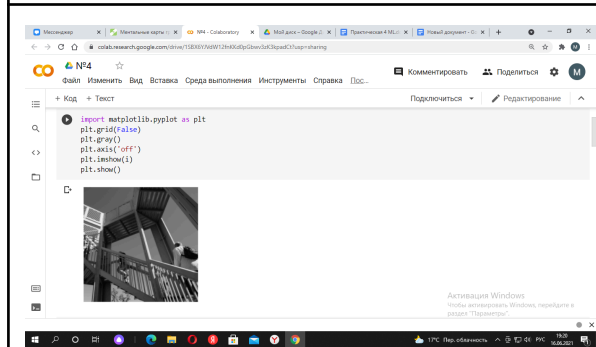
доцент

Яворский В.М.



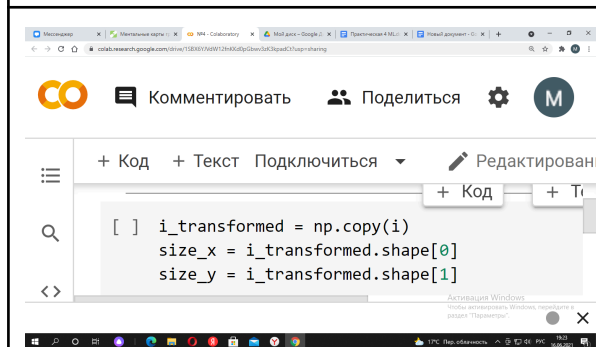
```
import cv2
import numpy as np
from scipy import misc
i = misc.ascent()
```

Загружаем изображение с помощью пакета “Scipy”



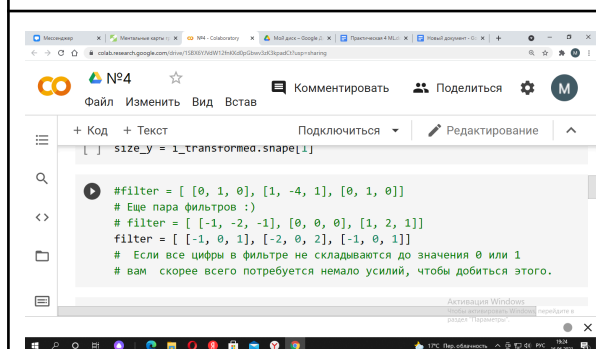
```
import matplotlib.pyplot as plt
plt.grid(False)
plt.gray()
plt.axis('off')
plt.imshow(i)
plt.show()
```

Использую библиотеку pyplot, чтобы нарисовать изображение



```
[ ] i_transformed = np.copy(i)
size_x = i_transformed.shape[0]
size_y = i_transformed.shape[1]
```

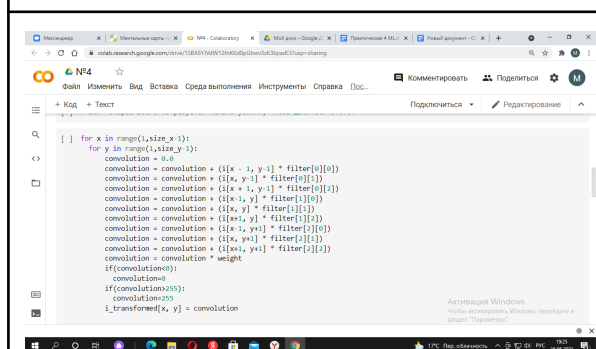
Изображение хранится в виде массива numpy, поэтому мы можем создать преобразованное новое изображение, просто скопировав этот массив. Давайте также получим размеры изображения, чтобы потом можно было пройти через значения массива/матрицы через итерирование в цикле.



```
[ ] size_y = i_transformed.shape[1]

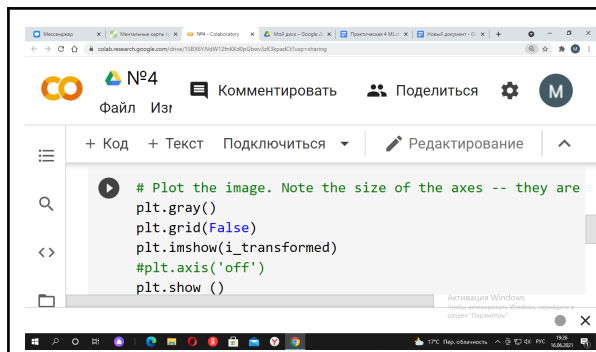
#filter = [ [0, 1, 0], [1, -4, 1], [0, 1, 0] ]
# Еще пара фильтров :)
# filter = [ [-1, -2, -1], [0, 0, 0], [1, 2, 1] ]
filter = [ [-1, 0, 1], [-2, 0, 2], [-1, 0, 1] ]
# Если все цифры в фильтре не складываются до значения 0 или 1
# вам скорее всего потребуется немало усилий, чтобы добиться этого.
```

Теперь мы можем создать фильтр в виде массива/матрицы размером 3x3.

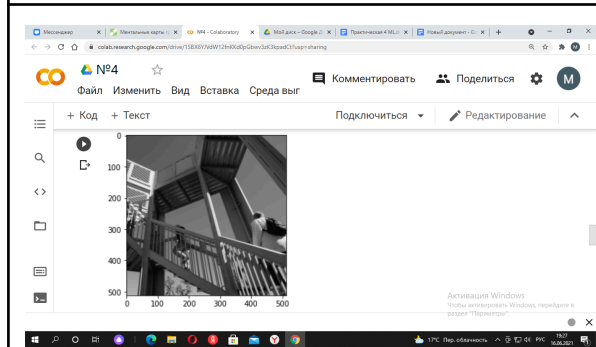


```
[ ] for x in range(1,size_x-1):
    for y in range(1,size_y-1):
        convolution = 0.0
        convolution = convolution + (i[x-1, y-1] * filter[0][0])
        convolution = convolution + (i[x, y-1] * filter[0][1])
        convolution = convolution + (i[x+1, y-1] * filter[0][2])
        convolution = convolution + (i[x-1, y] * filter[1][0])
        convolution = convolution + (i[x, y] * filter[1][1])
        convolution = convolution + (i[x+1, y] * filter[1][2])
        convolution = convolution + (i[x-1, y+1] * filter[2][0])
        convolution = convolution + (i[x, y+1] * filter[2][1])
        convolution = convolution + (i[x+1, y+1] * filter[2][2])
        convolution = convolution * weight
        if(convolution>0):
            if(convolution>255):
                i_transformed[x, y] = convolution
```

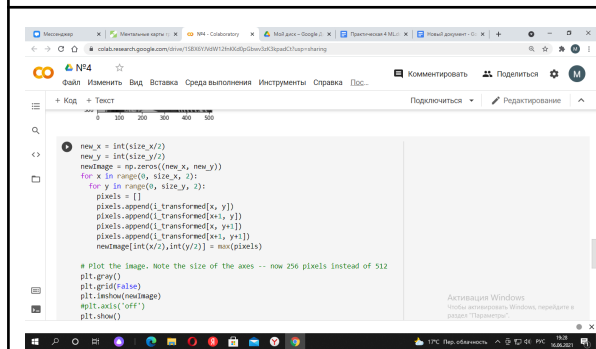
Теперь давайте создадим конволюцию.



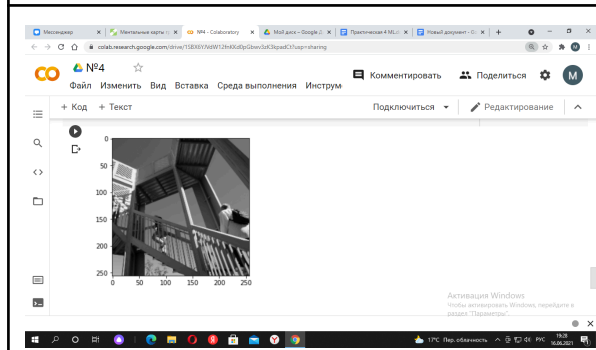
Теперь мы можем построить изображение, чтобы увидеть эффект свёртки!



Получаю изображение.



Этот код покажет (2, 2) операцию подвыборки. Идея здесь в том, чтобы провести итерирование по изображению, и посмотреть на пиксель, а также те, которые находятся в непосредственной близости справа, внизу и справа-внизу по диагонали. Возьмите из них самый большой по значению и оставьте его, загрузив в новое изображение. Таким образом, новое изображение будет 1/4 размера старого - при этом размеры по каждой из осей X и Y будут уменьшены вдвое. Вы увидите, что, несмотря на это сжатие в 1/4, признаки изображения сохраняются!



Получаю изображение.