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 №5

по дисциплине

«Информационные технологии в государственном и муниципальном управлении»

Выполнила:

студентка 1 курса

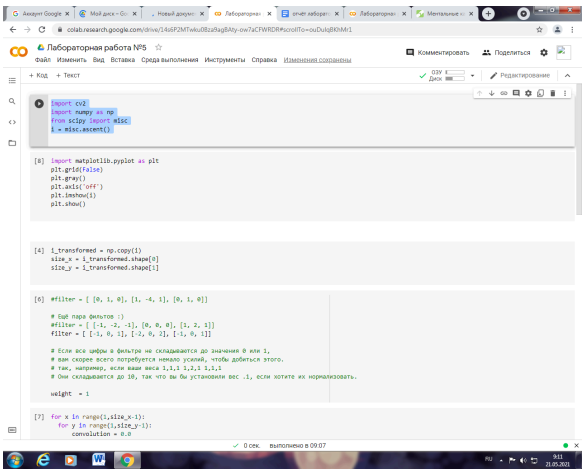
группы У20-1

Александрова Алина

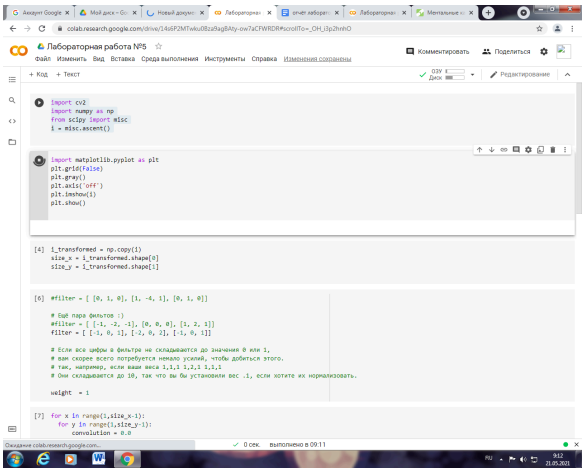
Проверил:

доцент

Яворский В.М.

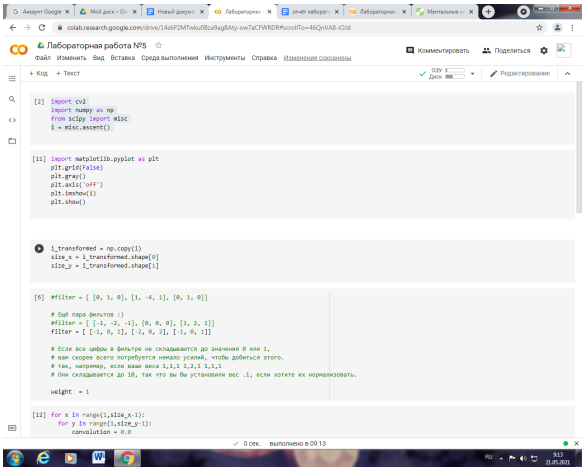


Создаём новый блокнот по ссылке: https://colab.research.google.com/; И называем блокнот.



Сначала мы загружаем изображение из пакета "Scipy". Для этого мы добавляем новый код:

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

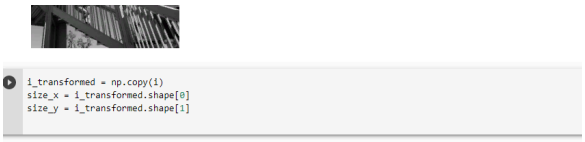


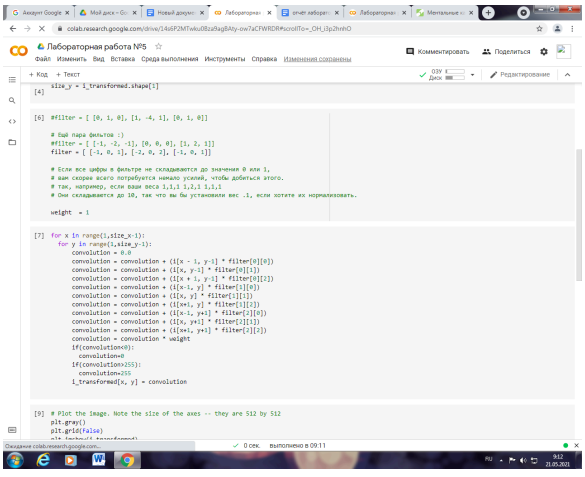
Далее мы используем библиотеку “pyplot” для того, чтобы нарисовать изображение, чтобы мы знали, как оно выглядит.

Добавляем новый код:

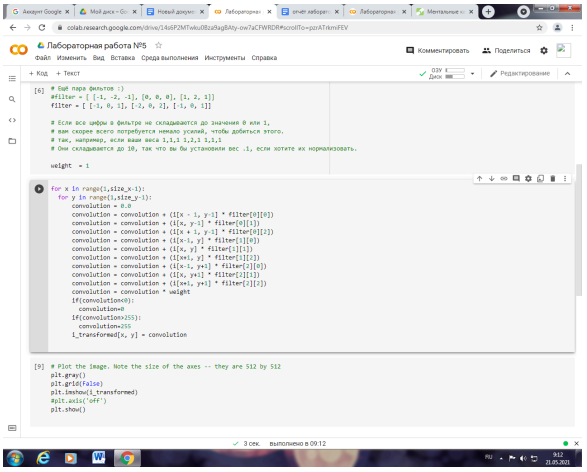
```
import matplotlib.pyplot as plt

plt.grid(False)

plt.gray()

plt.axis('off')

plt.imshow(i)

plt.show()
```

	Получаем изображение.

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

	Теперь мы можем создать фильтр в виде массива/матрицы размером 3x3.
	Добавляем новый код:
	`#filter = [ [0, 1, 0], [1, -4, 1], [0, 1, 0]]`
	
	`# Ещё пара фильтов :)`
	`#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.`

```
# так, например, если ваши веса 1 ,1,1 1,2,1 1,1,1
# Они складываются до 10, так что вы бы установили вес .1, если хотите их нормализовать.

weight  = 1
```

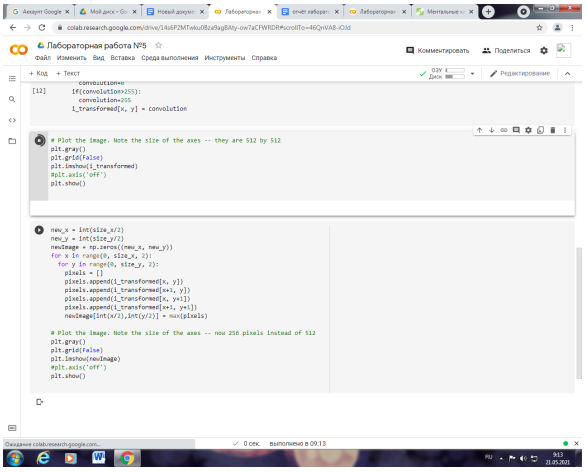


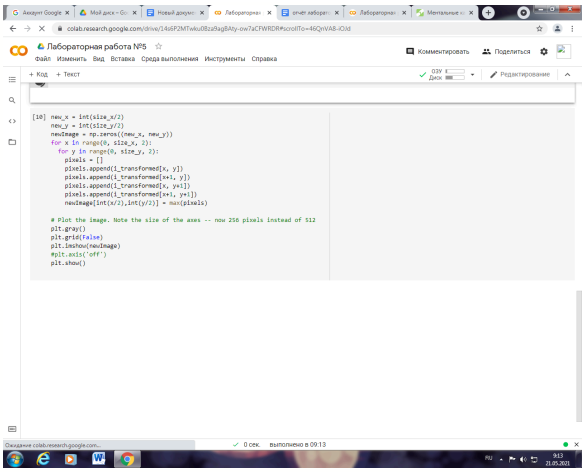
Теперь создаём конволюцию. Выполним итерацию по изображению, оставив поле в 1 пиксел, и умножим каждого из соседей текущего пиксела на значение, заданное в фильтре.

Добавляем новый код:

```
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):`
	
	`convolution=0`
	
	`if(convolution>255):`
	
	`convolution=255`
	
	`i_transformed[x, y] = convolution`

	**Шаг №8.**Теперь мы можем построить изображение,чтобы увидеть эффект свёртки. Добавляем новый код:
	
	`# Plot the image. Note the size of the axes -- they are 512 by 512`
	
	`plt.gray()`
	
	`plt.grid(False)`
	
	`plt.imshow(i_transformed)`
	
	`#plt.axis('off')`
	
	`plt.show()`
	
	и получаем изображение

	**Шаг №9.**
	Данный код покажет (2, 2) операцию подвыборки
	
	`new_x = int(size_x/2)`
	
	`new_y = int(size_y/2)`
	
	`newImage = np.zeros((new_x, new_y))`
	
	`for x in range(0, size_x, 2):`
	
	`for y in range(0, size_y, 2):`

```
    pixels = []


pixels.append(i_transformed[x,
y])


pixels.append(i_transformed[x+1,
y])


pixels.append(i_transformed[x,
y+1])


pixels.append(i_transformed[x+1,
y+1])

    newImage[int(x/2),int(y/2)] =
max(pixels)



# Plot the image. Note the size
of the axes -- now 256 pixels
instead of 512

plt.gray()

plt.grid(False)

plt.imshow(newImage)

#plt.axis('off')

plt.show()
```

После чего получаем изображение