

## Практическое задание №3

### Задача №10

**Условие:** Дан трёхмерный массив, содержащий изображение, размера (height, width, numChannels), а также вектор длины numChannels. Написать функцию, которая складывает каналы изображения с указанными весами, и возвращает результат в виде матрицы размера (height, width)

#### Решение:

Входные данные представляют из себя три строки: в первой содержатся все размеры: height, width, длина numChannels, во второй строке вектор numChannels с весами каналов изображения. В третьей – элементы трехмерной матрицы (в количестве height\* width\* numChannels), представленные в одной строке.

Пример входных данных:

2 2 3

1 3 1

1 1 2 4 5 6 1 2 3 4 5 6

Программа считывает данные. С помощью функции `numpy.reshape`, в которую подается вектор элементов трехмерной матрицы, и вектор размерностей, получается тензор третьего ранга - `dataImg`.

`dataImg:`

```
[[[1. 1. 2.]
```

```
 [4. 5. 6.]]
```

```
[[1. 2. 3.]
```

```
 [4. 5. 6.]]]
```

С помощью функции `dot`, из библиотеки `numpy`, происходит суммирование каналов, помноженных на их веса, каждого пикселя изображения. На выходе получается двумерный тензор `imgOutput` размера height, width.

Пример работы функции `dot` на входных данных:

$$\text{imgOutput}[0][0] = 6 = 1*1 + 3*1 + 1*2 = \text{dataImg}[0][0][0]*\text{numChannels}[0] + \text{dataImg}[0][0][1]*\text{numChannels}[1] + \text{dataImg}[0][0][2]*\text{numChannels}[2]$$
$$\text{imgOutput}[0][1] = 25 = 1*4 + 3*5 + 1*6 = \text{dataImg}[0][1][0]*\text{numChannels}[0] + \text{dataImg}[0][1][1]*\text{numChannels}[1] + \text{dataImg}[0][1][2]*\text{numChannels}[2]$$

...

Результат:

[[ 6 25]

[10 25]]

Запись данных в файл производится с помощью функции `savetxt`.