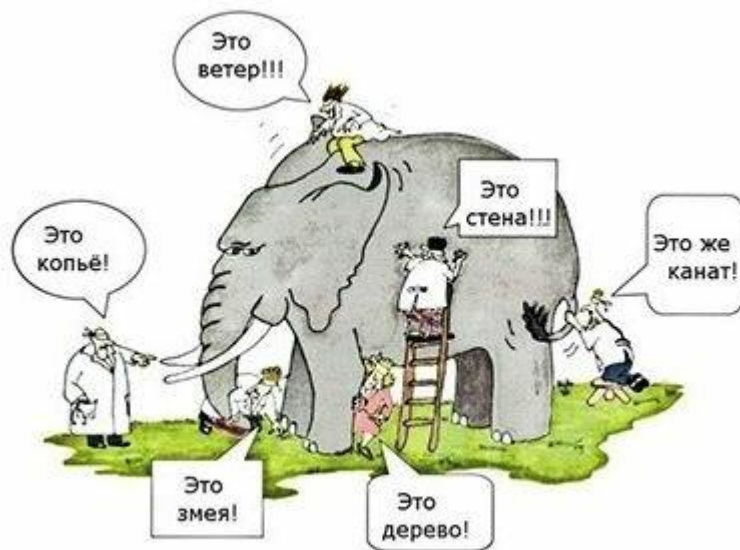


Семинар 3

Работа с изображениями

Как посмотреть на вопрос?



Библиотека PILOW (PIL)

- ▶ Чтение графического формата
- ▶ Преобразование графических форматов
- ▶ Редактирование изображение: вырезать, вставить, работа с пикселями

Библиотека PILOW (PIL)

- ▶ Пример работы с библиотекой:
- ▶ colab.research.google.com/drive/1Sdxc0nsYO5xlyVNru_irlgLop8TLfdPu#scrollTo=PtnKWkgDvXXh

NumPy для обработки изображений

NumPy—это одна из основных Python-библиотек с поддержкой массивов. Изображение представляет собой стандартный массив NumPy, содержащий пиксели точек данных. Таким образом, при выполнении основных NumPy-операций (срезы, маски) мы можем изменять пиксельные значения изображения. Само изображение можно загрузить через `skimage` и отобразить с помощью `Matplotlib`.

Возможности SciPy

- ▶ поиск минимумов и максимумов функций;
- ▶ вычисление интегралов функций;
- ▶ поддержка специальных функций;
- ▶ обработка сигналов;
- ▶ обработка изображений;
- ▶ работа с генетическими алгоритмами;
- ▶ решение обыкновенных дифференциальных уравнений;
- ▶ и др.
- ▶ Целевая аудитория — пользователи продуктов MATLAB и Scilab.

SciPy для обработки изображений

SciPy—это такой же важный научный модуль в Python, как и NumPy. Он подходит для решения основных задач по обработке и прочей работе с изображениями. В частности, в подмодуле [scipy.ndimage](#) доступны функции, которые работают в n-мерных массивах NumPy. Текущий пакет включает в себя функции для линейной и нелинейной фильтрации, бинарной морфологии, интерполяции B-сплайнами и измерений объектов.

<https://docs.scipy.org/doc/> - инфo по библиотеке

SciPy для обработки изображений

- ▶ Внимание на тренд
- ▶ Детектирование деталей
- ▶ Избавление от шумов

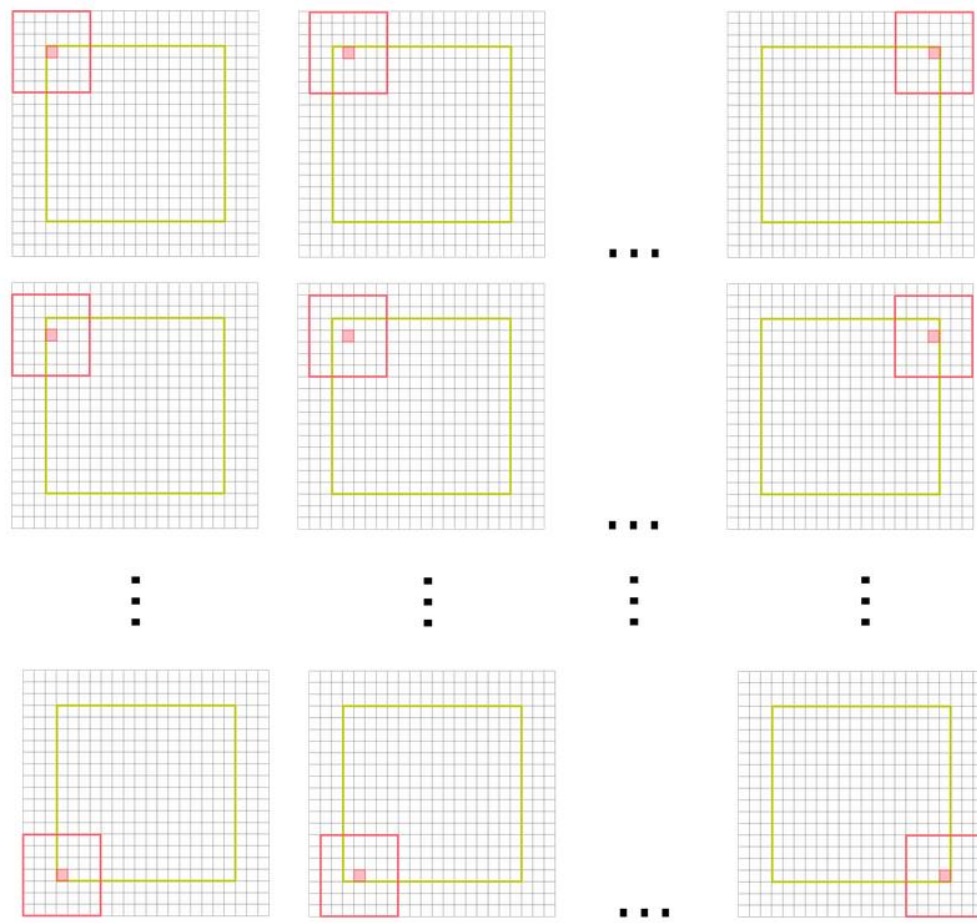
Пример работы с SciPy для анализа изображений

- ▶ colab.research.google.com/drive/1Sdxc0nsYO5xlyVNru_irlgLop8TLfdPu#scrollTo=PtnKWkgDvXXh

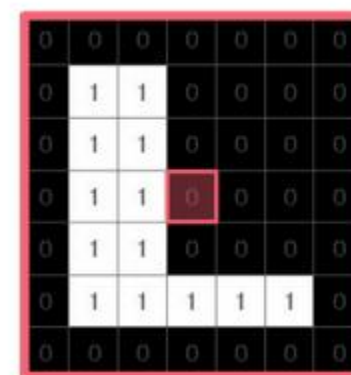
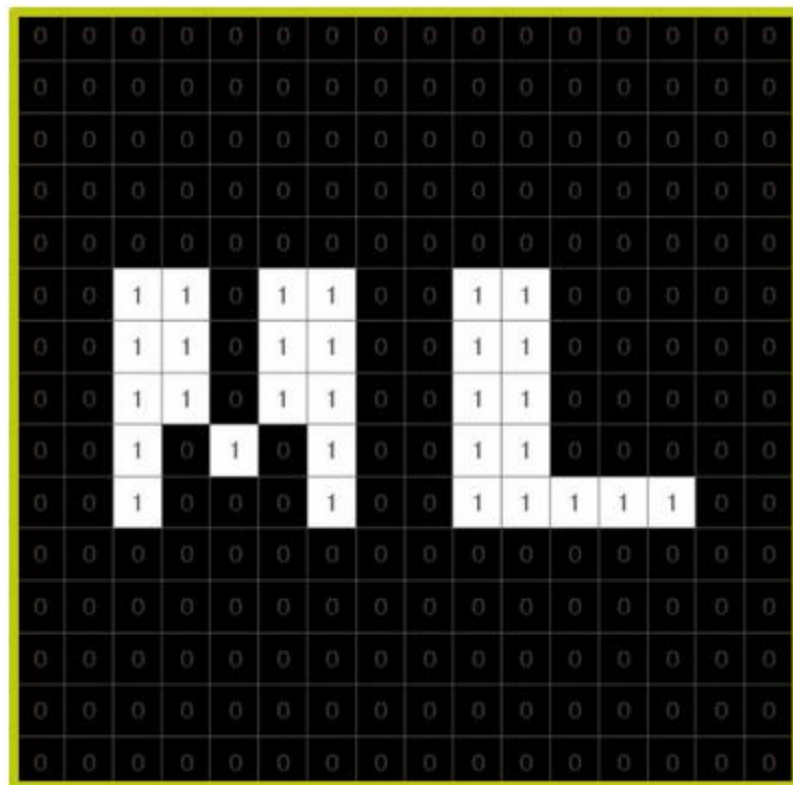
Свертка изображений

Свертка - это операция вычисления нового значения выбранного пикселя, учитывающая значения окружающих его пикселей. Для вычисления значения используется матрица, называемая *ядром свертки*. Обычно ядро свертки является квадратной матрицей $n \times n$, где n — нечетное, однако ничто не мешает сделать матрицу прямоугольной. Во время вычисления нового значения выбранного пикселя ядро свертки как бы «прикладывается» своим центром (именно тут важна нечетность размера матрицы) к данному пикселю. Окружающие пиксели так же накрываются ядром. Далее высчитывается сумма, где слагаемыми являются произведения значений пикселей на значения ячейки ядра, накрывшей данный пиксель. Полученное значение как раз и является новым значением выбранного пикселя. Если применить свертку к каждому пикселю изображения, то в результате получится некий эффект, зависящий от выбранного ядра свертки.

Работа ядра (матрицы)



Свертка



Пример работы ядра свертки

Пример:

пусть у нас есть клеточное поле, которое соответствует пикселям исходного изображения:

[illegible]

и есть ядро свёртки, представляющее собой матрицу 3×3 с якорем в центре и элементами:

```
[0][1][0]
[0][0][0]
[0][0][0]
```

Результат операции наложения нашего ядра на пиксель со значением 50:

[] [] [] [] [] [] [] [] [] []
 [] [48] [] [] [] [] [] [] [] [] []
 [] [] [] [] [] [] [] [] [] []
 [] [] [] [] [] [] [] [] [] []
 [] [] [] [] [] [] [] [] [] []

Всё очень просто: результат = $47*0 + 48*1 + 49*0 + 47*0 + 50*0 + 42*0 + 47*0 + 48*0 + 42*0 = 48$

A 20x20 grid of numbers from 0 to 9. A red box highlights a 5x5 subgrid from row 4 to row 8 and column 2 to column 6. A yellow box highlights a 10x10 subgrid from row 4 to row 13 and column 2 to column 11. The number 3 is located at row 5, column 4.

Эффекты с помощью ядра свертки

► # никаких изменений

```
kernel[0]=0;  
kernel[1]=0;  
kernel[2]=0;
```

```
kernel[3]=0;  
kernel[4]=1;  
kernel[5]=0;
```

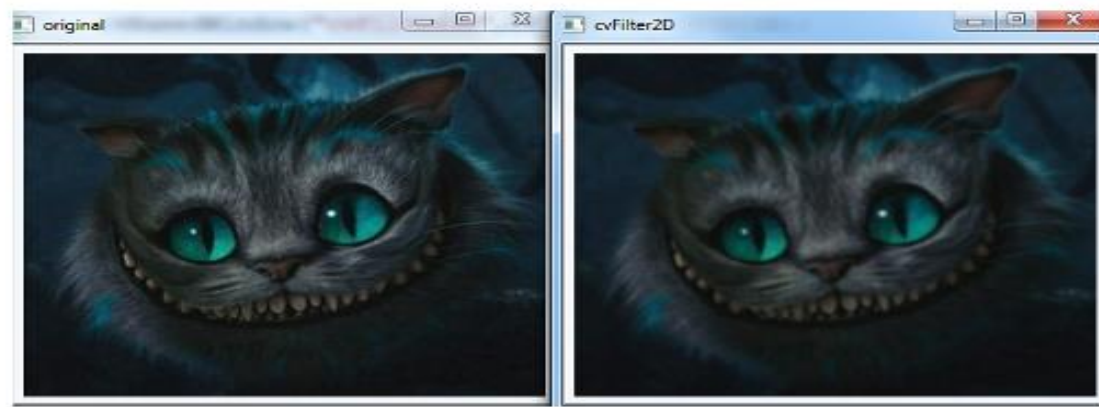
```
kernel[6]=0;  
kernel[7]=0;  
kernel[8]=0;
```

► # сглаживание

```
kernel[0]=0.1;  
kernel[1]=0.1;  
kernel[2]=0.1;
```

```
kernel[3]=0.1;  
kernel[4]=0.1;  
kernel[5]=0.1;
```

```
kernel[6]=0.1;  
kernel[7]=0.1;  
kernel[8]=0.1;
```



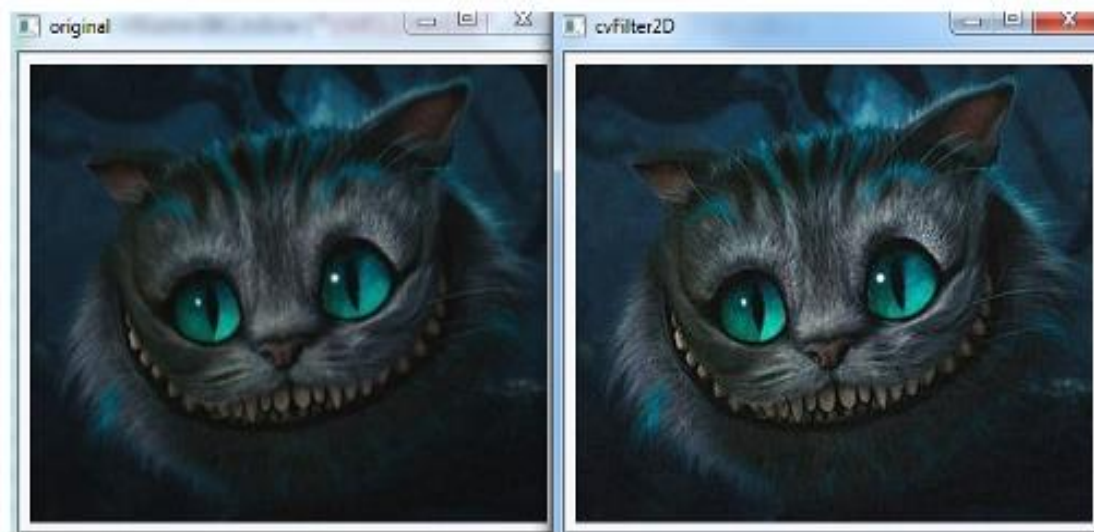
Эффекты с помощью ядра свертки

- # увеличение чёткости

```
kernel[0]=-0.1;  
kernel[1]=-0.1;  
kernel[2]=-0.1;
```

```
kernel[3]=-0.1;  
kernel[4]=2;  
kernel[5]=-0.1;
```

```
kernel[6]=-0.1;  
kernel[7]=-0.1;  
kernel[8]=-0.1;
```



Эффекты с помощью ядра свертки

► # увеличение яркости

```
kernel[0]=-0.1;  
kernel[1]=0.2;  
kernel[2]=-0.1;
```

```
kernel[3]=0.2;  
kernel[4]=3;  
kernel[5]=0.2;
```

```
kernel[6]=-0.1;  
kernel[7]=0.2;  
kernel[8]=-0.1;
```



Задание

- ▶ Загрузить «Smiles.jpg»
- ▶ Вырезать из исходной картинки лица 2 парней и 2 девушек (фрагмент)
- ▶ Сделать фрагмент серым
- ▶ Поместить выделенный фрагмент в черную рамку
- ▶ Выделить контуры изображения выбранным Вами фильтром