

Анализ изображений с помощью scikit-image

Амир Мирас

ВМК МГУ

Курс «Практикум на ЭВМ» для 317-ой группы

- Анализ изображений – извлечение значимой информации из изображений
- Что полезного можно извлечь?
 - Медицина (выявление аномалий, диагностика заболеваний)
 - Системы безопасности (обнаружение «подозрительных» предметов)
 - Военная промышленность (системы слежения и целенаведения)
 - Обнаружение и распознавание текста

- Библиотека обработки изображения для языка Python
- Открытый исходный код
- Написана на языках Python, Cython и C
- Алгоритмы сегментации, геометрических преобразований, фильтрации, морфологии, обнаружения признаков и т.д.
- Менее специализирована, чем другие библиотеки (например, OpenCV для компьютерного зрения)



scikit-image
image processing in python

```
>>> from skimage import data, io, filters
>>> camera_image = data.camera()
>>> type(camera_image)
numpy.ndarray
>>> camera_image.dtype
dtype('uint8')
>>> io.imshow(camera_image)
```



```
>>> filtered_image = filters.sobel(camera_image)
>>> filtered_image.dtype
dtype('float64')
>>> camera_image.max(), camera_image.min()
(255, 0)
>>> filtered_image.max(), filtered_image.min()
(0.5915023652179584, 0.0)
>>> io.imshow(filtered_image)
```



```
>>> camera_image().shape
(512, 512)
>>> coffee_image = data.coffee()
>>> coffee_image.shape
(400, 600, 3)
```

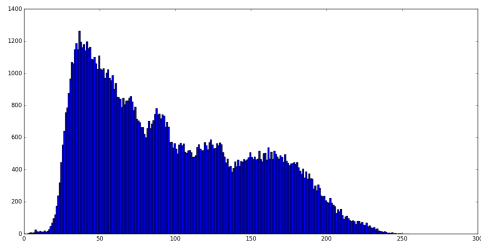


```
>>> red, green, blue = coffee_image.copy(),  
                        coffee_image.copy(),  
                        coffee_image.copy()  
>>> red[:, :, (1, 2)] = 0  
      green[:, :, (0, 2)] = 0  
      blue[:, :, (0, 1)] = 0
```

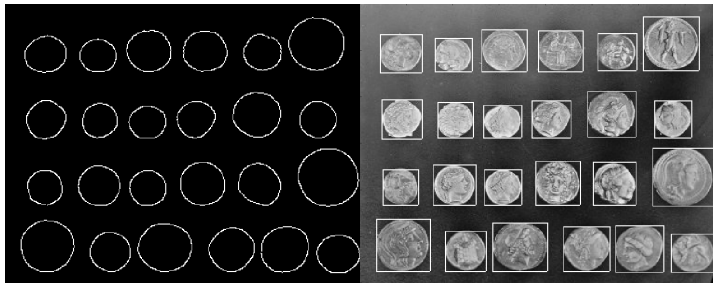


- Фильтрация ([skimage.filters](#))
- Математическая морфология ([skimage.morphology](#))
- Извлечение признаков ([skimage.feature](#))
- Геометрические преобразования ([skimage.transform](#))
- Сегментация ([skimage.segmentation](#))






```
>>> from skimage import feature
>>> from skimage import measure
>>> image = data.coins()
>>> plt.hist(image.flat, bins=np.arange(256))
```



```
>>> edges = feature.canny(image, sigma=3,  
                           low_threshold=10,  
                           high_threshold=80)  
>>> label_image = measure.label(edges)  
>>> image_copy = image.copy()  
    for region in measure.regionprops(label_image):  
        minr, minc, maxr, maxc = region.bbox  
        image_copy[minr:maxr, (minc, maxc)] = 255  
        image_copy[(minr, maxr), minc:maxc] = 255
```



- OpenCV
- Mahotas
- Image processing toolbox (Matlab)

-  Официальный сайт scikit-image
<http://scikit-image.org/>
-  Открытый код scikit-image
www.github.com/scikit-image/scikit-image
-  Image analysis in Python with scipy | SciPy 2014
www.youtube.com/watch?v=SE7h0IWD93Y
-  Scikit-image: image processing
www.scipy-lectures.org/packages/scikit-image/