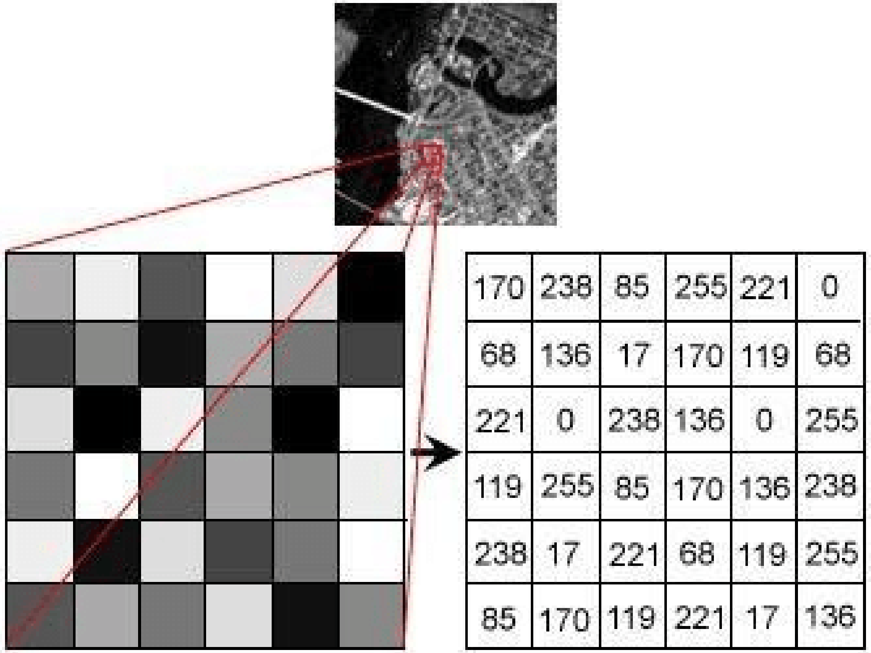
FAIM Python Course – Session 3

Image Processing

benjamin.titze@fmi.ch

Outline

- Prerequisites: Python 3.6 and Jupyter Notebook
  - Libraries: pillow (PIL), scikit-image, NumPy, SciPy, Matplotlib
- Overview: Digital image processing
- Introduction to the libraries
- Examples of common tasks (in Jupyter Notebook)
  - Loading, manipulating and saving images
  - Extracting image statistics
  - Applying filters/thresholds
  - Object detection



170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

Illustration: Goutam Das

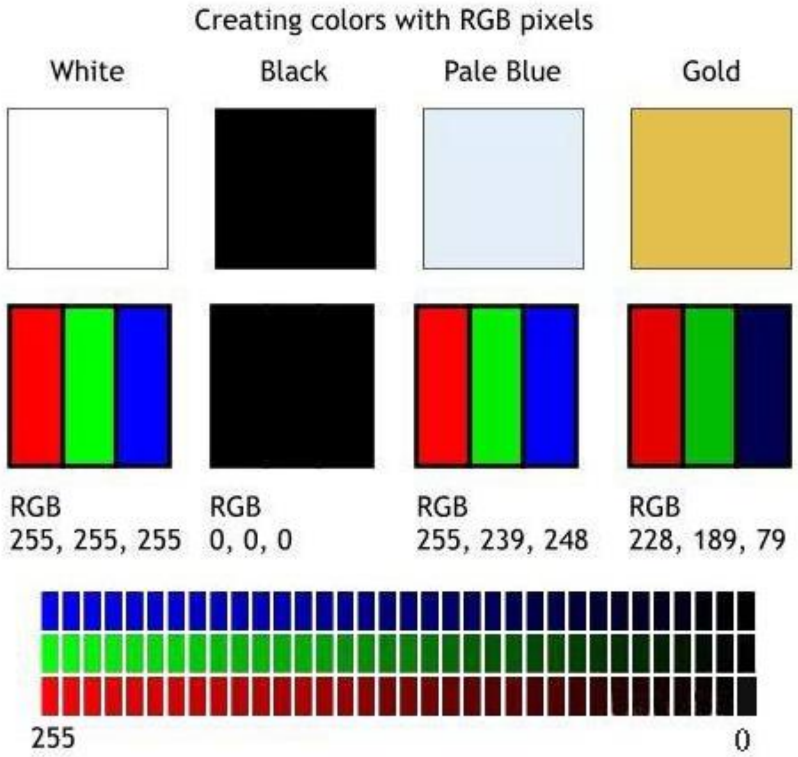


Illustration: www.panadisplay.com

A digital image is a 2D array of pixels.

Every pixel has a coordinate (x/y = column position/row position starting from 0/0 in upper left corner) and an intensity value (typically 8 bit or 16 bit for greyscale, or 3 × 8 bit = 24 bit for RGB).

(Bio)Image Processing/Analysis

Common tasks:

- I/O: Loading images into memory for processing, saving (processed) image data from memory to disk, converting between formats and bit-depths (tiff, png, jpg…; 8 bit, 16 bit…)
- Image manipulation: Brightness & contrast; Cropping, resizing, rotation, flipping, …
- Image statistics: Mean, standard deviation, histogram…
- Applying filters: Denoising, edge detection…
- Extraction of relevant information: Spot detection, segmentation…

Please note that this session is not aimed at teaching fundamental image processing concepts (see Jan’s introductory courses for that), but at giving you an overview how Python can be used for image processing.

NumPy

(pronounced *Num Pie*) – https://numpy.org
A library for fast manipulation of multidimensional numerical data. The basic type *ndarray* is a multidimensional array with zero-based indexing.

SciPy

(pronounced *Sigh Pie*) – https://www.scipy.org
A scientific computing library (built with NumPy arrays) with various modules for numerical integration, linear algebra, image processing, Fourier transforms, signal processing, statistical functions…

Recommended as default choice for most applications

scikit-image
image processing in python

https://scikit-image.org
Image processing toolkit with lots of useful functionality in one place. Started out as a SciPy toolkit (= ‘scikit’) in 2009. Works very well with NumPy and SciPy. As of 2020 probably the best choice for your image processing needs. Imported as **skimage**.

pillow
The friendly PIL fork

https://python-pillow.org
pillow is a fork of the Python Imaging Library (PIL), a project to provide basic imaging processing capabilities in Python. PIL was discontinued in 2011. The package is called ‘pillow’, but in Python it’s used with ‘import PIL’.

OpenCV

https://opencv.org
OpenCV (CV for Computer Vision) is a cross-platform library written in C/C++, started by Intel, now developed by the non-profit OpenCV foundation. Can be used in Python with the wrapper package **opencv-python**.

… and there are more packages to explore: https://opensource.com/article/19/3/python-image-manipulation-tools

Hands-on examples in Jupyter Notebook:

1. Load, transform and save images with Pillow (PIL)
   → loading-and-manipulating-images.ipynb
2. Images as NumPy arrays, image statistics, histograms
   → image-statistics.ipynb
3. Applying filters and thresholds
   → filtering-and-thresholds.ipynb
4. Blob detection (detect objects in Hubble image)
   → blob-detection.ipynb
5. Segmentation example (nuclear division time series)
   → segmentation-example.ipynb