

# Analyse d'images

Le traitement d'images permet de transformer des images. L'analyse d'images permet d'extraire des informations contenues dans une image. Il est aussi possible d'effectuer des tâches plus complexes de reconnaissance et d'analyse de scènes.

## Utilisation de la librairie Pillow

La librairie [Pillow](#) remplace la librairie PIL, avec des appels identiques.

### Exemple de base

Télécharger l'image exemple au format jpg dans le même répertoire que le programme python

[image\\_import-02.py](#)

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
"""
show jpg image
image jpg :
https://commons.wikimedia.org/wiki/File:Antoine_lavoisier_color.jpg
pillow : http://pillow.readthedocs.io/en/latest/index.html
"""

from PIL import Image
# install python3 module pillow using e.g. conda install

# Load the image file
img =
Image.open('/home/villersd/Images/photos/Antoine_lavoisier_color.jpg')
img.show()
```

### Exemple avec traitement numérique

Voici un exemple de lecture d'une image, de conversion en monochrome et ensuite en tableau de données numériques de NumPy. Une transformée de Fourier bidimensionnelle est effectuée, ainsi que quelques autres traitements, visualisations, ainsi qu'une sauvegarde :

[FFT-structures\\_2D\\_01.py](#)

```
#!/usr/bin/env python
```

```
# -*- coding: utf-8 -*-
"""
Digital Image processing with PIL

"""
# sources & refs :
http://stackoverflow.com/questions/2652415/fft-and-array-to-image-image-to-array-conversion
#
http://stackoverflow.com/questions/14577007/grayscale-image-to-numpy-array-for-fourier-transform
# http://optipng.sourceforge.net/pngtech/img/lena.png
#
http://stackoverflow.com/questions/14452824/saving-image-in-python-pil/14472089#14472089

import PIL
import Image
import numpy as np
import matplotlib.pyplot as plt

dirfile=""
#filename="x10001-512.tif"
filename="ech14x50001-512.jpg"
im = Image.open(str(dirfile+filename))
im = im.convert('L') #convert to grayscale
im.show()
data = np.asarray(im) # a is readonly
print type(data), data.ndim, data.shape, data.dtype
datafft = abs(np.fft.rfft2(data))
datafft[0, 0] = 0 # remove DC component for visualization
print type(datafft), datafft.ndim, datafft.shape, datafft.dtype
plt.imshow(np.abs(np.fft.fftshift(datafft)), interpolation='nearest')
plt.show()
plt.imshow(np.abs(np.fft.fftshift(datafft))[192:320, 192:320],
interpolation='nearest')
plt.show()
im2 = Image.fromarray(datafft)
im2.show()
im2.save('img2.gif')
```

L'[image de départ](#) correspond à un film mince de polydiméthylsiloxane traité sous plasma 30 minutes, de manière à obtenir une contrainte interne de compression dans le film (15% dans une seule direction), ce qui provoque la déformation de l'interface et l'apparition de rides (UMONS, lab. Interfaces et Fluides Complexes).

Pour approfondir l'utilisation de PIL, consulter ce [tutoriel en français](#) sur des bases de traitement d'images avec PIL et numpy

## Références

- [PIL](#) : Python Imaging Library et la [documentation sur effbot](#)
- [Pillow](#), un fork pour le remplacement et l'évolution de PIL

## Scikit-image

- Voir la galerie d'exemples sur le site <http://scikit-image.org/>

## OpenCV

- [OpenCV](#) : Open Source Computer Vision (site officiel)
- [SimpleCV](#) : interface simplifié à OpenCV : "computer vision made easy"

Références diverses :

- [http://www.bogotobogo.com/python/OpenCV\\_Python/python\\_opencv3\\_image\\_histogram\\_calHist.php](http://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_image_histogram_calHist.php)

## pymaging

- [pymaging](#) (pure python)

From:

<https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link:

[https://dvillers.umons.ac.be/wiki/teaching:progappchim:analyse\\_images](https://dvillers.umons.ac.be/wiki/teaching:progappchim:analyse_images)

Last update: **2021/03/09 11:25**

