

# Tutoriel de Traitement d'Image

S.Gibet

Année 2019-2020

Les programmes seront écrits en Python. Le programme *image\_base* permet de démarrer ce TP.

**scikit-image** (a.k.a. skimage): ensemble de fonctions pour le traitement d'image et la vision par ordinateur.

<http://scikit-image.org/docs/dev/api/skimage.html>

**scipy.ndimage**: Ce package contient des fonctions pour le traitement d'image multi-dimensionnel.

<https://docs.scipy.org/doc/scipy/reference/ndimage.html#module-scipy.ndimage>

## 1 Traitements de base

### 1.1 Acquérir une image (jpeg)

1. Acquérir une image  $256 * 256$  (jpeg) (vous pourrez utiliser l'image *chimpanze.jpg* fournie dans Moodle).
2. Afficher cette image.
3. Slicing – Récupérer un zoom de cette image et sauvegarder le dans une nouvelle image (io.imsave).
4. Appliquer un masque sur l'image (cercle ou carré extérieur noir ou blanc).

### 1.2 Echantillonnage

1. Acquérir l'image de l'exercice précédent.
2. Sous-échantillonner l'image en résolution :
  - $128 \times 128$
  - $64 \times 64$
  - $32 \times 32$
  - $16 \times 16$
3. Afficher l'image originale et dans une autre figure les sous-échantillonnées dans un subplot  $2 \times 2$ .
4. Faire une fonction *echantillonnage* avec en entrée l'image et le choix de la résolution.

### 1.3 Histogramme

1. Faire une fonction *histogramme*. L'histogramme représente la distribution des niveaux de gris d'une image (noir et blanc) ou la distribution des couleurs pour une image couleur.
2. Modifier votre fonction *histogramme* pour tenir compte de la normalisation (diviser par le maximum des intensités). Proposez un paramètre forçant la normalisation à une valeur constante donnée.
3. Afficher l'image obtenue avec différentes options de normalisation
4. (**Bonus**—) Changement de contraste. L'égalisation d'histogramme permet de modifier le contraste de l'image. L'idée est de calculer l'histogramme cumulé, et de transformer en conséquence chaque pixel de l'image (voir en fin de TP).

### 1.4 Convolution, détection de contour

Soit la matrice de convolution suivante :

$$M_c = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad (1)$$

1. Appliquer ce filtre à votre image et afficher la.
2. Quelle est la matrice de convolution pour un filtre gaussien (0, 0.625). Appliquer ce filtre et afficher l'image.
3. Faire une fonction *convolution* avec une image et une matrice personnalisable en entrée. Vous pourrez tester avec plusieurs filtres (voir sur wikipedia).

## 2 Filtrage et transformée de Fourier

### 2.1 Filtrage

1. Acquérir une image en nuance de gris et de résolution 256 \* 256 (jpeg).
2. Appliquer un filtre moyennneur sur cette image (vous pouvez utiliser la fonction *convolution* que vous avez développée précédemment).
3. Quel résultat produit le filtre ?

### 2.2 Transformée de Fourier

1. Afficher la représentation de l'amplitude (magnitude) et de la phase de l'image en utilisant la fonction *fft2*.
2. Faire un filtre passe-bas sur votre image.
3. Faire un filtre passe-haut sur votre image.

## Annexe

### 2.3 Égalisation d'histogramme et modification du contraste

On considère une image  $X$  en niveaux de gris codée sur  $M$  niveaux. Soit  $n_k$  le nombre d'occurrences du niveau  $x_k$ . La probabilité d'occurrence d'un pixel de niveau  $x_k$  dans l'image est :

$$p_x(x_k) = p(x = x_k) = n_k/N \quad (2)$$

pour tout  $k$  compris entre 0 et  $M - 1$ .  $N$  est le nombre total de pixels de l'images.  $p_x$  est l'histogramme normalisé sur  $[0,1]$ .

Pour améliorer le contraste dans une image, on peut appliquer une transformation  $T$  indépendamment sur chaque pixel de l'image, à partir de l'histogramme cumulé.

**Histogramme cumulé**— La transformation  $T$  est définie par :

$$T(x_k) = \frac{M-1}{N} \sum_{i=0}^k n_i \quad (3)$$