

Compte rendu du TP6 et TP7 - Image

Louis Allain

November 5, 2019

1 Traitements de base

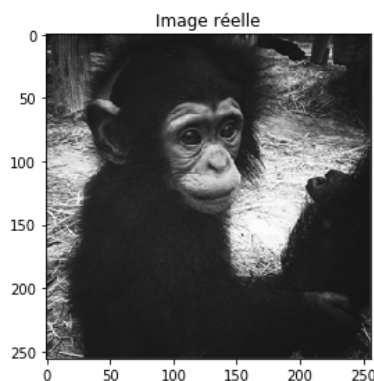
1.1 Acquérir une image

Réponses aux question :

1. Acquérir l'image.

```
chimp_img = io.imread('../res/chimpanze.jpg')
```

2. Afficher l'image réelle.



3. Enregistrer un zoom de l'image sous une autre image.



4. Appliquer un masque sur l'image.

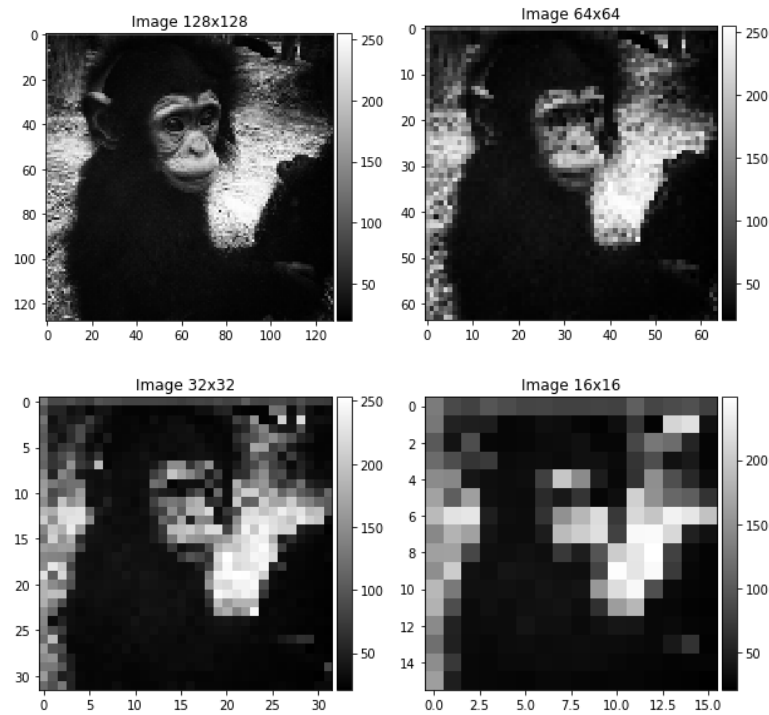
1.2 Echantillonnage

Réponses aux question :

1. Acquérir l'image de l'exo précédent.

```
chimp_img = io.imread('../res/chimpanze.jpg')
```

2. Sous-échantillonner l'image :



3. Fonction 'echantillonnage' :

```
def echantillonnage(img, factor):  
  
    ret = np.zeros(shape=(img.shape[0]//factor, img.shape[1]//factor))  
    for i in range(0, img.shape[0], factor):  
  
        for j in range(0, img.shape[1], factor):  
            ret[i//factor][j//factor] = img[i][j]  
    return ret
```

1.3 Histogramme

Réponses aux question :

1. Fonction histogramme avec normalisation :

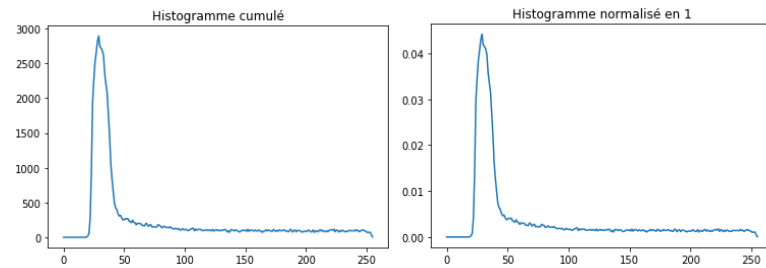
```
def histogramme(img, norme=1):  
  
    ret = np.zeros(256)  
    for i in range(0, img.shape[0]):  
        for j in range(0, img.shape[1]):
```

```
ret[img[i][j]] += 1

# Normalise si besoin
if(norme > 1):
    for i in range(0, ret.shape[0]):
        ret[i] = ret[i]/norme

return ret
```

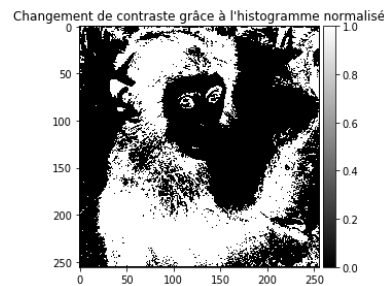
Voici les histogrammes obtenus :



2. Changement de contraste de l'image :

```
chimp_img_3 = chimp_img_2
h = histogramme(chimp_img_3, (256*256))
for i in range(0, chimp_img_3.shape[0]):
    for j in range(0, chimp_img_3.shape[1]):
        chimp_img_3[i][j] = chimp_img_3[i][j] * h[chimp_img_3[i][j]]
```

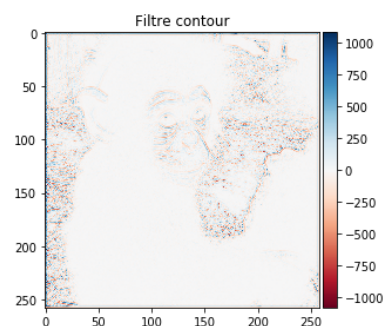
Voici l'image obtenue :



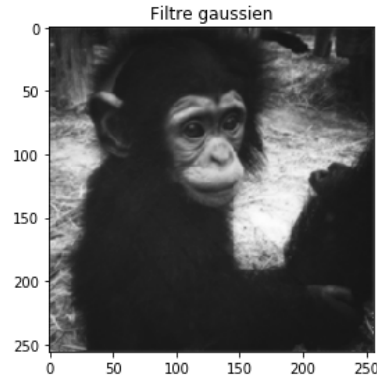
1.4 Convolution, détection de contour

Réponses aux question :

1. Appliquer un filtre de détection de contour :



2. Appliquer un filtre gaussien :



3. Fonction 'convolution' :

```
def convolution(img, Mc):  
  
    xMc = int((Mc.shape[1]-1)/2)  
    yMc = int((Mc.shape[0]-1)/2)  
    imgCp = img  
    m = int(img.shape[1] - xMc)  
  
    for i in range(xMc, m):  
        for j in range(yMc, img.shape[0]-yMc):  
            acc = 0.0  
            for k in range(-xMc, xMc+1):  
                for l in range(-yMc, yMc+1):  
                    acc = acc + img[j+l][i+k]*Mc[l+yMc][k+xMc]  
            imgCp[j][i] = acc  
    return imgCp
```

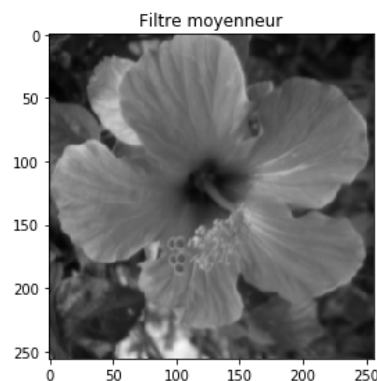
1.5 Filtre moyennneur

Réponses aux question :

1. Acquérir une image en nuance de gris :

```
fleur = io.imread('../res/fleur.jpg')
```

2. Appliquer un filtre moyennneur :



3. Le filtre précédent floute l'image, il agit comme un filtre passe bas.

2 Traitement de l'image et transformée de Fourier

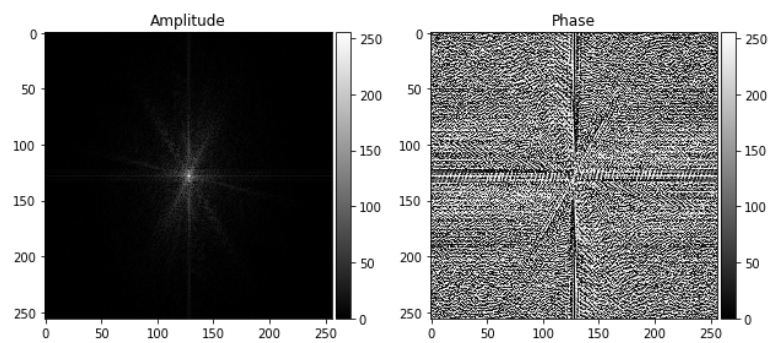
2.1 Transformée de Fourier

Réponses aux question :

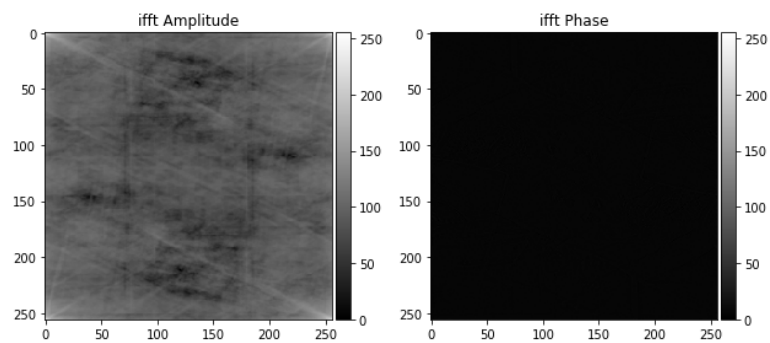
1. Je choisis d'utiliser cette image pour la suite :



2. Affichage de la phase et de l'amplitude de l'image avec `fft2` :



3. Retrouver la phase et l'amplitude avec `ifft2`



2.2 Filtrage par convolution

Réponses aux question :

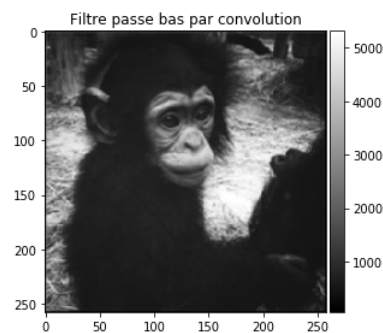
1. Je choisis d'utiliser cette image pour la suite :



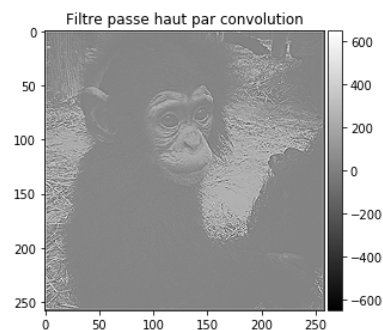
2. Application du filtre passe bas par convolution, code :

```
Mpb = np.array([
    [1, 3, 1],
    [3, 5, 3],
    [1, 3, 1]])
convolution(chimp, Mpb)
```

3. Résultat du filtre précédent :



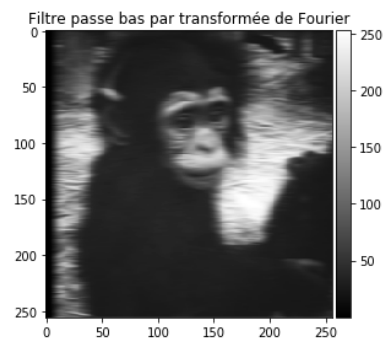
4. Résultat du filtre passe haut par convolution :



2.3 Filtrage par transformée de Fourier

Réponses aux question :

1. Résultat du filtre passe bas par transformée de Fourier :



2. Résultat du filtre passe haut par transformée de Fourier :

