TP4 - Attributs de texture

1 - Attributs statistiques

Charger les images présentes dans l'archive. Par exemple,

```
name = 'stained-wall1'
image = plt.imread( 'images/' + name + '.jpg' )
if image.ndim > 2:
    image = image[:,:,0]

plt.imshow(image, cmap='gray' )
```

Calculer l'histogramme correspondant à chaque image à l'aide de la fonction **np.histogram** (voir code ci-dessous).
Quelle valeur faut-il donner à **nbins** pour prendre en compte tous les niveaux de gris de l'image ?

Pour chaque image, calculer les quatre attributs statistiques suivants : moyenne, variance, kurtosis et entropie.

```
nbins = 10
counts, bin_edges =
np.histogram( np.asarray( image.ravel() ), bins=nbins )

counts = np.divide(counts,np.amax(counts))

plt.bar(bin_edges[:-1], counts, width = 1)
plt.xlim(min(bin_edges), max(bin_edges))
plt.show()
```

Comment évoluent ces attributs en fonction de **nbins** ? Conclure sur la possibilité d'utiliser ces attributs pour différencier les textures.

2 - Matrices de co-occurrence

Compléter la fonction **co_occurrence** qui calcule la matrice de co-occurrence d'une image **im** pour une configuration **u,v** donnée :

```
def co_occurrence( im, u, v):
            # à compléter
    return mat
```

Pour chaque image, calculer les matrices de co-occurrence pour les configurations (1,1), (10,10), (50,50) et (100,100). Quelle configuration permet le mieux de différencier les textures ?

3 - Attributs d'Haralick

Calculer l'énergie et l'entropie à partir des matrices de co-occurrences normalisées. Pour éviter une division par zéro, il est possible d'utiliser le flottant correspondant à la précision machine, donné par **np.finfo(float).eps**

Ecrire des fonctions permettant de calculer la corrélation et le moment différentiel inverse.
Il est possible de calculer la variance associée à une matrice de co-occurrence **c11_n** à partir de son histogramme grâce au code suivant :

```
counts, bin_edges =
np.histogram( np.asarray( c11_n.ravel() ), bins='auto' )
ny, nx = c11_n.shape
variance = np.dot(bin_edges[1:],np.power(counts,2)) -
np.power(np.dot(bin_edges[1:],counts),2)
```

Conclure sur la pertinence d'utiliser les attributs d'Haralick comme attributs de texture.