

TP1 - Données Multimedia

Campedel - Juillet 2014

CES Data Scientist



Objectifs

calculer une signature simple sur une image avec Python (et opencv)
visualiser l'espace des attributs extraits d'un ensemble d'images issues d'une même vidéo.

Les données

Il est toujours difficile de se constituer une base d'images significatives de l'application visée.
Comment en estimer sa taille ? son contenu ?

Des campagnes d'évaluation internationales permettent aux chercheurs et industriels de mesurer les performances de leurs approches sur des bases d'images, étiquetées.

Par exemple : [ImageClef 2013](#)

Une fois les challenges passés, les données sont généralement récupérables.

Il est également possible d'aller sur les sites de stockage de photos, par exemple

<http://www.freeimages.com/> (photos libres de droit).

Dans le cadre de cet enseignement, nous exploiterons les trames (frames en anglais) d'une vidéo.

Ressources bibliographiques

Environnement : Anaconda + opencv (wrapper python)

<https://docs.python.org/2/tutorial/>

<http://docs.continuum.io/anaconda/index.html>

<http://opencv-python-tutroals.readthedocs.org/>

<http://www.packtpub.com/opencv-computer-vision-with-python/book?tag=>

<http://matplotlib.org/>

Exercice 1 : extraire les trames à traiter

Extraire environ 50 images de l'une des deux vidéos.

Pour regarder la vidéo, vous pouvez utiliser VLC ou Elan.

Sous Linux, vous pouvez utiliser la commande :

```
>> mplayer -vo jpeg -ssstep 30 -frames 50 -vf scale -zoom -xy 320 file_name.avi
```

Si vous tapez la commande qui suit, soyez patient ...

```
>> man mplayer
```

La commande -ssstep 30 signifie que l'on prend la trame suivante après 30 secondes d'attente.

La commande `-vf scale -zoom -xy 320` signifie que l'on modifie la taille des images en imposant la plus grande dimension à 320 pixels.

La commande `-frames 50` limite l'extraction à 50 images.

Attention car les trames extraites portent un numéro qui ne reflète pas la position dans la vidéo. Il faut donc conserver l'information quelque part !

Exercice 2 : Jeu des 7 erreurs

Le script qui suit vous est proposé : hélas il présente 7 erreurs. A vous de le corriger !

```
import numpy as np
import cv2
import matplotlib.pyplot as plot

grayImage = cv2.imread('00000002.jpg', cv2.CV_LOAD_IMAGE_GRAYSCALE)

#rotation de l'image de 45 degrés

rows,cols = Image.shape

M = cv2.getRotationMatrix2D((rows/2,cols/2),45,1)
Image = cv2.warpAffine(grayImage,M,(cols,rows))

plt.figure(0)
plt.subplot(2,2,1)
plt.imshow(grayImage)
plt.title('image originale')
plt.subplot(2,1,2)
plt.imshow(destImage)
plt.title('image tournée')

#qu'observez-vous ?

m = numpy.moyenne(grayImage)
mr = numpy.moyenne(destImage)
v= numpy.variance(grayImage)
vr = numpy.variance(destImage)

plt.figure(1)
x=[m, mr]
y=[v, vr]
plt.plot(m, v, '.')
plt.annotate('orig', [m,v])
plt.plot(mr, vr, '.')
```

```
plt.annotate('tournee', [mr,vr])
plt.title('exploration de l''espace des attributs')
```

Exercice 3 : visualisation des attributs extraits sur l'ensemble des images

- Créer une liste avec les noms des fichiers des images à traiter : sous Linux, taper, dans le répertoire de travail (qui contient les images)
`>> ls *.jpg > list.txt`
cela créera un fichier texte contenant un nom de fichier par ligne. Si vos images sont d'un autre format que jpg, cette commande doit être adaptée. Si vous travaillez dans un autre répertoire, ne pas oublier d'indiquer le path des images : à la place de *.jpg, indiquer `mon_path/*.jpg`
- Extraire les attributs (moyenne, variance) pour chacune des images de la liste, convertie en niveaux de gris
- Visualiser l'ensemble des attributs extraits - Ces attributs appartiennent à un espace à deux dimensions (celle des moyennes et celle des variances) que nous pouvons donc facilement visualiser.
- Qu'observez-vous ? N'hésitez pas à zoomer sur certaines portions de l'image produite et à visualiser les images correspondantes (une visionneuse peut être utile à cet égard, ... cf Menu des Applications/Infographie)

Indice :

Exploitez l'exercice 2

Et ci-dessous une aide pour la lecture du fichier texte

```
f = open('list.txt', 'rt')
for line in f:
    w = line.split()[0]
    print(w)
    im=cv2.imread(w, cv2.CV_LOAD_IMAGE_GRAYSCALE)
    ....
f.close()
```