

Introduction à l'apprentissage automatique

Séance 6

Le but de ce TP est de découvrir une application de la vision par ordinateur, la reconnaissance d'images par vocabulaire visuel. On utilisera un sous-ensemble de la base de données Caltech101¹, constitué de 50 images de 6 catégories, disponible ici :

https://members.loria.fr/FSur/enseignement/apprauto/Caltech_small.zip

Le but est de prédire la catégorie d'une image de la base de test.

Examinez le fichier `TP6_code.py`, et constatez qu'il permet de charger 40 images d'apprentissage pour chacune des catégories, et 10 images de test. Dans la suite du TP, vous n'aurez essentiellement qu'à apporter des modifications à ce fichier.

1 « Méthode 0 » : description par l'histogramme des niveaux de gris

Dans ce premier exercice, on teste une méthode très basique de manière à avoir des scores de classification de référence.

Décrivez chaque image par l'histogramme normalisé de ses niveaux de gris (256 niveaux possibles dans une image 8 bits). Voir :

<https://docs.scipy.org/doc/numpy/reference/generated/numpy.histogram.html>

Testez différents algorithmes de classification : calculez au moins le score de classification sur la base de test, ainsi que la matrice de confusion.

On peut utiliser une SVM avec un noyau χ^2 à l'aide de :

```
SVM = svm.SVC(kernel=chi2_kernel)
```

avec l'import :

```
from sklearn.metrics.pairwise import chi2_kernel
```

Comparez les scores de classification obtenus avec celui attendu en tirant au hasard la classe prédite.

2 Vocabulaire visuel

Pour pouvoir utiliser l'implantation de SURF fournie dans la bibliothèque Opencv, exécutez en ligne de commande :

```
pip install opencv-python opencv-contrib-python
```

ou sur les machines de l'école (invite de commande Windows) :

```
pip install --user opencv-python opencv-contrib-python
```

1. http://www.vision.caltech.edu/Image_Datasets/Caltech101/

qui installe les bibliothèques `opencv-python` et `opencv-contrib-python`.

Créez la liste des descripteurs SURF des images de la base d'apprentissage.

Tutoriel SIFT (SURF est une variante, changez SIFT en SURF dans le code) :

https://docs.opencv.org/3.3.0/da/df5/tutorial_py_sift_intro.html

Après avoir créé l'objet `surf` par : `surf = cv2.xfeatures2d.SURF_create()`, on extraira de chaque image (par exemple l'image 17) ses descripteurs SURF (stockés dans `des`) par :

```
img = np.uint8(X_image_train[17].reshape(X_size_train[17]))
kp, des = surf.detectAndCompute(img, None)
```

Construisez le vocabulaire visuel par KMeans (créez 500 mots visuels, et utilisez `MiniBatchKMeans`).

Calculez un vecteur-signature (*codeword*) pour chaque image des bases d'apprentissage et de test, représentant la fréquence des mots visuels les plus proches de chaque descripteur SURF extrait de l'image.

Testez les différents algorithmes de classification.

3 Pour les plus rapides

Examinez l'effet du nombre de mots visuels et proposez une méthode de `gridsearch` pour le régler.

Remarquez que notre taux de reconnaissance est similaire au taux rapporté dans la dernière courbe de http://www.vision.caltech.edu/Image_Datasets/Caltech101/ (obtenu sur toute la base néanmoins).

Adaptez votre code pour traiter d'autres classes (attention, certaines classes sont beaucoup plus grosses, comme Faces, Airplanes, Motorcycles), puis tout Caltech101.