

TP3 : Feature Extraction / Selection

Machine learning

Cours : *Computer vision*

FIA3-GL-AL

Partie n°1 : Extraction de caractéristiques

Dans ce problème, notre but est de faire la classification d'images en utilisant trois classes de la base Caltech. Cette base est divisée en deux parties : train et test. 20 images par classe comme données d'apprentissage et 5 images par classe pour les tests.

Dans cette première partie nous allons voir plusieurs méthodes permettant d'extraire des caractéristiques à partir d'une image et de les stocker dans un vecteur.

Question 1 :

La première méthode consiste à extraire des caractéristiques HOG. Compléter le code pour extraire ces caractéristiques pour toutes les images d'apprentissage et de test. Apporter aussi les modifications nécessaires pour récupérer la classe de chaque image à partir de son chemin. La classe est le nom du dossier dans lequel se trouve chaque image. Ces classes sont utilisés pour construire le modèle d'apprentissage (supervisé). Et pendant la phase de test, elles sont utilisées pour évaluer les résultats.

Question 2 :

Vous allez comparer la première méthode basée sur HOG avec d'autres caractéristiques :

1. Vous allez juste prendre l'image et la convertir en un vecteur en utilisant la fonction `flatten()`
2. Vous pouvez aussi représenter chaque image par un histogramme de couleur dans l'espace HSV.

Vous pouvez utiliser les fonctions `opencv cv2.cvtColor(image, cv2.COLOR_BGR2HSV)` qui convertit l'image dans l'espace HSV et la fonction `cv2.calcHist([hsv], [0, 1], None, [180, 256], [0, 180, 0, 256])` pour calculer l'histogramme.

3. Pensez à comparer HOG features aux LBP features en faisant appel à la fonction `compute_lbp()` qui calcule l'histogramme issue des caractéristiques LBP d'une image en niveaux de gris.

Partie n°2 : Classification

Après avoir vu plusieurs façons de représenter les images sous forme de vecteurs, il est temps de passer à la classification. Pour cela nous allons utiliser Support Vector Machine (SVM) et K-Nearest Neighbor (KNN) de scikit-learn.

Evaluer les résultats de chaque méthode la partie 1 pour les deux classifieurs.

Partie n°3 : Selection de caractéristiques

PCA (Principal Components Analysis) est réalisée à l'aide d'une combinaison linéaire des attributs originaux en utilisant une décomposition en valeurs propres. Il s'agit donc de décomposer une matrice X pour projeter les données sur un espace de dimension réduite. En scikit-learn, on fait :

```
from sklearn.decomposition import PCA  
pca = PCA(n_components=N, whiten=True).fit(X)
```

avec N est la nouvelle dimension après réduction.

Cette méthode est souvent utilisée pour projeter les caractéristiques LBP dans un espace réduit afin d'améliorer les résultats.

Appliquer PCA sur les caractéristiques LBP et comparer les résultats.