

Les objectifs du projet :

Mettre en place 2 types de classifieurs : SVM et MLP en python

**Classification automatique supervisée
d'images de scènes naturelles**

1. Contexte et objectifs

L'objectif est de concevoir un système capable de classer automatiquement des images en différentes catégories. On dispose d'une banque de 2288 images numérotées. Chaque image possède 1 catégorie principale et 1 sous-catégorie.

La classe principale de l'image peut être :

- Artificielle (constructions humaines)
- Naturelle (image de nature)

La sous-classe de l'image peut-être :

- Côte (plage, mer...)
- Forêt
- Autoroute
- Ville (Photo prise de loin de l'ensemble de la ville)
- Montagne
- Paysage ouvert (landes, désert...)
- Rue
- Grand bâtiment

Les images se trouvent dans le fichier .zip. Les images ont été réduites à une taille de 128 x 128 pixels.

Chaque image est décrite à l'aide de 27 vecteurs de caractéristiques (features en anglais). Les 24 premiers vecteurs de caractéristiques sont issus d'un banc de filtres de Gabor et consistent à récupérer l'énergie dans chacune des bandes de fréquences (ici 24). Les 3 derniers vecteurs de caractéristiques sont issus de la luminance Y et de la chrominance Cb puis Cr.

L'objectif de ce projet est de proposer un classifieur parmi un SVM et un MLP qui permette de répondre le mieux au problème de classification posé (classification images naturelles/artificielles, classification par sous-catégories d'images).

Pour cela, commencer par partitionner la base de données en deux bases : apprentissage et test.

Visualiser les données de la base d'apprentissage en réalisant une ACP et/ou une ALD et en projetant les données sur les 2 premiers axes.

Ensuite, pour les 2 classifieurs (MLP et SVM), choisir la structure la plus adaptée au problème de classification. On utilisera comme critère de sélection une métrique calculée par validation croisée sur la base d'apprentissage.

Une fois la structure des classifieurs choisie, évaluer et comparer leurs performances sur la base de test. Conclure sur le choix des classifieurs retenus.

Vous pourrez évaluer la sensibilité des résultats à la taille de la base d'apprentissage. Vous pourrez aussi évaluer la perte en performances quand on supprime les 3 dernières caractéristiques, c'est-à-dire la luminance Y, la chrominance Cb et Cr.

Avant de programmer un MLP ou un SVM, il est nécessaire d'étudier sur le site de sikitlearn la documentation relative à ces classifieurs :

https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

https://scikit-learn.org/stable/modules/cross_validation.html

Pour le MLP, on fixera le nombre de couches cachées à 1. On optimisera le nombre de neurones dans la couche cachée et on choisira la fonction d'activation associée aux neurones.

Pour le SVM, on choisira le type de noyau et la constante de pénalité C.

Quelques lignes de code qui peuvent vous être utiles

Normalisation des données

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

Calcul du temps d'exécution

```
import time

start_time = time.time()
-- Apprentissage
end_time = time.time()
print('temps ecole = '+str(end_time - start_time))
```