

# Projet - Automne 2016

Reconnaissance de forme et analyse d'images avancée.  
IMN 601 - IMN712

**Date limite pour remettre votre travail : 12 décembre**

## Description

Vous devez explorer un certain nombre de méthodes de classification d'images, les valider de façon professionnelle tel que vu en classe et d'en faire rapport.

**Langage de programmation** Vous devez OBLIGATOIREMENT utiliser le langage de programmation **python** ainsi que les packages **sklearn** et **keras** pour la classification. **sklearn** propose des solutions pour la plupart des algorithmes vus au chapitre 3 alors que **keras** est un des nombreux packages python dédié au *deep learning*. Votre code doit être sur un serveur git avec de fréquents commits et pull requests. Votre code doit également être bien documenté et jouir d'un design minimalement correct.

**Données** Vous devez classer les images issues de 2 bases de données très connues, à savoir *cifar10* et *mnist*. *cifar10* contient 60,000 images couleur de taille 32x32 dont 50,000 sont pour l'apprentissage et 10,000 pour les tests. Les images de *cifar10* sont distribuées dans 10 classes. Pour ce qui est de *mnist*, elle contient 70,000 images de caractères manuscrits. Ces images sont en niveaux de gris, de taille 28x28 et distribuées en 10 classes. Ces bases de données peuvent être chargées comme suit :

```
from keras.datasets import cifar10, mnist
import matplotlib.pyplot as plt

(X_data,X_label),(Y_data,Y_label) = mnist.load_data()
plt.imshow(X_data[1,:,:])

(X_data,X_label),(Y_data,Y_label) = cifar10.load_data()
img = X_data[1,:,:,:]
img = img.transpose(1,2,0)
plt.figure()
plt.imshow(img)
```

**Algorithmes à tester** Pour IMN601, vous devez implémenter 2 algorithmes issus du chapitre 3 (et trois algorithmes pour IMN712). Cela inclut KNN, SVM, SVM à noyau, le perceptron, les arbres de décision, adaboost, les *random forests*, etc. Avec ces algorithmes, vous devrez tester les données d'entrée RGB et niveaux de gris comme "caractéristiques" ainsi qu'au moins une des caractéristiques vues au chapitre 4 à savoir SIFT, SIFT dense, HoG, BOW, ou GIST. Vous pouvez utiliser les packages **opencv**, **pyvlfeat** ou tout autre package pour gérer ces caractéristiques. Vous devez également implémenter un réseau à convolution et un perceptron multicouche à au moins 4 couches chacun. Pour IMN712, vous devez tester 2 réseaux à convolution différents.

Ces algorithmes devront être entraînés et testés sur les images de *cifar10* et *mnist*.

**Validation** Pour chaque algorithme, il vous faudra trouver les meilleurs hyperparamètres à l'aide d'une stratégie de type *grid search* telle que vue en classe. Vous devrez également identifier les cas *d'overfitting*.

**Rapport** Votre rapport doit contenir les informations suivantes

1. Une courte description des méthodes implémentées.
2. Les résultats du *grid search* pour chaque méthode.
3. *L'accuracy* (entraînement et test) pour la meilleure configuration de chaque algorithme sur cifar10 et mnist.
4. Les matrices de confusion pour la meilleure méthode pour cifar10 et mnist.
5. Discussion sur les résultats.

**Rapport** Un bonus de 5 points sera donné aux équipes qui testeront leurs algorithmes sur la base de données cifar100. Un autre bonus de 5 points sera donné pour chaque algorithme additionnel testé.

## Évaluation

Ce travail doit être fait par équipe de 2 pour imn601 et seul pour imn712. Le code doit être remis avec le rapport et avec un fichier README.txt décrivant l'exécution de votre programme. La pondération pour ce travail est la suivante :

1. Qualité du français : 10
2. Qualité du code + gestion git : 10
3. Algorithmes d'apprentissage + caractéristiques, chap 3 et 4 : 20
4. Algorithmes deep learning : 20
5. Validation : 15
6. Discussion : 15