

# Reconnaissance de chiffres en Python

## Objectif :

Le but des exercices de cette feuille est de réussir à créer un algorithme en Python permettant de reconnaître un chiffre à travers une image.

On implémentera ainsi différentes méthodes, plus ou moins simples, pour ce qui est des images on pourra utiliser cette [archive](#). (Pour l'extraire : `tar -xzf chiffres.tar.gz`)

## Préparation :

Avant de reconnaître des chiffres on doit pouvoir lire et manipuler des images en Python. Dans toute la feuille on considère une image de largeur  $l$  et de hauteur  $h$ .

1. Tout d'abord on aura besoin d'une fonction pour lire cette image depuis un fichier, on pourra par exemple utiliser la classe `Image` de la librairie `PIL`.  
(Pour l'installer : `pip3 install Pillow`)
2. Ensuite il faut une fonction pour passer de l'image à un objet simple à manipuler, on pourra la représenter comme une **matrice** de pixels de taille  $(1, l \cdot h)$ , par exemple avec la librairie `numpy`. On devra aussi transformer les couleurs en une valeur dans  $[0, 1]$ .  
(Remarque : On fera attention à ce que notre matrice soit bien en 2 dimensions pour pouvoir utiliser le produit matriciel)
3. Enfin on pourra faire une fonction pour récupérer une liste contenant tous les couples de références (`image, chiffre`) depuis un répertoire.  
(Remarque: pour lister tous les fichiers d'un répertoire on pourra utiliser la fonction `listdir(repertoire)` de la librairie `os`)

## Partie 1 : Les k plus proches voisins

Dans cette partie on va considérer une image comme un **point** ou un **vecteur** de  $[0, 1]^{l \cdot h}$ . Ainsi en calculant la **distance** entre deux points ou le **produit scalaire** de deux vecteurs, on peut obtenir une mesure de la ressemblance d'une image à une autre et utiliser la méthode des [k plus proches voisins](#).

1. D'abord on crée une fonction pour calculer la **distance** entre deux points de  $[0, 1]^{l \cdot h}$ .  
On pourra appliquer la formule de la distance euclidienne de la même manière que dans l'espace, ou une autre si vous trouvez plus intéressant.  
(Remarque : pour les performances on pourra enlever la racine carré dans la formule de la distance euclidienne qui n'est pas nécessaire ici)
2. On fera aussi une fonction pour calculer le **produit scalaire** de deux vecteurs de  $[0, 1]^{l \cdot h}$ .  
Encore une fois, la formule s'applique de la même manière que dans l'espace.
3. Enfin on crée une fonction pour reconnaître un chiffre depuis une image, pour ça on peut trouver les **k** images parmi les références dont les points sont les **moins distants**, ou dont les vecteurs se **ressemblent le plus**, et en déduire le chiffre.  
On pourra comparer les deux méthodes et voir laquelle fonctionne le mieux.

## Partie 2: Un réseau neuronal basique

TODO