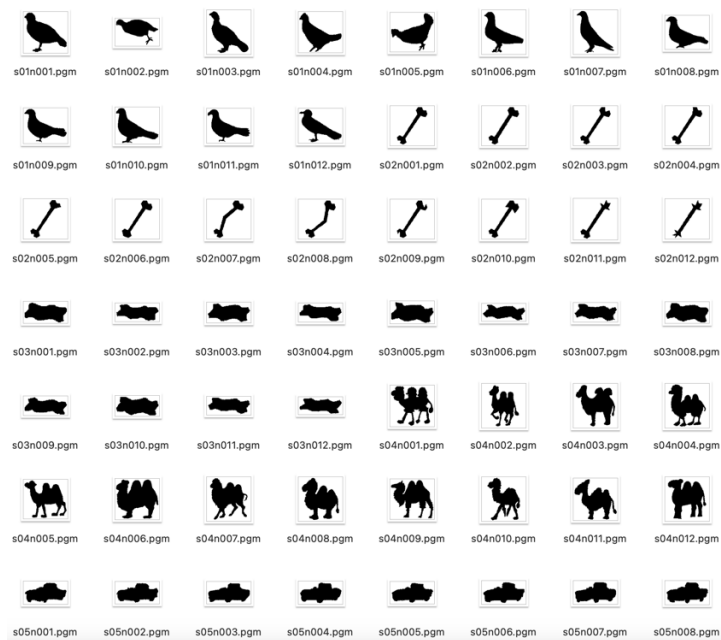


## Projet - Reconnaissance des Formes

### M1 Master Informatique – année 2024 / 2025

### Problématique

Nous considérons dans ce projet une base d'images, appelée SharvitB2, qui a servi de fondement à de nombreux travaux en reconnaissance des formes (robustesse à des occultations, comparaison d'approches polygonales, fusion de méthodes...). Cette petite base est composée de formes binaires représentant des classes d'objets à reconnaître. Elle comprend 18 classes (oiseau, clef, tortue, fourche...) de 12 échantillons pouvant avoir des déformations et des orientations variables (cf. Figure1).



**Figure 1. Échantillons de la base de formes SharvitB2 (18x12)**

Chaque forme est une image binaire stockée sous la forme SxxNyyy.pgm. Sxx correspond à la classe xx et Nyyy correspond à l'échantillon yyy de la classe. Ces fichiers au format pgm ne servent ici qu'à la visualisation.

Des représentations de ces formes (vecteurs caractéristiques) ont été obtenues en appliquant quatre méthodes classiques servant de fondement à de nombreux travaux en recherche en reconnaissance des formes :

- ART : représentation définie à partir d'une transformée radiale de l'image [1],

- ZRK : correspond aux moments de Zernike (déterminés à partir des polynômes complexes de Zernike calculés sur un disque unité) [2],
- GFD (définie à partir d'une transformée discrète polaire) : correspond à un *descripteur de Fourier générique* (en fonction de fréquences radiales et angulaires) [4],
- YNG : signature structurelle définie à partir des points du squelette de la forme [5].

Pour chaque image SxxNyyy.pgm de BDshape, des fichiers SxxNyyy.MET ont été générés pour faciliter les traitements. L'extension du fichier MET définit le type de *méthode* qui a été calculée sur l'image associée. C'est-à-dire dans ce projet : SxxNyyy.YNG (pour Yang) et SxxNyyy.GFD (pour les descripteurs de Fourier Générique),....

L'objectif du projet est *d'évaluer*, sur une petite base complexe, le comportement de ces méthodes classiques de reconnaissance des formes (notées  $\mathbf{M}^{\text{ART}}$ ,  $\mathbf{M}^{\text{GFD}}$ ...) en associant leur représentation à un *classifieur*.

**Remarque :** les fichiers correspondant à ces représentations sont accessibles à l'adresse : <http://helios2.mi.parisdescartes.fr/~lwendlin/RF2024/PROJET/> ainsi qu'une archive PGM qui contient toutes les formes (uniquement pour visualisation)

- 
- [1] Kim, W.-Y., Kim, Y.-S.: A new region-based shape descriptor. In: TR 15-01, Pisa, Italy (1999)
  - [2] Khotanzad, A., Hong, Y.H.: Invariant Image Recognition by Zernike. IEEE TPAMI 12(5), 489–497 (1990)
  - [3] R. Teague, Image analysis via the general theory of moments, J. Opt. Soc. Amer. 70(8) 920–930, (1979).
  - [4] Zhang, D., Lu, G.: Shape-based image retrieval using generic fourier descriptor. Signal Processing: Image Communication 17, 825–848 (2002)
  - [5] Yang, S.: Symbol Recognition via Statistical Integration of Pixel-Level Constraint Histograms: A New Descriptor. IEEE TPAMI 27(2), 278–281 (2005)

## Travail à faire

Le travail est à réaliser **en binôme** et comporte 4 étapes :

1. Lecture des données
2. Implémentation de 3 approches de classification (*voir annexe*) **en ne considérant que les 10 premières classes**
  - a. Approche des k-plus-proches voisins – supervisée (distance, choix de k...)
  - b. Approche des nuées dynamiques (ou *k-means*) – non supervisée
  - c. Au choix : SVM, MLP ou vote majoritaire (\*)
3. Protocole de test (lorsque c'est possible)
  - a. Choix du mode d'évaluation (découpage des données) pour les approches
  - b. Critères d'évaluation (matrice, F-mesure...)
  - c. Courbe précision/rappel (et AUC) avec K=12
4. Étude expérimentale
  - a. Analyse du comportement des méthodes en fonction des critères d'évaluations,
  - b. En considérant uniquement le taux de reconnaissance et le 2.a : comparer les résultats obtenus avec les 10 classes et ceux des 18 classes**
  - c. Discussion sur les résultats (améliorations possibles, impact des approches de classification...)
5. Conclusion et perspectives

(\*) **Non obligatoire pour le projet (en fonction du temps)** et vous pouvez utiliser des codes déjà existants

## Remarque :

- a) L'objectif de ce mini-projet est d'intégrer des notions développées en cours. Une interface n'est pas requise et l'exécution du code peut se faire en ligne de commande.
- b) Les études comparatives peuvent être distinctes suivant les approches de classification.

## Documents à rendre

1. Un dossier d'analyse (au format pdf) comportant les réponses aux questions précédentes et tout commentaire permettant de justifier votre démarche (entre 10 et 15 pages maximum).
2. Une archive du code commenté (C/java) pour toutes les approches.
3. Une présentation contenant une dizaine de slides maximum décrivant votre démarche et les problèmes rencontrés.

## Dates butoirs

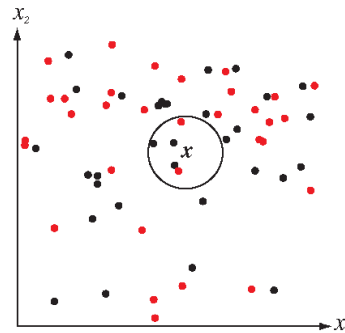
Le projet comporte deux dates butoirs :

10/12/2024 : remise du dossier d'analyse et du code (mail à [Laurent.Wendling@u-paris.fr](mailto:Laurent.Wendling@u-paris.fr))  
17/12/2024 : présentation (Zoom) de 10 minutes par binôme (envoi du document pdf deux jours au préalable)

## Annexe

### The k-Nearest Neighbor Classifier

- Given the training data  $D = \{x_1, \dots, x_n\}$  as a set of  $n$  labeled examples, the nearest neighbor classifier assigns a test point  $x$  the label associated with its **closest neighbor** in  $D$ .
- The  $k$ -nearest neighbor classifier classifies  $x$  by assigning it the label most frequently represented among the  $k$  nearest samples.



**Figure 2:** Classifier for  $k = 5$ .

- Closeness is defined using a **distance function**.

### Distance functions

- A general class of metrics for  $d$ -dimensional patterns is the **Minkowski metric**.

$$L_p(x, y) = \left( \sum_{i=1}^d |x_i - y_i|^p \right)^{1/p}$$

also referred to as the  **$L_p$  norm**.

- The **Euclidean distance** is the  $L_2$  norm

$$L_2(x, y) = \left( \sum_{i=1}^d |x_i - y_i|^2 \right)^{1/2}$$

- The **Manhattan or city block distance** is the  $L_1$  norm

$$L_1(x, y) = \sum_{i=1}^d |x_i - y_i|$$

### Squared-error Partitioning (clustering)

- Suppose that the given set of  $n$  patterns has somehow been partitioned into  $k$  clusters  $D_1, \dots, D_k$ .
- Let  $n_i$  be the number of samples in  $D_i$  and let  $m_i$  be the mean of those samples

$$m_i = \frac{1}{n_i} \sum_{x \in D_i} x$$

- Then, the sum-of-squared errors is defined by:

$$m_i = \sum_{i=1}^k \sum_{x \in D_i} \|x - m_i\|^2$$

- For a given cluster  $D_i$ , the mean vector  $m_i$  (centroid) is the best representative of the samples in  $D_i$ .
- A **general algorithm** for iterative squared-error partitioning:
  1. Select an **initial partition** with  $k$  clusters (*repeat steps 2 through 5 until the cluster membership stabilizes*).
  2. Generate a **new partition** by assigning each pattern to its **closest** cluster center.
  3. Compute **new cluster centers** as the centroids of the clusters.
  4. Repeat steps 2 and 3 until an **optimum value of the criterion function** is found (e.g., when a local minimum is found or a predefined number of iterations are completed).
  5. Adjust the number of clusters by **merging** and **splitting** existing clusters or by removing small or outlier clusters.

This algorithm, without step 5, is also known as the **k-means algorithm**.