

Projet - Reconnaissance des Formes M1 Master Informatique – année 2023 / 2024

Problématique

Nous considérons dans ce projet une base d'images **BDshape** composée de formes binaires représentant des classes d'objets. La base BDshape comporte 9 *classes* (animaux, avions...) de 11 *échantillons* (cf. Figure1). BDshape contient des situations complexes. Des formes sont *occultées* ; classe 4 (avion) et 5 (main). Des formes sont *partiellement* représentées : classe 2 (lapin), 3 (silhouette) et 5 (main). Des formes comportent des *distorsions* dans la classe 6 (outil). Et la classe 8 (animal) intègre des formes *hétérogènes*.

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.

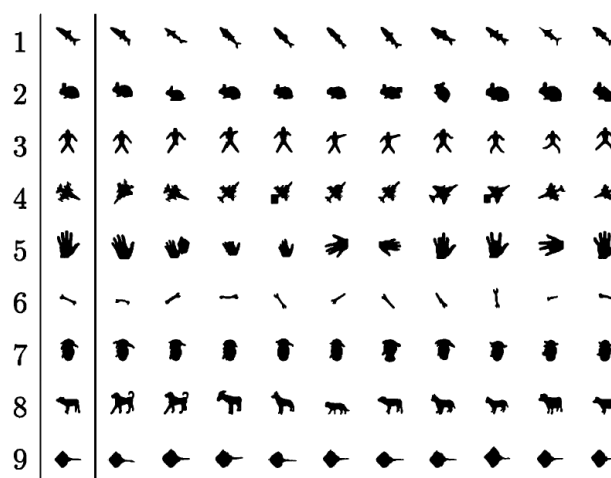


Figure 1. Base de formes BDshape (9x11)

Des représentations de ces formes ont été obtenues en appliquant quatre approches classiques servant de fondement à de nombreux travaux en analyse d'images.

- E34 (définie à partir des moments d'ordre 2) qui correspond au *degré d'ellipticité* calculé sur 16 coupes de la forme.
- GFD (définie à partir d'une transformée discrète polaire) qui correspond à un *descripteur de Fourier générique* (vecteur caractéristique de 36 valeurs correspondant à des fréquences radiales et angulaires).
- SA (définie à partir de la distance du barycentre au contour) qui correspond à une *Signature angulaire* calculée suivant 90 angles.
- F0 (définie à partir de forces constantes) qui correspond à une évaluation *des distorsions* de la forme suivant 128 directions.

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.E34 (pour l'ellipticité) 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 M^{E34} , M^{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/RF2023/PROJET/> ainsi qu'une archive PGM qui contient toutes les formes (uniquement pour visualisation)

Travail à faire

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

1. Lecture des données
2. Implémentation de 2 (ou 3) méthodes de classification (*voir annexe*)
 - a. Approche des k-plus-proches voisins – supervisé
 - b. Approche des nuées dynamiques (ou *k-means*) – non supervisé
 - c. Éventuellement une autre méthode de votre choix
3. Protocole de test
 - a. Choix du mode d'évaluation pour les approches
 - b. Critères d'évaluation (matrice de confusion, précision/rappel, temps...)
4. Discussion sur les résultats
 - a. Simplification éventuelle des méthodes,
 - b. Analyse des résultats,
 - c. Amélioration possible,

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 entre les méthodes 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).
3. Une présentation contenant une dizaine de slides maximum décrivant votre démarche et les problèmes rencontrés.

Dates butoirs

Le projet comporte deux dates butoirs :

12/12/2023 : remise du dossier d'analyse et de l'archive (mail à Laurent.Wendling@u-paris.fr)

19/12/2023 : présentation (Zoom) de 10 minutes par binôme (envoi du document pdf 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.

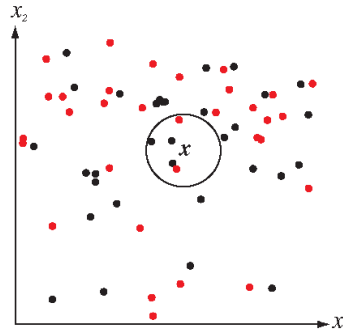


Figure 2: Classifier for $k = 5$.

- 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**.