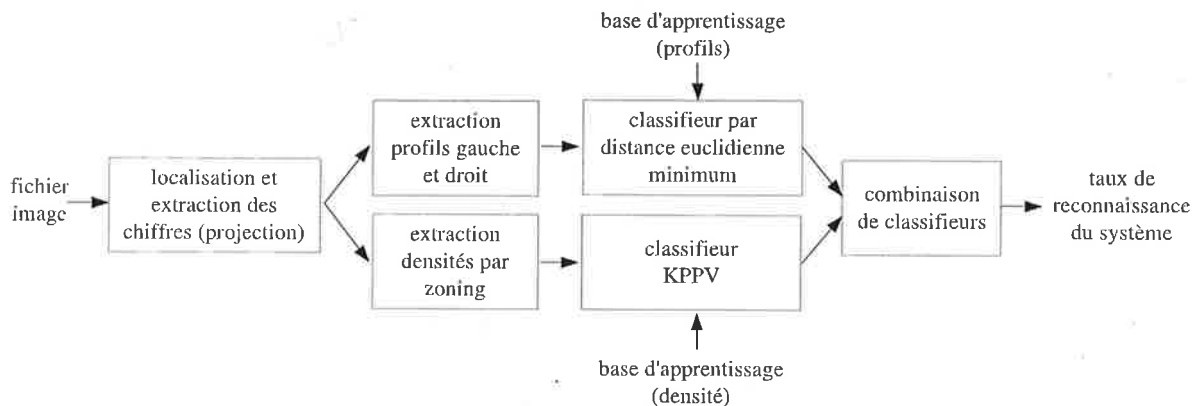


# Reconnaissance de chiffres manuscrits

## 1 Introduction

L'objectif de ce TP est d'appréhender les différentes étapes d'un système de reconnaissance de formes depuis les pré-traitements (localisation/extraction des formes à reconnaître) jusqu'à la reconnaissance (combinaison des résultats de reconnaissance de plusieurs classifieurs).

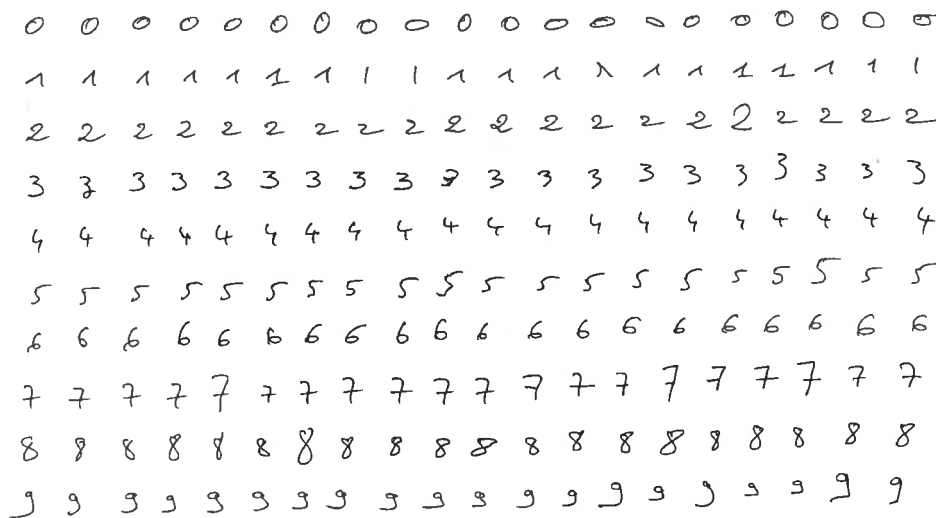
Il s'agit de développer, sous Matlab, un système complet de reconnaissance de chiffres manuscrits dont le schéma fonctionnel est le suivant :



NB : Les sources et un compte-rendu (par binôme) seront à remettre (voir détails en partie 9).

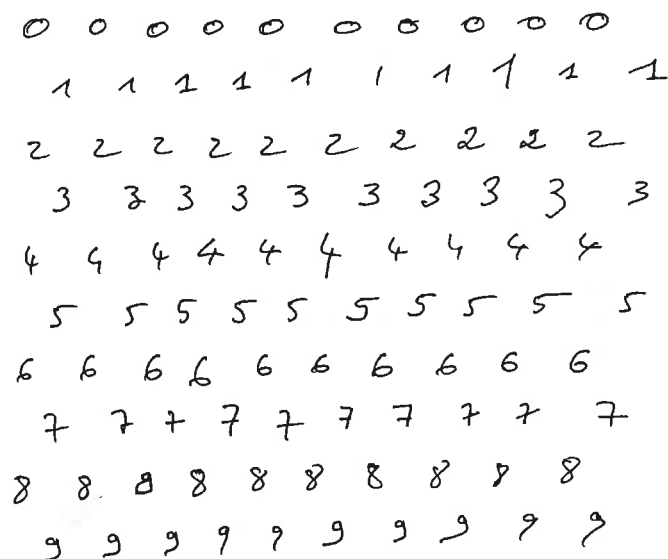
## 2 Données

On dispose dans le fichier image "app.tif" d'une image binaire contenant 200 chiffres manuscrits (20 exemples par classe) :



Les chiffres contenus dans cette image serviront de base d'apprentissage pour les deux classifieurs à développer.

Le fichier image "test.tif" est une image binaire contenant 100 chiffres (10 par classe) et permettra de tester le système de reconnaissance de formes développé.



### 3 Pré-traitements : localisation et extraction des chiffres manuscrits

- Principe

Dans l'exemple étudié, la façon la plus simple de localiser les chiffres consiste d'abord à rechercher, à partir de l'histogramme de projections horizontales de pixels noirs, les plages correspondant à un nombre de pixels noirs non nul. Le début et la fin de chaque plage détectée sur l'histogramme des projections horizontales définissent une ligne de chiffres dans l'image. En appliquant le même principe (mais à partir d'un histogramme de projections verticales) sur chaque ligne détectée, on peut alors déterminer l'emplacement de chaque chiffre de l'image.

- Manipulation

Ecrire un programme permettant de :

- rechercher les lignes dans une image binaire ;
- rechercher pour chaque ligne les colonnes correspondant aux chiffres ;
- déterminer, pour chaque chiffre, son rectangle englobant en prenant soin d'ajouter un bord blanc autour du chiffre (nécessaire pour l'extraction de caractéristiques) ;
- sauvegarder dans un fichier les coordonnées de ces rectangles englobants (coordonnées du coin haut gauche et du coin bas droit). Appliquer ce programme à la localisation des chiffres dans les deux images binaires fournies.

- Quelques fonctions Matlab utiles : `size`, `find`, `sum`, `rectangle`, `save`

- Analyse et conclusion

Quels sont les avantages et inconvénients de cette méthode de localisation par projection ? Préciser ses limites et notamment lorsque les formes à extraire ne sont pas idéalement disposées.

### 4 Classifieur1 : profils + classifieur par distance euclidienne minimum

- Principe

Pour cette première méthode de reconnaissance, les caractéristiques à extraire des chiffres sont leurs profils gauche et droit. Ces mesures sont obtenues de la façon suivante: on détermine sur un certain nombre de lignes – en général, réparties uniformément sur la hauteur du chiffre – la distance entre le bord gauche (resp. droit) du chiffre et le premier pixel noir rencontré sur cette ligne; l'ensemble de ces distances définit un profil gauche (resp. droit) du chiffre. On peut alors constituer un vecteur à  $d$  composantes en concaténant les deux profils ( $d/2$  distances par profil).

La méthode de décision proposée (classifieur par distance euclidienne minimum) consiste à attribuer au point de  $\mathbb{R}^d$ , représentant la forme à reconnaître, la classe dont la distance du point au centre de cette

classe est minimale. Il faut donc disposer, préalablement à toute reconnaissance, du centre de chacune des classes: c'est l'objet de la phase d'apprentissage.

- Manipulation

- Ecrire une fonction qui prend en entrée les coordonnées du chiffre dont on souhaite extraire les profils gauche et droit et qui renvoie un vecteur à  $d$  composantes. On pourra choisir dans un premier temps  $d/2 = 5$ . Les profils gauche et droit devront être normalisés (en les divisant par exemple par la largeur de l'image du chiffre traité) puisque les chiffres ne sont pas tous de même taille.
- Apprentissage du classifieur : Il s'agit de déterminer, pour chaque chiffre de l'image "app.tif", son vecteur de caractéristiques puis de calculer le centre (vecteur de  $\mathbb{R}^d$ ) de chacune des 10 classes et de les sauvegarder dans un fichier d'apprentissage.
- Décision : Il s'agit de fournir pour chaque chiffre à reconnaître de l'image "test.tif" un vecteur des distances du point à classer au centre de chacune des 10 classes. On pourra transformer ces vecteurs de distances en vecteurs de probabilité a posteriori par l'estimation suivante:

$$\hat{p}(\omega_i/x) = \frac{\exp(-\text{dist}(x, \omega_i))}{\sum_{j=0}^9 \exp(-\text{dist}(x, \omega_j))}$$

Ces vecteurs de probabilités seront à sauvegarder dans un fichier qui servira d'entrée à l'étape de combinaison de classifieurs.

- Quelques fonctions Matlab utiles : `linspace`, `isequal`, `mean`

- Analyse et conclusion

Evaluer les performances de ce premier classifieur (taux de reconnaissance). Discuter du choix des caractéristiques et notamment de l'influence sur les performances de ce classifieur du nombre de mesures retenues pour caractériser les profils. Discuter également de l'influence du type de distance choisi.

## 5 Classifieur2 : densités + KPPV

- Principe

Les caractéristiques utilisées dans cette deuxième méthode de reconnaissance sont basées sur la densité de pixels noirs calculée dans différentes zones (zoning) de l'image du chiffre. Ces mesures sont obtenues de la façon suivante: on "découpe" horizontalement et verticalement le rectangle englobant du chiffre en zones de tailles égales ; le nombre de pixels noirs dans chaque zone forme alors les composantes du vecteur de caractéristiques. En découplant par exemple l'image en  $n$  zones verticales (d'égale largeur) et  $m$  zones horizontales (d'égale hauteur), on obtient un vecteur à  $n \times m$  composantes.

La méthode de décision proposée (K plus proches voisins) nécessite de disposer, préalablement à toute reconnaissance, de l'ensemble des points formant la base d'apprentissage (fournie par la phase d'apprentissage). La règle des KPPV est alors la suivante: étant donnée une distance sur  $\mathbb{R}^d$ , calculer la distance du point à classer, représentant la forme à reconnaître, à tous les points de la base d'apprentissage et affecter à ce point la classe la plus représentée parmi ses  $K$  plus proches voisins.

- Manipulation

- Ecrire une fonction qui prend en entrée les coordonnées du chiffre dont on souhaite extraire les densités par zoning et qui renvoie un vecteur à  $d$  composantes. On pourra choisir dans un premier temps  $n = m = \sqrt{d} = 5$ . Les densités dans chaque zone devront être normalisées (en les divisant par exemple par la surface de la zone) puisque les chiffres ne sont pas tous de même taille.
- Apprentissage du classifieur : Il s'agit de déterminer, pour chaque chiffre de l'image "app.tif", son vecteur de caractéristiques puis de sauvegarder tous les vecteurs de caractéristiques dans un fichier d'apprentissage.
- Décision : Il s'agit de fournir pour chaque chiffre à reconnaître de l'image "test.tif" un vecteur des probabilités a posteriori d'appartenance à chacune des 10 classes, ces probabilités peuvent être obtenues par l'estimation suivante:

$$\hat{p}(\omega_i/x) = \frac{k_i}{k}$$

où  $k_i$  est le nombre de voisins appartenant à la classe  $\omega_i$ . Ces vecteurs de probabilités seront à sauvegarder dans un fichier qui servira d'entrée à l'étape de combinaison de classifieurs.

- Analyse et conclusion

Evaluer les performances de ce deuxième classifieur (taux de reconnaissance). Discuter du choix des caractéristiques et notamment de l'influence sur les performances de ce classifieur du nombre de zones retenues. Discuter également de l'influence des différentes valeurs de  $K$ .

## 6 Combinaison de classifieurs

- Principe

On dispose maintenant des décisions prises indépendamment par chacun des deux classifieurs, on peut donc combiner les résultats. Puisque les deux classifieurs fournissent des probabilités a posteriori d'appartenance à chacune des classes, les opérateurs de combinaison adaptés à ce type de problème sont des opérateurs de type mesure. Les opérateurs les plus simples sont la somme et le produit des probabilités.

Si  $p_1(\omega_i/x)$  et  $p_2(\omega_i/x)$  désignent les probabilités a posteriori affectées à la classe  $\omega_i$  par les classifieurs 1 et 2 respectivement, la probabilité a posteriori pour la classe  $\omega_i$  peut s'écrire :

$$p_s(\omega_i/x) = p_1(\omega_i/x) + p_2(\omega_i/x)$$

$$p_p(\omega_i/x) = p_1(\omega_i/x)p_2(\omega_i/x)$$

pour les opérateurs somme et produit respectivement.

- Manipulation

- (i) Combinaison des deux classifieurs par les opérateurs somme et produit : A partir des deux fichiers contenant les décisions des deux classifieurs, il s'agit de fournir, pour chaque chiffre à reconnaître, le vecteur final des probabilités d'appartenance aux 10 classes.

- Analyse et conclusion

Evaluer les performances de ces deux méthodes de combinaison. Discuter de la différence des performances obtenues par les deux opérateurs de combinaison en référence aux performances obtenues par les deux classifieurs pris isolément.

## 7 Conclusion

Quels sont les avantages et les inconvénients de ce système de reconnaissance de chiffres? Basez votre discussion sur le choix des caractéristiques et des méthodes de décision proposées et leurs optimisations possibles. Discuter des tailles relatives des bases d'apprentissage et de test et de leur influence sur les performances de chacun des classifieurs et du système complet de reconnaissance.

## 8 Travail supplémentaire facultatif

Programmer d'autres jeux de caractéristiques et/ou programmer d'autres classifieurs (distance de Mahalanobis, perceptron, MLP, ...) et/ou programmer d'autres opérateurs de combinaison.

Etudier l'influence de ces différents choix sur les performances du système complet de reconnaissance.

## 9 Consignes pour le compte-rendu

Un compte-rendu du TP (au format PDF) sur la reconnaissance de chiffres manuscrits et les codes sources du programme sont à remettre pour le **Lundi 2 Janvier 2017** au plus tard (1 par **binôme**).

Quelques consignes sur le compte-rendu et les sources à remettre :

- Un rapport n'est pas un listing de code additionné de quelques copier-coller de courbes. Ne négligez pas l'importance du compte-rendu et le temps à y consacrer.
- Le rapport devra notamment décrire les tests et évaluations effectués. L'analyse et l'interprétation des résultats obtenus sont essentielles.
- Dans une introduction, les objectifs seront rappelés.
- Dans une conclusion, vous ferez un bilan des développements réalisés, des résultats obtenus, des analyses et conclusions tirées. Vous pourrez également suggérer des améliorations et perspectives à ce travail.
- Décrire les structures de données utilisées pour les variables "complexes" utilisées (par exemple, la structure choisie pour sauvegarder les extrémités du rectangle englobant de chaque chiffre).
- Chaque résultat (image, graphe ou valeur numérique) doit être commenté : description de ce qui est représenté, commentaire et analyse des résultats.
- Les sources comprennent les fonctions (commentées) développées et un script (commenté) permettant d'exécuter l'ensemble des tests effectués et de visualiser les résultats (voire une interface graphique).

- Les graphes doivent être annotés : donnez un titre, une légende et indiquez ce que représentent les axes. Choisissez des couleurs et symboles qui aident à la compréhension.

Le rapport au format PDF ainsi que les sources seront remis :

- dans une archive ZIP nommée comme suit : **A2016\_TPIN54\_NomEtudiant1\_NomEtudiant2.zip**
- au plus tard le **Lundi 2 Janvier 2017**
- par mail à l'adresse : **cindy.cappelle@utbm.fr**
- objet du mail : **[IN54] TP NomEtudiant1 NomEtudiant2**