# Classification supervisée d'images de document et contraintes industrielles

Fabien Carmagnac, Pierre Héroux, Eric Trupin

# Classification supervisée d'images de document et contraintes industrielles

Fabien Carmagnac[1,2] – Pierre Héroux[1] – Éric Trupin[1]

[1] Laboratoire PSI - CNRS FRE 2645 - Université de Rouen
UFR des Sciences et Techniques
F-76 821 Mont-Saint-Aignan cedex

[2] A2iA SA
40 bis rue Fabert
75 007 Paris cedex - France

Fabien.Carmagnac@a2ia.com
{Pierre.Heroux,Eric.Trupin}@univ-rouen.fr

**Résumé** : *Cet article présente une méthode originale pour la classification supervisée répondant à des contraintes industrielles fortes. Cette méthode propose en effet une solution aux problèmes des bases d'apprentissage réduites, de vitesse de traitement et de sélection de caractéristiques en fonction du problème à traiter. On montre comment le calcul d'une seule distance, au sens d'une métrique dans un espace de caractéristiques, entre l'élément à classer et un point de vue permet d'évaluer une fonction d'appartenance pour toutes les classes candidates. Cette idée est exploitée dans un algorithme itératif implanté au sein du module de classification d'une application de lecture automatique des champs manuscrits et typographiés sur des images de document. Des résultats expérimentaux sont présentés et comparés avec des approches classiques.*

**Mots-clés** : Classification supervisée, images de document, application industrielle

## 1 Introduction

Si de plus en plus d'applications commerciales reprennent à leur compte les travaux de recherche, les contraintes industrielles ne cessent de proposer de nouveaux défis. Si des applications de lecture automatique de documents (code postaux, chèques, formulaire) ont vu le jour, elles ont également mis en lumière certains besoins spécifiques concernant les temps de traitement, la généralisation et la robustesse malgré des bases d'apprentissage restreintes. Cet article propose une méthode de classification d'images de document adaptée à ces contraintes. Cette méthode est implantée afin d'étendre les fonctionnalités d'une application existante (A2iA FieldReader). Initialement, cette application réalisait, la lecture automatique des champs manuscrits et typographiés sur des documents de la même classe. Un modèle de lecture du document contient des informations relatives à la localisation, la nature, la syntaxe, la signification et des règles de cohérence entre les différents champs. Ce modèle de lecture permet de guider, de valider et donc d'améliorer la lecture automatique. L'extension proposée consiste à doter le système d'un module de classification d'images de document qui lui permettra de traiter des documents de classes différentes en associant à chacune d'elles un modèle de lecture différent. Le module de classification doit alors répondre à un certain nombre de contraintes. En effet, afin que le produit reste viable au niveau commercial il doit être :

– simple d'utilisation de telle sorte qu'un utilisateur non spécialiste en analyse de document ou en classification puisse facilement le prendre en main ;
– souple afin qu'il puisse s'adapter à différents problèmes (nombre et nature des classes à discriminer) ;
– rapide, l'étape de classification ne devant pas se traduire par un allongement trop conséquent du temps de traitement ;
– robuste malgré le fait que les utilisateurs ne consacrent que très peu de temps à l'annotation de base d'apprentissage.

Afin de tendre vers ce dernier objectif, nous avons conçu un module d'aide à l'annotation de bases d'apprentissage basée sur un algorithme de classification semi-supervisée. La description de cet outil fait l'objet d'une autre communication du CIFED [CAR 04]. En aval, se place un module de classification supervisée, dont la description fait l'objet du présent article, en se basant sur les annotations produites.

La section 2 précise les implications des contraintes que doit observer le module de classification. La section 3 énonce des définitions nécessaires à la compréhension de l'approche dont le principe est donné en section 4. Des résultats expérimentaux sont comparés à des approches classiques en section 5.

## 2 Contexte et contraintes

Nous donnons dans cette section, les implications des contraintes industrielles sur les choix pour la méthode de classification. En premier lieu, le module de classification doit s'adapter aux besoins spécifiques du client. Nous ne devons donc présupposer ni le nombre, ni la nature des classes. Il n'est alors pas possible de connaître *a priori* quelles seront les caractéristiques les plus discriminantes, ce choix devant alors être effectué en ligne. Il n'est pas non plus possible de présupposer des compétences des utilisateurs en traitement

d'image de document ou en classification.
Dans notre cas, la notion de classe de documents est définie par les traitements qui sont appliqués ultérieurement, guidés par le modèle de lecture. Certaines images de classes différentes peuvent se révéler d'un aspect très proches, alors que des images de la même classe peuvent avoir des structures dissemblables comme le montrent les exemples de la figure 1. Par ailleurs, en fonction du problème (champs à lire définis dans le masque de lecture), deux mêmes images peuvent se trouver ou non dans la même classe.
Le temps de classification doit rester acceptable, et ce quel que soit le nombre de classes à discriminer. Il doit donc être une fonction sub-linéaire du nombre de classes. La robustesse doit également être assurée malgré des bases d'apprentissage réduites, interdisant ainsi l'utilisation des techniques neuromimétiques. Enfin, le caractère incrémental en nombre de classes permettant de faire évoluer le système doit être un objectif qui doit poursuivi.

# 3 Définitions

Cette section introduit la terminologie utilisée dans la description du principe de classification. Les concepts de classe de documents, de caractéristiques et de distance entre images de documents y sont détaillés.
Pour un problème de classification particulier, on dispose d'une base d'apprentissage contenant des images de document ; cet ensemble est noté $ImageSet$. Ces images représentent les classes à discriminer. L'ensemble des classes du problèmes est noté $ClassSet$.
Les caractéristiques utilisées peuvent être numériques, syntaxiques et/ou structurelles. Il est possible d'associer à chaque jeu de caractéristiques plusieurs métriques permettant alors de calculer autant de distances entre deux images de document (distance euclidienne, distance de Hamming, distance du Max, distance du Min [RIB 98], distance d'édition [WAG 74], distance entre graphe [BUN 90]). L'espace associé à un jeu de caractéristiques $F$ est noté $FeatSpace_F$. $M_F$ est une métrique associée au jeu de caractéristiques $F$.

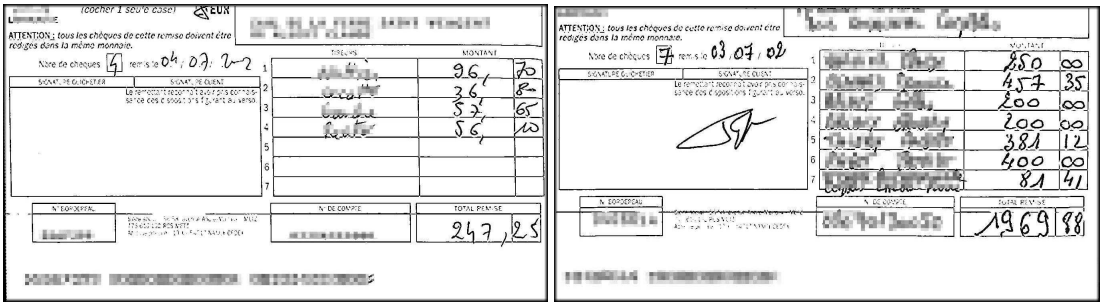
$$\begin{array}{rcl} F & : & ImageSet \rightarrow FeatSpace_F \\ M_F & : & FeatSpace_F \times FeatSpace_F \rightarrow [0,1] \end{array} \quad (1)$$

On peut alors calculer selon chaque métrique $M_F$ une distance entre deux images $I_1$ et $I_2$. Une distance calculée entre images provenant de la même classe est appelée *distance intra-classe*. Une distance calculée entre images provenant de deux classes différentes est appelée *distance inter-classe*.

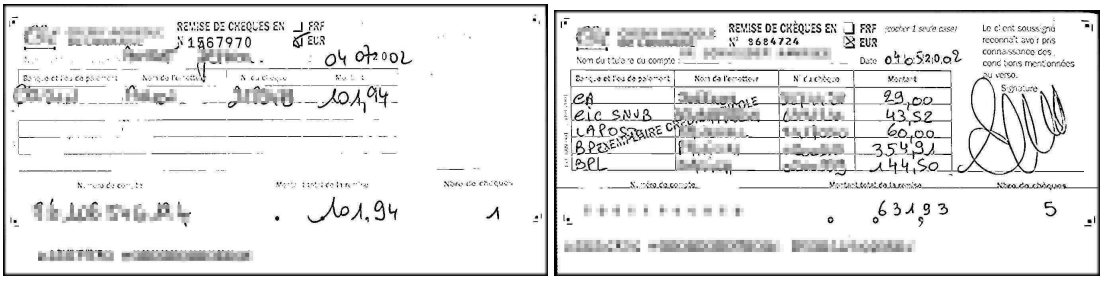
$$\begin{array}{rcl} D_{M_F} : ImageSet \times ImageSet & \rightarrow & [0,1] \\ D_{M_F}(I_1, I_2) & = & M_F(F(I_1), F(I_2)) \end{array} \quad (2)$$

# 4 Principe de classification

Dans les méthodes classiques, les images de documents sont représentées par des points dans différents espaces de caractéristiques [BEL 92] [FUK 90] [MIL 93] [TOU 74]. Les classes d'images sont alors représentées par des nuages de points. Notre approche est différente des méthodes de classification couramment utilisées [COV 67], [DUD 73], [FU 82] (réseaux de neurones, plus proches voisins, arbres de décision, machines à support vecteur), qui cherchent à établir les frontières entre les classes dans l'espace de caractéristiques, car l'espace des distances entre les vecteurs de caractéristiques est privilégié. Ce principe est inspiré des travaux sur l'optimisation de la recherche de plus proches voisins développé par Vidal [VID 94], Micó et al. [MIC 94] puis Moreno-Seco et al. [MOR 02] qui travaillent dans un espace de métriques permettant ainsi d'unifier des caractéristiques numériques, syntaxiques et structurelles. La distance entre deux points du même espace de caractéristiques est représentée par un point dans un espace des distances unidimensionnel. Un point dans cet espace des distances représente la distance au sens d'une métrique entre deux points de l'espace de caractéristiques à $n$ dimensions.
Le représentant $I^*_{M_F,C}$ d'une classe $C$ au sens d'une métrique $M$ dans un espace de caractéristiques $F$ définit un référentiel dans lequel se situent toutes les classes $C' \in ClassSet$ dans $FeatSpace_F$. Grâce à la métrique $M_F$, on peut calculer les distances entre $C$ et toutes les autres classes dans ce référentiel.
Soit $\tilde{I}$ une image de document dont on cherche la classe. Soit $\tilde{C}$ sa classe réelle. La position de $\tilde{I}$ dans le référentiel est alors déterminée en calculant selon $M_F$ la distance $D_{(M_F,C)}(\tilde{I})$ entre $\tilde{I}$ et $C$. Bien qu'une seule distance soit calculée, on obtient une information concernant l'appartenance de $\tilde{I}$ à toutes les classes. En effet, plus $\tilde{I}$ est éloignée d'une classe dans ce référentiel, plus la probabilité qu'elle en soit un membre est faible.
La figure 2 illustre le mécanisme de projection des classes et du point $\tilde{I}$ à classer depuis l'espace des caractéristiques (partie supérieure des figures) vers l'espace des distances (axe horizontal). Sur chacune de ces figures représentant le même espace de caractéristiques, un point de vue différent est choisi. Les classes sont circonscrites par des couronnes centrées sur le point de vue. Chaque couronne est projetée comme un intervalle sur l'espace des distances.
La phase d'apprentissage consiste à déterminer les intervalles projetés pour l'ensemble des points de vue. Elle est effectuée en deux temps. Lors de la première étape, pour chaque espace de caractéristiques, les représentants de chaque classe sont déterminés à partir des distances calculées entre tous les prototypes de la base d'apprentissage de la classe considérée (distances intra-classe). Le représentant peut être choisi de plusieurs façon (centre de gravité de la classe, prototype minimisant la variance intra-classe, prototype minimisant la plus grande distance intra-classe induite). Lors de la seconde étape, on calcule la distance entre ces représentants et les prototypes des autres classes. On détermine ainsi les intervalles de distance projeté. Des fonctions d'appartenance floues prenant en compte la dynamique sont induites de ces intervalles. Cette fonction vaut 1 à l'intérieur de l'intervalle et décroît avec l'éloignement de façon inversement proportionnelle à la largeur de l'intervalle.
La classification est réalisée en filtrant de façon itérative la liste des classes candidates (initialisée avec l'ensemble des classes). À chaque itération, on sélectionne parmi tous les points de vue celui qui permet de minimiser la probabilité qu'un point de l'espace des caractéristique se trouve pro-

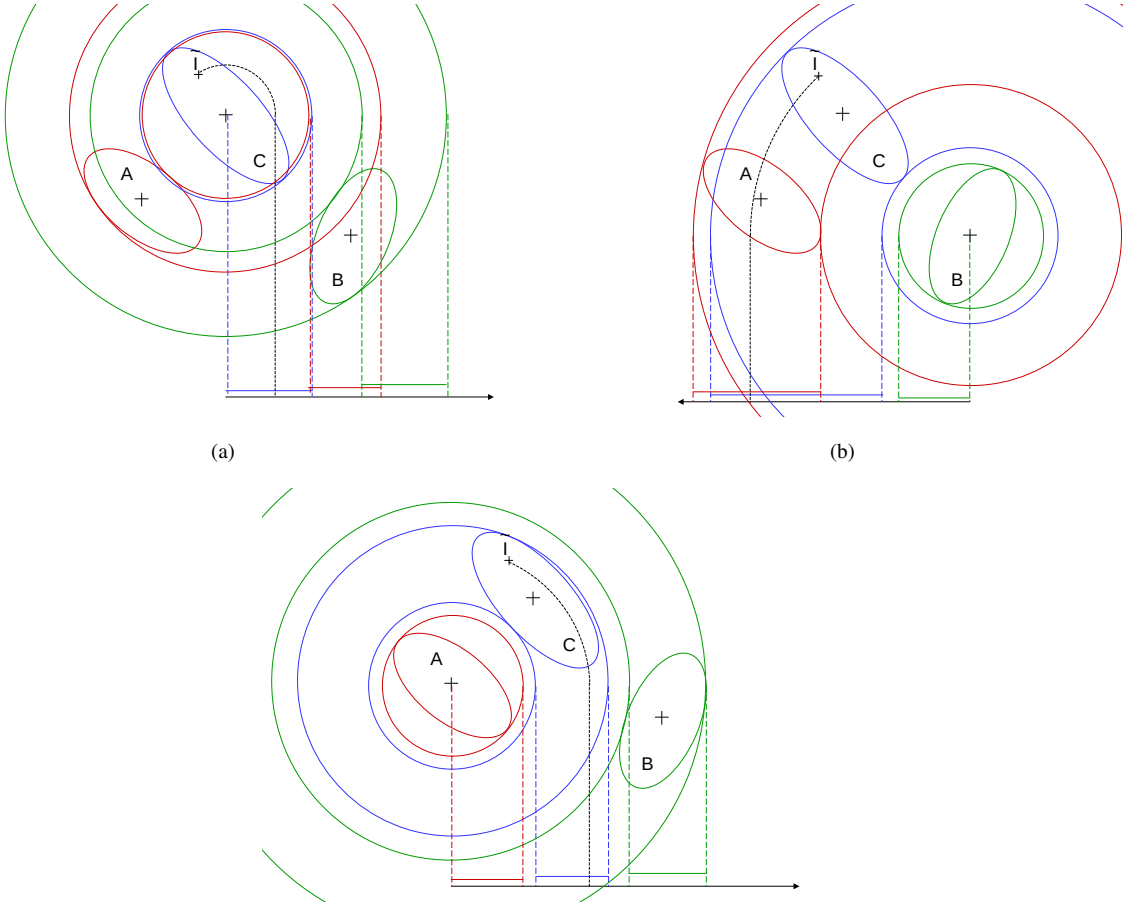


(a) Deux images de la classe 1



(b) Deux images de la classe 2

FIG. 1 – Exemples d'images de DB0




(a) (b)

(c)

FIG. 2 – Principe et choix du point de vue

jeté dans plusieurs intervalles. Ce point de vue est celui qui permet *a priori* la meilleure discrimination entre classes. Le point de vue représenté sur la figure 2(c) est le meilleur des trois, en effet, les intervalles engendrés par la projection des couronnes centrées sur le représentant de la classe $C$ ne présente aucun recouvrement. Ainsi, un point de l'espace de caractéristiques sera projeté dans, au plus, un intervalle. Dans un cas réel, avec un nombre de classes plus important, ce critère permet de minimiser le nombre d'intervalles dans lequel un point de l'espace des caractéristiques sera projeté. Le filtrage de la liste des classes candidates revient à supprimer les intervalles correspondants pour chaque point de vue, la sélection du meilleur point de vue doit être réitérée à chaque passe de l'algorithme.
Dans un deuxième temps, on calcule la distance entre le point à classer et le point de vue sélectionné selon la métrique correspondante. Cette distance correspond à l'abscisse dans l'espace projeté des distances. Les figures 2(a), 2(b) et 2(c) montrent comment le point $\tilde{I}$ est projeté sur chaque point de vue. On constate, en particulier qu'un choix erroné de point de vue (Fig. 2(a) et 2(b)) a pour effet de projeter $\tilde{I}$ dans deux intervalles. Cependant cela permet d'éliminer l'hypothèse que ce point puisse appartenir à la classe B. La liste des classes candidates serait alors filtrée en conséquence. En revanche, le choix du meilleur point de vue (Fig. 2(c)) permet d'éliminer, en un seul calcul de distance, les hypothèses des classes A et B.

**Algorithme 4.1**

```
Candidates ← ClassSet
TantQue Card(Candidates) > 1
 choisir le meilleur point de vue (M_F, C)
 calculer d=D_(M_F,C) (Ĩ)
 Pour toutes les classes calculer la
fonction d'appartenance de Ĩ
 Filtrer la liste des classes candidates
  (critère : seuil de la fonction
d'appartenance)
finTantQue
si Card(Candidates) > 0
 alors C=Candidates[0] sinon rejeter Ĩ
```

# 5 Résultats expérimentaux

La version actuelle du module de classification dispose de 6 extracteurs de caractéristiques (relatives à l'image [CLA 00] [UNS 93] et des éléments de structure du document [HÉR 98] [AZO 95]). Les métriques sont basées sur la distance euclidienne pour les caractéristiques numériques et sur la distance d'édition pour les caractéristiques ayant trait à la structure du document. Dans les tests présentés dans cette section, le point de vue sélectionné est celui minimisant l'écart-type de la distance intra-classe.
Les tests effectués portent sur 4 problèmes de classification d'images représentant des remises de chèques (Fig. 1, 4 et 5) des formulaires (Fig. 3 de [CAR 04] publié dans ces mêmes actes) et des tickets d'embarquement (Fig. 3). Les résultats expérimentaux figurent dans le tableau 1. Aucune reconfiguration n'a été opérée entre les 4 tests. Comme indiqué en section 2, la notion de classe est définie par la position et la nature des champs à lire. Les prétraitements appliqués à l'image (binarisation) sont une autre source de variabilité. La figure 1 montre des exemples d'images de deux classes différentes. Enfin, l'apprentissage est réalisé avec un faible nombre d'exemples (entre 5 et 10 images par classe). Le taux modeste de classification obtenu sur la base $DB4$ était prévisible. En effet, après l'apprentissage, le module a signalé que deux couples de classes ne pouvaient être discriminé car dans aucun des points de vue les intervalles n'étaient distincts. Cependant, la difficulté du problème est évidente lorsqu'on regarde la proximité des différentes classes.
Les résultats présentés dans le tableau 1 ont été comparés à ceux donnés par un classifeur $k$-ppv. Dans un premier temps, un classifeur a été implanté pour chacun des 6 jeux de caractéristiques et pour plusieurs valeurs de $k$ (1, 3, 5 et 10). Ce dernier paramètre ne semble pas avoir d'influence considérable. Nous avons constaté que certaines caractéristiques semblaient bien adaptées pour certains jeux d'images et absolument pas pour d'autres. En effet, le meilleur résultat a été de 95% pour un jeu de caractéristiques sur une base, alors que ce même jeu a donné des résultats inférieurs à 50% sur d'autres bases. Dans un second temps, toutes les caractéristiques ont été rassemblées dans un unique vecteur. Les résultats obtenus ont été inférieurs à 40% pour chacune des bases testées. Celà illustre le fait que les classifieurs $k$-ppv doivent être appliqués en ayant mené une étude, dépendante de la base, sur la pertinence des caractéristiques.
Cette comparaison valide notre approche de sélection dynamique de caractéristiques. Par ailleurs, là où le classifieur $k$-ppv nécessite le calcul d'une distance avec chaque point de la base d'apprentissage, notre approche ne nécessite qu'un calcul par itération, les extracteurs de caractéristiques correspondants n'étant déclenchés que lorsque celà s'avère nécessaire.

# 6 Conclusion

Cet article présente une stratégie originale de classification appliquée au problème de la classification d'images de documents. Cette stratégie est implantée comme module au sein d'un produit dont le but est la lecture automatique de champs manuscrits sur des images de documents hétérogènes provenant de différentes classes.
La stratégie implantée tente de satisfaire des contraites induites par une problématique industrielle. Il s'agit, en effet, d'appréhender des problèmes de classification d'images de document pour lesquels on ne peut présumer ni de la nature ni du nombre de classe Ces classes étant définies en fonction des traitements ultérieurs, la variabilité intra-classe est parfois importante. Une autre difficulté provient du fait que les bases d'apprentissage sont réduites à environ 5 à 10 images par classe.
Les données produites pendant la phase d'apprentissage permettent de choisir un point de vue (espace de caractéristique, métrique et origine de la comparaison) qui permettent de discriminer au mieux les classes candidates. La distance, au sens d'une métrique et d'un espace de caractéristique, entre le représentant d'une classe et une image de classe inconnue, donne non seulement une information sur l'appartenance de

| | Nombre d'images | Nombre de classes | Nombre d'itérations | Bonne classification | Taux de rejet | Taux de confusion |
|---|---|---|---|---|---|---|
| DB0 | 25125 | 2 | 1 | 100% | 0% | 0% |
| DB1 | 123 | 8 | 2 to 4 | 96% | 4% | 0% |
| DB2 | 65689 | 6 | 2 to 3 | 92% | 4% | 4% |
| DB3 | 8770 | 6 | 2 to 3 | 95% | 4% | 1% |
| DB4 | 2550 | 14 | 2 à 4 | 70% | 25% | 5% |

TAB. 1 – Résultats

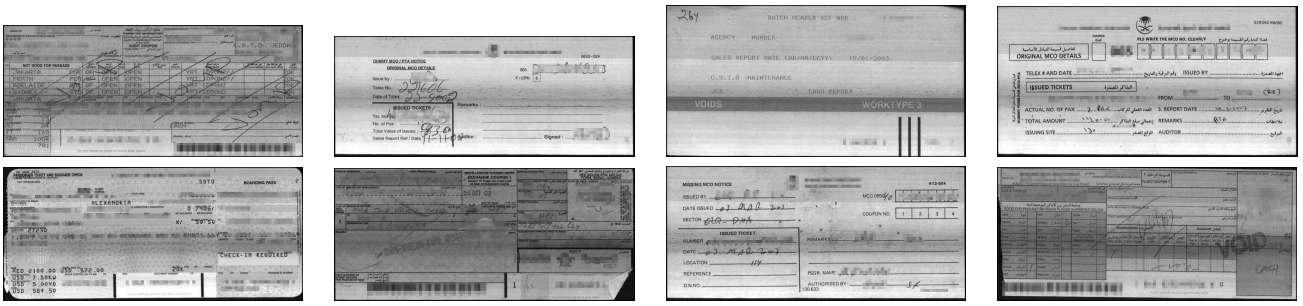


FIG. 3 – Un exemple pour les 8 classes de DB1


FIG. 4 – Un exemple pour les 6 classes de DB2


FIG. 5 – Un exemple pour les 6 classes de DB3

l'image à la classe considérée, mais également sur l'appartenance aux autres classes. La classification s'opère en filtrant de façon itérative un ensemble de classes candidates jusqu'à ne plus avoir qu'une classe ou plus du tout (le document est rejeté).

Cette stratégie présente un certain nombre d'avantages par rapport aux méthodes classiques. Tout d'abord, le processus de classification est une fonction sub-linéaire du nombre de classes. D'autre part, le module de classification peut facilement être configuré. En particulier, différentes règles peuvent premettre de filtrer la liste des classes candidates (selon la valeur de la fonction d'appartenance ou éliminer une proportion et donc fixer le nombre maximum d'itérations). Enfin, de nouveaux extracteurs de caractéristiques et les différentes métriques associées peuvent facilement être intégrés. Ces nouvelles caractéristiques (éventuellement dédiées à la résolution d'un problème) sont rendues disponibles. Elles seront utilisées ou pas dans le processus de classification sans que cela n'ait d'influence sur le temps de classification. On pourrait même imaginer analyser pour chaque problème quelles ont été les caractéristiques utilisées et dans quel contexte. Par exemple, on pourrait en conclure qu'un jeu de caractéristiques n'est pas utile, ou qu'il n'est utile que pour discriminer deux classes particulières difficilement séparables par les autres caractéristiques.

Le principal avantage de raisonner dans un espace de distance est que cela permet d'utiliser des caractéristiques de différentes natures. Des caractéristiques numériques, syntaxiques ou structurelles peuvent être utilisées au sein d'un même jeu de caractéristiques si on peut lui associer, au moins, une métrique permettant de calculer une distance entre deux images.

La phase d'apprentissage est incrémentale. Si une nouvelle classe doit être ajoutée, les distances calculées précédemment peuvent être conservées ; seules les distances avec les nouveaux documents doivent être calculées. Enfin, de nouveaux vecteurs de caractéristiques ou de nouvelles métriques peuvent également être ajoutés sans remise en cause des données précédemment extraites. Il semble que cette stratégie puisse être appliquée à d'autres problèmes de classification supervisée.

Les résultats présentés portant sur plusieurs exemples de bases semblent intéressants. Ils pourraient cependant être améliorés sur plusieurs aspects (critère de choix du représentant, détermination dynamique du meilleur point de vue...). D'autres perspectives d'amélioration concernent les vecteurs de caractéristiques utilisés. De nouveaux vecteurs de caractéristiques pourraient être obtenus par combinaison de ceux existants. Les plus discriminants pourraient être déterminés par exemple par l'utilisation d'algorithmes génétiques appliqués sur les données de la base d'apprentissage.

# Références


[AZO 95] AZOKLY A. S., Uniform Approach for Recognition of Physical Layout of Documents based on White Space Analysis, PhD thesis, Université de Fribourg, 1995.

[BEL 92] BELAID A., BELAID Y., *Reconnaissance des Formes : Méthodes et Applications*, InterEditions, 1992.

[BUN 90] BUNKE H., Syntactic and Structural Pattern Recognition Theory and Applications, *Series in computer science*, vol. 7, 1990.

[CAR 04] CARMAGNAC F., HÉROUX P., TRUPIN E., Classification Semi-Supervisée basée sur des Algorithmes de CAH Multi-métriques, *Actes du Huitième Colloque International Francophone sur l'Écrit et le Document* CIFED'04, 2004.

[CLA 00] CLAVIER E., Stratégies de tri : un système de tri des formulaires, Thèse de Doctorat, Université de Caen, 2000.

[COV 67] COVER T. M., HART P. E., Nearest Neighbor Pattern Classification, *IEEE Transaction on Information Theory*, vol. 13, n° 1, 1967, pp. 21-27.

[DUD 73] DUDA R. O., HART P. E., *Pattern Classification and Scene Analysis*, Wiley-Interscience Publication, 1973.

[FU 82] FU K. S., *Syntactic Pattern Recognition and Applications*, Prentice Hall, 1982.

[FUK 90] FUKUNAGA K., *Introduction to Statistical Pattern Recognition*, Academic Press Inc., $2^{nd}$ édition, 1990.

[HÉR 98] HÉROUX P., DIANA S., RIBERT A., TRUPIN E., Classification Methods Study for Automatic Form Class Identification, *14th IAPR International Conference on Pattern Recognition ICPR'98*, Brisbane, Australie, 1998, International Association on Pattern Recognition, pp. 926-928.

[MIC 94] MICÓ M. L., A New Version of the Nearest-Neighbour Approximating and Eliminating Search Algorithme (AESA) with linear preprocessing time and memory requirements, *Pattern Recognition Letters*, vol. 15, 1994, pp. 9-17.

[MIL 93] MILGRAM M., *Reconnaissance des Formes. Méthodes Numériques et Connexionnistes*, Armand Colin, Paris, 1993.

[MOR 02] MORENO-SECO F., MICÓ L., ONCINA J., Extending LAESA fast Nearest Neighbour to find the k-Nearest Neighbours, *Structural, Syntactic and Statistical Pattern Recognition*, n° 2396 Lecture Notes in Computer Science, pp. 691-699, Springer-Verlag, 2002.

[RIB 98] RIBERT A., Structuration évolutive de données : Application à la construction de classifieurs distribués, Thèse de Doctorat, Université de Rouen, 1998.

[TOU 74] TOU G., *Pattern Recognition Principles*, Addison-Wesley, 1974.

[UNS 93] UNSER M., ALDROUBI A., GERFEN C. R., A Multiresolution Image Registration Procedure Using Spline Pyramids, *Wavelet Applications in Signal and Image Processing*, vol. 2034, SPIE, 1993, pp. 160-170.

[VID 94] VIDAL E., New Formulation and Improvents of the Nearest-Neighbour Approximating and Eliminating Search Algorithm (AESA), *Pattern Recognition Letters*, vol. 15, 1994, pp. 1-7.

[WAG 74] WAGNER R. A., FISCHER M. J., The String-to-String Correction Problem, *Journal of the ACM*, vol. 21, n° 1, 1974, pp. 168-173.