Une méthode générique pour la classification automatique d'images à partir des pixels

Raphaël Marée*, Pierre Geurts*, Louis Wehenkel*

* Département d'Électricité, Électronique et Informatique, Institut Montefiore, B-4000 Sart-Tilman, Belgique
{Raphael.Maree,P.Geurts,L.Wehenkel}@ulg.ac.be,
http ://www.montefiore.ulg.ac.be/~maree/

Résumé. Dans cet article, nous évaluons une approche générique de classification automatique d'images. Elle repose sur une méthode d'apprentissage récente qui construit des ensembles d'arbres de décision par sélection aléatoire des tests directement sur les valeurs basiques des pixels. Nous proposons une variante, également générique, qui réalise une augmentation fictive de la taille des échantillons par extraction et classification de sous-fenêtres des images. Ces deux approches sont évaluées et comparées sur quatre bases de données publiques de problèmes courants : la reconnaissance de chiffres manuscrits (MNIST), de visages (ORL), d'objets 3D (COIL-20) et de textures (OUTEX).

1 Introduction

La classification automatique d'images a de nombreuses applications dans le domaine du contrôle qualité, de la biométrie (reconnaissance de visages), de la médecine, de la bureautique (reconnaissance de caractères), de la géologie (reconnaissance de textures de sols), ...

Ce problème est particulièrement difficile pour les méthodes traditionnelles d'apprentissage supervisé principalement à cause du grand nombre de variables d'entrées qui servent à décrire les images (les pixels). En effet, en présence d'un grand nombre de variables, les méthodes d'apprentissage souffrent la plupart du temps d'une grande variance qui dégrade leur précision et de plus elles présentent des temps de calcul très élevés. Pour gérer ce problème de dimensionnalité, la classification d'images repose généralement sur un pré-traitement spécifique à chaque problème qui réduit sa complexité en extrayant des caractéristiques pertinentes. Celles-ci sont ensuite utilisées en entrée d'une méthode traditionnelle d'apprentissage automatique éventuellement ajustée pour l'application. Il résulte de cette approche qu'une variation des conditions d'acquisition des images ou l'apparition d'un nouveau sous-problème implique d'adapter manuellement le pré-traitement et ce pour chaque nouveau problème, en tenant compte des spécificités de l'application.

Parallèlement, les récentes avancées en apprentissage automatique ont fait apparaître des méthodes capables de traiter des problèmes de plus en plus complexes sans utiliser aucune information a priori sur le domaine d'application. Elles rivalisent souvent avec les méthodes propres à ces domaines qui, elles, résultent pourtant d'une adaptation importante au problème d'application. Dans ce contexte, notre étude a pour

but d'évaluer la possibilité de proposer une méthode générique pour la classification d'images, de manière à s'affranchir de l'étape souvent laborieuse de pré-traitement.

Dans ce but, notre démarche a été de choisir un certain nombre de problèmes représentatifs du domaine de la classification d'images et d'utiliser la même méthode sur chacun d'eux sans aucun pré-traitement spécifique, c'est-à-dire en travaillant directement à partir des valeurs des pixels. Parmi les méthodes récentes potentiellement capables de traiter ces problèmes complexes, nous avons choisi une approche basée sur des ensembles d'arbres de décision et nous l'avons également combinée avec une technique simple et générique basée sur l'extraction de sous-fenêtres dans les images.

La structure de l'article présente les différentes étapes de cette approche. Dans la section 2, nous décrivons les bases de données qui ont été utilisées pour la validation sur divers types de problèmes représentatifs du but général de classification d'images. Au sein de la section 3, nous tentons de dégager les caractéristiques qui doivent être remplies par une méthode d'apprentissage générique pour le traitement d'images. Nous décrivons ensuite la méthode d'ensembles d'arbres de décision que nous avons choisi d'évaluer ainsi que la technique d'extraction de sous-fenêtres à laquelle nous l'avons combinée. La section 4 présente les résultats de nos expérimentations qui sont comparés à ceux d'approches spécifiques présentées dans la littérature.

2 Bases de données d'images

Dans cette section nous décrivons les bases de données d'images qui ont été utilisées pour l'évaluation. Outre le fait que ces bases de données sont publiques et gratuites, nous pensons qu'elles constituent un panel représentatif des problèmes courants de classification d'images. Les caractéristiques de ces bases de données sont résumées dans le tableau 1.

Base de données	Nb. objets	Nb. attributs	Nb. classes
MNIST	70000	784 ($28 * 28 * 1$)	10
ORL	400	10304 ($92 * 112 * 1$)	40
Coil-20	1440	16384 ($128 * 128 * 1$)	20
OUTEX	864	49152 ($128 * 128 * 3$)	54

TAB. 1 – Résumé des bases de données d'images étudiées.

2.1 MNIST

La base de données MNIST [1] contient 70000 images de chiffres écrits à la main par 500 personnes, de taille normalisée et centrée en images de 28×28 pixels avec 256 niveaux de gris par pixel. Le but est de construire un modèle qui classe les chiffres. Les différentes écritures se distinguent par des traits fins ou épais, des caractères penchés,... Un aperçu de la base de données est proposé à la figure 1.

[1] http ://yann.lecun.com/exdb/mnist/

FIG. 1 – Aperçu de la base de données de chiffres manuscrits MNIST : 20 chiffres extraits aléatoirement.

FIG. 2 – Aperçu de la base de données de visages ORL : 2 poses pour 3 sujets.

2.2 ORL

La base de données ORL (AT&T Laboratories Cambridge)[2] contient des images du visage de 40 personnes, chacune de ces personnes a été photographiée à 10 reprises, sous différentes conditions relatives à l'éclairage, l'expression faciale (yeux ouverts ou fermés, sourire ou non, avec ou sans lunettes) et de faibles différences d'inclinaison de la tête. Chaque image est de taille 92×112 pixels avec 256 niveaux de gris par pixel. Le but est de reconnaître les personnes. Quelques exemples de cette base de données sont montrés à la figure 2.

2.3 COIL-20

La bibliothèque d'images de l'université de Columbia [3] propose des bases de données d'images d'objets 3D. COIL-20 est un ensemble d'images représentant 20 objets. Il contient 1440 images normalisées, prises par une caméra fixe, avec les objets se trouvant sur une platine motorisée réalisant une rotation de 360 degrés par intervalles de 5 degrés. Cela correspond donc à 72 images par objet et chaque image est de taille 128×128 pixels avec 256 niveaux de gris par pixel. Le but est de reconnaître les différents objets (boîtes, flacons, animaux en plastique, ...) de la base de données dont un aperçu est présenté à la figure 3.

2.4 OUTEX Contrib_TC_0006

OUTEX (Ojala et al. 2002) est une structure pour l'évaluation empirique d'algorithmes d'analyse de textures . La base de données Contrib_TC_0006 incluse dans cet environnement [4] est construite à partir des textures VisTeX [5]. Elle contient 54 tex-

[2] http ://www.uk.research.att.com/facedatabase.html
[3] http ://www.cs.columbia.edu/CAVE/
[4] http ://www.outex.oulu.fi/outex.php
[5] http ://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html

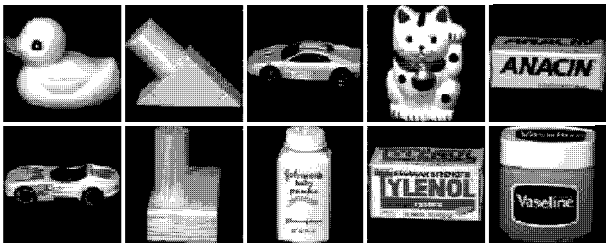

FIG. 3 – Aperçu de la base de données d'objets COIL-20 : 1 pose pour 10 objets.

tures en couleurs avec 16 sous-images de taille 128 × 128 pour chaque texture VisTeX originale. Le but est de reconnaître les différentes textures (bois, métal, eau, fleurs, ...).

3 Une méthode générique pour la classification automatique d'images

Dans cette section nous résumons les caractéristiques génériques que nous pensons devoir être satisfaites par une méthode d'apprentissage pour l'exploitation de base de données d'images. Ensuite, nous présentons brièvement la méthode d'ensembles d'arbres extrêmement aléatoires proposée dans (Geurts 2002) et utilisée telle quelle lors des premières expérimentations. Comme les résultats le montreront dans la section 4, cette méthode ne marche pas de manière satisfaisante pour certains problèmes. Nous proposons alors une variante, avec extraction de sous-fenêtres, qui satisfait également les conditions génériques et qui tient compte de la contiguïté des pixels.

3.1 Caractéristiques nécessaires

De manière plus formelle, un algorithme générique pour la classification d'images prend en entrée un ensemble d'images pré-classifiées : $LS = \{(A^i, c^i), i = 1, \ldots, N\}$, où A^i est une matrice $W_x \times W_y$ décrivant l'image et $c^i \in \{1, \ldots, M\}$ est sa classe (parmi M classes). L'élément $a^i_{k,l}$ de la matrice A^i $(k = 1, \ldots, W_x, l = 1, \ldots, W_y)$ décrit le pixel de l'image i à la position (k, l) par un seul entier dans le cas d'images en niveau de gris ou par 3 valeurs RGB entières dans le cas d'images couleur.

Pour pouvoir exploiter les informations brutes des pixels, la méthode d'apprentissage doit donc être capable de gérer efficacement un grand nombre de données : en termes d'objets, c'est-à-dire le nombre d'images, mais surtout en termes d'attributs : pour une image de taille 128 × 128 en niveaux de gris il y a $128 * 128 = 16384$ colonnes d'attributs par objet, c'est-à-dire le nombre de valeurs de pixels, et 3 fois plus lorsque l'image est en couleurs. La plupart des méthodes traditionnelles d'apprentissage automatique peuvent difficilement traiter de manière efficace un si grand nombre de données et/ou leur précision est pénalisée par une variance élevée (pour les arbres de décision, voir (Geurts 2002)). C'est pourquoi ces méthodes sont habituellement

précédées par une étape de pré-traitement spécifique au problème d'appplication qui extrait en nombre réduit des paramètres qui sont ensuite utilisés comme attributs de la méthode. Dans notre approche, la méthode d'apprentissage doit au contraire pouvoir traiter des vecteurs contenant un grand nombre de valeurs numériques et cela sans information supplémentaire. Ainsi, aucun pré-traitement ne devrait être nécessaire à l'exception d'une procédure identique de conversion des images (gif, jpeg, ...) en valeurs entières basiques (ppm).

Une méthode générique pour la classification d'images doit donc principalement pouvoir résoudre les problèmes d'efficacité et de variance des méthodes traditionnelles. Récemment, de nouvelles méthodes satisfaisant ces critères (machines à support vectoriel, ensembles d'arbres, ...) sont apparues, et nous présentons maintenant l'une d'entre elles, ainsi qu'une variante, que nous validerons au sein de la section 4.

3.2 Ensembles d'arbres extrêmement aléatoires

Les méthodes d'ensemble ont actuellement beaucoup de succès en apprentissage. Ces méthodes consistent à améliorer une méthode d'apprentissage existante en combinant les prédictions de plusieurs modèles construits à l'aide de cette méthode à partir d'un même échantillon d'apprentissage. Elles sont particulièrement efficace en combinaison avec la méthode d'arbre de décision (Breiman 1984) qui, sinon, a souvent une précision peu compétitive avec d'autres méthodes. Dans cet article, nous proposons d'utiliser une méthode particulière de construction d'ensemble d'arbres de décision proposée dans (Geurts 2002). Cette méthode consiste à construire les arbres constituant l'ensemble en sélectionnant leurs paramètres (i.e. en classification d'images, les positions des pixels dont on teste les valeurs et les seuils auquel on compare ces valeurs) de manière complètement aléatoire. L'algorithme de construction d'un arbre particularisé au problème de la classification d'images est décrit dans la table 3.2[6]. Même si un arbre extrêmement aléatoire est souvent moins bon qu'un arbre classique seul, le fait d'en moyenner plusieurs améliore alors la précision (par réduction importante de la variance) en comparaison avec les arbres classiques ou d'autres méthodes d'ensembles (par exemple le bagging, (Geurts 2002)).

Nous pensons que les méthodes basées sur des ensembles d'arbres de décision et plus particulièrement la variante décrite ici constituent un bon point de départ dans notre recherche d'une méthode générique pour la classification d'images pour plusieurs raisons. D'abord, ce sont des méthodes générales qui ne font aucune hypothèse a priori sur les données. Elles ont été appliquées avec succès à de nombreux problèmes correspondant à des domaines d'application très divers (voir par exemple (Roli et Kittler 2002) pour quelques applications récentes). De plus, la version aléatoire proposée ici est particulièrement intéressante dans le contexte des images pour des raisons d'efficacité computationnelle. En effet, le choix aléatoire des tests à l'intérieur de l'arbre de décision rend la complexité de l'algorithme indépendante du nombre de variables d'entrée et d'ordre $N \log(N)$ par rapport à la taille de l'échantillon.

[6]Le seuil sur le score des tests, s_th, permet de rejeter le choix aléatoire de tests trop mauvais. Néanmoins, en pratique, la précision de la méthode n'est pas fortement influencée par la valeur de ce paramètre. Dans toutes nos expérimentations, où nous utilisons la mesure de score proposée dans (Wehenkel 1998), sa valeur sera dès lors fixée à 0.1.

Construire_arbre_aléatoire(LS) :

- Si LS contient des images appartenant toutes à la même classe, renvoyer une feuille étiquettée avec cette classe.
- Sinon :
  1. Faire $[a_{k,l} < a_{th}]$=Choisir_test_aléatoire(LS) ;
  2. Diviser LS en LS_{left} et LS_{right} selon le test $[a_{k,l} < a_{th}]$ et construire les sous-arbres $\mathcal{T}_{left}$ = Construire_arbre_aléatoire(LS_{left}) et $\mathcal{T}_{right}$ = Construire_arbre_aléatoire(LS_{left}) à partir de ces sous-ensembles ;
  3. Créer un nœud avec le test $[a_{k,l} < a_{th}]$, attacher $\mathcal{T}_{left}$ et $\mathcal{T}_{right}$ comme successeurs de ce noeud et renvoyer l'arbre résultant.

Choisir_test_aléatoire(LS) :

1. Sélectionner aléatoirement une position (k, l) ;
2. Sélectionner aléatoirement un seuil a_{th} selon une distribution $N(\mu_{k,l}, \sigma_{k,l})$, où $\mu_{k,l}$ et $\sigma_{k,l}$ sont respectivement la moyenne et l'écart-type des valeurs de pixels $a_{k,l}$ dans LS ;
3. Si le score de ce test est supérieur à un seuil donné s_{th}, renvoyer le test $[a_{k,l} < a_{th}]$;
4. Sinon, retourner à l'étape 1 et sélectionner une autre position. Si toutes les positions ont déjà été considérées, renvoyer le meilleur test obtenu jusqu'ici.

TAB. 2 – Algorithme d'induction d'arbre aléatoire pour la classification d'images.

3.3 Extraction de sous-fenêtres

Même si la méthode précédente est efficace pour traiter un grand nombre de variables d'entrée, la complexité des arbres (c-à-d le nombre de variables qui sont combinées le long d'une branche pour faire une prédiction) est néanmoins limitée par la taille de l'échantillon. Pour certains problèmes, lorsque le nombre d'images est faible par rapport au nombre total de pixels, l'algorithme ne dispose pas de suffisamment de moyens pour fournir des modèles acceptables. Pour traiter ce problème, nous avons adopté une approche également générique et assez courante en classification d'images (voir par exemple (Hoque et Fairhurst 2000) ou (Dahmen et al. 2001)). Elle consiste à augmenter artificiellement le nombre de cas d'apprentissage en construisant des modèles sur des sous-fenêtres extraites à partir des images et en combinant les prédictions relatives aux sous-fenêtres pour classer une nouvelle image.

Avec cette technique, la construction d'un arbre aléatoire de l'ensemble se fait en deux étapes :

- Pour une taille de fenêtre fixée $w_1 \times w_2$, on extrait aléatoirement un certain nombre N_w de fenêtres à partir des images de la base de données. On associe à chacune de ces fenêtres la classe de l'image dont elle est extraite.
- On construit un arbre aléatoire à partir de ces N_w fenêtres en utilisant les valeurs des $w_1 * w_2$ pixels qui les décrivent.

Pour faire un prédiction avec un ensemble d'arbres aléatoires construits sur des sous-fenêtres, on utilisera la procédure suivante :

- On extrait toutes les sous-fenêtres possibles de taille $w_1 \times w_2$ à partir de l'image qu'on désire classer.
- On associe à l'image la classe majoritaire parmi les classes attribuées aux sous-fenêtres par l'ensemble d'arbres aléatoires.

Évidemment, ces modifications augmentent les temps de calcul des phases d'apprentissage et d'évaluation.

4 Expérimentations

Diverses expérimentations ont été réalisées avec la méthode d'ensembles d'arbres extrêmement aléatoires et sa variante avec sous-fenêtres sur les bases de données de la section 2. Nous comparons brièvement nos résultats avec ceux de plusieurs approches trouvées dans la littérature dont, pour chaque problème, la meilleure que nous connaissons. Néanmoins, les méthodologies de validation étant différentes d'une publication à l'autre, ces comparaisons doivent être considérées avec précaution comme le montrent les variations de certains de nos résultats.

Le seul paramètre utilisé par la méthode d'arbres extrêmement aléatoires est le nombre T d'arbres à construire. Cependant, les arbres étant indépendants, plus on construit d'arbres aléatoires, plus on améliore la précision. Dans nos expérimentations, nous avons utilisé pour chaque problème un nombre d'arbres suffisant pour que l'erreur sur l'ensemble de test soit stabilisée.

Pour la construction d'arbres sur des sous-fenêtres, les paramètres additionnels sont la taille de la fenêtre $w_1 \times w_2$ et le nombre N_w de sous-fenêtres extraites lors de la construction. Comme pour le nombre d'arbres de l'ensemble, la précision est une fonction croissante de N_w. Nous avons donc choisi une valeur $N_w = 120000$ pour tous les problèmes sauf pour MNIST où nous avons fixé N_w à 360000. Sur ce problème, la taille importante de l'échantillon de base motive une valeur plus élevée. Une valeur supérieure pourrait encore améliorer la précision mais deviendrait rédibitoire en termes de temps de calcul. La précision par contre est très sensible à la taille de la fenêtre qui est utilisée et la taille optimale est également dépendante du problème. Une manière de fixer automatiquement cette taille est par exemple d'utiliser la validation croisée. Néanmoins, nous avons déterminé cette valeur empiriquement.

4.1 MNIST

Dans la littérature, le protocole d'expérimentation pour cette base de données consiste généralement à utiliser les 60000 premières images pour l'ensemble d'apprentissage et les 10000 dernières pour l'ensemble de test. Nous avons suivi ce schéma, la validation croisée n'étant pas nécessaire en présence d'un si grand nombre de cas.

Plusieurs méthodes ont été proposées dans la littérature (LeCun et al. 1998) pour aborder ce problème dont des méthodes d'apprentissage automatique comme les plus proches voisins et les réseaux de neurones, généralement précédées par une phase de pré-traitement, et les machines à support vectoriel. En effet, certaines de ces approches

ont utilisé un pré-traitement spécifique à la reconnaissance de caractères destiné à augmenter la taille de l'échantillon en générant des versions déformées des images originales (combinaisons de décalages, inclinaisons, dimensionnements, compressions) ou consistant à corriger l'inclinaison des chiffres. Le taux d'erreur des méthodes référencées par (LeCun et al. 1998) varie de 12% à 0.7%. L'utilisation par (Burges et Schölkopf 1997) de machines à support vectoriel sans information a priori donne 1.1% d'erreur. Le tableau 3 résume nos résultats. Avec les arbres extrêmement aléatoires nous obtenons 3.26% et sa variante avec sous-fenêtres atteint 2.63% (avec $N_w = 360000$).

Méthode	**Taux d'erreur**
Arbres aléatoires ($T = 50$)	3.26%
Extraction de fenêtres (T=10, $w_1 = w_2 = 24$)	2.63%
Boosted LeNet-4 + distortions (LeCun et al. 1998)	0.7%

TAB. 3 – Résultats sur la base de données MNIST.

4.2 ORL

Étant donné le peu d'images qui constituent cette base de données et l'absence d'un protocole de test public, nous avons ici utilisé la validation croisée afin de mieux nous rendre compte des capacités de généralisation de nos deux méthodes.

Dans la littérature, divers algorithmes ont été testés sur cette base de données tels que des modèles de Markov cachés (Nefian et Hayes 1999), des réseaux de neurones à convolution (Lawrence et al. 1997), des machines à support vectoriel (Guo et al. 2000) et des variantes des plus proches voisins (Paredes et Perez-Cortes 2001). Ces études utilisent diverses étapes de pré-traitement pour réduire la dimensionnalité du problème (décomposition en images teintées avec des nuances de gris différentes et extractions de caractéristiques, sélection de pixels informatifs basée sur la variance de sous-fenêtres locales, opérations de gradient, ...). Ces expérimentations utilisent souvent des méthodologies différentes (moyenne de 3, 4 ou 5 exécutions sur des groupes de données distincts) ce qui rend la comparaison difficile. Les résultats rencontrés varient de 7.5% à 0% mais le nombre d'exécutions n'est pas précisé dans ce dernier cas.

Dans notre cas, nous avons moyenné les résultats de 100 exécutions en utilisant chaque fois 5 images distinctes tirées aléatoirement par classe pour chaque ensemble d'apprentissage et l'autre moitié des images pour chaque ensemble de test. Le tableau 4 donne la moyenne et l'écart-type pour chacune de nos deux méthodes. Pour les ensembles d'arbres aléatoires, la moyenne du taux d'erreur est de 4.56%. Avec l'extraction de sous-fenêtres, le taux d'erreur moyen est de 2.13%.

Méthode	**Taux d'erreur**
Arbres aléatoires ($T = 500$)	4.56% ± 1.43
Extraction de fenêtres ($T = 10$, $w_1 = w_2 = 32$)	2.13% ± 1.18
Local Features-based Nearest Neighbor (Paredes 2001)	0%

TAB. 4 – Résultats sur la base de données ORL.

4.3 COIL-20

Ce problème a été abordé dans la littérature à l'aide de différentes techniques pour la reconnaissance d'objets 3D utilisant des techniques de réduction paramétrique de l'espace d'entrée (Murase et Nayar 1995) ou un un pré-traitement qui caractérise structurellement les pixels des images (Jedynak et Fleuret 1996). Le protocole d'évaluation de ces méthodes consiste généralement à utiliser une pose sur deux par objet (donc à intervalles de 10 degrés, cfr. 2.3) pour l'ensemble d'apprentissage et l'autre moitié des images pour l'ensemble de test. Les taux d'erreur relevés dans la littérature varient de 6.53% à 0%. Nous obtenons également avec nos deux méthodes 0% d'erreur en utilisant ce protocole (P1). Malgré le fait que toutes les méthodes proposées dans la littérature ne sont pas excellentes, nous considérons ce problème trop simple.

Afin de rendre ce problème plus complexe, nous avons utilisé un autre protocole de test (P2) qui réalise 100 exécutions avec pour chacune d'elles un ensemble d'apprentissage composé de 36 poses tirées aléatoirement (et non à intervalles réguliers) pour chaque objet, l'autre moitié des images étant assignées à l'ensemble de test. Le tableau 5 résume nos résultats pour ce protocole. La moyenne du taux d'erreur avec la méthode d'ensemble d'arbres aléatoires est alors de 1.26%. Pour la technique avec sous-fenêtres, on obtient un taux d'erreur moyen de 0.51%.

Méthode	**Taux d'erreur P2**
Arbres aléatoires ($T = 500$)	$1.26\% \pm 0.60$
Extraction de fenêtres ($T = 10$, $w_1 = w_2 = 32$)	$0.51\% \pm 0.35$

TAB. 5 – Résultats sur la base de données COIL-20 selon le protocole complexe.

4.4 OUTEX Contrib_TC_0006

Cette base de données présente un petit nombre d'objets, un très grand nombre d'attributs (les images sont en couleurs) et de classes. L'environnement OUTEX spécifie les objets utilisés pour l'ensemble d'apprentissage et l'ensemble de test (8 images par classe par ensemble), nous avons donc adopté ce protocole puisqu'il est précis et public.

Dans (Mäenpää et al. 2002) sont évaluées plusieurs méthodes d'extraction de caractéristiques et de transformation des images de textures colorées en amont d'une méthode traditionnelle des plus proches voisins. Leurs résultats sur cet ensemble de données varient de 9.5% à 0.2% de taux d'erreur.

Le tableau 6 résume nos résultats. La méthode d'ensemble d'arbres extrêmement aléatoires donne un taux d'erreur relativement décevant de 64.35 %, ce que nous expliquons par le déséquilibre de la taille de l'échantillon. Par contre, la variante avec sous-fenêtres réduit le taux d'erreur jusqu'à 2.78%. Cette amélioration peut s'expliquer par la nature de l'application. En effet, étant donné qu'une texture est souvent basée sur la répétition de motifs, on peut extraire à partir des images originales des fenêtres très petites qui sont relativement bien classées par les modèles puisqu'elles contiennent intrinsèquement toute l'information suffisante sur la classe de l'image.

Méthode	Taux d'erreur
Arbres aléatoires ($T = 1000$)	64.35 %
Extraction de fenêtres ($T = 10$, $w_1 = w_2 = 4$)	2.78%
RGB histograms 3-D 32^3 raw + KNN (Mäenpää et al. 2002)	0.2%

TAB. 6 – Résultats sur la base de données OUTEX Contrib_TC_00006.

4.5 Discussion

4.5.1 Précision

Les résultats de nos expériences sont comparables à ceux d'autres approches voire meilleurs que bon nombre d'entre elles, mais ils sont néanmoins légèrement inférieurs en précision aux meilleurs résultats publiés dans la littérature. Toutefois, nous pensons pouvoir relativiser cette infériorité. En effet, sur la base de données ORL, étant donné la faible taille de l'échantillon, les résultats sont fortement instables en fonction du choix de l'ensemble d'apprentissage. Or, les résultats de la littérature reposent sur un nombre non précisé ou faible d'exécutions. La comparaison ne peut donc être qu'indicative. En ce qui concerne MNIST, nos deux méthodes sont clairement en deçà de certaines méthodes spécifiques ou d'approches basées sur les machines à support vectoriel. Dans le cas de la base de données OUTEX, seules des méthodes avec pré-traitement spécifique ont été évaluées dans la littérature, il serait intéressant de connaître les résultats d'approches utilisant les machines à support vectoriel. Enfin, rappelons que le but de notre méthode est d'être générique : jusqu'à quel point la réflexion et le temps nécessaires au pré-traitement traditionnel justifient-ils une amélioration de la précision par rapport à une méthode générique est une question laissée aux praticiens. Notons que, comme le grand nombre de publications pour la classification d'images le laisse penser, il n'existe pas de choix universel des caractéristiques à extraire pour un problème donné. De plus, le pré-traitement n'est généralement pas trivial et ne mène pas systématiquement à des résultats excellents. Ainsi, notre approche est meilleure que certaines méthodes spécifiques pour les problèmes que nous avons considérés.

4.5.2 Temps de calcul

En raison des choix aléatoires, la première méthode est très efficace et cela même si un grand nombre d'arbres doivent être construits. Par exemple, la construction de 500 arbres pour le protocole P1 de COIL-20 demande seulement 3s[7]. Sur la base de données MNIST, la construction de 50 arbres demande environ 11 minutes. De même, le test d'une nouvelle image est très rapide (moins d'une milliseconde quel que soit le problème).

Par contre, les temps de calcul de la variante utilisant les sous-fenêtres sont plus importants puisque la taille de l'échantillon d'apprentissage augmente. De même, le test est plus coûteux étant donné qu'il requiert la propagation de toutes les sous-fenêtres d'une image dans l'ensemble d'arbres. À titre indicatif, sur le problème COIL-

[7]Le code est écrit en C et tourne sur un Pentium 4 1.6GHz avec 512Mb de mémoire.

20, l'échantillon d'apprentissage passe de 720 à 120000 éléments et la construction d'un ensemble de 10 arbres sur les sous-fenêtres demande 2m30s. Pour classer une image, le test de ses 9409 fenêtres de taille 32×32 à l'aide de l'ensemble d'arbres demande environ une demi-seconde. De notre point de vue, ces valeurs restent néanmoins raisonnables. Aussi, nous pensons que le temps de test pourrait être réduit sans perte de précision en effectuant un ré-échantillonnage des sous-fenêtres.

5 Conclusions et perspectives

Dans cet article, nous avons évalué une approche générique pour la classification automatique d'images. Dans un premier temps, nous avons appliqué une méthode d'ensemble basée sur des arbres extrêmement aléatoires directement sur les valeurs basiques des pixels. Cette méthode donne des résultats acceptables tout en présentant des temps de calcul particulièrement attrayants. Dans un second temps, nous avons proposé une variante également générique qui consiste à extraire et classer des sous-fenêtres à partir des images originales. Cette technique améliore systématiquement la précision au prix de temps de calcul plus importants.

Sur les quatre bases de données, même si nos résultats sont comparables à ceux des méthodes actuelles les plus adroites, ils sont néanmoins légèrement inférieurs aux meilleurs résultats connus à ce jour. Afin de chercher à savoir si cette sous-optimalité est le prix à payer pour avoir une méthode générique, nous envisageons d'explorer d'autres variantes pour améliorer encore la précision.

6 Remerciements

Raphaël Marée est financé par un contrat First-Doctorat de la Région Wallonne en collaboration avec SA Automation & Robotics. Pierre Geurts est chargé de recherches au FNRS, Belgique.

Références

Breiman L., Friedman J.H., Olsen, R.A. et Stone, C.J. (1984), Classification and Regression Trees, Wadsworth International, 1984.

Burges C. J.C. et Schölkopf B. (1997), Improving the Accuracy and Speed of Support Vector Machines, Advances in Neural Information Processing Systems, Vol. 9, pp 375-382, The MIT Press, 1997.

Dahmen J., Keysers D. et Ney H. (2001), Combined Classification of Handwritten Digits Using the 'Virtual Test Sample Method', Proc. Second International Workshop on Multiple Classificers, pp 99-108, 2001.

Geurts P. (2002), Contributions to decision tree induction : bias/variance tradeoff and time series classification, PhD. Thesis, Department of Electrical Engineering and Computer Science, University of Liège, 2002.

Guo G., Li S. et Chan K. (2000), Face recognition by support vector machines, Proc. International Conference on Automatic Face and Gesture Recognition, pp 196-201, 2000.

Hoque M. et Fairhurst M.C. (2000), A Moving Window Classifier for off-line character recognition, Proc. Seventh International Workshop on Frontiers in Handwriting Recognition, pp 595-600, 2000.

Jedynak B. et Fleuret F. (1996), Reconnaissance d'objets 3D à l'aide d'arbres de classification, Conférence Internationale IMAGE'COM, France, 1996.

Lawrence S., Giles C.L., Tsoi A.C. et Back A.D. (1997), Face Recognition : A Convolutional Neural Network Approach, IEEE Transactions on Neural Networks, Vol. 8 (1), pp 98-113, 1997.

LeCun Y., Bottou L., Bengio Y. et Haffner P. (1998), Gradient-based learning applied to document recognition, Proc. of the IEEE, Vol. 86 (11), pp 2278-2324, 1998.

Mäenpää T., Pietikäinen M. et Viertola J. (2002), Separating color and pattern information for color texture discrimination, Proc. 16th International Conference on Pattern Recognition, 2002.

Murase H. et Nayar S.K. (1995), Visual Learning and Recognition of 3D Objects from Appearance, International Journal of Computer Vision, Vol. 14 (1), pp 5-24, 1995.

Nefian A. et Hayes M. (1999), Face recognition using an embedded HMM, Proc. IEEE Conference on Audio and Video-based Biometric Person Authentication, pp 19-24, 1999.

Ojala T., Mäenpää T., Pietikäinen M., Viertola J., Kyllönen J. et Huovinen S. (2002), OUTEX - New framework for empirical evaluation of texture analysis algorithms, Proc. 16th International Conference on Pattern Recognition, pp 701-706, 2002.

Paredes R. et Perez-Cortes A. (2001), Local representations and a direct voting scheme for face recognition, Pattern Recognition in Information Systems, Proc. 1st International Workshop on Pattern Recognition in Information Systems, pp 71-79, 2001.

Roli F. et Kittler J. (2002), Multiple Classifier Systems, Proc. Third International Workshop on Multiple Classifier Systems, 2002.

Wehenkel L. A. (1997), Automatic Learning Techniques in Power Systems, Kluwer Academic Publishers, 1997.

Summary

In this paper, we evaluate a generic and automatic approach for image classification. This approach is based on a novel automatic learning algorithm which builds ensembles of decision trees from pixel values only, by selecting decision tree tests at random. We further combine this algorithm with a generic technique which consists in extracting and classifying subwindows from the original images and hence artificially augments the learning sample size. These two approaches are validated and compared on four publicly available datasets corresponding to representative applications of image classification problems : handwritten digits (MNIST), faces (ORL), 3D objects (COIL-20), textures (OUTEX).