

Classification automatique d'images par arbres de décision

Raphaël Marée

Thèse de Doctorat en Informatique
Institut Montefiore, Université de Liège

8 février 2005

Plan de la thèse

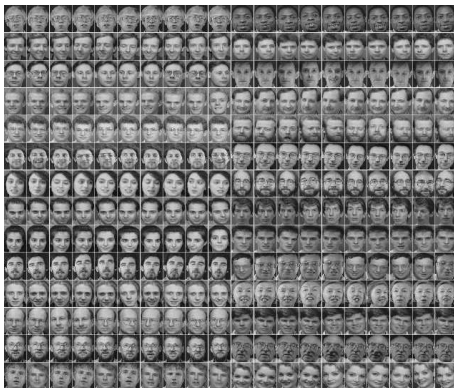
- Introduction
 - Classification d'images
 - Objectifs de la thèse
 - Apprentissage automatique supervisé
- Approches étudiées
 - Globale
 - Variantes locales
- Discussion
- Conclusion

Classification d'images

- **Contexte** : technologies d'acquisition d'images et de stockage engendrent de nombreuses bases de données d'images
- **But** : classer systématiquement des images selon des classes bien définies
 - **Exemples de classe** : objet, logo, cellule, type de situation
 - **Applications** : diagnostic médical, étude de phénomènes naturels, contrôle qualité, catalogue d'images, surveillance, ...

Exemples

- Classification de visages (*Etienne, Virginie, Thierry, ...*)



Exemples

- Classification de cellules (*stomatocyte, discocyte, échinocyte*)



- Classification de pollen (*Parietaria judaica, Urtica membranacea, Urtica ureas*)



- Classification de radiographies (*abdomen, membre, sein, crâne, poitrine, colonne vertébrale*)



Problèmes pour l'humain

- Temps requis
- Faculté de concentration
- Répétitivité
- Lenteur (nb. images/jour ?)
- *“Être prêt à travailler en horaire décalé. Accepter de faire un travail répétitif. Ne pas être daltonien. Être résistant au stress.”*

Classification *automatique* d'images : approche classique

- Constitution d'un échantillon d'images *étiquetées*, c-à-d à chaque image correspond une classe
- Mise au point d'une procédure informatique pour classer ces images

- Évaluation des performances sur échantillon indépendant de nouvelles images

Classification *automatique* d'images : approche classique

- Constitution d'un échantillon d'images *étiquetées*,
c-à-d à chaque image correspond une classe
- Mise au point d'une procédure informatique pour classer ces images
 - 1. Extraction de caractéristiques
 - 2. Description par un ensemble de valeurs numériques
(attributs)
- Évaluation des performances sur échantillon indépendant de nouvelles images

Classification *automatique* d'images : approche classique

- Constitution d'un échantillon d'images *étiquetées*, c-à-d à chaque image correspond une classe
- Mise au point d'une procédure informatique pour classer ces images
 - 1. Extraction de caractéristiques
 - 2. Description par un ensemble de valeurs numériques (attributs)
- Évaluation des performances sur échantillon indépendant de nouvelles images
 - 1. Extraction, 2. description
 - Classification de chaque image par comparaison de ses attributs avec ceux des images de l'échantillon étiqueté

Exemples

- Classification de caractères manuscrits (*0, 1, 2, ..., 9*)



- Classification de visages (*Etienne, Virginie, Thierry, ...*)



Exemples

- Classification de textures (*Métal, Briques, Fleurs, Grains, ...*)



Exemples

- Classification d'objets (*Flacon X, Bouteille Y, Fruit Z, ...*)



Inconvénients de l'approche classique

- Définition et extraction de caractéristiques
 - Caractéristiques idéales non connues a priori
 - Nouvelle réflexion, adaptation
 - Nouveau problème
 - Nouvelles classes
 - Variation des conditions (robustesse) : changements d'illumination, orientation, échelle, bruit, encombrement, occultations
 - Écarte potentiellement de l'information

Thèse

- Est-il possible de mettre au point une méthode de classification d'images autonome, générique et robuste par apprentissage automatique ?
- Pas de pré-traitement : application directe de méthodes d'apprentissage automatique à la représentation pixels
- Évaluation sur plusieurs problèmes et adaptation
 - Précision
 - Robustesse

Apprentissage automatique

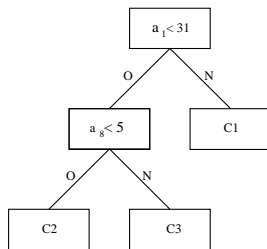
- Mise au point d'algorithmes pour l'exploitation d'observations issues d'un environnement, construction de procédures de décision, de modèles.
- **Applications** : médecine, industrie, finance, réseaux, ...
- **Apprentissage supervisé** : échantillon d'entrées-sorties, construction d'une règle (modèle) qui représente la relation entrées-sorties sous-jacente.

Apprentissage automatique supervisé

- Construction d'un modèle (arbre de décision, ...) à partir d'un échantillon d'observations étiquetées (*LS*)

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	Classe
60	19	18	17	0	1	1	1	C1
60	3	22	23	1	29	11	23	C1
75	9	2	1	3	77	46	3	C1
2	10	10	2	234	0	0	0	C2
3	7	9	18	5	0	0	0	C2
2	14	5	10	8	10	8	10	C3
65	3	20	21	2	0	1	1	?

⇒
 APPRENTISSAGE



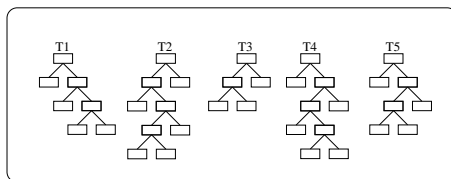
Classification par apprentissage automatique supervisé

- Évaluation du taux d'erreur du modèle sur un échantillon de test (TS) représentatif, indépendant du LS
- Précision, sources d'erreur :
 - Sous-apprentissage (biais) : modèle trop simple
 - Sur-apprentissage (variance) : modèle trop précis

Méthodes d'apprentissage supervisé

- K plus proches voisins
- Mono-arbre (CART) [BFOS84]
- Ensembles d'arbres
 - Bagging [Bre96]
 - Boosting [FRS96]
 - Random Forests [Bre01]
 - Extra-Trees [Geu02]
- Machines à vecteurs de support (SVM) [Vap95]

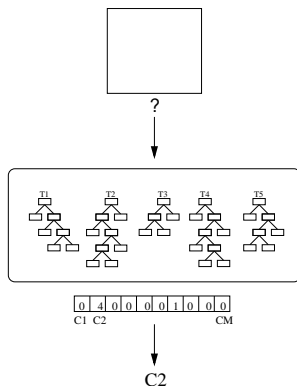
Extra-Trees : apprentissage



- Extra-Trees [Geu02]
 - Construction indépendante des arbres
 - Choix aléatoire des tests (attributs et seuils)
 - Très rapide
 - Arbre complet (classification parfaite du LS)

Extra-Trees : Prédiction

- Agrégation



Approche globale : plan

Approche globale de classification d'images

- Principe
- Évaluation
 - Bases de données
- Discussion

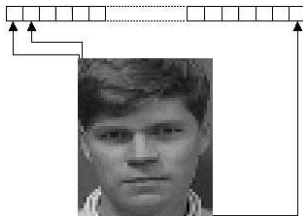
Principe

- Ensemble d'apprentissage de N images pré-classifiées,





$$LS = \{(\mathbf{a}^i, c^i), i = 1, \dots, N\}$$

\mathbf{a}^i : vecteur de valeurs de pixels de l'image

c^i : classe de l'image



Bases de données et protocoles

Bases de données	# images	# attributs	# classes
 MNIST	70000	784 (28 * 28 * 1)	10
 ORL	400	10304 (92 * 112 * 1)	40
 COIL-100	7200	3072 (32 * 32 * 3)	100
 OUTEX	864	49152 (128 * 128 * 3)	54

- MNIST : $LS = 60000$ premières images ; $TS = 10000$ dernières images
- ORL : Validation croisée stratifiée : 10 tirages aléatoires $LS = 360$; $TS = 40$
- COIL-100 : $LS = 1800$ images ; $TS = 5400$ images (36 autres images par objet)
- OUTEX : $LS = 432$ images (8 images par texture) ; $TS = 432$ images (8 images par texture)

Résultats : taux d'erreur MNIST



70000 images, 784 variables, 10 classes

Méthode	Taux d'erreur
Mono-Arbre	11.50%
K Plus proches voisins	5.66%
Bagging	4.42%
Extra-Trees	3.17%
Random Forests	3.00%
Boosting	2.29%
SVMs	1.95%
<i>IDM, 3-NN [KGN04]</i>	0.50%

Résultats : taux d'erreur ORL



400 images, 10304 variables, 40 classes

Méthode	Taux d'erreur
Mono-Arbre	29.25% \pm 6.89
Bagging	9.50% \pm 5.70
Boosting	3.75% \pm 2.79
K Plus proche voisins	2.25% \pm 2.36
Extra-Trees	1.25% \pm 1.68
Random Forests	1.25% \pm 1.68
SVMs	1.25% \pm 1.25

Résultats : taux d'erreur COIL-100



7200 images, 3072 variables, 100 classes

Méthode	Taux d'erreur
Mono-Arbre	20.80%
Bagging	2.24%
Extra-Trees	1.96%
K Plus proche voisins	1.94%
Random Forests	1.17%
Boosting	0.54%
SVMs	0.44%
<i>Local Affine Frames [OM02]</i>	0.10%

Résultats : taux d'erreur OUTEX



864 images, 49152 variables, 54 classes

Méthode	Taux d'erreur
Mono-Arbre	89.35%
K Plus proche voisins	80.79%
Bagging	73.15%
SVMs	71.99%
Boosting	69.44%
Random Forests	66.90%
Extra-Trees	65.05%
<i>RGB Histograms [MPV02]</i>	0.20%

Résultats : résumé taux d'erreur

- Random Forests, Extra-Trees, Boosting, SVM : bonnes performances pour 3 problèmes sur 4
- Précision insatisfaisante pour textures
- Faible robustesse

Résultats : temps d'exécution

Temps d'apprentissage

MNIST (60000, 28×28)

Méthode	Temps
Boosting	$\pm 6h$
Bagging	$\pm 5h$
SVMs	$\pm 28m$
Random Forests	$\pm 20m$
Extra-Trees	$\pm 12m$
Mono-arbre	$\pm 7m$

COIL-100 (1800, $32 \times 32 \times 3$)

Méthode	Temps
Boosting	$\pm 5h$
Bagging	$\pm 2h$
Random Forests	$\pm 52m$
Mono-arbre	$\pm 3m$
SVMs	$\pm 1m$
Extra-Trees	$\pm 9s$

Temps de prédiction (par image)

MNIST

Méthode	Temps
SVM	49msec
Extra-Trees	0.6msec

COIL-100

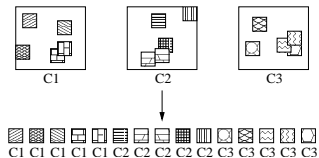
Méthode	Temps
SVM	37msec
Extra-Trees	4msec

Approche locale : plan

Approche locale par extraction aléatoire de fenêtres

- Version de base
 - Principes
 - Évaluation : précision, robustesse
 - Discussion
- Variantes robustes

Approche locale : phase d'apprentissage



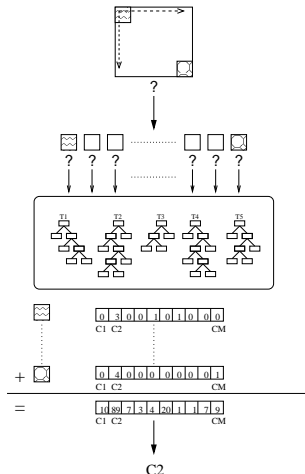
Ensemble d'apprentissage de N_w fenêtres $w \times w$ pré-classifiées,

$$LS = \{(\mathbf{a}^i, c^i), i = 1, \dots, N_w\}$$

\mathbf{a}^i : vecteur de valeurs de pixels de la fenêtre

c^i : classe de l'image dont est issue la fenêtre

Approche locale : prédiction



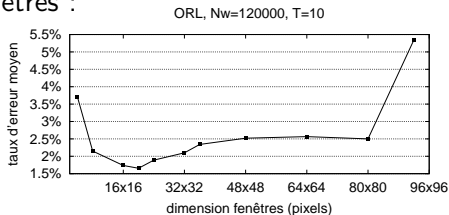
Résultats : taux d'erreur

BDs	Extra-Trees	Extra-Trees et fenêtres	État de l'art
MNIST	3.26%	2.63%	0.5% [DKN04]
ORL	4.56% ± 1.43	1.66% ± 1.08	2% [Rav04]
COIL-100	1.96%	0.37%	0.1% [OM02]
OUTEX	65.05%	2.78%	0.2% [MPV02]

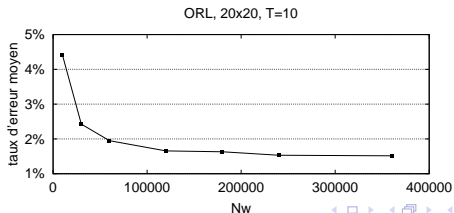


Influence des paramètres : illustration sur ORL

- Taille des fenêtres :



- Nombre de fenêtres d'apprentissage :



Résultats : temps d'exécution

- **Apprentissage** : dépend des paramètres
MNIST : 6h, ORL : 37s, COIL-100 : 1h, OUTEX : 11m
- **Prédiction** : dépend des paramètres et du parcours
 - *Parcours exhaustif*



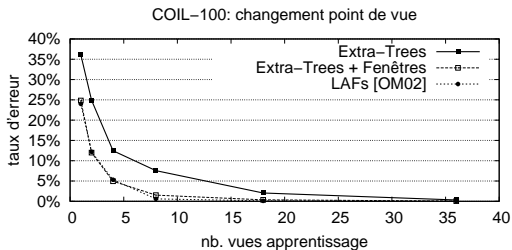
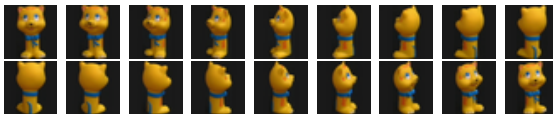
MNIST : 2msec, ORL : 354msec
COIL-100 : 14msec, OUTEX : 800msec

- *Parcours aléatoire*



MNIST : 1msec, ORL : 10msec
COIL-100 : 5msec, OUTEX : 33msec

Robustesse : changement de point de vue



Robustesse : orientation, occultations, échelle

- **Changements d'orientation 2D**

Rotation de 20° : Globale : $> 20\%$ d'erreurs

Locale : $< 5\%$ d'erreurs

Mais $> 30\%$ d'erreurs quand rotation de 45°

- **Occultations**

Occultation de 40% : Globale : $> 15\%$ d'erreurs

Locale : $< 10\%$ d'erreurs

- **Changements d'échelle**



7% d'erreurs quand facteur 1.5

Discussion approche locale de base

Approche locale par extraction aléatoire de fenêtres

- Amélioration de la précision
- Amélioration de la robustesse
- Paramètres
 - Taille des fenêtres : effet non monotone
 - Nombre de fenêtres : effet monotone
 - Nombre d'arbres \ll approche globale

Approche locale : plan variantes robustes

Approche locale par extraction et transformation aléatoires de fenêtres

- Variantes robustes
 - Motivations
 - Principes
 - Évaluation
 - Discussion

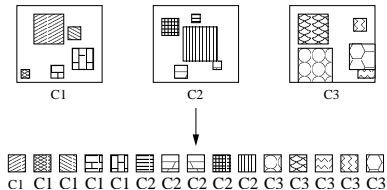
Motivations

En situations non contrôlées, échelle et orientation non connues a priori

- Méthodes locales de la littérature :
 - Extraction de régions, de points d'intérêts
 - Normalisation, description invariante
- Notre approche :
 - Apprentissage à partir de fenêtres transformées

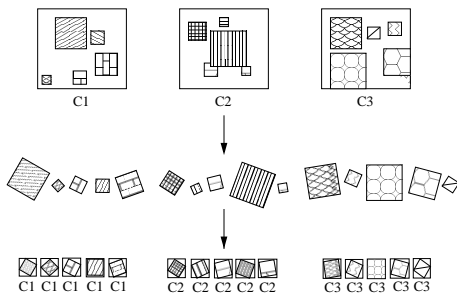
Variante 1 : fenêtres de tailles aléatoires

- Extraction de fenêtres de tailles aléatoires
- Transformation en une taille unique



Variante 2 : fenêtres de tailles et d'orientations aléatoires

- Extraction de fenêtres de tailles aléatoires
- Transformation aléatoire de l'orientation
- Transformation en une taille unique



Robustesse : échelle, orientation

- **Changement d'échelle**

Changement d'échelle	Version de base	Variante 1
16 × 16	71.72%	0.44%
32 × 32	0.37%	0.50%
48 × 48	7.15%	0.56%
128 × 128	-	0.44%

- **Changement d'orientation**

Changement d'orientation	Version de base	Variante 2
0°	0.37%	3.72%
30°	8.85%	-
45°	31.69%	5.87%

Variante 3 : représentation HSV

- Représentation HSV des pixels au lieu de RGB
 - Précision : approche globale et locale de base

	OUTEX		COIL-100	
	RGB	HSV	RGB	HSV
Globale	64.35%	41.20%	1.96%	1.59%
Locale	2.78%	1.16%	0.37%	0.26%

- Robustesse : variante locale 2

	COIL-100	
Changement d'orientation	RGB	HSV
0°	3.72%	0.87%
45°	5.87%	1.56%

Résumé résultats variantes

- Représentation HSV
 - Amélioration générale de la précision et de la robustesse par rapport à RGB
- Fenêtres tailles aléatoires et orientation fixe
 - Changement d'échelle : robustesse
 - Changement de point de vue : amélioration
 - Changement d'orientation : pas d'amélioration
- Fenêtres tailles et orientations aléatoires
 - Changement d'échelle : robustesse
 - Changement de point de vue : amélioration
 - Changement d'orientation : robustesse

Méthodes locales : en résumé

- Précision
 - Paramètres pour compromis précision/efficacité
 - Nombre de fenêtres (apprentissage, test)
 - Nombre d'arbres
- Robustesse
 - Paramètres pour compromis précision/robustesse
 - Variantes 1,2
- Extra-Trees
 - Scalable

Discussion : plan

- Étude empirique
 - Autres bases de données
 - Résumé
- Pourquoi ça marche ?
- Problèmes non investigués
- Application industrielle de l'approche

Autres applications : ETH-80

Classification de 8 classes d'objets (*Pomme, Poire, Tomate, ...*)



Autres applications : ZuBuD

Classification de 201 bâtiments de Zürich (*Bât 1, Bât 2, ...*)



Autres applications : WANG

Classification de 10 thèmes de photographies (*Plage, bus, éléphant, cheval, ...*)



Résumé des bases de données

- Bases de données largement utilisées, protocoles précis
 - **Caractéristiques** : de 28×28 à 640×480 pixels, de 10 à 201 classes, de 400 à 70000 images
 - **Types** : chiffres manuscrits, visages, objets, textures, immeubles, thèmes
 - **Variations** : changements de point de vue, d'illumination, d'échelle, d'orientation, encombrements et occultations

Pourquoi ça marche ?

- Construction et agrégation d'arbres
 - Amélioration de la précision
 - Réduction de la variance en présence d'un grand nombre de variables (pixels)
- Extraction et agrégation de fenêtres
 - Dissocie les attributs des positions fixes dans les images
 - Fenêtres transformées : augmentation de la variabilité du LS
 - Tolérance aux occultations et aux encombrements
- HSV
 - Limite les effets des changements d'illumination

Application industrielle de l'approche

- Projet européen pour la classification de défauts sur des réflecteurs de phares



- Exemples de défauts :



- Approche : utilisation des Extra-Trees à partir de la représentation pixels combinée avec des caractéristiques qui décrivent le voisinage

Conclusion

- Contributions
- Perspectives

Contributions

- Nouvelle approche de classification d'images
 - Basée sur apprentissage automatique supervisé
 - Extraction aléatoire de fenêtres
 - Transformations de fenêtres
 - Représentation pixels
- Étude empirique sur 7 problèmes divers
 - Précision, robustesse

Perspectives

- Évaluation
 - Autres méthodes d'apprentissage : SVM, Boosting
 - Robustesse
 - Autres transformations de fenêtres
 - Autres techniques d'extraction de régions invariantes
 - Autres bases de données : dégradations accrues
- Autres types de données
 - Textes
 - Séquences génétiques



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



L. Breiman.
Bagging predictors.
Machine Learning, 24(2) :123–140, 1996.



L. Breiman.
Random forests.
Machine learning, 45 :5–32, 2001.



T. Deselaers, D. Keysers, and H. Ney.
Classification error rate for quantitative evaluation of content-based image retrieval systems.
In *Proc. 17th International Conference on Pattern Recognition (ICPR)*, 2004.



Y. Freund and E. Robert Schapire.
Experiments with a new boosting algorithm.
In *Proc. Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.



P. Geurts.
Contributions to decision tree induction : bias/variance tradeoff and time series classification.
Phd. thesis, Department of Electrical Engineering and Computer Science, University of Liège, May 2002.



D. Keysers, C. Gollan, and H. Ney.
Local context in non-linear deformation models for handwritten character recognition.
In *Proc. International Conference on Pattern Recognition*, 2004.



T. Mäenpää, M. Pietikäinen, and J. Viertola.
Separating color and pattern information for color texture discrimination.

In *Proc. 16th International Conference on Pattern Recognition*, 2002.



S. Obdržálek and J. Matas.

Object recognition using local affine frames on distinguished regions.

In *Electronic Proceedings of the 13th British Machine Vision Conference, University of Cardiff*, 2002.



S. Ravela.

Shaping receptive fields for affine invariance.

In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR2004)*, June 2004.



V. Vapnik.

The nature of statistical learning theory.

Springer Verlag, 1995.