

# Classification et sélection automatique de caractéristiques de textures

Marine Campedel\*, Eric Moulines\*\*

Ecole Nationale Supérieure des Télécommunications  
Laboratoire de Traitement du Signal et des Images  
46, rue Barrault, 75013 Paris

\* Marine.Campedel@enst.fr,  
<http://www.tsi.enst.fr/~campedel>

\*\* Eric.Moulines@enst.fr  
<http://www.tsi.enst.fr/~moulines>

Cette étude est financée par le Centre National d'Etude Spatiale (CNES).

**Résumé.** Choisir a priori les caractéristiques pertinentes pour une application donnée n'est pas aisé. En particulier, les outils de classification d'images utilisent des modèles variés pour représenter les textures. Nous proposons de choisir les modèles de texture les plus pertinents à l'aide d'une procédure automatique de sélection de caractéristiques. Nous comparons pour cela l'efficacité de plusieurs algorithmes récents en évaluant les performances de différents classificateurs exploitant ces sélections. Nous démontrons l'intérêt d'une telle procédure de sélection à partir des images de Brodatz.

## 1 Introduction

Face à l'accroissement rapide des tailles des bases de données, en particulier des bases d'images, il est nécessaire de développer de nouveaux algorithmes de traitement facilitant à la fois le stockage et l'indexation de ces données. Nous nous intéressons dans ce travail aux algorithmes de sélection de caractéristiques (appelées aussi descripteurs), qui permettent d'extraire une information non redondante et pertinente, en vue d'une exploitation efficace des bases de données. Ces algorithmes font l'objet d'une littérature abondante depuis une dizaine d'années [Kohavi et John, 1997, Blum et Langley, 1997, Guyon et Elisseeff, 2003] .

Les algorithmes de sélection sont répartis en deux groupes principaux : les 'filters' et les 'wrappers'. Les premiers exploitent les propriétés intrinsèques des caractéristiques utilisées, sans référence à une quelconque application. Les seconds, au contraire, définissent la pertinence des caractéristiques par l'intermédiaire d'une prédiction de la performance du système final. Les premiers sont généralement moins coûteux mais aussi moins efficaces que les seconds.

Les performances des algorithmes dépendent de l'évaluation de la pertinence des descripteurs mais aussi de la stratégie de recherche du sous-ensemble de caractéristiques. Cette recherche ne peut pas être exhaustive (il faudrait explorer  $2^D$  combinaisons

pour un ensemble initial de  $D$  caractéristiques), différentes stratégies d'exploration de l'espace des caractéristiques sont donc présentées dans la littérature, en particulier reposant sur l'usage d'algorithmes génétiques [Kim *et al.*, 2002] ou de stratégies incrémentales (forward) et/ou décrémentationales (backward) [Kohavi et John, 1997]. Il s'agit de progressivement ajouter (ou respectivement supprimer) une caractéristique en comparant le bénéfice (resp. la perte) apporté par l'utilisation de cette caractéristique. Nous nous plaçons ici dans le cadre supervisé. Le principe de la sélection est donc de sélectionner le sous-ensemble de caractéristiques permettant de discriminer au mieux les différentes classes de données. Dans le cas 'filter', les algorithmes calculent un poids pour chaque caractéristique, significatif de cette capacité à rassembler les éléments d'une même classe et séparer les éléments étiquetés différemment. Nous avons choisi de tester l'algorithme ReliefF, initialement présenté par [Kira et Rendell, 1992, Robnik-Sikonja et Kononenko, 2003]. Dans le cas 'wrapper', la sélection est directement reliée à la performance d'un classificateur. Nous avons choisi trois algorithmes différents: Fisher (appelé aussi LDA ou analyse linéaire discriminante), RFE [Guyon, 2002, Guyon et Elisseeff, 2003] et L0 [Weston *et al.*, 2003]. Ces trois algorithmes calculent la pertinence de chaque caractéristique à l'aide des poids estimés par un classificateur linéaire (Fisher ou SVM). Nous nous intéressons aux machines à vecteurs de support (SVM) car elles limitent le risque de surapprentissage du fait de leur capacité de régularisation (ce risque étant particulièrement important lorsque le nombre de caractéristiques, i.e. la dimension, est grand face à au nombre de données).

Les quatre algorithmes étudiés reposent donc sur l'estimation de poids (scores) correspondant à chaque caractéristique. Ces poids sont utilisés pour ordonner puis sélectionner les  $K$  (parmi  $D$ ) descripteurs les plus pertinents. Seule la méthode L0 permet une estimation du nombre minimal  $K$  de caractéristiques nécessaires.

Notre objectif est de comparer les performances de ces différents algorithmes à l'aide de performances de classification (KNN, Fisher, SVM).

Le problème de choisir les bons descripteurs pour la classification d'images est un problème récurrent dans la littérature [Sebe et Lew, 2000, Rui *et al.*, 2001]. Nous proposons donc d'y répondre à l'aide de techniques de sélection automatiques. Nous appliquons nos différentes procédures de sélection sur des images de textures issues de la base Brodatz, afin de déterminer les caractéristiques les plus discriminantes, parmi un ensemble calculé sur des matrices de co-occurrence (coefficients dits d'Haralick [Haralick *et al.*, 1973]), des filtres de Gabor et diverses ondelettes.

L'ensemble de nos simulations s'effectue à l'aide de l'outil Matlab Spider développé par Elisseeff et Weston [Weston *et al.*, 2004]. Les textures d'images sont calculées à partir des implantations de Boland [Boland, 1998], de la librairie d'ondelettes de Pyr [Simoncelli, 2001] et des 'contourlets' de [Do et Vetterli, 2003].

## 2 Algorithmes de sélection

Les algorithmes supervisés s'appliquent sur des bases de données étiquetées. Soient les données  $x_i, i = 1, \dots, N$  et les étiquettes associées  $y_i$ . Nous ne traitons que des

étiquettes discrètes. Les données sont numériques et multivaluées sur un espace initial de dimension  $D$  ( $x_i \in R^D$ ).

Le principe de l'algorithme de sélection supervisé est d'extraire un sous-ensemble de  $K$  caractéristiques dans l'ensemble de taille  $D$  initial, permettant de représenter au mieux les données étiquetées, donc en particulier permettant de classifier au mieux ces données. Les algorithmes que nous avons étudiés associent tous un score de pertinence à chaque type de caractéristique, ce qui permet de les ordonner et de sélectionner les  $K$  meilleurs.

## 2.1 ReliefF

Cet algorithme, introduit sous le nom de Relief dans [Kira et Rendell, 1992] puis amélioré et adapté au cas multi-classes par Kononenko sous le nom de ReliefF, ne se contente pas d'éliminer la redondance mais définit un critère de pertinence. Ce critère mesure la capacité de chaque caractéristique à regrouper les données de même étiquette et discriminer celles ayant des étiquettes différentes. L'algorithme est décrit ci-dessous. L'analyse de ReliefF est effectuée dans [Robnik-Sikonja et Kononenko, 2003].

- Initialisation des poids (scores) :  $w_d = 0, d = 1, \dots, D$
- Pour  $i = 1 : m$  (m limité à  $N$ )
  - tirer aléatoirement une donnée de  $x_i$ ,
  - trouver les  $k$  plus proches voisins de  $x_i$  ayant la même étiquette (*hits*),
  - trouver les  $k$  plus proches voisins de  $x_i$  ayant une étiquette différente, dans chaque classe  $c$  (*misses<sub>c</sub>*),
  - pour chaque caractéristique  $d$ , mettre à jour le poids selon la formule ci-dessous.

$$w_d = w_d - \sum_{j=1}^k \frac{\text{diff}(x_i, d, \text{hits}_j)}{mK} \quad (1)$$

$$+ \sum_{c \neq \text{class}(x_i)} \frac{P(c)}{1 - P(\text{class}(x_i))} \sum_{j=1}^k \frac{\text{diff}(x_i, d, \text{misses}_{cj})}{mK} \quad (2)$$

La distance utilisée est définie par :

$$\text{diff}(x_i, d, x_j) = \frac{|x_{id} - x_{jd}|}{\max(d) - \min(d)} \quad (3)$$

$\max(d)$  ( resp.  $\min(d)$  ) désigne la valeur maximale (resp. minimale) que peut prendre la caractéristique désignée par l'index  $d$ , sur l'ensemble des données.  $x_{id}$  est la valeur de la  $d$ ème caractéristique de la donnée  $x_i$ .

Le poids d'une caractéristique est d'autant plus grand que les données issues de la même classe ont des valeurs proches et que les données issues de classes différentes sont bien séparées. Cet algorithme est rapide, sa complexité est en  $O(N^2 D)$ . Il n'y a pas de recherche de sous-espace, ReliefF produit un score permettant d'ordonner les caractéristiques selon le critère de pertinence défini ci-dessus.

## 2.2 Fisher

Le deuxième algorithme choisi repose sur l'analyse discriminante linéaire (LDA) de Fisher. De façon générale, un problème de discrimination linéaire, à deux classes,

revient à séparer l'espace des données en deux grâce à un hyperplan. Choisir la classe (à valeur dans  $-1,1$ ) d'une donnée consiste alors à déterminer de quel côté de l'hyperplan elle se situe, ce qui se traduit sous la forme :

$$g(x_i) = \text{sign}(w.x_i + b) \quad (4)$$

$x_i \in R^D, i = 1 \dots N, w \in R^D, b \in R$  est appelé le biais.

L'objectif de la LDA est de trouver une projection des données sur une droite, qui sépare au mieux les données, i.e. en maximisant le rapport des variances inter et intra classes des projections. Lorsque l'on fait l'hypothèse que les caractéristiques suivent, sur chaque classe (+ et -), une distribution gaussienne  $N(\mu^{(+)}, \sigma^{(+)})$  et  $N(\mu^{(-)}, \sigma^{(-)})$ , et qu'elles sont décorréélées, il vient :

$$w_d = \frac{(\mu_d^{(+)} - \mu_d^{(-)})^2}{\sigma_d^{(+)^2} + \sigma_d^{(-)^2}} \quad (5)$$

Le formule peut être étendue au cas multiclasse en considérant qu'il s'agit d'un ensemble de problèmes à deux classes du type 1 contre tous. Nous utilisons l'implantation de l'algorithme fournie par Spider [Weston *et al.*, 2004].

## 2.3 SVM-RFE

Cet algorithme de sélection est présenté dans [Guyon, 2002]. Il repose lui-aussi sur l'estimation de poids relatifs à l'optimisation d'un problème de discrimination linéaire, ce problème étant résolu à l'aide d'une machine à vecteurs de support (SVM).

### 2.3.1 Principe des SVM

Une présentation claire et détaillée des SVM peut être obtenue notamment dans [Cortes et Vapnik, 1995, Schoelkopf et Smola, 2002].

Si un ensemble de données d'apprentissage est linéairement séparable, alors le classificateur SVM maximise la marge de séparation, ce qui revient à résoudre le problème suivant :

$$\min_{w,b} \|w\|_2^2 \quad (6)$$

avec la contrainte

$$y_i(w.x_i + b) \geq 1$$

Les vecteurs supports sont les données  $x_i$  qui vérifient :

$$y_i(w.x_i + b) = 1$$

Sous sa forme duale, le problème apparaît comme quadratique:

$$\max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j x_i . x_j \quad (7)$$

sous les contraintes

$$\sum_{i=1}^N \alpha_i y_i = 0, \alpha_i > 0, i = 1, \dots, N$$

On a la relation :

$$w_d = \sum_i \alpha_i y_i x_{id}$$

Les  $\alpha_i$  sont tous nuls exceptés ceux correspondant aux vecteurs supports.

La robustesse des SVMs provient de l'ajout d'une contrainte de régularisation, sous la forme :

$$\min_{w,b} ||w||_2^2 + C \sum_{i=1}^N \epsilon_i \quad (8)$$

sous les contraintes

$$y_i(w \cdot x_i + b) \geq 1 - \epsilon_i, \epsilon_i > 0, i = 1, \dots, N$$

Nous utilisons dans notre étude l'implantation des SVMs appelée libsvm version 2.5 [Chang et Lin, 2004] (C=1000). La plupart des problèmes ne se contentent pas de deux classes de données. Nous avons choisi de considérer les problèmes multi-classes comme une accumulation de problèmes à deux classes (une contre toutes les autres). Il existe d'autres versions multiclassées de SVM dont [Elisseff, 2000], mais nous ne les avons pas utilisées dans cette étude. Dans notre cas, les poids finalement associés aux caractéristiques sont moyennés sur l'ensemble des problèmes 1-contre-tous.

### 2.3.2 Récursion

Il est montré dans [Guyon, 2002, Rakotomamonjy, 2003] que le coût de suppression d'une caractéristique est de l'ordre de  $w_d^2$ . La procédure de sélection est décrémente et élimine donc progressivement les caractéristiques de faible poids. L'algorithme est décomposé en trois étapes :

Tant qu'il reste des caractéristiques,

- Entraîner le classificateur SVM (linéaire) ;
- Calcul des  $w_i^2$  ;
- Supprimer la (les) caractéristique(s) correspondant au(x) poids le(s) plus faible(s).

Nous utilisons l'implantation faite dans l'outil Spider. La procédure est grandement accélérée lorsque plusieurs caractéristiques sont éliminées simultanément et lorsque l'on stoppe la boucle d'élimination dès obtention du nombre désiré de caractéristiques.

## 2.4 Minimisation de L0 (l0)

L'algorithme L0 présenté dans [Weston *et al.*, 2003], utilise lui aussi les poids estimés par un classificateur SVM. L'idée générale est cependant très différente de SVM-RFE, puisqu'il s'agit dans ce cas de favoriser la mise à zero du plus grand nombre de

## Sélection de modèle de textures

poids. Les auteurs proposent donc de minimiser la norme L0 des poids, i.e. de trouver l'ensemble minimal de caractéristiques ayant un poids non nul.

$$\min_w \|w\|_0^0 = \min_w \text{card}(w_i | w_i \neq 0) \quad (9)$$

Il est démontré dans [Weston *et al.*, 2003] que la minimisation de la norme L0 est approchée par :

$$\min_w \sum_{j=1}^D \ln(\epsilon + |w_j|)$$

Ce problème est résolu par une procédure itérative (convergeant vers un minimum local) :

- Initialisation des poids  $z_i = 1, i = 1, \dots, D$
- Résoudre le problème :

$$\min_w \sum_{j=1}^D |w_j|$$

avec la contrainte  $y_i(w.(x_i * z) + b) \geq 1$

- Mise à jour des poids  $z = z * \bar{w}$ ,  $\bar{w}$  étant la solution du problème précédent.

La notation  $*$  correspond au vecteur des produits terme à terme :

$$t * w = (t_1 w_1, \dots, t_D w_D)$$

Finalement, les auteurs mettent en évidence l'efficacité de l'algorithme, lorsque l'on remplace la norme L1 des poids par la norme L2, ce qui permet d'utiliser une SVM pour résoudre le problème intermédiaire.

Nous utilisons l'algorithme développé par les auteurs, dans l'environnement Spider.

## 3 Critères d'évaluation

Afin de comparer les résultats des quatre algorithmes présentés ci-dessus, nous avons choisi d'évaluer les performances en terme d'erreur de classification à l'aide de trois algorithmes classiques (Knn, Fisher, SVM). En pratique, nous effectuons une validation croisée : les quatre cinquièmes des données sont utilisés pour la sélection des caractéristiques et l'apprentissage des classificateurs, la partie restante étant utilisée pour l'évaluation. La performance est mesurée par l'erreur de classification moyennée sur les cinq ensembles (disjoints) de test.

Etant donnés les algorithmes de sélections utilisés, nous considérons les trois classificateurs dont ils semblent les plus proches :

- K-voisins ('knn') : le nombre de voisins utilisé est fixé à la partie entière de la racine carrée du nombre de données d'apprentissage par classe (12 dans le cas synthétique et 4 dans le cas des images de Brodatz).
- Classificateur de Fisher ('fisher')
- SVM : nous utilisons le même modèle que pour les sélections SVM-RFE et 10 (libsvm 2.5, noyau linéaire, C=1000).

## 4 Simulations

Les premières simulations concernent un problème purement synthétique, mettant en évidence les avantages et inconvénients des différentes procédures de sélection. Les suivantes appliquent deux procédures de sélection rapides à la sélection automatique de modèles de texture.

### 4.1 Comparaison des performances de sélection sur une base synthétique

Nous testons tout d'abord notre procédure d'évaluation sur un problème synthétique. Les données sont au nombre de 400, exprimées sur 50 caractéristiques, tirées uniformément sur  $[0, 1]$ . Seules les deux premières caractéristiques permettent une séparation linéaire parfaite des données. Les dimensions 3 et 4 sont des versions faiblement bruitées des deux premières. Les dimensions 5 à 8 sont des versions bruitées des quatre premières. Toutes les autres dimensions sont du bruit. Les évaluations produisent les résultats de la figure 1.

Différentes remarques peuvent être faites :

- La performance de classification obtenue par le classificateur SVM est toujours meilleure, quel que soit l'algorithme de sélection utilisé en amont. Ceci est dû à sa capacité intrinsèque à gérer les éléments bruités.
- Les sélections de deux caractéristiques parmi les cinquante ne sont pas correctes dans le cas de Relieff et Fisher. Tous les indicateurs mettent en évidence la supériorité de svm-rfe et l0, qui comme l'indique l'erreur quasi nulle du classificateur SVM, sélectionnent les deux caractéristiques pertinentes ou bien leurs versions faiblement bruitées.
- Lorsque 8 caractéristiques sont sélectionnées, les indicateurs réagissent de façon différente aux sélections. Toutes les sélections contiennent les deux dimensions pertinentes, ce qui permet de comprendre pourquoi le classificateur SVM obtient la même performance quelle que soit la sélection. Par contre, il semble que svm-rfe et l0 préfèrent sélectionner des caractéristiques de bruit plutôt que des caractéristiques redondantes, ce qui altère les performances de knn et fisher.
- Pour 25 caractéristiques sélectionnées, les performances sont équivalentes quelle que soit la procédure de sélection. Il est intéressant de constater que la présence de caractéristiques bruitées altère les performances de classification, exceptée celle de fisher, qui pondère très faiblement ces composantes.
- Les algorithmes svm-rfe et l0 ne produisent pas des évaluations significativement différentes excepté le cas de 8 sélections. Cependant l0 est un algorithme beaucoup plus lent qui, puisqu'il pondère itérativement les données, peut ne plus converger (nous l'avons limité à quatre itérations).

## Sélection de modèle de textures

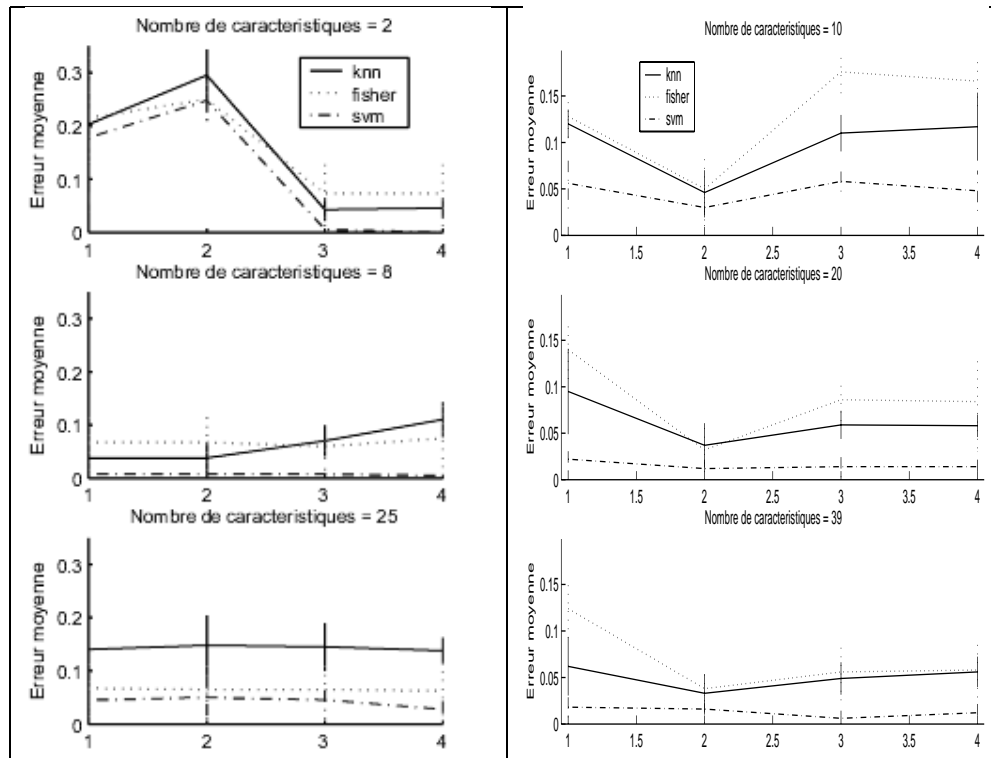


FIG. 1 – Valeurs moyennes et écarts types (barres verticales) obtenus sur 5 tirages. A gauche, le problème est synthétique, les méthodes de sélection sont désignées par un index correspondant à : 1=relieff 2=fisher 3=svm-rfe 4=l0. A droite, sélection sur les images de Brodatz.



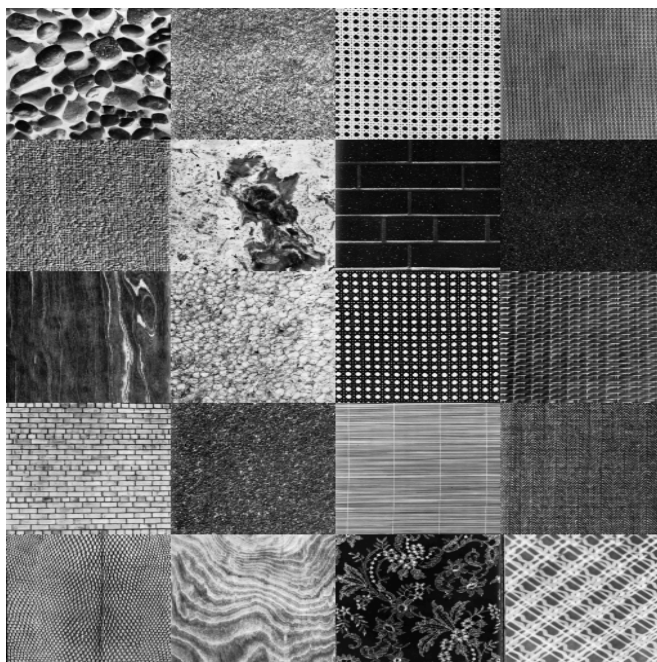


FIG. 2 – Sélection aléatoire de 20 images de la base de textures de Brodatz.

## 4.2 Application au traitement d'images de textures (Brodatz)

L'objectif de cette simulation n'est pas de rendre un verdict définitif sur le choix des modèles de textures. Il s'agit de décrire une méthodologie de sélection automatique. Bien souvent les experts ont des préférences différentes quant au modèle de texture à utiliser. Estimer simultanément tous ces modèles introduit une redondance préjudiciable au niveau du stockage des caractéristiques comme au niveau de la performance de classification. Nous mettons donc en évidence l'intérêt d'une sélection automatique des caractéristiques, parmi un ensemble présélectionné.

Comme base de travail nous avons choisi la base des images de texture de Brodatz. Nous avons tiré aléatoirement 20 images parmi les 111 dont nous disposons. Ces images, présentées figure 2, ont une taille initiale de 640x640. Nous les avons chacune décomposées en 25 imagerie disjointes de taille 128x128. Chaque imagerie est ensuite décrite par un vecteur de caractéristiques, résultant de la concaténation de plusieurs vecteurs de textures.

La littérature fait état de nombreux travaux sur les textures, nous avons sélectionné les coefficients d'Haralick, les Gabors et autres ondelettes, l'ensemble est résumé dans le tableau 4.2. Dans le cas de procédure de filtrage ou d'ondelettes, nous prenons arbitrairement la moyenne et la variance des sorties de chaque sous-bande. Les caractéristiques sont normalisées (moyenne nulle et variance unité) sur l'ensemble des 111 images.

## Sélection de modèle de textures

Modèle	Nb	Détails
Haralick	78	[Haralick <i>et al.</i> , 1973] Matrice de co-occurrence. 13 premiers coefficients obtenus pour quatre orientations ( $0 \frac{\pi}{4}$ , $\frac{\pi}{2}$ et $\frac{3\pi}{4}$ ) et distance $d=1$ + moyennes et variances correspondantes.
Gabor	24	Filtrage sur 3 échelles de 0.1 à 0.5 et 4 orientations
Steerable wavelet	48	[Simoncelli, 2001]
Contourlet	56	[Do et Vetterli, 2003]
Qmf	26	4 échelles

TAB. 1 – Description des différentes caractéristiques calculées pour chaque imagerie

	knn(k=4)		fisher		svm	
	m	$\sigma$	m	$\sigma$	m	$\sigma$
tout	0.065	0.019	0.040	0.021	0.024	0.018
Haralick	0.045	0.015	0.046	0.023	0.016	0.017
Gabor	0.074	0.017	0.064	0.022	0.064	0.040
Steer	0.286	0.051	0.166	0.034	0.198	0.041
Contourlet	0.235	0.026	0.048	0.026	0.092	0.027
qmf	0.206	0.054	0.050	0.022	0.138	0.035

TAB. 2 – Performance de classification sur chaque modèle

Nous pouvons tout d'abord comparer les performances de classification sur chaque modèle de texture. Le tableau 2 résume les moyennes et écart-types ( $m$ ,  $\sigma$ ) obtenus en validation croisée. Nous constatons que, parmi les caractéristiques que nous avons initialement choisies<sup>1</sup>, les coefficients d'Haralick permettent d'obtenir la meilleure performance, quel que soit le classificateur utilisé. De plus cette performance est meilleure que celle obtenue avec l'ensemble des descripteurs.

Nous appliquons maintenant les procédures de sélection aux 78 caractéristiques d'Haralick. Les résultats sont présentés figure 1, pour des ensembles de caractéristiques de taille 10, 20 et 39. Nous constatons que la sélection de 20 dimensions sur les 78 permet d'obtenir des performances de classification équivalentes. Par contre, une sélection de 10 dégrade les performances de classification.

Concernant la sélection de 20, nous constatons par ailleurs que fisher sélectionne à 90% les mêmes caractéristiques d'un ensemble d'apprentissage sur l'autre, alors que svm-rfe n'en conserve que 65%. La figure 4.2 présente la répartition des caractéristiques sélectionnées sur les 13 types de coefficients, pour chacun des cinq tirages de la validation croisée. On constate que relief, svm-rfe et l0 sont beaucoup plus sélectives sur le

1. Des paramètres mieux choisis pour les autres modèles de texture, en particulier les filtres de Gabor, auraient sans doute pu permettre d'égaliser ces performances

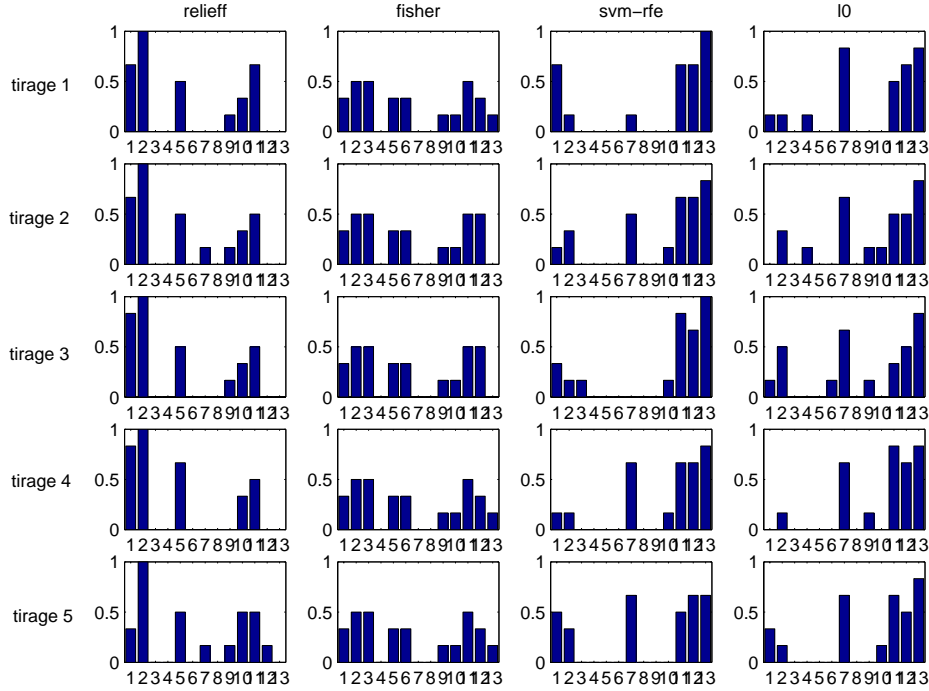


FIG. 3 – Répartition des sélections de 20 selon les 13 types de coefficients

type des données et tendent à échanger des caractéristiques de même catégorie, d'un tirage sur l'autre. Sachant de par l'étude du cas synthétique que svm-rfe et l0 préfèrent éliminer les caractéristiques redondantes, contrairement à relieff et fisher qui préfèrent éliminer les composantes non pertinentes, nous pouvons essayer d'identifier les unes et les autres. Par exemple, la caractéristique de type 13 est fortement sélectionnée par svm-rfe et l0 et très peu par fisher voire pas du tout par relieff; il s'agit probablement d'une composante de type "bruit". Similairement, la composante 5 n'est jamais sélectionnée par svm-rfe ou l0 mais est conservée par relieff et fisher, ce qui nous indique qu'elle fait partie vraisemblablement des caractéristiques redondantes. En outre, parmi les 13 types de caractéristiques, certains sont considérés comme inutiles par toutes les procédures (type 8). Par ailleurs, les sous-ensembles de primitives sélectionnées induisent des performances de classification SVM équivalentes, ce qui signifie qu'aucune des méthodes de sélection n'a perdu d'information utile et que le nombre idéal de caractéristiques se situe entre 10 et 20.

## 5 Conclusion et perspectives

Nous avons montré dans cette étude la nécessité d'appliquer une procédure de sélection automatique de caractéristiques en vue d'une tâche de classification. Nous

avons comparé différents algorithmes de sélection (supervisés) récents à l'aide des erreurs de classification induites. Nous avons montré leur efficacité à l'aide d'un problème synthétique ainsi que d'un problème de classification d'images de textures.

Nous préconisons l'usage d'une procédure de sélection, non seulement pour réduire l'espace de stockage et améliorer les performances de classification, mais aussi pour justifier le choix d'un modèle donné de caractéristiques. Dans l'étude présentée ci-dessus, nous avons appliqué cette stratégie à la sélection de modèles de textures.

Dans la plupart des systèmes impliquant un grand nombre de données et de caractéristiques, les étiquettes associées aux données ne sont pas connues. Nous nous intéressons donc maintenant aux algorithmes de sélection non supervisés, qui exploitent la capacité à clusteriser des données multivaluées ainsi qu'aux heuristiques permettant d'évaluer ces algorithmes. Ces travaux sont menés, pour le CNES, dans le but de produire des outils de traitement automatique en vue de l'indexation de bases d'images satellitaires de très grande taille.

## Références

- [Blum et Langley, 1997] A. L. Blum et P. Langley. Selection of relevant features and examples in machine learning. *Artif. Intell.*, 97(1-2):245–271, 1997.
- [Boland, 1998] M. Boland. Programmation en c des coefficients d'haralick. 1998.
- [Chang et Lin, 2004] C.C. Chang et C.J. Lin. Libsvm 2.5. 2004.
- [Cortes et Vapnik, 1995] C. Cortes et V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, 1995.
- [Do et Vetterli, 2003] M.N. Do et M. Vetterli. Contourlets. In *Beyond Wavelets*. Academic Press, 2003.
- [Elisseeff, 2000] A. Elisseeff. *Etude de la complexité et contrôle de la capacité des systèmes d'apprentissage : SVM multi-classe, réseaux de régularisation et réseaux de neurones multicouches*. PhD thesis, ENS Lyon, 2000.
- [Guyon et Elisseeff, 2003] I. Guyon et A. Elisseeff. An introduction to feature and variable selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
- [Guyon, 2002] I. Guyon. Gene selection for cancer classification using support vector machines. *Mach. Learn.*, 46:389–422, jan 2002.
- [Haralick *et al.*, 1973] R.M. Haralick, K. Shanmugam, et I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 6:610–621, 1973.
- [Kim *et al.*, 2002] Y. S. Kim, W. N. Street, et F. Menczer. Evolutionary model selection in unsupervised learning. *Intelligent Data Analysis*, 6:531–556, 2002.
- [Kira et Rendell, 1992] K. Kira et L.A. Rendell. A practical approach to feature selection. pages 249–256, 1992.
- [Kohavi et John, 1997] R. Kohavi et G. H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1-2):273–324, 1997.
- [Rakotomamonjy, 2003] A. Rakotomamonjy. Variable selection using svm-based criteria. *Journal of Machine Learning Research*, 3:1357–1370, 2003.

- [Robnik-Sikonja et Kononenko, 2003] M. Robnik-Sikonja et I. Kononenko. Theoretical and empirical analysis of relief and rrelief. *Mach. Learn.*, 53(1-2):23–69, 2003.
- [Rui *et al.*, 2001] Y. Rui, T.S. Huang, et S.F. Chang. Image retrieval: current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10:39–62, 2001.
- [Schoelkopf et Smola, 2002] B. Schoelkopf et A. Smola. *Learning with Kernels*. 2002.
- [Sebe et Lew, 2000] N. Sebe et M.S. Lew. Wavelet based texture classification. volume 3, 2000.
- [Simoncelli, 2001] E.P. Simoncelli. Matlab tools for multi-scale image processing. 2001.
- [Weston *et al.*, 2003] J. Weston, A. Elisseeff, B. Scholkopf, et M. Tipping. Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439–1461, 2003.
- [Weston *et al.*, 2004] J. Weston, A. Elisseeff, G. Bakir, et Fabian Sinz. The spider for matlab - v1.4. 2004.