

---

# Découverte de règles de classification par approche évolutive : application aux images de télédétection

Jerzy Korczak — Arnaud Quirin

LSIIT, CNRS/Université Louis Pasteur  
Pôle API, Boulevard Sébastien Brant  
F-67400 Illkirch cedex  
{korcza,quirin}@lsiit.u-strasbg.fr

---

*RÉSUMÉ.* Nous décrivons dans cet article une nouvelle méthode de classification d'images de télédétection. Ces images contiennent des données volumineuses et complexes : les informations sont parfois bruitées, parfois erronées. Avec notre approche, des règles de classification sont découvertes par un algorithme évolutif au lieu d'appliquer un algorithme de classification choisi a priori. Durant le processus évolutif de découverte, les règles de classification sont créées en utilisant des images de télédétection brutes, une expertise contenue dans des images classifiées, et des statistiques sur les concepts thématiques à découvrir. Les règles découvertes sont simples à interpréter, efficaces, robustes et résistantes au bruit. Nous détaillons dans l'article l'algorithme évolutif conçu, ainsi que des validations sur des images de télédétection qui couvrent non seulement la zone urbaine de Strasbourg, mais aussi une zone de végétation dans la lagune de Venise.

*ABSTRACT.* In this article, a new method of classification of remote sensing images is described. Usually, these images contain voluminous, complex, and sometimes erroneous and noisy data. In our approach, classification rules are discovered by evolution-based process instead of applying a priori chosen classification algorithm. During the evolution process, classification rules are created using raw remote sensing images, the expertise encoded in classified zones of images, and statistics about related thematic objects. The discovered rules are simple to interpret, efficient, robust and noise resistant. In the paper, the evolution-based approach is detailed and validated on remote sensing images covering not only urban zones of Strasbourg, France, but also vegetation zones of the lagoon of Venice.

*MOTS-CLÉS :* Images de télédétection, règles de classification, image haute résolution, image hyperspectrale, apprentissage supervisé, apprentissage évolutif, algorithme génétique, système de classificateurs.

*KEYWORDS:* Remote sensing image, classification rules, high resolution image, hyperspectral image, supervised learning, evolutionary learning, genetic algorithm, classifiers system.

---

## 1. Introduction

En général, la classification d'images de télédétection consiste à découvrir des concepts thématiques comme *eau* ou *végétation clairsemée* à partir d'une image brute et à produire une carte détaillée (appelée classification) de la disposition spatiale de ces classes. La mise en place d'algorithmes de classification d'images efficaces et robustes est l'une des demandes les plus importantes de la part des utilisateurs d'images de télédétection. Depuis des années, de gros efforts ont été développés pour élaborer de nouveaux algorithmes de classification et pour raffiner les méthodes utilisées pour classifier les séries de données statistiques (Bock, 1999). Cependant peu d'algorithmes statistiques ou déterministes ont démontré une fiabilité exemplaire et souvent la meilleure méthode reste l'expertise humaine qui est extrêmement chère.

C'est pourquoi nous proposons une nouvelle méthode d'explication automatique de classification en synthétisant les relations existantes entre l'image brute et l'image classifiée. Cette synthèse est décrite par des règles de classification décidant de l'appartenance ou non de chaque pixel à une classe donnée. Dans cet article, une nouvelle approche guidée par les données est proposée pour découvrir de telles règles de classification en exploitant l'idée des systèmes de classifieurs évolutifs. La seule source d'informations nécessaire est l'image de télédétection et sa classification préalable, fournie par un expert. Les images peuvent être produites par de nombreux satellites (p. ex. SPOT, LANDSAT, DAIS) utilisant différents systèmes de capteurs et sont donc de résolutions spectrale et spatiale variées (Weber, 1995).

L'objectif de nos recherches est de proposer une nouvelle méthode de classification - évolutive - qui, contrairement aux algorithmes de classification habituels, permettra de générer de manière supervisée des classifieurs autonomes. Les classifieurs sont, d'une manière générale, un ensemble de règles de classification découvertes par un système de classifieurs (DeJong, 1988 ; Wilson, 1999). Comme nous l'avons dit, ce système est guidé par les données car les règles de classifications produites sont capables de s'adapter elles-mêmes aux informations disponibles, à l'environnement, et à l'évolution des classes.

La littérature en télédétection présente différentes approches de classification, notamment : la classification pixel par pixel (Fjørtoft, 1996), l'approche par zone (Kurita, 1993), l'approche par objet (Korczak, 1999).

Notre approche est basée sur l'information spectrale des pixels : les règles de classification ainsi découvertes ne sont donc capables que de trouver les classes spectrales, pas les classes sémantiques. Cette composante spectrale de la description des classes est essentielle pour mieux reconnaître les classes thématiques, elle en constitue le premier pas. Le système de classifieurs proposé peut être adapté facilement afin de proposer une représentation plus sophistiquée des objets manipulés. Mais cela nécessite un effort de recherche plus appuyé sur la reconnaissance des caractéristiques des objets.

La population initiale de classifieurs est aléatoirement créée à partir de l'image et d'un ensemble donné de classes. Ensuite, elle évolue grâce à l'algorithmique génétique jusqu'à ce qu'une solution acceptable soit trouvée. Ce modèle de système de classifieurs a été implanté dans un prototype nommé ICU (Quirin, 2002), qui a été utilisé pour valider notre approche. Dans ICU, nous avons adapté et étendu des idées existantes et reconnues, comme XCS (Wilson, 1999), les S-classifieurs, les systèmes de classifieurs flous (Rendon, 1997). Nous avons été aussi inspirés par les travaux de (Riolo, 1988) sur la gratification et la pénalisation, et de (Richards, 1995) sur l'exploration de l'espace de recherche des classifieurs.

La structure de l'article est la suivante : la partie 2 introduit la notion fondamentale de classifieurs évolutifs, la partie 3 détaille le processus de découverte des règles de classification et explique le comportement des fonctions génétiques de l'algorithme. Nous présentons dans la partie 4, trois études de cas sur des données réelles de télédétection.

## 2. Définition d'un système de classifieurs évolutifs en télédétection

Un système de classifieurs intègre apprentissage symbolique et évolution. Des *règles*, sortes d'expressions symboliques permettant de décrire au classifieur la conduite à tenir dans un cas particulier sont apprises, voire oubliées, en fonction de leurs performances et de l'environnement courant. Ces règles ne seront introduites ni par le programmeur, ni par un expert du domaine.

Un système de classifieur est dit *évolutif* s'il présente une capacité d'adaptation : il peut alors modifier son comportement et ses connaissances en fonction de la situation de l'environnement à un instant donné. Par exemple en télédétection, la taille des classes peut évoluer si après une première classification, on découvre des classes de pixels non classifiés, ou correspondant à plusieurs classes (pixels mixtes). Lorsqu'un classifieur intégrera l'un de ces pixels restants à l'une ou l'autre des classes, il faudra adapter dynamiquement l'algorithme : certaines règles qui ne traitaient que les cas simples (pas de pixels mixtes) deviendront désuètes, et d'autre part, il nous faudra créer de nouvelles règles efficaces pour les autres cas.

Ensuite, un classifieur, dans le sens composant d'un système de classifieur, est une sorte de règle représentant les connaissances du système. Voici la représentation d'une de ces règles (Wilson, 1999) : si *<condition>* alors *<action>* où chaque partie est un message comportant des 0 ou des 1, permettant de coder des informations élémentaires, comme une couleur, une forme ou tout autre constante. La partie *<condition>* est utilisée comme un message d'entrée dans le système, correspondant à une condition à remplir pour déclencher cette règle. La partie *<action>* est le message de sortie correspondant, codant l'action à effectuer lorsque la règle est activée.

A la différence de certaines méthodes d'analyse effectuées dans le domaine de la télédétection, nous voudrions pouvoir disposer de ce que l'on nomme des *règles de classification*, c'est-à-dire que le système produise de lui-même une certaine forme d'information permettant à partir de ces mêmes pixels de déduire les classes thématiques des zones couvertes, mais que cette forme d'information soit du mieux possible réutilisable sur une partie non traitée de l'image, sans recommencer à chaque fois le long processus d'apprentissage. Ces règles devraient, après la phase d'apprentissage, *s'évaluer* (i.e. s'appliquer sur tous les pixels de l'image) rapidement et tenir compte de toute l'information spectrale que peut contenir un pixel. Les règles produites devraient être d'une manière générale *simple* et elles devront être maximale spécifiquement, c'est-à-dire qu'elles couvriront le maximum de pixels de la classe qui lui sera propre, en minimisant le nombre de pixels d'autres classes.

Nous venons de le voir, le but de nos classifieurs est d'associer à un pixel caractérisé par un *vecteur spectral* (ensemble des valeurs de réflectance de ce pixel pour chaque bande) l'instance de la classe thématique correspondante. Il nous faut donc une structure de donnée capable de mémoriser une telle association.

Tout d'abord, rappelons qu'un pixel est en fait un vecteur spectral, définissant une valeur de réflectance pour les  $n$  bandes de l'image :

$$\langle \text{pixel} \rangle := [ b_1 b_2 \dots b_n ]$$

La condition d'une règle repose sur la notion d'*intervalle spectral*. Un tel intervalle est un couple de nombres entiers, qui permet de découper l'espace des valeurs spectrales en deux espaces : celui des valeurs souhaitées pour les pixels d'une même classe, et le reste.

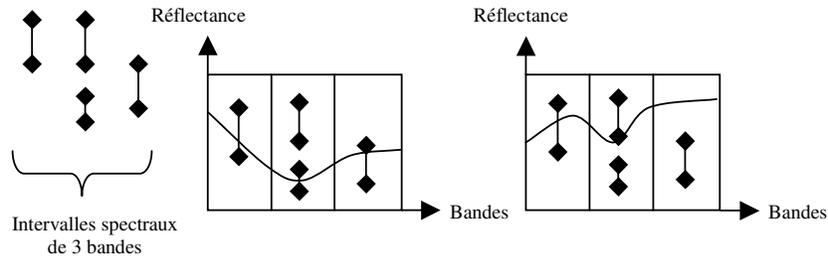
Pour généraliser le concept et mieux cibler les classes, nous affectons à chaque bande disponible dans l'image brute un ensemble d'intervalles éventuellement disjoints.

Voici, avant de rentrer dans les détails, la forme complète de la partie *<condition>* d'une règle :

$$\langle \text{condition} \rangle := E_1 \wedge E_2 \wedge \dots \wedge E_n$$

où  $n$  est le nombre total de bandes et  $E_i$  sont des ensembles d'intervalles définis pour toutes les bandes présentes dans les données sources. Chaque ensemble définit un ou plusieurs intervalles spectraux de la forme suivante :  $E_i := [m_1 ; M_1] \vee [m_2 ; M_2] \vee \dots \vee [m_p ; M_p]$  où  $[m_j ; M_j]$  est appelé un intervalle spectral,  $m_j$  et  $M_j$  sont respectivement les réflectances minimale et maximale autorisées d'un pixel pour la bande  $i$ , pour que celui-ci active la règle (dans ce cas,  $m_j \leq b_j \leq M_j$ ).

Pour satisfaire la règle, un pixel doit donc satisfaire *tous* les ensembles d'intervalles spectraux pour chaque bande et pour satisfaire un ensemble d'intervalles, il doit satisfaire l'un des intervalles de cet ensemble. Les règles définissent donc des conjonctions de disjonctions d'intervalles (Fig. 1).



**Figure 1.** Correspondance entre une règle et le spectre des pixels

La figure à gauche montre les intervalles spectraux définis dans une règle pour une classe C. Les figures suivantes montrent le spectre de deux pixels dans une image qui active ou non la règle. On classe ainsi ou non le pixel dans la classe C.

Cette représentation a surtout été choisie parce qu'elle apporte la simplicité dans les résultats présentés à l'utilisateur, et par le fait que l'expression de multiples contraintes reste tout de même compacte. De plus la notion d'ensemble d'intervalles est à même de saisir l'uniformité des pixels qui appartiendraient à la même classe. On peut bien sûr imaginer des structures plus complexes, par exemple en utilisant le voisinage des pixels ou bien sous forme d'équations mathématiques munies de multiples opérateurs.

### 3. De la création à l'évolution

#### 3.1. Algorithme génétique

L'algorithmique génétique est utilisée en général pour résoudre des problèmes n'ayant pas de méthode de résolution classique ou ceux ayant un espace de contraintes - parfois contradictoires - trop vaste pour être parcouru dans un temps raisonnable. Le principe général consiste à créer aléatoirement des solutions acceptables pour le problème donné puis à les soumettre à un processus d'évolution lors duquel elles vont s'adapter (mutations, hybridations). Seules les meilleures solutions subsistent.

Afin de faciliter la création de règles de classification, l'algorithme initialise les valeurs des attributs en fonction des classes attribuées par l'expert sur chaque zone validée de l'image, en considérant à chaque fois la valeur de réflectance minimum et maximum calculée pour chaque bande. Ensuite on confie ce *pool* initial à un algorithme génétique qui est chargé de le faire évoluer vers une solution quasi-optimale. Notre système de classificateurs est en fait un ensemble de classificateurs, répondant chacun à une classe, et évoluant de manière distincte.

Pour répondre au critère de qualité, il nous faut un système de notation basé sur l'image experte : les classifieurs sont jugés en fonction de la surface qu'ils couvrent par rapport au modèle de l'expert, et l'expert mis à part, sur le degré de chevauchement des surfaces entre elles. Il en découle une note réelle entre 0 et 1. Les règles sont donc sélectionnées en fonction de la note obtenue pour une certaine classe, mais il est aussi possible d'utiliser des notes globales en définissant un *arbre thématique*, permettant de rassembler plusieurs classes.

On le voit donc, un algorithme déterministe de découverte de ces règles est tout simplement proscrit. Nous avons donc choisi l'algorithmique génétique, connue pour sa robustesse à traiter ce type de problème d'optimisation et pour ses qualités puissantes d'exploration et de recherche.

---

#### **Algorithme 1.** *Déroulement d'un algorithme génétique*

---

R est une règle de classification et P, P' et P'' sont des populations de règles.

```

R := REGLE_INITIALE(images) // Création d'une règle selon les images
P := INITIALISATION(R) // Création d'un pool aléatoire de règles
EVALUATION(P) // Calcul de la fonction fitness pour chaque règle
répéter tant_que CRITERE_ARRÊT(P) = faux
    P' := SELECTION_CROISEMENT(P) // Sous-population à croiser
    P' := CROISEMENT(P') ∪ RECOPIE(P) // Exploitation des règles
    P'' := SELECTION_MUTATION(P') // Sous-population à muter
    P'' := MUTATION(P'') ∪ RECOPIE(P') // Exploration des règles
    EVALUATION(P'')
    P := RECYCLAGE(P, P'') // Recyclage générationnel
fin répéter

```

**Résultat :** R, une règle de classification pour une classe donnée

---

On le voit dans l'algorithme, nous avons conçu notre système de classifieurs comme un ensemble de classifieurs indépendants, ce qui permet d'éviter qu'un classifieur performant engloutisse un classifieur qui mettrait plus de temps à s'améliorer. L'utilisateur peut aussi définir une hiérarchie de classifieurs afin d'avoir une information plus précise sur les qualités de certains classifieurs, permettant éventuellement de relancer l'algorithme avec des paramètres mieux adaptés au traitement des classes à affiner.

### **3.2. La fonction d'évaluation**

La fonction d'évaluation permet de différencier les bons des mauvais classifieurs, et donc de guider l'évolution génétique de la population. La fonction calculant la *fitness* des classifieurs dépendant fortement du domaine d'application, elle se détermine à partir de la classification obtenue par les règles ( $I_{r\grave{e}gle}$ ) et celle demandée par l'expert ( $I_{expert}$ ). Nous présentons dans le Tableau 1 les quantités évaluées (cardinal de l'ensemble des pixels correspondant à chaque cas).

**Tableau 1.** Calcul de la fonction d'évaluation

		Image classifiée par la règle $R$ ( $I_{r\grave{e}gle}$ )	
		Pixels activant $R$	$R$ inactive
Image classifiée par l'expert $E$ ( $I_{expert}$ )	Pixels activant $E$	$P_{exp}^{reg}$	$\overline{P}_{exp}^{reg}$
	$E$ inactif	$\overline{P}_{exp}^{reg}$	$P_{exp}^{reg}$

Dans l'implémentation nous avons proposé la fonction *fitness* suivante :

$$N_{finale} = C_{classe} \cdot N_{classe} + \overline{C}_{classe} \cdot \overline{N}_{classe} \quad [1]$$

où  $N_{classe} = \frac{P_{exp}^{reg}}{P_{exp}^{reg} + \overline{P}_{exp}^{reg}}$ ,  $\overline{N}_{classe} = \frac{\overline{P}_{exp}^{reg}}{\overline{P}_{exp}^{reg} + P_{exp}^{reg}}$ , et  $C_{classe}$  et  $\overline{C}_{classe}$  sont

appelés coefficient de reclassement et rééquilibre certains cas de classes sous-représentées à l'extrême, ils valent  $\frac{1}{2}$  par défaut.

Notre fonction *fitness* a les avantages suivants : indépendance du parcours des pixels, invariabilité par rapport à la taille des classes et résolution du problème des classes sous ou sur-représentées.

Il est important que le processus d'évolution converge. Le critère de terminaison de l'algorithme s'appuie sur certains critères statistiques, définis de telle sorte qu'il soit probable que le meilleur classifieur est proche d'un maximum global ou que la population n'évoluera plus. Nous utilisons notamment la détection de palier (arrêt si la *fitness* se stabilise). Soit  $Q_k$  la qualité du meilleur classifieur du pool génétique obtenu au cours de la  $k$ -ième dernière génération, avec  $k \geq 1$ , et  $Q_0$  la qualité du meilleur classifieur de la génération courante. L'algorithme est interrompu si l'inégalité suivante est vérifiée :

$$\left| \frac{\sum_{k=1}^P Q_k}{P} - Q_0 \right| \leq E \quad [2]$$

où  $P$  représente la longueur maximale d'un palier et  $E$  la variation maximale de ce palier par rapport à la note courante : plus  $E$  est petit, plus le palier devra être constant afin de pouvoir être détecté.

### 3.3. Opérateurs génétiques

La séparation de l'algorithme génétique en couches (sélection, croisement, mutation, ...) nous est apparu comme une nécessité à la fois du point de vue de la

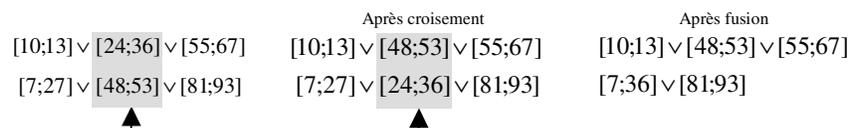
programmation, mais donne en outre un certain pouvoir de modularité à l'utilisateur. Chaque couche réalise une opération précise et retransmet le résultat à la couche suivante qui le traite à son tour.

### 3.3.1. La sélection des classifieurs

Nous distinguons plusieurs occasions dans lequel l'algorithme va devoir sélectionner un classifieur, ce qui nous donne autant de fonctions de sélection à appeler : la sélection pour la reproduction, pour la recopie, celle du classifieur à garder et celle du classifieur à éliminer. Les stratégies proposées à l'utilisateur sont le rang, l'élitisme aléatoire, l'élitisme pur, l'eugénisme pur et le hasard (Blickle, 1995).

### 3.3.2. Le croisement des parents

L'opérateur de croisement est utilisé afin d'exploiter les qualités des classifieurs dont on dispose. Les propriétés des parents sont exploitées lors du croisement de deux bons classifieurs pour découvrir quels intervalles doivent être gardés ou supprimés, à l'aide d'un opérateur spécifique que nous avons défini, consistant à réaliser un croisement (échange) uniforme des intervalles au sein des règles. La figure 2 montre un exemple du résultat du croisement (après validation et fusion) pour une bande donnée de deux règles. Cependant, la fusion conduit à long terme à une diminution du nombre d'intervalles dans les règles (simplification implicite engendrée par l'opérateur de croisement), ainsi qu'une perte d'information : en effet, il est peut être important pour la génération suivante de conserver le fait d'avoir deux intervalles distincts, pour ne pas éliminer de règles prématurément, sans qu'elles aient eut le temps de démontrer leur potentiel.



**Figure 2.** Fusion après un croisement

### 3.3.3. La mutation des classifieurs

La mutation permet d'explorer l'espace encore laissé vierge par l'opérateur précédent. Les règles sont en fait un ensemble relativement important de variables, surtout lorsque l'on traite des images hyperspectrales. Nous avons donc choisi d'appliquer l'opérateur de mutation à plusieurs niveaux, c'est-à-dire que l'on va à la fois agir sur une bande complète, sur un intervalle ou sur une borne. Outre la capacité d'exploration, cet unique opérateur qui en fait en comprend plusieurs, permet de *corriger* les erreurs commises par l'opérateur de croisement, grâce à la grande finesse des modifications qu'il procure. Au lieu d'écraser les bornes par des valeurs aléatoires, on les déplace, ce qui est beaucoup plus justifiable.

La mutation ne concerne qu'un et un seul objet de la règle à la fois (bande, intervalle, borne). Ceci permet à l'algorithme de tester les modifications les unes après les autres et à la fonction *fitness* de privilégier le type de modification et l'effet correspondant au cas par cas. A chaque fois, l'algorithme choisit aléatoirement l'opérateur à appliquer.

La mutation opère aux différents niveaux de représentation d'une règle : bande, intervalle, bornes. La mutation de bande consiste à supprimer dans la règle les contraintes liées à certaines bandes sélectionnées. La mutation d'intervalle permet d'ajouter, supprimer ou découper un intervalle en deux. Le découpage puis la suppression permet donc de cibler exactement le *tube spectral* dans lequel les valeurs spectrales des pixels peuvent discriminer une certaine classe. Enfin, la modification ponctuelle d'une ou deux bornes d'un intervalle prolonge et affine l'idée de ciblage réalisée par les autres types de mutation.

### 3.3.4. Le recyclage générationnel

Il existe plusieurs stratégies pour produire le pool final qui survivra à la génération suivante à partir de celle des parents, notamment : la stratégie révolutionnaire et la stratégie préservative.

Souvent, les meilleurs classifieurs de l'ancienne population remplacent les plus mauvais de la population modifiée. Cette stratégie permet d'être certains que les classifieurs dont la qualité est maximale passent directement à la population suivante. Cependant elle présente le risque de voir des classifieurs perdurer à travers les âges : ceci n'est a priori pas gênant, sauf dans le cas d'un pool génétique faible avec quelques classifieurs de performances moyennes qui bénéficieraient alors d'une immunité bien discutable.

## 4. Etude de cas

### 4.1. Expérimentations

L'approche présentée a été validée sur de nombreuses images de télédétection : des images haute résolution SPOT (3 canaux, 8 bits/composante, résolution 1,30m, Fig. 3), des images hyperspectrales DAIS (DAIS, 2001) ou ROSIS (80 à 100 canaux, 16 bits/composante, résolution 3m, Fig. 4) et des images expertes, obtenues par segmentation automatique auprès d'outils statistiques. Le premier exemple présenté illustre la présence de détails perturbateurs sur une image haute résolution. A l'exemple urbain, nous opposons le cas rural du second exemple dans lequel les pixels mixtes de végétation de la lagune de Venise offrent une difficulté supplémentaire. L'apprentissage a été réalisé à partir de données classées de la moitié inférieure de l'image, et la validation sur l'image entière. Pour le premier cas (192\*224 pixels), voici un extrait des règles obtenues : [15;37] [9;40] [1;15] pour *Eau*, [8;38] [0;21] [10;35] pour *Ombre* et [14;78] [0;25] [31;87] pour *Végétation*.

Nous présentons l'image classifiée obtenue à partir de nos classifieurs associée à un commentaire sur la performance des règles et les classifications obtenues.

***Image de Strasbourg, Stade Vauban (image haute résolution)***



**Figure 3.** Image classifiée

*Commentaire* : Image complexe par un niveau de détail qui fausse la classification : 300000 classifieurs générés et évalués, 11 classes apprises, performance de 90,74 %. Bons résultats cependant pour les classes EAU/OMBRE habituellement difficiles. Les règles obtenues découpent correctement les zones interstitielles (ombre des arbres, terrains). Les mêmes règles ont obtenu une performance de 89 % sur Strasbourg entier (image 23 fois plus large).

***Image de Venise, San Felice (image hyperspectrale)***



**Figure 4.** Image classifiée

*Commentaire* : Très bons résultats détaillés malgré les 80 canaux de l'image et ceux bruités : 10000 classifieurs générés et évalués, 12 classes apprises, performance de 96,13 %. Ici les classes en milieu rural étaient sémantiquement très proches (différentes sortes d'algues) mais la très bonne qualité de l'expert a permis à notre système de classifieurs d'éviter les difficultés et d'obtenir la meilleure performance de cette étude de cas. Nous avons utilisé des paramètres très robustes dont le revers est de conduire à un apprentissage plutôt long.

**4.2. Discussion**

Par ces deux exemples, nous avons illustré la capacité de nos classifieurs à intégrer des données à la fois hétérogènes et complexes (dimensions, nombre de bandes, bruit) en délaissant les bandes inexploitable. L'apprentissage est sans doute plutôt long, mais cela vient de la taille des images, de l'algorithmique génétique, et de notre désir de privilégier des paramètres plus *sûrs* que *rapides*. Les résultats obtenus sont d'ailleurs assez bons, au moins par rapport à l'expertise à notre disposition, et l'algorithme peut conduire dans certains cas à son amélioration. Les pixels mal classés proviennent de données bruitées (capteurs défectueux ou perturbations atmosphérique), d'une résolution élevée pour la Fig. 3 (le niveau de détail conduit à un nombre réel beaucoup plus élevé de classes peu représentées), et

d'erreurs sur l'expertise utilisée. En cas de chevauchement de plusieurs règles, ou de l'absence de règles pour classifier un pixel, nous utilisons une méthode statistique permettant de choisir la meilleure règle, en minimisant la distance entre le vecteur spectral du pixel et les valeurs de la règle.

Néanmoins, nous avons vu que le cas hyperspectral révélait quelques limitations à cette structuration : nous pensons que des règles fondées sur des centres spectraux entourés de nuages de probabilité captant tel ou tel pixel seraient plus à même de modéliser ce type d'image. La structure utilisée a été malgré tout capable de créer des règles mutuellement exclusives et maximales, classifiant la totalité des pixels, alors que cette condition n'est pas évidente à produire car elle n'est pas décrite explicitement dans le modèle.

Enfin nous terminerons par citer la bonne corrélation entre les résultats que nous avons obtenus et certaines études statistiques de l'image (spectrogramme, exclusion des bandes bruitées sans qu'il a été nécessaire de les signaler). En général, la réalité du terrain, même implicite, est donc reproduite fidèlement dans les règles de classification découvertes par notre algorithme. En effet, elles sont à la fois adaptées aux grands espaces (terrains de sport) comme aux plus petits détails (arbres).

## 5. Conclusions et perspectives

Au cours de notre projet, nous avons mené à bien une étude des concepts entrant d'une part dans la définition d'un système capable de découvrir des règles décrivant les classes thématiques contenues dans une image de télédétection et d'autre part, dans la définition d'un mécanisme d'intégration de ces connaissances dans un système de classifieurs. Ces règles sont à la fois pratiques et simples pour l'utilisateur, et suffisamment génériques pour permettant de réitérer le processus de nombreuses fois, sans avoir à recommencer l'apprentissage. De plus, elles peuvent être appliquées sur d'autres images, présentant les mêmes caractéristiques (ensoleillement, nombre de canaux, ...) ou sur des flux vidéos, ce qui présente un grand intérêt.

Les résultats des expérimentations sont très encourageants, par rapport aux difficultés posées par le contenu très riche de telles images et par la qualité médiocre de l'expertise. Notre système est donc robuste, sûr et la simplicité des règles permet tout de même de reproduire fidèlement la réalité, adaptée aux grandes et petites classes. Il a donné lieu à un prototype simple d'emploi, actuellement disponible sur le site <http://hydria.u-strasbg.fr>.

Dans l'avenir le plus proche, nous nous intéresserons à l'extension du formalisme d'encodage des règles pour y inclure d'autres types de connaissances (spatiale, temporelle, 3D, structures, ...) et à l'affinement des opérateurs génétiques. Nous cherchons à améliorer les temps de réponse de l'algorithme d'une part par le

paramétrage plus efficace de l'algorithme, et d'autre part par l'implémentation de notre système sur une grille de processeurs.

#### Remerciements

Nous tenons à remercier Pierre Gañçarski, Cédric Wemmert, Christiane Weber, Anne Puissant, Massimo Menenti et Jélila Labeled pour leurs conseils et leurs commentaires qu'ils ont apportés à la réalisation de ce projet de recherche.

#### 6. Références

- T. Blicke, L. Thiele, *A Comparison of Selection Schemes used in Genetic Algorithms*, Computer Engineering and Communication Networks Lab, TIK-Report Nr. 11, Second Edition, Swiss Federal Institute of Technology, Zurich, 1995.
- H. H. Bock, E. Diday, (eds.) Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data, [in] *Studies in Classification, Data Analysis and Knowledge Organization*, vol. 15, Springer-Verlag, Heidelberg, 1999.
- DAIS, M. Wooding, *Proceedings of the Final Results Workshop on DAISEX (Digital Airborne Spectrometer EXperiment)*, ESTEC, Noordwijk, 2001.
- K. A. DeJong, Learning with Genetic Algorithms: An Overview, *Machine Learning*, vol. 3, pp. 121-138, 1988.
- R. Fjørtoft, P. Marthon, A. Lopes, F. Sery, D. Ducrot-Gambart, E. Cubero-Castan, Region-Based Enhancement and Analysis of SAR Images, [in] *Proc. of ICIP'96*, vol. 3, Lausanne, pp. 879-882, 1996.
- J. Korczak, N. Louis, Synthesis of Conceptual Hierarchies Applied to Remote Sensing, [in] *Proc. of SPIE*, Image and Signal Processing for Remote Sensing IV, Barcelona, pp. 397-406, 1999.
- T. Kurita, N. Otsu, Texture Classification by Higher Order Local Autocorrelation Features, [in] *Proc. of Asian Conf. on Computer Vision*, Osaka, pp. 175-178, 1993.
- A. Quirin, Découverte de règles de classification : classifieurs évolutifs, Mémoire DEA d'Informatique, Université Louis Pasteur, LSIT UMR-7005 CNRS, Strasbourg, 2002.
- M. V. Rendon, *Reinforcement Learning in the Fuzzy Classifier System*, Reporte de Investigaci No. CIA-RI-031, ITESM, Campus Monterrey, Centro de Inteligencia Artificial, 1997.
- R. A. Richards, *Zeroth-Order Shape Optimization Utilizing A Learning Classifier System*, <http://www.stanford.edu/~buc/SPHINcsX/book.html>, Stanford, 1995.
- R. L. Riolo, Empirical Studies of Default Hierarchies and Sequences of Rules in Learning Classifier Systems, PhD Dissertation, Comp. Sc. and Eng. Dept, Univ. of Michigan, 1988.
- C. Weber, *Images satellitaires et milieu urbain*, Hermès, Paris, 1995.
- S. W. Wilson, State of XCS Classifier System Research, [in] *Proc. of IWLCS-99*, Orlando, 1999.