# Partitionnement non supervisé de données de grandes dimensions spatiale et spectrale pour l'aide à la décision

Jihan Alameddine

# THESE DE DOCTORAT DE

L'UNIVERSITE DE RENNES 1

ECOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies de l'Information et de la Communication*
Spécialité : Signal, Image, Vision

Par
**Jihan ALAMEDDINE**

# Partitionnement non supervisé de données de grandes dimensions spatiale et spectrale pour l'aide à la décision

**Thèse présentée et soutenue à l'université de Rennes 1, le 15 Novembre 2022**
**Unité de recherche : 6164 IETR**
**Thèse N° :**

**Rapporteurs avant soutenance** :

| | |
|---|---|
| Gabriela CIUPERCA | MC-HDR / Université Claude Bernard Lyon 1 |
| Anissa MOKRAOUI | Professeur / Université Sorbonne Paris Nord |

**Composition du Jury :**

| | | |
|---|---|---|
| Président : | Didier COQUIN | Professeur / Université de Savoie Mont Blanc |
| Examinateurs : | Jenny BENOIS-PINEAU | Professeur / Université Bordeaux 1 |
| | Gabriela CIUPERCA | MC-HDR / Université Claude Bernard Lyon 1 |
| | Anissa MOKRAOUI | Professeur / Université Sorbonne Paris Nord |
| Dir. de thèse : | Kacem CHEHDI | Professeur / Université de Rennes 1 |
| Co-encadrant : | Claude CARIOU | MC / Université de Rennes 1 |

UNIVERSITÉ DE RENNES 1

**Titre :** Partitionnement non supervisé de données de grandes dimensions spatiale et spectrale pour l'aide à la décision.

**Mots clés :** partitionnement non supervisé, données de grande taille, images hyperspectrales, aide à la décision, validation, sélection autonome d'échantillons d'apprentissage.

**Résumé :** Le partitionnement d'un ensemble de données de grande taille, où le nombre de classes, d'échantillons d'apprentissage et d'autres connaissances *a priori* ne sont pas disponibles, pose un défi considérable. Ainsi, la conception de méthodes de partitionnement fiables, où toutes les décisions sont prises uniquement sur la base du tableau de données croisant objets/attributs est un problème complexe. Pour apporter une solution à ce problème, nous nous intéressons dans la cadre de cette thèse au développement de méthodes de partitionnement non supervisées et non paramétriques adaptées aux données de grande taille quel que soit le domaine applicatif. La première partie consacrée aux travaux de l'état de l'art présente d'abord les principaux critères d'évaluation d'une partition et donne ensuite une synthèse des principales méthodes de partitionnement mettant en évidence leurs avantages et leurs limites.
La seconde partie présente les trois approches de partitionnement non supervisées développées. Pour confirmer leur caractère général, elles ont été appliquées à trois domaines : l'environnement, la reconnaissance des visages avec expressions et la médecine, avec des données acquises par des capteurs différents. Les évaluations montrent le succès des méthodes développées au vue de la pertinence des résultats. En effet, sans aucune intervention de l'utilisateur, les performances sont meilleures que celles des méthodes semi-supervisées et non supervisées les plus efficaces de l'état de l'art.

**Title:** Unsupervised partitioning of large spatial and spectral data for decision making.

**Keywords:** unsupervised partitioning, large-size data, hyperspectral images, decision making, validation, autonomous selection of learning samples.

**Abstract:** Partitioning a large-size dataset, where the number of classes, training samples, and other *a priori* knowledge are not available, poses a considerable challenge. Thus, designing reliable partitioning methods, where all decisions are made based on the objects/attributes cross data table only, is a complex problem. In order to address this problem, this thesis focuses on the development of unsupervised and nonparametric partitioning methods. Moreover, they are adapted to large-size data whatever the application domain.
The first part dedicated to the state-of-the-art work first presents the main criteria for evaluating a partition and then gives a synthesis of the main partitioning methods highlighting their advantages and their limitations. The second part presents the three unsupervised partitioning approaches developed. To confirm their general character, they were applied to three domains: environment, face recognition with expressions and medicine; with data acquired by different sensors. The evaluations show the success of the developed methods in view of the relevance of the results. Indeed, without any user intervention, the performances are better than the most efficient semi-supervised and unsupervised methods of the state-of-the-art.

# Table de matière

**Notation** ..... 6
**Liste des acronymes** ..... 7
**Résumé**.. ..... 10
**Position du problème et objectifs** ..... 11
**Partie I** ..... 16
**Etat de l'art** ..... 16
**Chapitre 1 : Critères d'évaluation d'une méthode de partitionnement** ..... 17
1.1 Introduction ..... 17
1.2 Critères d'évaluation non supervisés ..... 17
1.2.1 Critère d'évaluation de Levine et Nazif (*LN*) ..... 18
1.2.2 Indices mesurant la compacité d'une partition ..... 18
1.2.3 Indices mesurant la séparabilité d'une partition ..... 20
1.2.4 Indices hybrides ..... 20
1.2.5 Indice mesurant la connectivité ..... 25
1.3 Critères d'évaluation supervisés ..... 26
1.4 Discussion ..... 29
**Chapitre 2 : Partitionnement semi-supervisé et non supervisé** ..... 30
2.1 Introduction ..... 30
2.2 Méthodes de partitionnement ..... 32
2.2.1 Méthodes semi-supervisées ..... 32
2.2.2 Méthodes non supervisées ..... 35
2.3 Analyse numérique de la sensibilité de l'AP ..... 44
2.3.1 Base de données 1 ..... 44
2.3.2 Base de données 2 ..... 46
2.4 Discussion ..... 48
**Conclusion de la Partie I** ..... 50
**Partie II** ..... 52
**Approches de partitionnement non supervisées et non paramétriques développées** ..... 52
**Chapitre 3 : Choix du critère de similarité** ..... 53
3.1 Introduction ..... 53
3.2 Travaux associés ..... 54
3.3 Analyse théorique des critères de similarité ..... 57
3.4 Validation numérique ..... 61
3.5 Discussion ..... 66
**Chapitre 4 : Méthodes de partitionnement non supervisées et non paramétriques par propagation d'affinité pour des données de grande taille** ..... 68
4.1 Introduction ..... 68
4.2 Méthode de partitionnement hiérarchique et non supervisée par AP (HUP-OAP) ..... 69
4.2.1 Optimisation de l'AP ..... 69
4.2.2 Partitionnement des données de grande taille par bloc et hiérarchisation ..... 77
4.2.3 Evaluation numérique ..... 81
4.2.4 Conclusion ..... 90
4.3 Méthode HUP-OAP avec réduction de la taille de la matrice de similarité ..... 90
4.3.1 Réformulation du critère de disponibilité ..... 91
4.3.2 Transformation des données réduisant la taille la matrice de similarité ..... 103
4.3.3 Partitionnement des données transformées et hiérarchisation ..... 107
4.3.4 Evaluation numérique ..... 109
4.4 Discussion ..... 116
**Chapitre 5 : Méthode non supervisée et autonome pour la sélection des échantillons d'apprentissage** ..... 117
5.1 Introduction ..... 117
5.2 Approche proposée ..... 119
5.2.1 Travaux associés ..... 119
5.2.2 Sélection des échantillons d'apprentissage ..... 123
5.3 Evaluation numérique ..... 124

5.3.1 Évaluation sur les données IRIS ..... 126
5.3.2 Évaluation sur l'image hyperspectrale synthétique ..... 135
5.3.3 Evaluation sur l'image hyperspectrale aérienne réelle de grande taille ..... 141
5.4 Discussion ..... 148
**Chapitre 6 : Méthode de partitionnement non supervisée et non paramétrique par association directe et indirecte pour des données de grande taille** ..... 150
6.1 Introduction ..... 150
6.2 Travaux associés ..... 151
6.3 Méthode proposée ..... 154
6.3.1 Notations et Définitions ..... 154
6.3.2 Approche hiérarchique et non supervsée par association directe et indirecte ..... 158
6.4 Evaluation numérique ..... 160
6.4.1 Partitionnement de l'image synthétique ..... 160
6.4.2 Partitionnement de l'image réelle de grande taille ..... 162
6.5 Discussion ..... 164
**Chapitre 7 : Application des méthodes développées à trois domaines : l'environnement, la reconnaissance faciale avec expressions et la médecine** ..... 166
7.1 Introduction ..... 166
7.2 Application à l'environnement-Détection de plantes invasives ..... 167
7.2.1 Application à une image hyperspectrale synthétique ..... 167
7.2.2 Application à une image réelle hyperspectrale ..... 169
7.3 Application à la reconnaissance faciale avec expressions ..... 180
7.3.1 Application à la base de données JAFFE ..... 180
7.3.2 Application à la base de données YALE ..... 185
7.3.3 Application à la base de données ORL ..... 188
7.4 Applications médicales ..... 190
7.4.1 Application à la détection de tumeur cérébrale par IRM ..... 190
7.4.2 Application à la détection de mélanomes ..... 191
7.5 Discussion ..... 192
**Conclusion générale** ..... 193
**Annexes** ..... 195
**Annexe 1 Transformation de caractéristiques visuelles invariante à l'échelle (SIFT)** ..... 195
**Annexe 2 Bases de données expressions faciales** ..... 200
**Annexe 3 Bases de données médicales** ..... 206
**Références** ..... 210
**Liste des Figures** ..... 216
**Liste des Tableaux** ..... 218

# Remerciements

Je voudrais tout d'abord remercier le département des Côtes d'Armor et l'équipe « Traitement des Signaux & Images Multicomposantes et Multimodales » de l'IETR pour le co-financement de cette thèse.

Je tiens à remercier particulièrement et vivement, le professeur Kacem CHEHDI, directeur de cette thèse, pour ses encouragements à débuter ce travail, pour la place qu'il m'a permis de prendre au sein du laboratoire, pour sa confiance tout au long de mon cheminement, pour sa disponibilité, pour ses conseils avisés, ses orientations explicites et implicites. Grâce à lui, j'ai pu mener cette thèse à son terme.

Je remercie toute l'équipe « Traitement des Signaux & Images Multicomposantes et Multimodales » de m'avoir accueillie pour la préparation de cette thèse. Et plus particulièrement Claude CARIOU et Josias LEFEVRE pour toutes les discussions et conseils et Mme Joëlle THEPAULT pour le support administratif.

Mes remerciements vont ensuite à Gabriela CIUPERCA et Anissa MOKRAOUI pour m'avoir fait l'honneur de rapporter sur ce travail. Je souhaite également remercier les examinateurs Jenny BENOIS-PINEAU et Didier COQUIN d'avoir accepté de faire partie du jury.

Je tiens à remercier toute ma famille qui a tout fait pour m'aider et qui m'a supportée dans tout ce que j'ai entrepris et mes amis pour leurs soutiens.

Encore un grand merci à tous.

# Notation

$X = \{x_1, x_2, \dots, x_i, \dots, x_N\}$ : l'ensemble des $N$ individus (ou objets) à partitionner

$A_i = (a_{i1}, a_{i2}, \dots, a_{iB})$ : vecteur d'attributs quantitatifs représentant l'individu $x_i$

$B$ : nombre d'attributs de $x_i$

$H(x_i)$ : fréquence d'apparition de l'individu $x_i$

$N_c$ : nombre estimé de classes

$C_i$ : $i^{ème}$ classe formée

$P = \{C_1, C_2, \dots, C_i, \dots, C_{N_c}\}$ : partition formée par une méthode de partitionnement

$N_{C_i}$ : nombre d'individus dans la classe $C_i$

$K$ : nombre de classes de la vérité de terrain

$V = \{V_1, V_2, \dots, V_j, \dots, V_K\}$ : vérité de terrain

$V_j$ : $j^{ème}$ classe de la vérité de terrain

$N_{V_j}$ : nombre d'individus dans la classe $V_j$

$G_i$ : centre de gravité de la classe $C_i$

$m$ : paramètre de flou

$S$ : matrice de similarité

$R$ : matrice responsabilité

$A$ : matrice de disponibilité

$p$ : paramètre de préférence

$\lambda$ : facteur d'amortissement

$L_q$ : norme d'ordre $q$

$d(.,.)$ : distance

$d_q(.,.)$ : distance associée à la norme $L_q$

# Liste des acronymes

## Partie I

**AP :** Affinity Propagation

**CCR :** Correct Classification Rate

**CH :** Calinski-Harabasz index

**DB :** Davies-Bouldin index

**FCM :** Fuzzy C-Means

**LBG :** Linde-Buzo-Gray algorithm

***LN* :** Levine-Nazif index

**MEAP :** Multi-Exemplar Affinity Propagation

**MLBG :** Modified Linde-Buzo-Gray algorithm

**NMI :** Normalized Mutual Information

**NW-FCM :** New Weighted Fuzzy C-Means

**OFCM :** Optimized Fuzzy C-Means

**RMSSD :** Root-Mean-Square Standard Deviation index

**RS :** R-Squared index

**Sil :** Silhouette index

**VT :** Vérité de Terrain

**XB :** Xie-Benie index

## Partie II

### Méthodes de partitionnement non supervisées développées

- ***Méthode 1***

**UP-OAP :** Unsupervised Partitioning by Optimized Affinity Propagation

**HUP-OAP :** Hierarchical UP-OAP

- ***Méthode 2***

**UP-OAPM :** Unsupervised Partitioning by Optimized Affinity Propagation with Modified availability

**HUP-OAPM-RSM :** Hierarchical UP-OAPM with Reduced Similarity Matrix

- ***Méthode 3***

**UP-DIA :** Unsupervised Partitioning by Direct and Indirect Association

**HUP-DIA :** Hierarchical UP-DIA

### Méthode de Sélection d'échantillons d'apprentissage développée

**TS-HUP-OAP :** Training Samples by Hierarchical and Unsupervised Partitioning by Optimized Affinity Propagation

**H-TS :** Homogeneity Training Samples

**MSH-TS :** Moderately Strong Homogeneity Training Samples

**SH-TS :** Strong Homogeneity Training Samples

**OGT-TS :** Original Ground Truth where 30% of the objects in each class are randomly selected as "Training Samples"

**CGT-TS :** Corrected Ground Truth where 30% of the objects in each class are randomly selected as "Training Samples"

### Acronymes des méthodes de l'état de l'art

**ANN :** Artificial Neural Network

**BPERN :** Back Propagation Elman Recurrent Network

**CBSNN :** Curiosity-Based Spiking Neural Network

**CCR-SCVT :** Correct Classification Rate avec Subdivision des Classes de la VT

**CNN :** Convolutional Neural Network

**CRN :** Chemical Reaction Network

**CS :** Cuckoo Search

**CSERN :** Cuckoo Search for Elman Recurring Network

**DBSCAN :** Density-Based Spatial Clustering of Applications with Noise

**ERN :** Elman's Recurrent Network

**HCS :** Highly Connected Subgraphs

**JAFFE :** JApanese Female Facial Expression

**KNN :** K-Nearest Neighbors

**LE-LSTM :** Laplacian Eigenmap Long Short-Term Memory

**MOGA-OCD :** Multi-Objective Genetic Algorithm for Overlapping Community Detection

**MST :** Minimum Spanning Tree

**ORL :** Olivetti Research Laboratory

**RECOME :** RElative COre MErge

**RNNKD :** Relative *K* Nearest Neighbor Kernel Density

**RNN :** Recurrent Neural Network

**RWR :** Random Walk with Restart

**SAM :** Spectral Angle Mapper

**SCMAG :** Subspace Clustering in Multi-valued Attributed Graph

**SID :** Spectral Information Divergence

**SIFT :** Scale Invariant Feature Transform

**SNGC :** Shared Neighbors Graph Clustering algorithm

**SNN :** Spiking Neural Network

**S-OFCM :** Stable and Optimized Fuzzy C-Means

**SSC-OMP :** Sparse Subspace Clustering by Orthogonal Matching Pursuit

**SVM :** Support Vector Machine

**U-OFCM :** Unsupervised and Optimized Fuzzy C-Means

**V-NIR :** Visible-Near Infra Rred

# Résumé

Le partitionnement d'un ensemble de données de grande taille, où le nombre de classes, les échantillons d'apprentissage et d'autres connaissances *a priori* ne sont pas disponibles, pose un défi considérable. Ainsi, la conception des méthodes de partitionnement fiables, où toutes les décisions sont prises uniquement sur la base du tableau de données croisant objets/attributs est un problème difficile à résoudre. En effet, à mesure que les ensembles de données augmentent respectivement en taille et en dimension, la gestion des ressources, le temps d'exécution et l'espace mémoire deviennent importants, ce qui explique l'intérêt croissant que connait le partitionnement des données. Il existe par conséquent dans la littérature une abondance des méthodes de partitionnement pour de nombreux domaines applicatifs tels que : la géologie, l'agriculture, le militaire, la médecine, l'environnement et la sécurité. Cependant, ce problème est loin d'être résolu.

Pour apporter une solution, relever les défis et atteindre les objectifs fixés, nous nous intéressons dans la cadre de cette thèse au développement de méthodes de partitionnement non supervisées et non paramétriques applicables aux données de grande taille quel que soit le domaine.

La première partie de cette thèse consacrée aux travaux de l'état de l'art présente d'abord les principaux critères d'évaluation d'une partition, ensuite une synthèse des principales méthodes de partitionnement mettant en évidence leurs avantages et leurs limites. Les approches non supervisées, non paramétriques et hiérarchiques s'avèrent les plus adaptées au problème posé où aucune connaissance *a priori* sur les données à partitionner n'est admise.

La seconde partie présente les trois approches de partitionnement non supervisées, non paramétriques et hiérarchique développées. Ces approches présentent plusieurs avantages, à savoir : (1) aucune connaissance *a priori* n'est requise ; (2) la stabilité des résultats grâce à leur caractère déterministe ; (3) la sélection optimisée de l'exemplaire de chaque classe autour duquel les individus sont agrégés ; (4) un temps de calcul très faible avec un traitement par blocs; (5) l'applicabilité aux données quelle que soit leur taille; (6) la possibilité d'élaborer plusieurs partitions hiérarchiques en indiquant la plus pertinente selon un critère objectif ; et enfin, (7) la possibilité de sélectionner objectivement les échantillons des classes pour un système d'apprentissage. Pour ce dernier point, une méthode autonome et non supervisée est développée dans les cas où les échantillons d'apprentissage sont indisponibles, peu fiables ou insuffisants.

Les méthodes de partitionnement non supervisées développées ont été appliquées avec succès à trois domaines applicatifs : l'environnement, la reconnaissance faciale avec expressions et la médecine dont les données ont été acquises avec des capteurs différents. Ces évaluations montrent la pertinence et l'objectivité des résultats obtenus sans aucune intervention de l'utilisateur. De plus, leurs performances sont meilleures que celles des méthodes semi-supervisées telles que le Fuzzy C-Means (FCM), le FCM stable et optimisé (S-OFCM) et le *K*-means. Elles surpassent également les méthodes non supervisées telles que le Fuzzy C-Means non supervisée et optimisée (U-OFCM) et la propagation d'affinité originale (AP).

## Position du problème et objectifs

Le développement des systèmes décisionnels ou d'aide à la décision optimisés pour apporter des solutions fiables et objectives aux besoins exigeants suscités par de nombreux secteurs applicatifs est un domaine de recherche actif. Cela concerne par exemple l'environnement, la sécurité alimentaire (qualité sanitaire des aliments), la médecine (surveillance, diagnostic), l'industrie, la défense, la biologie, l'aérospatial et l'agriculture.

Pour répondre aux exigences de ces secteurs, la qualité de l'analyse du contenu informationnel des données acquises dans le cadre de chaque secteur est une étape clé dans la conception de ces systèmes. La résolution de ce problème passe en général par un partitionnement objectif des données en vue de faciliter les prises de décision et leur compréhension par les utilisateurs. Pour pouvoir répondre aux besoins soulevés par des applications réelles, le partitionnement de données a connu donc un intérêt croissant, ce qui explique l'abondance des méthodes proposées. Nous notons cependant que l'analyse des données par des méthodes de partitionnement est loin d'être résolu efficacement, c'est-à-dire avec objectivité et pertinence. Ce problème complexe nécessite donc une démarche rigoureuse pour pouvoir apporter des solutions appropriées aux réels problèmes posés. Cette rigueur doit porter sur le vocable employé et sur la démarche scientifique suivie afin de stopper toutes les confusions.

L'analyse efficace et la génération de nouvelles connaissances à partir de ces masses de données nécessitent l'utilisation bien adaptée des techniques de classification. Ainsi, pour répondre efficacement aux besoins de tous ces domaines, toute méthode de partitionnement doit être menée en respectant la nature physique des données. Une multitude de méthodes publiées ne respectent pas l'information précise fournie par ces données, où les vérités de terrain utilisées sont simplifiées et la classification des méthodes existantes est parfois ambiguë et non cohérente. Par exemple, des méthodes introduisant le nombre de classes dans le processus de partitionnement sont qualifiées comme non supervisées. Or il est faux de les considérer rigoureusement comme telles car l'introduction du nombre de classes par l'utilisateur est une opération supervisée.

Afin de lever toutes les confusions, nous classons selon des critères liés à la nature des connaissances introduites par les utilisateurs les méthodes de partitionnement en trois grandes catégories au lieu de deux (supervisée et non supervisée), comme souvent proposé dans la

littérature. La confusion est due au fait que les méthodes qui nécessitent une connaissance *a priori* du nombre de classes sont dans la même catégorie que celles qui estiment le nombre de classes. Ainsi, nous avons subdivisé la catégorie des méthodes non supervisées en deux catégories (méthodes semi-supervisées et non supervisées) pour avoir une meilleure dissimilarité entre les catégories. Il est important de rappeler que les méthodes de chaque catégorie peuvent être hiérarchiques ou non-hiérarchiques et paramétriques ou non-paramétriques.

Les méthodes de partitionnement non supervisées et non paramétriques présentent plusieurs avantages par rapport aux approches supervisées et semi-supervisées, telle la non nécessité de préciser par l'utilisateur, à l'avance les classes à discriminer et les échantillons d'apprentissage (des relevés de terrain). En fait, les méthodes supervisées nécessitent des échantillons d'apprentissage pour effectuer le processus de partitionnement. Par conséquent, leurs performances sont fortement liées à la pertinence des échantillons d'apprentissage sélectionnés. La difficulté d'obtenir des échantillons d'apprentissage fiables a toujours été l'un des principaux facteurs empêchant ces méthodes d'atteindre une précision élevée. De plus, le choix des échantillons d'apprentissage reste souvent subjectif et difficile à faire. Ces connaissances *a priori* ne permettent pas la découverte de nouvelles classes pertinentes. Dans ce cas, l'introduction de ces connaissances comme données d'entrée peut être considérée comme une contrainte et ne reflète pas souvent la réalité pratique. En outre, les méthodes paramétriques nécessitent le réglage subjectif d'un ou plusieurs paramètres. S'il est conseillé de choisir un ensemble optimal de paramètres, il est extrêmement difficile d'y parvenir, car ils varient fortement d'un environnement à l'autre et d'un type de données à l'autre.

Nous précisons que dans cette thèse nous utilisons le terme "partitionnement" car il est à la fois plus pratique et plus approprié pour décrire les méthodes d'analyses des données développées. En effet, il est plus pratique de dire par exemple « partitionnement d'images » au lieu d'utiliser un terme long, tel que "classification des pixels d'images" et ainsi éviter le terme "classification d'images", qui est un autre problème.

**Définition générale :** Le "***partitionnement des données***" consiste à diviser un ensemble d'objets décrits par des caractéristiques quantitatives et/ou qualitatives en différents "sous-ensembles" ou classes homogènes, selon un critère de similarité dans le sens où les objets de chaque sous-ensemble partagent des caractéristiques communes. Les classes obtenues forment une partition.

Le partitionnement des données peut être effectué avec ou sans apprentissage. Dans ce dernier cas, toutes les classes sont localisées et les nouvelles classes détectées, peuvent être analysées, identifiées et répertoriées pour les ajouter aux classes connues en vue d'un éventuel apprentissage ou non. Ce cas de figure se trouve dans plusieurs domaines (environnement, médecine, sécurité, biologie, etc.).

Le partitionnement d'une image ou de données consiste à créer une partition définie formellement comme suit :

Soit $X = \{x_1, x_2, \dots, x_N\}$ l'ensemble de $N$ individus à partitionner, où chaque individu $x_i$ est caractérisé par un vecteur d'attributs quantitatifs $A_i = \{a_{i1}, a_{i2}, \dots, a_{iB}\}$, avec $B$ le nombre d'attributs. Nous notons que l'ensemble $X$ est présenté sous forme d'un tableau de données croisant $N$ individus × $B$ attributs quantitatifs. Dans le cas de partitionnement d'une image, celle-ci sera représentée aussi sous forme d'un tableau de données.

Le processus de division de $X$ en $K$ classes, $C_i$, consiste à créer une partition, $P$, suivant un ou plusieurs critères d'optimisation objectifs avec : $P = \{C_1, C_2, \dots, C_K\}$.

Cette partition $P$ va donc mettre en évidence les différentes classes de l'ensemble de données $X$.

La qualité d'une partition va dépendre du degré d'homogénéité des classes formées et par conséquent de leur nombre. Ainsi, afin d'obtenir un résultat de partitionnement correct, trois conditions doivent être vérifiées simultanément :

***Exhaustivité :*** tous les individus de l'ensemble de données doivent être associés à une classe.

$$\forall\, x_i \in X, \exists\, C_j \text{ tel que } x_i \in C_j$$

***Séparabilité :*** les classes doivent être suffisamment différenciables pour qu'un individu ne puisse être associé qu'à une seule classe.

$$\bigcup_{i=1}^{K} C_i = X \text{ et } \bigcap_{i=1}^{K} C_i = \emptyset$$

***Pertinence :*** l'association d'un individu à une classe doit être effectuée selon un critère objectif d'optimisation.

Dans le cadre de cette thèse, nous cherchons à développer des approches de partitionnement prenant en considération la réalité physique des données à partitionner et qui doivent vérifier les conditions suivantes :

- Non supervisées, sans aucune connaissance *a priori* (ni nombre de classes, ni d'échantillons d'apprentissage, ou seuils d'agrégation, ou nombre d'itérations, etc.) ;
- Non paramétriques ;
- Applicables à des données de grande taille ;
- Applicables aux différents types de données représentées par des attributs quantitatifs quel que soit le domaine visé ;
- Stable avec des résultats probants.

Cette thèse est donc organisée en deux parties. La première partie qui porte sur les travaux de l'état de l'art, est composée de deux chapitres. La seconde partie, composée de cinq chapitres, est consacrée aux nouveaux développements.

Le ***premier chapitre*** porte sur les critères d'évaluation des méthodes de partitionnement. Les principaux critères supervisés et non supervisés sont analysés.

Le ***deuxième chapitre*** est dédié à l'état de l'art des méthodes de partitionnement semi et non supervisées. L'objectif de ce chapitre est d'analyser les principales méthodes de partitionnement, afin de mettre en avant leurs avantages et inconvénients et ainsi montrer la nécessité de développer des nouvelles approches de partitionnement.

Le ***troisième chapitre*** porte sur le choix de critère de similarité pour une méthode de partitionnement. Une analyse théorique et numérique est conduite sur les différents indices de similarité utilisés en partitionnement, pour pouvoir choisir l'indice le plus performant pour les approches développées.

Le ***quatrième chapitre*** présente les deux premières approches de partitionnement développées. Tout d'abord nous démontrons théoriquement les inconvénients de la méthode de partitionnement non supervisée de l'état de l'art retenue. Ensuite, des modifications importantes sont apportées pour pouvoir l'appliquer à des données de grande taille.

Le ***cinquième chapitre*** aborde le problème de la sélection des échantillons d'apprentissage et propose une méthode autonome et non supervisée pour la sélection des échantillons d'apprentissage fiable pour les méthodes supervisées en utilisant la méthode de partitionnement développée dans le chapitre précédent.

Dans le ***sixième chapitre***, nous développons une nouvelle approche de partitionnement basée sur de nouveaux critères d'optimisation mesurant la connectivité entre les individus à classer, qu'on nomme critères par association directe et association indirecte.

Dans le ***septième chapitre*** nous présentons et nous comparons chaque méthode de partitionnement développée dans le cadre de cette thèse sur des données correspondant à des applications différentes pour montrer leurs généralités. Trois domaines applicatifs sont considérés : environnement (localisation des structures paysagères par imagerie hyperspectrale), reconnaissance faciale avec expressions et médecine (détection de tumeur cérébrale par IRM et dermatologie).

Enfin, nous concluons les travaux menés et donnons des perspectives.

# Partie I

# Etat de l’art

L'objectif de cette partie est de donner une synthèse des principales méthodes de partitionnement de l’état de l’art mettant en évidence leurs avantages et limites. Cette partie est organisée en deux chapitres. Le *premier chapitre* décrit et discute les critères d'évaluation supervisés et non supervisés pour évaluer les résultats d’un partitionnement. Le *second chapitre* met particulièrement l’accent sur les principales approches de partitionnement nécessitant le minimum de connaissances *a priori* à savoir les méthodes semi-supervisées et aucune connaissance *a priori* pour désigner les méthodes non supervisées.

# Chapitre 1 : Critères d'évaluation d'une méthode de partitionnement

## 1.1 Introduction

Les processus d'évaluation et de validation des résultats des méthodes de partitionnement sont des tâches essentielles pour mesurer en toute objectivité les performances d'un algorithme. Pour évaluer ces performances et quantifier la qualité d'une partition, deux types d'indices peuvent être employés : les indices externes et les indices internes [1]. La principale différence réside dans l'utilisation ou non d'informations externes dans les processus d'évaluation et de validation d'une partition. Les indices externes, qui sont des critères supervisés, mesurent la concordance entre deux partitions où la première est connue *a priori* (VT), considérée comme référence et la seconde est le résultat d'un algorithme de partitionnement [2]. Cependant, les connaissances *a priori* sur les données ne sont pas toujours disponibles et lorsqu'elles existent, elles peuvent être biaisées. Tandis que les indices internes, qui sont des critères non supervisés, permettent d'évaluer une partition en utilisant des quantités et des caractéristiques inhérentes de l'ensemble de données considéré, c'est-à-dire reposent sur la structure intrinsèque des données. La plupart des indices d'évaluation interne quantifient la qualité d'un partitionnement en termes de compacité et de séparation entre les classes.

Dans ce chapitre, nous analysons les principaux critères internes (non supervisés) et externes (supervisés) pour l'évaluation des résultats de partitionnement. Il est à noter que ces critères peuvent également être utilisés comme indices d'agrégation pour affecter un individu à une classe ou bien comme critère d'arrêt.

## 1.2 Critères d'évaluation non supervisés

Les critères d'évaluation internes, ou non supervisés, sont basés sur les mesures des propriétés des classes pour identifier les meilleures partitions. Une classe bien formée doit être compacte et la plus éloignée par rapport aux autres classes, c'est-à-dire qu'elle doit avoir une similarité intra-classe élevée et une faible similarité interclasse. Ces indices peuvent être classés en quatre familles : indices mesurant la compacité, indices mesurant la séparabilité, indices hybrides et indice de connectivité. Nous rappelons ici les principaux.

### 1.2.1 Critère d'évaluation de Levine et Nazif (*LN*)

Levine et Nazif [3] proposent une mesure de performance d'une partition basée sur le calcul de la somme des contrastes des régions ou classes $C_i$ pondérées par leurs aires $Air(C_i)$. Le contraste d'une région est défini à partir des contrastes existants avec les régions adjacentes.

$$LN = \frac{\sum_{C_i} Air(C_i) LN_i}{\sum_{C_i} Air(C_i)} \tag{1.1}$$

$$LN_i = \sum_{C_i} \frac{l_{ij}}{\mathrm{Pr}(C_i)} \left| \frac{\mu_i - \mu_j}{\mu_i + \mu_j} \right| \tag{1.2}$$

Avec $\mu_i$ le centre de gravité de la région $C_i$, $l_{ij}$ la longueur de la frontière commune entre $C_i$ et $C_j$ et $\mathrm{Pr}(C_i)$ le périmètre de la région $C_i$.

Plus cette mesure est élevée, meilleure est la qualité des résultats de partitionnement.

### 1.2.2 Indices mesurant la compacité d'une partition

Ces indices mesurent la proximité entre les différents objets de la même classe. L'une des mesures les plus couramment utilisées pour la compacité est la variance.

**Définition 1.1.** Soient $C_i$ et $C_j$ deux classes issues d'un processus de partitionnement, $C_i$ est plus compacte que $C_j$ si $var_i < var_j$, avec $var_i$ est la variance de la classe $C_i$ définie par :

$$var_i = \frac{1}{N_{C_i}} \sum_{x_i \in C_i} d^2(x_i, G_i) \tag{1.3}$$

où $N_{C_i}$ est le nombre d'individus de la classe $C_i$ et $G_i$ est son centre de gravité.

En outre, il existe de nombreuses mesures basées sur la distance pour estimer la compacité d'une classe, telle que la distance maximale ou moyenne. Les indices les plus connus sont les suivants :

- <u>Inertie intra-classe</u> [4] : ce critère à minimiser indique le degré d'homogénéité entre les individus $x_i$ et $x_j$ appartenant à la même classe $C_i$. Il calcule leurs distances par rapport au point représentant le profil de la classe. Il est défini par :

$$Intra = \frac{1}{N_c} \sum_{C_i} \frac{1}{2N_{C_i}} \sum_{x_i \in C_i} \sum_{x_j \in C_i} d^2(x_i, x_j) \tag{1.4}$$

où $N_c$ le nombre de classes d'une partition.

- *Root-mean-square standard deviation* [5] : ce critère à minimiser permet également la mesure de l'homogénéité des classes formées. Il est défini par :

$$RMSSD = \left\{ \sum_{i=1}^{N_c} \sum_{x_i \in C_i} d^2\,(x_i, G_i) \,/\, [B \times \sum_{i=1}^{N_c} (N_{C_i} - 1)] \right\}^{\frac{1}{2}} \tag{1.5}$$

avec $B$ le nombre d'attributs.

Cet indice diminue quand le nombre de classes augmente. En pratique, $N_c \ll N$, alors $\sum_{i=1}^{N_c}(N_{C_i} - 1) = N - N_c$ peut être vu comme un nombre constant. Alors $RMSSD = \sqrt{\sum_{i=1}^{N_c} \sum_{x_i \in C_i} d^2(x_i, G_i)/\text{constante}}$ avec $\sum_{i=1}^{N_c} \sum_{x_i \in C_i} d^2(x_i, G_i)$ est la somme de l'erreur quadratique qui diminue quand le nombre de classes augmente.

- Calinski-Harabasz index [6] : ce critère pondère la variance intra-classe par le nombre de classes et permet donc une évaluation plus objective tenant compte de la population d'une classe formée. Sa maximisation correspond à la partition optimale et il est défini comme suit :

$$CH = \frac{\sum_{i=1}^{N_c} N_{C_i} d^2(G_i, G_X)/(N_c - 1)}{\sum_{i=1}^{N_c} \sum_{x_i \in C_i} d^2(x_i, G_i)/(N - N_c)} \tag{1.6}$$

où $G_X$ est le centre de gravité de l'ensemble de données $X$ et $G_i$ est le centre de gravité de la classe $C_i$.

Cet indice a pour effet uniforme de diviser les individus en tailles relativement égales et il n'a pas de bonnes performances lorsqu'il s'agit de jeux de données distribués asymétriques. De plus, le nombre optimal de classes est affecté en présence de données aberrantes. En fait, comme $(N - N_c) \,/\, (N_c - 1)$ est constant pour le même nombre de classes $N_c$, seul $\frac{\sum_{i=1}^{N_c} N_{C_i} d^2(G_i, G_X)}{\sum_{i=1}^{N_c} \sum_{x_i \in C_i} d^2(x_i, G_i)}$ doit être pris en compte. En présence des données aberrantes,

$\sum_{i=1}^{N_c} \sum_{x_i \in C_i} d^2(x_i, G_i)$ augmente de manière plus significative que $\sum_{i=1}^{N_c} N_{C_i} d^2(G_i, G_X)$. Par conséquent, pour le même $N_c$, $CH$ diminuera sous l'influence des données aberrantes, ce qui rend la valeur de cet indice instable.

### 1.2.3 Indices mesurant la séparabilité d'une partition

Ces indices mesurent à quel degré une classe est distincte ou bien séparée par rapport aux autres classes. Par exemple, les distances entre les centres des classes ou les distances minimales entre les individus dans différentes classes sont largement utilisées comme des mesures de séparation. En outre, les mesures basées sur la densité sont utilisées dans certains indices. Ces mesures sont faciles à calculer et permettent de détecter correctement les classes de forme hyper-sphérique.

- Inertie interclasse [4] : ce critère à maximiser mesure le degré d'hétérogénéité ou de séparabilité entre les classes. Il est défini par :

$$Inter = \frac{1}{N_c} \sum_{C_i} N_{C_i} d^2(G_i, G_X) \tag{1.7}$$

où $G_X$ est le centre de gravité de $X$ et $G_i$ est le centre de gravité de $C_i$.

La valeur de cet indice a le défaut d'augmenter quand le nombre de classes augmente.

- R-squared [7] : il mesure le degré de différence entre les classes.

$$RS = [\sum_{x_i \in X} d^2(x_i, G_X) - \sum_{i=1}^{N_c} \sum_{x_i \in C_i} d^2(x_i, G_i)] / \sum_{x_i \in X} d^2(x_i, G_X) \tag{1.8}$$

avec $G_X$ le centre de gravité de $X$ et $G_i$ le centre de gravité de $C_i$.

### 1.2.4 Indices hybrides

Ces indices permettent de mesurer la compacité et la séparabilité des classes en même temps.

- Indice de Dunn [8] : ce critère est à maximiser en calculant le rapport de la distance minimum entre deux individus classés ensemble sur la distance maximum qui sépare deux éléments classés différemment.

$$Dunn = \min_{i}\left\{\min_{j,j\neq i}\left\{\frac{\min\limits_{x_i\in C_i, x_j\in C_j} d(x_i,x_j)}{\max\limits_{k}\left\{\max\limits_{x_i,x_j\in C_k} d(x_i,x_j)\right\}}\right\}\right\} \tag{1.9}$$

Cet indice présente deux inconvénients : sa sensibilité à la présence des données aberrantes et son coût calculatoire. Lorsqu'il y a des données aberrantes, la séparation entre les classes peut fortement diminuer car elle utilise uniquement la distance minimale, plutôt que la distance moyenne entre les individus de différentes classes. De plus, le nombre de classes n'est pas optimal si l'ensemble de données ne présente pas de formes sphériques.

- Indice de Davies-Bouldin [9] : il est basé sur la minimisation du rapport des dispersions intra-classe et de la séparation interclasse. La meilleure partition est celle qui minimise la moyenne de la valeur calculée pour chaque classe. En d'autres termes, la meilleure partition est celle qui minimise la similarité entre les classes.

$$DB = \frac{1}{N_c}\max_{\substack{1\leq j\leq N_c\\ j\neq i}}\left\{\frac{S_a(C_i)+S_a(C_j)}{d_a(C_i,C_j)}\right\} \tag{1.10}$$

$$S_a(C_k) = \frac{1}{N_c}\sum_{x_i\in C_i} d(x_i, G_i) \tag{1.11}$$

$$d_a(C_i, C_j) = d(G_i, G_j) \tag{1.12}$$

avec $S_a$ la séparation interclasse et $d_a$ la dispersion intra-classe.

Cet indice tient alors compte à la fois de la compacité et de la séparabilité des classes, la valeur de cet indice est d'autant plus faible que les classes sont compactes et bien séparées. L'indice Davies-Bouldin peut être utilisé pour estimer le nombre de classes en cherchant son minimum global. Cet indice favorise les classes hyper-sphériques, c'est-à-dire si l'ensemble de données est de forme arbitraire, il n'est pas probant.

Par contre, il peut être utilisé pour comparer des partitions avec des nombres de classes identiques ou différentes.

- Indice $\mathfrak{T}$ [10] : cet indice est défini par :

$$\mathfrak{T} = \left(\frac{1}{N_c} \times \frac{\mathcal{E}_1}{\mathcal{E}_k} \times \max_{i,j} d(G_i, G_j)\right)^q \tag{1.13}$$

$$\mathcal{E}_1 = \sum_{x_i \in X} d(x_i, G_X) \tag{1.14}$$

$$\mathcal{E}_k = \sum_{k=1}^{N_c} \sum_{x_j \in C_k} d(G_k, x_j) \tag{1.15}$$

L'indice $\mathfrak{T}$ est une composition de trois facteurs, à savoir $\frac{1}{N_c}$, $\frac{\mathcal{E}_1}{\mathcal{E}_k}$ et $\max_{i,j} d(G_i, G_j)$. Le premier facteur permet de réduire la valeur de l'indice $\mathfrak{T}$ lorsque la valeur de $N_c$ augmente. Le second facteur est le rapport entre $\mathcal{E}_1$, qui est constant pour un ensemble de données donné et $\mathcal{E}_k$, qui diminue avec l'augmentation de $N_c$. Par conséquent, à cause de ce terme, cet indice augmente lorsque $\mathcal{E}_k$ diminue. Cela, à son tour, indique que la formation d'un plus grand nombre de classes, de nature compacte, serait encouragée. Enfin, le troisième facteur, qui mesure la séparation maximale entre deux classes sur toutes les paires de classes possibles, augmentera avec la valeur de $N_c$. Cependant, cette valeur est supérieure à la séparation maximale entre deux points de l'ensemble de données. Ainsi, les trois facteurs se concurrencent et s'équilibrent de manière critique. La puissance $q$ permet de contrôler le contraste entre les différentes configurations de classes et elle est généralement égale à deux ($q = 2$).

- Indice SD [11] : ce critère est basé sur le concept de dispersion moyenne et de séparation totale entre les classes. Le premier terme évalue la compacité basée sur la variance des classes et le second terme évalue la différence de séparation basée sur la distance entre les centres des classes. La valeur de cet indice est la somme de ces deux termes et le nombre optimal de classes est obtenu en minimisant la valeur de $SD$.

$$SD = Scat(N_c) + Dis(N_c) \tag{1.16}$$

avec :

$$Scat(N_c) = \frac{1}{N_c}\sum_{i=1}^{N_c} \| \ var_i \ \| / \| \ var_X \ \| \tag{1.17}$$

$$Dis(N_c) = \frac{\max_{i,j} d(G_i, G_j)}{\min_{i,j} d(G_i, G_j)} \sum_{i=1}^{N_c} \left( \sum_{j=1}^{N_c} d(G_i, G_j) \right)^{-1} \tag{1.18}$$

– $var_X$ : représente la variance de l'ensemble de données et est définie par :

$$var_X = \frac{1}{N}\sum_{i=1}^{N} d^2(x_i, G_X) \tag{1.19}$$

– $var_i$ : représente la variance du $i^{ème}$ classe $C_i$ et est définie par :

$$var_i = \frac{1}{N_{C_i}} \sum_{x_i \in C_i} d^2(x_i, G_i) \tag{1.20}$$

– $\|var\| = (var^T \cdot var)^{1/2}$ : norme de la variance

La plus petite valeur de $Scat(N_c)$ indique que la partition contient des classes compactes. Généralement la valeur de $Dist(N_c)$ augmente avec le nombre de classes et est aussi influencée par la structure géométrique des classes.

- *Indice de Silhouette* [12] : cet indice procède à l'échelle microscopique, c'est à dire qu'il s'intéresse aux individus en particulier et non pas aux classes. Le but de ce critère est de vérifier si chaque individu a été bien classé.

$$Sil = \frac{1}{N_c}\sum_{i=1}^{N_c} N_{C_i} \sum_{x_i \in C_i} sil(x_i) \tag{1.21}$$

$$sil(x_i) = \frac{b(x_i) - a(x_i)}{\max[b(x_i), a(x_i)]} \tag{1.22}$$

$$a(x_i) = \frac{1}{N_{C_i}} \sum_{x_j \in C_i, j \neq i} d(x_i, x_j) \tag{1.23}$$

$$b(x_i) = \min_{j, j \neq i}[\frac{1}{N_{C_j}} \sum_{x_j \in C_j} d(x_i, x_j)] \tag{1.24}$$

avec :

- Cohésion $a(x_i)$ (dissimilarité moyenne) : distance moyenne de $x_i$ à tous les autres individus de la même classe, plus $a(x_i)$ est petit, meilleure est l'assignation de $x_i$ à sa classe ;
- Séparation $b(x_i)$ : distance moyenne de $x_i$ aux individus dans les autres classes ;
- Silhouette $sil(x_i)$ : $sil(x_i) = [-1, +1]$, $-1 =$ mauvais, $0 =$ indifférent et $1 =$ meilleur.

Cet indice utilise la distance minimale moyenne entre les classes comme séparation interclasse. Pour un ensemble de données avec des sous-classes, c'est-à-dire des classes proches les unes des autres, la séparation interclasse atteindra sa valeur maximale lorsque des sous-classes proches les unes des autres sont considérées comme une seule et une même grande classe. Par conséquent, le mauvais nombre de classes optimal sera choisi en raison des sous-classes. Ainsi, cet indice ne peut fonctionner que dans les conditions des classes sphériques.

- *Indice de Xie-Beni* [13] : il définit la séparation interclasse comme la distance minimale entre les centres des classes et la compacité intra-classe comme la dispersion moyenne de tous les individus.

$$XB = \frac{1}{N}\left[\frac{\sum_{i=1}^{N_c}\sum_{x_i \in C_i} d^2(x_i, G_i)}{\min\limits_{i,j\neq i} d^2(G_i, G_j)}\right] \tag{1.25}$$

La partition optimale est atteinte lorsque le minimum de l'indice est trouvé. Cet indice décroit linéairement quand $N_c$ devient proche de $N$.

### 1.2.5 Indice mesurant la connectivité

Cet indice est défini comme une fonction quantifiant l'appartenance d'un individu aux voisinages des autres individus de l'ensemble des données [14]. L'indice de connectivité d'un individu est d'autant plus élevé que cet individu appartient à un nombre important de voisinages des autres individus.

L'indice de connectivité de $x_i \in X$ est défini comme :

$$I_C(x_i) = \sum_{j=1}^{N} \mathcal{M}\left(x_i, V_k(x_j)\right) \tag{1.26}$$

avec :

- $V_k(x_j)$ : voisinage de l'individu $x_i$, c'est l'ensemble des $k$ plus proches voisins dans $X$ de $x_i$, en utilisant une distance quelconque,
- $\mathcal{M}\left(x_i, V_k(x_j)\right)$ est une fonction d'appartenance à $V_k(x_j)$ évaluée au point $x_j$. Cette fonction est choisie de sorte qu'elle vaut 1 si $x_i \in V_k(x_j)$ et 0 sinon. Autrement dit, la connectivité d'un individu $x_i$ correspond au nombre de données auxquelles $x_i$ est connecté.

Cette fonction permet de quantifier la liaison d'un individu dans l'ensemble de données $X$ et révèle ainsi les structures des classes de $X$.

Ce critère de connectivité est directement informatif sur l'appartenance d'une donnée à une classe. Le partitionnement utilisant cet indice conserve les avantages des méthodes basées sur la densité (pas d'hypothèse sur la forme et le nombre des classes), mais il permet de plus de différencier sans difficulté des classes de faible densité et/ou de forte densité.

## 1.3 Critères d'évaluation supervisés

Les critères d'évaluation externes, ou supervisés, consistent à comparer les résultats de partitionnement à un résultat connu de l'extérieur, comme les vérités de terrain. Ils mesurent la correspondance entre les labels des classes obtenus par une méthode de partitionnement et les labels des classes fournis par la VT. Les principaux critères sont les suivants :

- Taux de classification correcte (CCR) (en %) : ce critère utilise l'information sur la partition obtenue par une méthode de partitionnement et le label de la classe de la VT sur toutes les paires de points. Il est calculé comme suit :

$$CCR = \frac{1}{N} \times [\sum_{i=1}^{K} Z_i] \times 100 \tag{1.27}$$

où $N$ est le nombre total d'individus dans l'ensemble de données $X$, $K$ est le nombre de classes de la VT et $Z_i$ est le nombre de points de la VT correctement classés dans chaque classe $V_j$ de la VT.

Les valeurs de ce critère sont comprises dans l'intervalle [0, 100] et des valeurs plus grandes indiquent une meilleure qualité de partitionnement.

- F-mesure [15] : la F-mesure est un mélange de deux indices, qui sont la Précision et le Rappel. La précision ($Pr$), qui mesure l'homogénéité des classes obtenues par une méthode de partitionnement par rapport aux classes de la VT connues *a priori* et le rappel ($Ra$), qui évalue la complétude des classes obtenues par rapport aux classes de la VT. La Précision et le Rappel sont calculés comme suit :

$$Pr(V_i, C_j) = \frac{N_{ij}}{N_{C_j}} \tag{1.28}$$

$$Ra(V_i, C_j) = \frac{N_{ij}}{N_{V_i}} \tag{1.29}$$

où $N_{ij}$ est le nombre d'individus de la classe $V_i$ de la VT qui sont dans la classe $C_j$ obtenue par une méthode de partitionnement, c.à.d. $N_{ij} = |V_i \cap C_j|$, $N_{C_j}$ est le nombre d'individus dans la classe $C_j$ et $N_{V_i}$ est le nombre d'individus dans la classe $V_i$.

La F-mesure de la classe $V_i$ de la VT et de la classe $C_j$ obtenue est donnée par :

$$F(V_i, C_j) = \frac{2 \times Pr(V_i, C_j) \times Ra(V_i, C_j)}{Pr(V_i, C_j) + Ra(V_i, C_j)} \tag{1.30}$$

Les valeurs de ce critère sont comprises dans l'intervalle [0, 1] et des valeurs plus grandes indiquent une meilleure qualité de partitionnement.

- Pureté [16] : la pureté quantifie la mesure dans laquelle une classe $C_i$ contient des entités provenant d'une seule partition. En d'autres termes, elle mesure la "pureté" de chaque classe. La pureté de la classe $C_i$, notée $Pu_i$, est définie comme suit :

$$Pu_i = \frac{1}{N_{C_i}} \max_{j=1,..,N} N_{ij} \tag{1.31}$$

La pureté d'une partition $P$ est définie comme la somme pondérée des valeurs de pureté de chaque classe :

$$Pu = \sum_{i=1}^{N_c} \frac{N_{C_i}}{N} Pu_i \tag{1.32}$$

où $N_{ij}$ est le nombre d'individus de la classe $V_i$ de la VT qui sont dans la classe $C_j$ obtenue, $N_{C_i}$ est le nombre d'individus dans la classe $C_i$, $N_{C_i}$ est le nombre d'individus dans la classe $C_i$ et $N$ est le nombre total d'individus dans $X$.

Plus la pureté de $P$ est grande, plus la concordance avec la VT est meilleure. La valeur maximale de la pureté est 1, lorsque chaque classe comprend des points d'une seule partition. Lorsque le nombre de classes estimé est égal au nombre de classes de la VT ($N_c = K$), une valeur de pureté de 1 indique un partitionnement idéal, avec une correspondance parfaite entre les classes obtenues et les classes de la VT. Cependant, la pureté peut être de 1 même pour $N_c > K$, lorsque chacune

des classes est un sous-ensemble des classes de la VT. Lorsque $N_c < K$, la pureté ne peut jamais être 1, car au moins une classe doit contenir des points de plus.

- Mesures basées sur l'entropie [17] :

L'entropie mesure la pureté des labels des classes de la partition $P$ et est définie comme suit :

$$En(VT\backslash P) = -\sum_{i=1}^{N_c}\sum_{j=1}^{K} p_{ij}\log\left(\frac{p_{ij}}{P_{C_i}}\right) \tag{1.33}$$

où $p_{ij} = \frac{N_{ij}}{N}$ est la probabilité qu'un individu dans la classe $C_i$ appartienne à la classe $V_j$ de la VT, $p_{C_i} = \frac{N_{C_i}}{N}$ est la probabilité de la classe $C_i$.

Plus les membres d'une classe sont répartis en différentes classes, plus l'entropie est élevée. Pour un partitionnement parfait, la valeur de l'entropie est nulle, tandis que la mauvaise valeur possible de l'entropie est $\log(N_c)$.

- Information mutuelle normalisée [18] :

L'information mutuelle vise à quantifier la quantité d'informations partagées entre la partition $P$ et la partition de la VT donnée par l'utilisateur. Elle est définie comme suit :

$$IM(P, VT) = \sum_{i=1}^{N_c}\sum_{j=1}^{K} p_{ij}\log\left(\frac{p_{ij}}{p_{C_i}\cdot p_{V_j}}\right) \tag{1.34}$$

où $p_{V_j} = \frac{N_{V_j}}{N}$ est la probabilité de la classe $V_j$ de la VT.

Elle mesure la dépendance entre la probabilité conjointe observée $p_{ij}$ de $P$ et $VT$ et la probabilité conjointe attendue $p_{C_i}\cdot p_{V_j}$ sous l'hypothèse d'indépendance. Quand $P$ et $VT$ sont indépendants alors $p_{ij} = p_{C_i}\cdot p_{V_j}$ et donc $IM(P, VT) = 0$. Cependant, il n'y a pas de limite supérieure pour l'information mutuelle.

L'information mutuelle normalisée ($NMI$) est définie comme suit :

$$NMI(P,VT) = \frac{IM(P,VT)}{\sqrt{En(P) \cdot En(VT)}} \tag{1.35}$$

avec :

$$En(P) = \sum_{i=1}^{N_c} p_{C_i} \log\left(p_{C_i}\right) \tag{1.36}$$

et

$$En(VT) = \sum_{j=1}^{K} p_{V_j} \log\left(p_{V_j}\right) \tag{1.37}$$

La valeur $NMI$ se situe dans la plage $[0, 1]$. Les valeurs proches de 1 indiquent un bon partitionnement.

## 1.4 Discussion

Dans cette section, nous avons analysé les critères d'évaluation supervisés et non supervisés. La plupart des critères sont efficaces et simples à utiliser pour évaluer un résultat de partitionnement. Dans [19], une analyse de neuf critères non supervisés parmi ceux cités dans la section précédente a été menée pour examiner leur comportement en fonction de l'évolution du paramètre de préférence $p$ de la méthode de l'AP qui sera présentée dans le chapitre suivant. Les résultats de partitionnement obtenus montrent que le critère de Levine et Nazif ($LN$) donne les résultats les plus probants à la fois pour le taux moyen de classification correcte et pour l'estimation du nombre exact de classes. Ce critère sera par conséquent retenu dans les méthodes développées présentées après.

# Chapitre 2 : Partitionnement semi-supervisé et non supervisé

## 2.1 Introduction

Le partitionnement des données ou des images est une étape très importante dans la conception des systèmes décisionnels. Son intérêt dans divers domaines, tels que la géologie [20], la médecine [21], [22], [23], [24], [25], [26], la production industrielle [27], [28] et la sécurité [29], a nécessité le développement de nombreuses méthodes. Nous avons répertorié ces méthodes en trois catégories au lieu de deux (supervisées et non supervisées), comme souvent proposé dans la littérature, où les méthodes qui nécessitent une connaissance *a priori* du nombre de classes sont répertoriées dans la même catégorie que celles qui estiment le nombre de classes.

Pour une meilleure discrimination, nous avons subdivisé la catégorie des méthodes dites non supervisées de l'état de l'art, en deux catégories (méthodes semi-supervisées et non supervisées), sur la base des connaissances *a priori* introduites par l'utilisateur. Il est important de préciser que les méthodes de chaque catégorie peuvent être hiérarchiques ou non hiérarchiques et paramétriques ou non paramétriques. Par conséquent, nous considérons trois principales catégories des méthodes de partitionnement qui sont supervisées, semi-supervisées et non supervisées :

– **Les méthodes supervisées** nécessitent des échantillons d'apprentissage pour accomplir la tâche de partitionnement. Les méthodes telles que le Maximum de Vraisemblance [30] et les Machines à Vecteur de Support [31] sont les méthodes les plus couramment utilisées dans cette catégorie.

Nous notons que les informations requises par les méthodes de cette catégorie ne sont pas toujours disponibles pour de nombreuses applications et même si elles existent elles ne sont pas toujours fiables [32], [33].

– **Les méthodes semi-supervisées**, considérées comme non supervisées dans la littérature, nécessitent le nombre de classes et des valeurs seuils ou d'autres paramètres empiriques de l'opérateur. Le *K*-means [34] est un algorithme de base semi-supervisé, exigeant que le nombre de classes soit fixé par l'utilisateur. Le Fuzzy C-Means (FCM) [35], [36] dérivé de l'algorithme *K*-means est un autre algorithme semi-supervisé, ainsi que Linde-Buzo-Gray (LBG) [37].

La connaissance du nombre de classes requises par les méthodes semi-supervisées est difficilement accessible en pratique. Par exemple, avec de grands espaces paysagers, qu'ils soient accessibles ou non, le nombre de classes permettant une analyse fine de toutes les structures est impossible. Évidemment, nous pouvons toujours fournir un certain nombre de classes, qui peuvent être supérieures ou inférieures à la réalité, mais ces approches sont en contradiction avec les informations coûteuses fournies par les capteurs à haute précision spatiale et spectrale de plus en plus utilisés. Comme indiqué dans [32], [33] et [38], la connaissance *a priori* du nombre de classes est un processus subjectif par nature, qui exclut un jugement absolu quant à la pertinence de toute analyse de données. Par conséquent, ces méthodes ne permettent pas la découverte de nouvelles classes pertinentes. Dans ce cas, la connaissance du nombre de classes dans les méthodes semi-supervisées peut être considérée comme une contrainte et ne reflète pas souvent la réalité, comme celle des échantillons d'apprentissage utilisés dans les méthodes supervisées.

– **Les méthodes non supervisées** ne nécessitent ni le nombre de classes, ni des échantillons d'apprentissage associés ou toute autre connaissance *a priori*. Cette catégorie est appelée analyse exploratoire des données par Rui Xu et D. Wunsch [38]. Pour cette raison, elle répond à un réel besoin généré par certains domaines d'applications, comme l'environnement, où les zones à analyser sont très vastes ou difficiles d’accès. Elles fournissent des solutions précises, objectives et cohérentes qui traduisent le contenu informationnel réel des images ou des données. Cette catégorie de méthodes est plus adaptée pour explorer l'analyse de données, ou pour apprendre un nouvel objet et découvrir un nouveau phénomène.

L'une des méthodes non supervisées les plus élaborées est la propagation d'affinité (AP) [39]. Cette méthode a fait l'objet d'une attention particulière en raison de ses deux avantages principaux : *i*) elle peut être utilisée selon deux modes, à savoir, non supervisé ou semi-supervisé et *ii*) elle est insensible à l'initialisation.

En tenant compte de la brève analyse générale des types de méthodes de partitionnement, nous donnons ci-après plus de détails sur les principales méthodes semi-supervisées et non supervisées. Après une analyse approfondie de l’ensemble des méthodes, nous mettons en lumière leurs avantages et inconvénients afin de définir l’orientation des travaux de recherche de partitionnement développés dans le cadre de cette thèse.

## 2.2 Méthodes de partitionnement

Le partitionnement est un processus d'exploration de données qui aide l'utilisateur à découvrir et à comprendre la structure d'un ensemble de données. Il s'agit de partitionner les individus d'un ensemble de données en groupes distincts, de sorte que les individus d'un même groupe soient similaires, alors que les individus de groupes distincts sont dissimilaires. La qualité des résultats obtenus dépend de plusieurs critères comme la mesure de similarité et le type de la méthode utilisée (semi ou non supervisée, paramétrique ou non, etc.).

Nous présentons dans la section suivante les principales méthodes de partitionnement semi-supervisées et non supervisées.

### 2.2.1 Méthodes semi-supervisées

Dans cette sous-section, les principaux algorithmes de partitionnement semi-supervisé sont présentés et analysés.

#### *2.2.1.1 K-means*

Le *K*-means [34] est l'un des algorithmes de base semi-supervisés le plus connu, nécessitant que le nombre de classes ($N_c$) soit fixé *a priori* par l'utilisateur. Cet algorithme vise à minimiser une fonction objectif (somme de l'erreur quadratique), qui est définie comme suit :

$$f = \sum_{i=1}^{N_c} \sum_{x_i \in C_i} d^2(x_i, G_i) \tag{2.1}$$

où $x_i$ désigne les individus qui seront associés à des classes, $N_c$ est le nombre de classes désiré, $G_i$ correspond au centre de la $i^{ème}$ classe $C_i$ et $d(.,.)$ est la distance euclidienne entre deux individus.

Les étapes principales de cet algorithme sont présentées dans l'Algorithme 2.1.

**Algorithme 2.1. *K*-means**

**Entrée :**

- Tableau de données ($N$ objets $\times$ $B$ attributs) représentant l'ensemble des individus à partitionner
- Nombre de classes $N_c$

**Etape 1 :** Sélectionner aléatoirement $N_c$ individus et les considérer comme les centres initiaux des classes.
**Etape 2 :** Attribuer chaque individu à la classe qui a le centre le plus proche.

**Etape 3 :** Mettre à jour les centres des classes de façon à minimiser la fonction objectif $f$ de l'équation (2.1). Lorsque tous les individus ont été affectés, recalculer les positions des $N_c$ centres.
**Etape 4 :** Répéter les étapes 2 et 3 jusqu'à ce que les centres restent inchangés (jusqu'à la convergence de l'algorithme).

**Sortie :** Partition des données en $N_c$ classes

Le problème principal de l'algorithme *K*-means est que le résultat de partitionnement dépend fortement de l'initialisation des centres de classes qui sont sélectionnés au hasard et de leur nombre, ce qui le rend instable [38], [40], [41], [42]. De plus, le *K*-means ne converge pas vers le minimum global mais converge vers un minimum local. Pour résoudre les derniers problèmes, de nombreuses itérations de *K*-means sont nécessaires.

Plusieurs extensions de cet algorithme ont été proposées [37], [43], [44]. Ces algorithmes prennent plus de temps que le *K*-means et sont toujours affectés par le choix initial des centres de classes.

#### *2.2.1.2 Fuzzy C-Means (FCM)*

Le FCM est dérivé de l'algorithme *K*-means en ajoutant une opération de fuzzification pour résoudre des problèmes de partitionnement ambigus [35]. FCM a été développé à l'origine par Dunn [45] et généralisé par Bezdek [46]. Cet algorithme itératif attribue un degré d'appartenance à une classe pour un point de données, en fonction de la similitude du point de données avec une classe particulière par rapport à toutes les autres classes.

Le FCM consiste à affecter un objet $x_i$ à la classe $C_k$ de centre $G_k$ en minimisant la fonction objectif suivante :

$$f = \sum_{k=1}^{N_c} \sum_{i=1}^{N} (u_{ki})^m d^2(x_i, G_k) \tag{2.2}$$

avec la contrainte :

$$\sum_{i=1}^{N} u_{ki} = 1, \forall\, i \tag{2.3}$$

où $N_c$ est le nombre de classes désiré, $u_{ki}$ représente le degré d'appartenance d'un individu $x_i$ au $k^{ème}$ classe. Plus $u_{ki}$ est grand, plus l'appartenance à la classe est forte, avec $u_{ki} \in [0, 1]$. $m$ appelé

coefficient de flou, est un paramètre contrôlant le degré de flou dans la partition $(m \geq 1)$ et $G_k$ correspond au centre de la $k^{ème}$ classe floue.

La fonction objectif est minimisée lorsque des points de données proches du centroïde de leurs classes se voient attribuer des valeurs d'appartenance élevées et des valeurs d'appartenance faibles sont attribuées à des points de données éloignés du centroïde. Les centres des classes et les fonctions d'appartenance sont mis à jour par les expressions suivantes :

$$G_k = \sum_{i=1}^{N} \frac{(u_{ki})^m}{\sum_{j=1}^{N} (u_{kj})^m} x_i \tag{2.4}$$

$$u_{ki} = \left[ \sum_{j=1}^{N_c} \left( \frac{d^2(x_i, G_k)}{d^2(x_i, G_j)} \right)^{1/m-1} \right]^{-1} \tag{2.5}$$

Cet algorithme nécessite le nombre de classes comme connaissance *a priori* et il est également sensible aux centres de classes initiaux et au choix du paramètre de fuzzification.

Les quatre étapes principales de cet algorithme sont les suivantes (Algorithme 2.2) :

**Algorithme 2.2. FCM**

**Entrée :**

- Tableau de données ($N$ objets $\times$ $B$ attributs) représentant l'ensemble des individus à partitionner
- Nombre de classes $N_c$

**Etape 1 :** Initialiser la matrice d'appartenance $u_{ki}$, $1 \leq k \leq N, 1 \leq i \leq N_c$, avec des valeurs aléatoires comprises entre 0 et 1 satisfaisant la contrainte de l'équation (2.3).
**Etape 2 :** Calculer les centres de classes $G_k$ suivant l'équation (2.4).
**Etape 3 :** Mettre à jour le degré d'appartenance $u_{ki}$ suivant l'équation (2.5).
**Etape 4 :** Répéter les étapes 2 et 3 jusqu'à que l'algorithme converge, c'est-à-dire jusqu'à que la différence entre la matrice d'appartenance actuelle et la matrice d'appartenance précédente est inférieure à une valeur de tolérance spécifiée ou que le nombre d'itérations atteint la valeur maximale spécifiée par l'utilisateur.

**Sortie :** Partition des données en $N_c$ classes

### 2.2.2 Méthodes non supervisées

Dans cette sous-section nous présentons les principales méthodes non supervisées et non paramétriques. Nous rappelons que pour le partitionnement non supervisé, nous ne disposons d'aucune autre information que les données elles-mêmes sans aucune connaissance *a priori* (c'-à-d. ni le nombre de classes, ni d'échantillons d'apprentissage, ni modèles ou des valeurs de seuils).

#### *2.2.2.1 FCM non supervisée (NW-FCM)*

Une version non supervisée du FCM est proposée dans [47] pour accroitre la précision et la stabilité du FCM standard, adaptée aux problèmes de reconnaissance de formes multi-classes en haute dimension. Elle s'appuie sur deux concepts : la pondération non supervisée des centres de classes à partir de l'extraction non paramétrique d'attributs pondérés et l'extraction d'attributs par analyse discriminante. L'avantage de cette méthode réside dans ses caractéristiques non supervisée et non paramétrique, grâce auxquelles le système détermine automatiquement le nombre de classes ; de plus, elle est plus robuste que l'algorithme FCM standard. Toutefois, elle n'est pas complètement stable car, suivant la complexité des images traitées, le niveau de variabilité peut être élevé.

#### *2.2.2.2 FCM optimisée (OFCM)*

La méthode OFCM [48] est une version stable et non supervisée de l'algorithme FCM. L'originalité de cet algorithme provient de : (1) l'introduction d'une procédure incrémentale adaptative pour initialiser les centres de classes, ce qui rend l'algorithme stable et déterministe ; par conséquent, les résultats de partitionnement ne varient pas d'une exécution à l'autre et (2) l'utilisation d'un critère d'évaluation non supervisé permet d'estimer le nombre optimal de classes. Cet algorithme itératif comprend les quatre étapes principales suivantes :

**Etape 1 : *Choix de la classe à subdiviser***

Au début, la classe à diviser est l'ensemble de données $X$, alors $N_c = 1$. Lorsque $N_c > 1$, la classe la plus étendue est choisie. Pour choisir une classe candidate à subdiviser, une mesure de dispersion pour chaque classe $C_i, i = 1,2, \dots, N_c$, est calculée comme suit :

$$\text{Dispersion}(C_i) = \frac{1}{N_{C_i}} \sum_{j=1}^{N_{C_i}} d\left(G_i, x_i^j\right) \tag{2.6}$$

où $G_i$ et $N_{C_i}$ sont le centre de gravité et le nombre d'individus de la classe $C_i$ respectivement, $d(.,.)$ est la distance euclidienne et $x_i^j$ est le $j^{ème}$ individu de la classe $C_i$.

Chaque fois que $N_c > 1$, la classe la plus étendue est choisie comme classe candidate pour une division ultérieure, si le critère d'évaluation non supervisé de l'étape 4 est vérifié.

**Etape 2 :** ***Choix initiaux des centres de classes***

La classe candidate sélectionnée est subdivisée en deux sous-classes ; son centre de gravité est choisi comme premier centre de sous-classe, tandis que le second centre de sous-classe est un élément choisi au hasard dans la même classe.

**Etape 3 :** ***Partitionnement avec réglage fin du centre de classe***

L'ensemble de données est partitionné en utilisant le FCM standard. Pour rendre l'approche indépendante des centres initiaux des classes, un processus de réglage fin de suppression-insertion est utilisé.

**Etape 4 :** ***Evaluation du partitionnement intermédiaire obtenu à l'aide d'un critère non supervisé***

Cette étape finale valide ou rejette le partitionnement obtenu avec $N_c + 1$ classes. Cette étape permet d'estimer la partition finale de l'ensemble $X$ et le nombre associé de classes.

Pour valider le résultat de partitionnement $P_{N_c+1}$ de l'ensemble $X$ après la division de la classe la plus étendue, la condition suivante doit être remplie :

$$D\left(P_{N_c+1}\right) - D\left(P_{N_c}\right) > \eta \times D\left(P_{N_c}\right) \tag{2.7}$$

où $\eta$ est une valeur liée à la précision garantissant l'arrêt de l'algorithme de subdivision et $D\left(P_{N_c}\right)$ est défini comme suit :

$$D\left(P_{N_c}\right) = \frac{1 + \overline{D}\left(P_{N_c}\right) - \underline{D}\left(P_{N_c}\right)}{2} \tag{2.8}$$

où $\overline{D}(P_{N_c})$ et $\underline{D}(P_{N_c})$ sont les dispersions globales interclasse et intra-classes de la partition $P_{N_c}$, respectivement.

$\underline{D}(P_{N_c})$ est calculée à partir de la dispersion intra-classe $\underline{D}(C_i)$ de chaque classe de la partition $P_{N_c}$ :

$$\underline{D}(P_{N_c}) = \frac{1}{N_c}\sum_{i=1}^{N_c}\frac{N_{C_i}}{N}\underline{D}(C_i) \tag{2.9}$$

$\overline{D}(P_{N_c})$ est calculée en incluant la dispersion entre chaque classe et les autres classes de $P_{N_c}$ :

$$\overline{D}(P_{N_c}) = \frac{1}{N_c}\sum_{i=1}^{N_c}\frac{N_{C_i}}{N}\overline{D}(C_i) \tag{2.10}$$

Si le critère de l'équation (2.7) est satisfait, le partitionnement en $N_c + 1$ classes est validé, alors il faut passer à l'étape 1. Si le critère n'est pas satisfait, il faut passer à l'étape 1 et changer la classe à subdiviser, en choisissant la suivante la plus étendue. L'algorithme s'arrête dans le cas où aucune des classes ne satisfait au critère, c'est-à-dire si aucune classe de la partition courante n'est divisible, le nombre courant de classes est considéré comme optimal.

#### *2.2.2.3 LBG modifiée (MLBG)*

La méthode MLBG proposée par Rosenberger et Chehdi [49] est une optimisation de la méthode semi-supervisée LBG [37]. La méthode MLBG ne nécessite pas la connaissance du nombre de classes mais seulement d'un majorant de celui-ci. Elle est caractérisée par les propriétés suivantes :

- évaluation et possibilité de remise en cause d'un résultat intermédiaire,
- correction d'un résultat courant en exploitant les résultats antérieurs,
- évaluation de la solution sans connaissance de la VT.

Pour déterminer la meilleure partition de $X$, quatre étapes essentielles sont nécessaires comme le précise l'Algorithme 2.3.

**Algorithme 2.3. MLBG**

**Entrée :** Tableau de données ($N$ objets $\times$ $B$ attributs) représentant l'ensemble des individus à partitionner

**Etape 1 :** Normaliser les données.
**Etape 2 :** Choisir la classe à subdiviser : à l'état initiale ($k = 1$), la classe à subdiviser est l'ensemble $X$. A chaque itération ($k > 1$), la classe qui maximise une fonction de dispersion est partitionnée.
**Etape 3 :** Choisir le noyau de division d'une classe, composés du barycentre de la classe à partitionner et du point le plus distant du barycentre appartenant à la même classe.
**Etape 4 :** Partitionner en $k + 1$ classes par la méthode $K$-means.
**Etape 5 :** Evaluer la partition de l'ensemble $X$ suivant le critère de dispersion intra-classe de l'équation (2.7) :
**Si** le critère est satisfait
Valider la partition de $k + 1$ classes
Relancer l'étape 1 pour essayer de créer $k + 2$ classes
**Sinon** Remettre à cause le choix des noyaux et retourner à l'étape 2 en intégrant les échecs précédents.
**Fin si**
**Si** aucun noyau n'engendre une partition valide
Relancer l'étape 1
**Sinon** Arrêter
**Fin si**

**Sortie :** Partition de l'ensemble $X$ en $N_c$ classes.

Le point sensible de cette méthode réside dans l'initialisation des noyaux. En effet, le choix des noyaux des classes influe sur la convergence et la qualité du résultat.

#### *2.2.2.4 La propagation d'affinité*

L'une des méthodes non supervisées les plus élaborées est la propagation d'affinité (AP) [39]. Cette méthode a fait l'objet d'une attention particulière en raison de ses deux principaux avantages : *i*) elle peut être utilisée selon deux modes, à savoir, non supervisée ou semi-supervisée et *ii*) elle est insensible à l'initialisation. En raison de ces avantages, elle est devenue largement utilisée dans de nombreux domaines applicatifs, tels que la surveillance et la sécurité de l'environnement [50], [51], [52], [53], [54], la gestion des données multimédias et la reconnaissance de formes : catégorisation d'images [55], [56], [57], [58], reconnaissance de chiffres manuscrits [55], [59], suggestion d'interrogation visuelle [60], similitudes des ensembles de protéines [61] et analyse des données génétiques [62], [63].

#### 2.2.2.4.1 Algorithme de l'AP

Dans l'algorithme de l'AP développé par Frey et Dueck [39], deux procédures de transmission de messages, appelées responsabilité et disponibilité, sont utilisées pour échanger des messages entre les individus. Ces messages bien élaborés permettent d'identifier de manière itérative le meilleur exemplaire (ou représentant) de chaque classe autour duquel les objets vont être agrégés. La responsabilité, $r(x_i, x_k)$, est envoyée de l'individu $x_i$ à l'exemplaire candidat $x_k$ et reflète à quel point il serait approprié que l'individu $x_k$ soit l'exemplaire de l'individu $x_i$. La disponibilité, $a\ (x_i, x_k)$, est envoyée de l'exemplaire candidat $x_k$ à l'individu $x_i$ et reflète à quel point il serait approprié pour l'individu $x_i$ de choisir l'exemplaire candidat $x_k$ comme exemplaire. Pour calculer les deux critères, la matrice de similarité est utilisée comme l'opposée de la distance euclidienne au carré :

$$s(x_i, x_k) = -d_2^2(x_i, x_k), \forall\ i \neq k \tag{2.11}$$

Les éléments diagonaux $s(x_k, x_k)$ de la matrice $S$, reflétant la pertinence *a priori* du choix de l'individu $x_k$ pour servir d'exemplaire et connus sous le nom de préférence, ne sont pas calculés de la même manière que les éléments $s(x_i, x_k)$, pour $i \neq k$. Plus précisément, $s(x_k, x_k) = p$ (paramètre de préférence) est initialisé à la valeur minimale ou médiane de la matrice de similarité $S$ pour $i \neq k$ :

$$s(x_k, x_k) = p, \forall\ k \tag{2.12}$$

Pour l'ensemble $X = \{x_1, x_2, \dots, x_N\}$ de $N$ individus à partitionner, $S$, $R$ et $A$ désignent les matrices de similarité, de responsabilité et de disponibilité de taille $N \times N$, respectivement. $s(x_i, x_k), r(x_i, x_k)$ et $a(x_i, x_k)$ sont leurs éléments respectifs pour les individus $x_i$ et $x_k$. Mathématiquement, la responsabilité $r(x_i, x_k)$ et la disponibilité $a(x_i, x_k)$ sont définies comme suit :

$$r(x_i, x_k)_{i \neq k} = s(x_i, x_k) - \max_{k', k' \neq k} [s(x_i, x_{k'}) + a(x_i, x_{k'})] \tag{2.13}$$

$$r(x_k, x_k) = p - \max_{k', k' \neq k} [s(x_k, x_{k'}) + a(x_k, x_{k'})] \tag{2.14}$$

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{i', i' \neq \{i,k\}} \max[0, r(x_{i'}, x_k)]\} \tag{2.15}$$

$$a(x_k, x_k) = \sum_{k', k' \neq k} \max[0, r(x_{k'}, x_k)] \tag{2.16}$$

Pour chaque itération courante, $l$, les responsabilités et les disponibilités sont estimées comme suit :

$$\hat{r}(x_i, x_k)_l = \lambda\, \hat{r}(x_i, x_k)_{l-1} + (1 - \lambda)\, r(x_i, x_k)_l \tag{2.17}$$

$$\hat{a}(x_i, x_k)_l = \lambda\, \hat{a}(x_i, x_k)_{l-1} + (1 - \lambda)\, a(x_i, x_k)_l \tag{2.18}$$

où $\lambda$ est facteur d'amortissement ($\lambda \in ]0,1[$).

Dans ce processus itératif, les responsabilités et les disponibilités sont combinées pour identifier l'exemplaire de chaque classe à former. Le critère qui identifie l'individu $x_k$ comme exemplaire de l'individu $x_i$ est :

$$E^*(x_i) = \underset{k}{\operatorname{argmax}}\ [\hat{r}(x_i, x_k) + \hat{a}(x_i, x_k)] \tag{2.19}$$

On observe que l'utilisation de l'algorithme AP (voir Algorithme 2.4) nécessite la valeur du facteur d'amortissement $\lambda$ et le choix entre deux possibilités de la valeur du paramètre de préférence $p$ comme entrées, qui peut être fixé à la valeur minimale ou à la valeur médiane de la matrice de similarité, quel que soit l'individu. Le choix des valeurs de ces deux paramètres conditionne les résultats du partitionnement ; par conséquent, l'optimalité des résultats n'est pas toujours garantie.

**Algorithme 2.4. AP originale**

**Entrée :**

- Tableau de données ($N$ objets $\times B$ attributs) représentant l'ensemble des individus à partitionner
- **Paramètres à définir par l'utilisateur :**
  - Valeur du facteur d'amortissement $\boldsymbol{\lambda}$ ($\lambda \in ]0,1[$ )
  - Valeur du paramètre de préférence $\boldsymbol{p}$ *fixée à la valeur minimale ou médiane de la matrice de similarité S*

**Étapes préliminaires :**

— Calcul de la matrice de similarité $S$ de taille $N \times N$
$s(x_i, x_k) = -d_2^2(x_i, x_k)$, où $d_2$ est la distance euclidienne
— Identification de la valeur du paramètre de préférence $p$ selon le choix fixé
— Initialisation : $r(x_i, x_k) = 0$, $a(x_i, x_k) = 0$

**Procédure :**

1. Remplacer les éléments diagonaux de $S$ par la valeur de $p$
2. Calculer toutes les responsabilités compte tenu des disponibilités selon les équations (2.13), (2.14) et (2.17)
3. Calculer toutes les disponibilités compte tenu des responsabilités selon les équations (2.15), (2.16) et (2.18)
4. Combiner les disponibilités et les responsabilités selon l'équation (2.19) pour chaque individu $x_i$ pour identifier son exemplaire $x_k$ qui maximise $[\hat{r}(x_i, x_k) + \hat{a}(x_i, x_k)]$
5. **Si** les exemplaires ne changent pas, passer à l'étape (6)
   **Sinon** répéter étapes (2) à (4) jusqu'à convergence
   **Fin si**
6. Associer chaque individu à son exemplaire le plus proche et arrêter

**Sortie** : Partition $P$ de $N_c$ classes et exemplaire de chaque classe

---

#### 2.2.2.4.2 Modification de l'AP

Malgré son grand succès dans de nombreux domaines applicatifs en fournissant des résultats de partitionnement pertinents, l'utilisation de l'AP reste limitée aux images ou données de petite taille. Ainsi, l'algorithme AP peut être encore amélioré, d'une part, pour pouvoir l'appliquer à des ensembles de données de grande taille et d'autre part, pour obtenir de meilleurs résultats de partitionnement sans aucune connaissance *a priori*, en adaptant le facteur d'amortissement et le paramètre de préférence.

Plusieurs études ont été menées pour remédier aux inconvénients de l'AP en proposant différentes stratégies. Dans [50], une nouvelle version de l'AP est proposée en ajoutant un terme à la fonction objectif, pour identifier le nombre approprié de classes et extraire la connaissance du domaine source. Cette méthode améliore les performances de partitionnement des données cibles lorsque celles-ci sont insuffisantes. Deux autres extensions de l'AP sont proposées dans [51] et [52]. Le premier est pour la résolution d'entités multi-sources et le second est un regroupement des stations radio à distance utilisant AP pour une transmission conjointe en temps réel dans le réseau d'accès radio cloud. Dans [55], une approche de partitionnement par AP rapide est proposée, en

prenant en compte simultanément les informations de structure locale et globale contenues dans les données. Pour toutes ces extensions de l'AP, le paramètre de préférence et le facteur d'amortissement restent non adaptatifs.

Pour ajuster le paramètre de préférence, $p$, une solution est proposée dans [34]. La valeur initiale de chaque paramètre de préférence, $p_j$, est déterminée indépendamment en fonction de la distribution des données et $p_j$ est automatiquement ajustée pendant le processus en fixant deux seuils. Outre la connaissance de ces deux seuils, le problème de son application aux grands ensembles de données demeure.

Dans [64], une extension de l'AP est détaillée pour atteindre des complexités raisonnables de temps et d'espace. Cette extension utilise la propagation d'informations locales et globales et fournit des implémentations parallèles basées sur le modèle MapReduce pour accélérer le traitement dans les réseaux à grande échelle. La méthode proposée est développée pour découvrir les communautés dans les réseaux sociaux. Les expérimentations sont menées par les auteurs en définissant le paramètre de préférence et le facteur d'amortissement.

Dans un autre contexte, l'AP est utilisé comme étape de partitionnement préliminaire [65], [66]. Dans [65], Zhou et al. ont introduit l'AP pour pré-regrouper les images d'entraînement et sélectionner des images représentatives pour former des instances d'entraînement négatives. Dans [66], l'AP a été utilisé pour trouver des structures communautaires pour un algorithme évolutif multi-objectif. L'algorithme AP a été appliqué comme étape préliminaire pour accélérer la convergence et assurer la solution optimale au sens de Pareto. Une extension du modèle mono-exemplaire de l'AP à un multi-exemplaire a également été développée, nommée MEAP [59]. MEAP détermine le nombre d'exemplaires dans chaque classe associée à un super exemplaire pour identifier les sous-classes dans une catégorie. L'application de ces trois méthodes reste limitée au partitionnement de données de petite taille.

D'autres modifications de l'AP ont été apportées concernant la matrice de similarité [62], [67], [68]. Dans [62], un algorithme de classification de sous-espaces basé sur l'AP a été développé pour résoudre le problème d'initialisation des centres de classes. Les auteurs introduisent une pondération des d'attributs dans le critère de similarité de l'AP. Une nouvelle étape est ajoutée dans le processus de l'AP pour mettre à jour de manière itérative les poids d'attributs en fonction de la partition actuelle des données. L'amplitude relative des poids d'attributs est utilisée pour identifier

les sous-espaces dans lesquels les classes sont incorporées. Dans [67], une nouvelle version de l'AP utilisant une matrice de similarité adaptative a été proposée, où ses éléments montrent les probabilités de leadership entre les points de données. La matrice de similarité symétrique est transformée de manière adaptative en une matrice asymétrique. Dans [68], les paramètres liés à la communication sont introduits dans la fonction de similarité de l'AP. L'algorithme proposé utilise un mécanisme pondéré pour évaluer quantitativement l'effet sur la stabilité d'une classe lorsqu'un individu y est fusionné et améliore la formation des classes par l'AP en utilisant un critère d'évaluation. Ces méthodes ne sont pas non plus adaptées aux données de grande taille. De plus, le facteur d'amortissement et le paramètre de préférence sont fixés par l'utilisateur.

L'AP a été combiné avec d'autres méthodes pour améliorer la tâche d'apprentissage [69], [70]. Dans [69], une stratégie d'apprentissage de classification intégrée améliorée basée sur l'AP est proposée. Cette méthode combine les algorithmes de pic de densité, *K*-means et AP pour classifier les données avec différentes formes et densités. Premièrement, l'algorithme de pic de densité est utilisé pour obtenir les points centraux initiaux de l'algorithme *K*-means. Ensuite, le *K*-means est appliqué pour regrouper les données pour former plusieurs petites sous-classes sphériques. Enfin, l'AP fusionne ou divise les sous-classes formées par le *K*-means. Dans [70], un nouveau modèle de classification automatique de données textuelles volumineuses est développé. Il est basé sur des techniques inspirées de l'apprentissage actif et de la classification qui peuvent générer des informations puissantes et préparer les données pour l'apprentissage automatique avec un effort manuel minimal. Pour ces deux méthodes, le paramètre de préférence n'est pas adaptatif et le facteur d'amortissement est fixé par l'utilisateur. De plus, d'autres connaissances *a priori* sont nécessaires pour la tâche d'apprentissage.

Finalement, dans [71], Chehdi et al. ont suggéré une solution pour permettre à l'AP d'être appliqué à des images de grande taille. Cette solution se compose de deux étapes : la réduction du nombre des pixels à partitionner et l'estimation du nombre correct de classes via l'optimisation du paramètre de préférence. Pour réduire le nombre de pixels, l'image hyperspectrale est divisée en blocs et l'étape de réduction est ensuite appliquée indépendamment à l'intérieur de chaque bloc. Ensuite, l'AP est appliqué uniquement sur les pixels non dupliqués et les exemplaires des pixels dupliqués. Cependant, dans sa version actuelle, l'AP ne prend pas en compte le nombre d'individus

identiques dans le calcul du critère de disponibilité. Dans ce cas, les résultats ne sont pas les mêmes avec ou sans la présence d'individus identiques.

## 2.3 Analyse numérique de la sensibilité de l'AP

Pour mettre en évidence les inconvénients de l'AP, nous présentons dans cette sous-section des résultats numériques en utilisant deux bases de données. La première est un ensemble de données synthétique et la deuxième est une image hyperspectrale synthétique de petite taille permettant d'exécuter l'AP. L'AP a été appliqué à chaque ensemble de données en fixant la valeur du paramètre de préférence $p$ d'abord à la valeur minimale ($p_{min}$), puis à la valeur médiane ($p_{med}$) de la matrice de similarité $S$ et la valeur du paramètre λ a été fixée à 0.9.

### 2.3.1 Base de données 1

Cette base de données est composée de sept ensembles d'individus ($J_1$, $J_2$, $J_3$, $J_4$, $J_5$, $J_6$ et $J_7$), chaque ensemble étant composé de 3 classes parfaitement distinctes. $J_1$ est un ensemble de 8 individus où chacun est caractérisé par deux attributs comme illustré dans la Figure 1. Les jeux de données $J_2$, $J_3$, $J_4$, $J_5$, $J_6$ et $J_7$ sont des variantes de $J_1$ en introduisant un nombre variable d'individus identiques, comme précisé dans le Tableau 1. Les ensembles de données $J_2$, $J_3$, $J_4$, $J_5$, $J_6$ et $J_7$ sont composés respectivement de 11, 12, 14, 16, 16 et 23 individus parmi lesquels 4, 8, 10, 10, 15 et 19 sont identiques. Le nombre d'individus identiques est indiqué en gras.

Les résultats de partitionnement par l'AP de ces sept jeux de données sont présentés dans le Tableau 2. Pour chaque ensemble, le nombre de classes estimé ($N_c$), les individus constituant chaque classe et leur exemplaire sont spécifiés.

Ces résultats montrent que le nombre estimé de classes en présence d'objets dupliqués n'est pas toujours le même. Par exemple, pour les ensembles $J_3$ et $J_4$, le nombre estimé de classes n'est pas correct, et pour $J_1$, $J_2$ et $J_7$ il est correctement estimé lorsque la valeur de $p$ est la valeur minimale. Pour tous les cas où le nombre de classes est mal estimé, les partitions obtenues sont invalides car les classes $C_1$ et $C_2$ sont agrégées. Ces résultats confirment donc que l'AP dans sa version originale ne prend pas en compte la présence d'individus identiques lors du calcul des matrices de responsabilité et de disponibilité et que les résultats changent en présence ou absence des individus identiques.

| Classes | Individus | Attributs | |
|---|---|---|---|
| | | $\alpha_1$ | $\alpha_2$ |
| $C_1$ | $x_1$ | 1.25 | 2.01 |
| | $x_2$ | 1.11 | 2.07 |
| | $x_3$ | 1.19 | 2.05 |
| $C_2$ | $x_4$ | 15.80 | 7.25 |
| | $x_5$ | 15.90 | 7.50 |
| $C_3$ | $x_6$ | 24.00 | 13.00 |
| | $x_7$ | 24.12 | 13.65 |
| | $x_8$ | 24.34 | 13.82 |

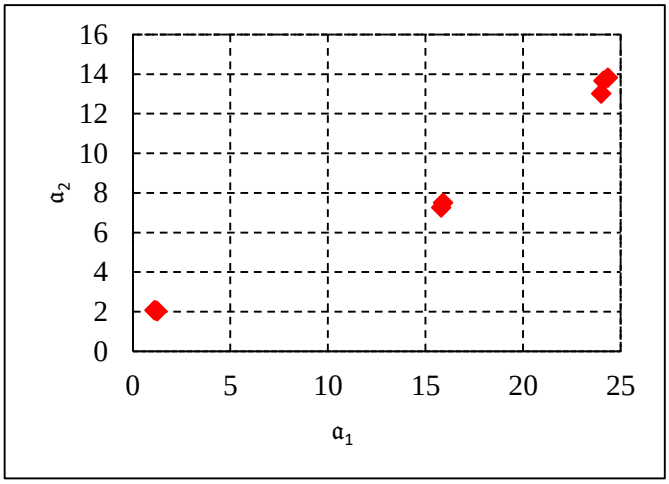



**Figure 1.** Ensemble de données ($J_1$) caractérisées par deux attributs sans la présence des individus identiques et leur représentation par rapport aux attributs $\alpha_1$ et $\alpha_2$.

**Tableau 1.** Base de données 1.

| Ensembles de données | Nombre d'individus | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|---|
| $J_1$ | 8 | $x_1, x_2, x_3$ | $x_4, x_5$ | $x_6, x_7, x_8$ |
| $J_2$ | 11 | $\boldsymbol{x_1}\times\mathbf{4}, x_2, x_3$ | $x_4, x_5$ | $x_6, x_7, x_8$ |
| $J_3$ | 12 | $\boldsymbol{x_1}\times\mathbf{2}, \boldsymbol{x_2}\times\mathbf{2}, x_3$ | $\boldsymbol{x_4}\times\mathbf{2}, x_5$ | $x_6, x_7, \boldsymbol{x_8}\times\mathbf{2}$ |
| $J_4$ | 14 | $\boldsymbol{x_1}\times\mathbf{3}, x_2\times\mathbf{2}, x_3$ | $\boldsymbol{x_4}\times\mathbf{3}, x_5$ | $x_6, x_7, \boldsymbol{x_8}\times\mathbf{2}$ |
| $J_5$ | 16 | $\boldsymbol{x_1}\times\mathbf{7}, x_2, x_3$ | $x_4, \boldsymbol{x_5}\times\mathbf{3}$ | $x_6, x_7, x_8$ |
| $J_6$ | 16 | $\boldsymbol{x_1}\times\mathbf{2}, \boldsymbol{x_2}\times\mathbf{2}, \boldsymbol{x_3}\times\mathbf{2}$ | $x_4, \boldsymbol{x_5}\times\mathbf{3}$ | $\boldsymbol{x_6}\times\mathbf{2}, \boldsymbol{x_7}\times\mathbf{2}, \boldsymbol{x_8}\times\mathbf{2}$ |
| $J_7$ | 23 | $\boldsymbol{x_1}\times\mathbf{7}, \boldsymbol{x_2}\times\mathbf{3}, x_3$ | $x_4, \boldsymbol{x_5}\times\mathbf{5}$ | $\boldsymbol{x_6}\times\mathbf{4}, x_7, x_8$ |

**Tableau 2.** Résultats de partitionnement par l'AP sur les sept ensembles de données.

| Ensembles de données | $p_{min}$ | | | $p_{med}$ | | |
|---|---|---|---|---|---|---|
| | $N_c$ | Exemplaires | Classes formées | $N_c$ | Exemplaires | Classes formées |
| $J_1$ | **3** | $x_3, x_5, x_7$ | $\{x_1, x_2, x_3\}; \{x_4, x_5\}; \{x_6, x_7, x_8\}$ | 2 | $x_2, x_7$ | $\{x_1, x_2, x_3, x_4, x_5\}; \{x_6, x_7, x_8\}$ |
| $J_2$ | **3** | $x_3, x_5, x_7$ | $\{x_1 \times 3, x_2 \times 1, x_3 \times 1\}$; $\{x_4 \times 1, x_5 \times 1\}$;$\{x_6 \times 1, x_7 \times 1, x_8 \times 1\}$ | 2 | $x_2, x_7$ | $\{x_1 \times 3, x_2 \times 1, x_3 \times 1, x_4 \times 1, x_5 \times 1\}$; ;$\{x_6 \times 1, x_7 \times 1, x_8 \times 1\}$ |
| $J_3$ | 2 | $x_2, x_7$ | $\{x_1 \times 2, x_2 \times 2, x_3 \times 1, x_4 \times 2, x_5 \times 1\}$; $\{x_6 \times 1, x_7 \times 1, x_8 \times 2\}$ | 2 | $x_2, x_7$ | $\{x_1 \times 2, x_2 \times 2, x_3 \times 1, x_4 \times 2, x_5 \times 1\}$; $\{x_6 \times 1, x_7 \times 1, x_8 \times 2\}$ |
| $J_4$ | 2 | $x_3, x_5$ | $\{x_1 \times 3, x_2 \times 2, x_3 \times 1, x_4 \times 3, x_5 \times 1\}$; $\{x_6 \times 1, x_7 \times 1, x_8 \times 2\}$ | 2 | $x_2, x_7$ | $\{x_1 \times 3, x_2 \times 2, x_3 \times 1, x_4 \times 3, x_5 \times 1\}$; $\{x_6 \times 1, x_7 \times 1, x_8 \times 2\}$ |
| $J_5$ | 2 | $x_1, x_7$ | $\{x_1 \times 7, x_2 \times 7, x_3 \times 1, x_4 \times 7, x_5 \times 3\}$; $\{x_6 \times 1, x_7 \times 1, x_8 \times 2\}$ | 2 | $x_2, x_7$ | $\{x_1 \times 7, x_2 \times 7, x_3 \times 1, x_4 \times 7, x_5 \times 3\}$; $\{x_6 \times 1, x_7 \times 1, x_8 \times 2\}$ |
| $J_6$ | 2 | $x_3, x_7$ | $\{x_1 \times 2, x_2 \times 2, x_3 \times 2, x_4 \times 1, x_5 \times 3\}$; $\{x_6 \times 3, x_7 \times 3, x_8 \times 3\}$ | **3** | $x_2, x_4, x_7$ | $\{x_1 \times 2, x_2 \times 2, x_3 \times 2\}; \{x_4 \times 1, x_5 \times 3\}$; $\{x_6 \times 3, x_7 \times 3, x_8 \times 3\}$ |
| $J_7$ | **3** | $x_3, x_5, x_7$ | $\{x_1 \times 7, x_2 \times 3, x_3 \times 1\}$; $\{x_4 \times 1, x_5 \times 5\}$;$\{x_6 \times 4, x_7 \times 1, x_8 \times 1\}$ | 2 | $x_2, x_7$ | $\{x_1 \times 7, x_2 \times 3, x_3 \times 1, x_4 \times 1, x_5 \times 5\}$; $\{x_6 \times 4, x_7 \times 1, x_8 \times 1\}$ |
| Score | 3/7 | | | 1/7 | | |

### 2.3.2 Base de données 2

Pour cette deuxième base de données, nous avons utilisé des données issues d'une image aérienne réelle hyperspectrale de taille 8 075×9 748 pixels et de 54 bandes spectrales couvrant le spectre visible/proche infrarouge [400, 970] nm (cf. Tableau 3). La résolution spatiale au sol de cette image est de 0.50 m. Cette image a été acquise dans la région de Murcia (Espagne) en octobre 2010 à l'aide du spectromètre imageur AISA Eagle de l'équipe TSI2M (Lannion) dans le cadre d'un projet en partenariat avec INFRAECO, entreprise hispano-chilienne spécialisée dans la gestion environnementale. L'image hyperspectrale que nous avons traitée ici est de petite dimension spatiale (64×64) pour mieux illustrer l'intérêt de l'utilisation de l'AP sans couvrir un espace important dans le document. Elle a été construite à partir d'un ensemble des régions pour lesquelles nous disposions d'une VT, c'est-à-dire mesures spectrales et observations. Nous avons sélectionné cinq classes parmi celles de la VT pour construire cette image. A cette image, nous avons associé une VT comme le montre la Figure 2. Les données de l'image de la Figure 2 (b) ont été prélevées aléatoirement à partir des zones 1, 2, 3, 4 et 5 de la Figure 2 (a). Ces zones correspondent respectivement aux 5 principales classes « Rivière », « *Pinus halepensis* », « Pêchers », « *Arundo donax* » et « Bâtiments » comme le montre la Figure 2 (c) et le Tableau 4. L'ensemble des données de la VT ne sert qu'à l'évaluation des performances des algorithmes développés et à leur validation.

**Tableau 3.** Numéro des bandes spectrales et longueur d'onde associée des images hyperspectrales de la Figure 2.

| Numéro des bandes | 1 | 10 | 20 | 30 | 40 | 50 | 60 |
|---|---|---|---|---|---|---|---|
| Longueur d'onde (nm) | 398.23 | 477.47 | 567.82 | 660.49 | 753.81 | 848.84 | 943.88 |

**(a)** Image hyperspectrale originale (mode RVB) (400×400 pixels)

**(b)** Image hyperspectrale test construite (RVB) (64×64 pixels)

**(c)** Image des labels des classes de la VT (64×64 pixels)

**Figure 2.** Image originale, image test construite et image VT.

**Tableau 4.** Détails des 5 classes de la VT de l'image synthétique hyperspectrale de la Figure 2.

| Labels | Classes | Nombre de pixels |
|---|---|---|
| | Rivière | 452 |
| | *Pinus halepensis* | 1068 |
| | Pêchers | 1189 |
| | *Arundo donax* | 500 |
| | Bâtiments | 887 |

Pour cette base de données, chaque pixel est caractérisé par une mesure physique donnée par le capteur hyperspectral. Pour conserver cette caractérisation, ces mesures sont directement intégrées sans transformation pour le calcul de la similarité entre pixels. L'application de l'AP pour partitionner l'image hyperspectrale donne les résultats présentés dans le Tableau 5 et la Figure 3.

Le Tableau 5 montre la forte variabilité des résultats de partitionnement en fonction du choix de la valeur du paramètre de préférence $p$. Par exemple, avec $p = p_{min}$ et $\lambda = 0.9$, ou $\lambda = 0.8$, ou $\lambda = 0.7$, le nombre estimé de classes est respectivement 12, 660 et 895. Dans le cas où $p = p_{med}$ et $\lambda = 0.9$, ou $\lambda = 0.8$, ou $\lambda = 0.7$, le nombre estimé de classes est 20, 144 et 481 respectivement. La Figure 3 (a) montre les résultats pour $\lambda = 0.9$ et les valeurs de $p = p_{min}$ et $p = p_{med}$. Dans ces cas, le nombre estimé de classes est respectivement 12 et 20. Lorsque la valeur de $p$ est fixée à la valeur médiane de $S$, le nombre estimé de classes est supérieur à celui obtenu avec une valeur de $p = p_{min}$.

De plus, le Tableau 5 confirme l'effet de la présence d'individus identiques sur le résultat du partitionnement. En fait, nous n'avons appliqué l'AP qu'à un ensemble réduit composé des pixels non identiques et des représentants des pixels identiques. Pour obtenir la partition finale, les pixels non compris dans cet ensemble réduit, sont affectés aux classes de leur exemplaire [71]. La partition finale est illustrée dans la Figure 3 (b). Dans ce cas, le nombre de classes pour $p_{min}$ et $p_{med}$ est respectivement de 7 et 13, tandis que l'AP appliqué sur l'image complète donne 12 et 20 classes respectivement.

Les résultats obtenus sur les bases de données 1 et 2 confirment la sensibilité de l'AP en fonction du choix de la valeur du paramètre $p$, du facteur d'amortissement $\lambda$ et enfin, de la présence de tous les individus à partitionner ou uniquement de la présence des objets non dupliqués et des exemplaires des dupliqués.

**Tableau 5.** Résultats d'estimation du nombre de classes ($N_c$) et le nombre d'itérations ($N_{iter}$) par l'AP suivant λ (a : application de l'AP à tous les pixels de l'image de la Figure 2, b : application de l'AP aux pixels non dupliqués et exemplaires des pixels dupliqués).

| $\lambda$ | $N_c$ | | | | $N_{iter}$ | | | |
|---|---|---|---|---|---|---|---|---|
| | $p_{min}$ | | $p_{med}$ | | $p_{min}$ | | $p_{med}$ | |
| | a | b | a | b | a | b | a | b |
| 0.5 | 3654 | 710 | 1665 | 206 | 1000 | 1000 | 1000 | 1000 |
| 0.6 | 2226 | 7 | 734 | 14 | 1000 | 1000 | 1000 | 1000 |
| 0.7 | 895 | 7 | 481 | 54 | 1000 | 1000 | 1000 | 1000 |
| 0.8 | 660 | 6 | 144 | 13 | 1000 | 1000 | 1000 | 699 |
| 0.9 | 12 | 7 | 20 | 13 | 284 | 131 | 220 | 145 |
| 0.99 | 20 | 7 | 3 | 13 | 873 | 394 | 398 | 587 |

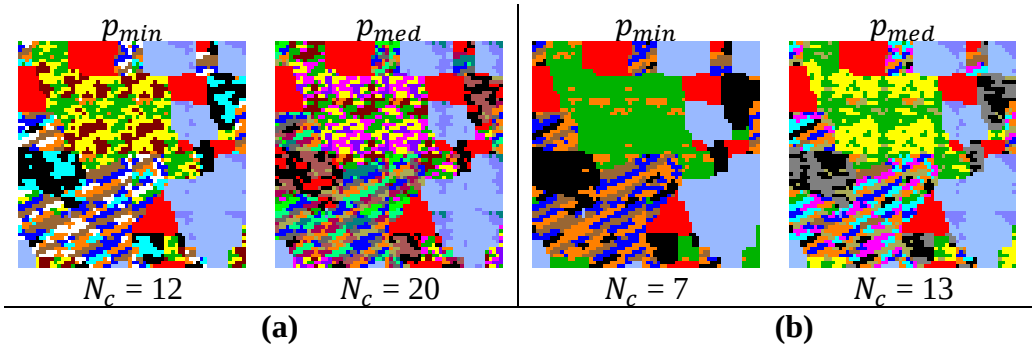



**Figure 3.** Partitionnement de l'image hyperspectrale de la Figure 2 par l'AP (λ = 0.9). (a) : avec tous les pixels de l'image ; (b) : avec les pixels non dupliqués et l'ensemble des exemplaires des pixels dupliqués.

## 2.4 Discussion

Plusieurs méthodes sont présentées dans ce chapitre qui sont semi et non supervisées, chaque méthode a ses propres avantages et inconvénients. Les méthodes semi-supervisées nécessitent la connaissance *a priori* du nombre de classes. Or cette connaissance est imprécise et pas souvent accessible. Dans ce cas, la connaissance du nombre de classes peut être considérée comme une contrainte.

L'un des algorithmes non supervisés les plus élaborés est la propagation d'affinité (AP). Cet algorithme a fait l'objet d'une attention particulière en raison de ses deux principaux avantages : il peut être utilisé suivant deux modes, non supervisé ou semi-supervisé avec des résultats stables. Malgré ses bons résultats, il est cependant sensible à la présence des individus identiques dans l'ensemble de données si on ne les prend pas en considération lors de toute simplification, aux choix des valeurs du paramètre de préférence, $p$, et du facteur d'amortissement, $\lambda$. De plus, le

critère de recherche de représentants n’est pas pertinent. Également, l'application de l’AP sur de données de grande taille, telles que des images aériennes issues d'un imageur multispectral ou hyperspectral, reste impossible en raison de sa complexité temporelle de calcul, qui a une relation quadratique avec le nombre de pixels à classer. Grâce à ses avantages et performances quant à la qualité des résultats de partitionnement et son caractère non supervisé, nous avons retenu son principe dans les méthodes de partitionnement développées. Dans la suite de cette thèse, nous apporterons des solutions pour remédier à ses inconvénients et surtout l’adapter à des données de grande taille tout en optimisant le temps de calcul et l’espace mémoire.

## Conclusion de la Partie I

Dans cette partie nous avons présenté un état de l'art portant sur les critères d'évaluation et les méthodes de partitionnement. D'après cet état de l'art nous pouvons tirer plusieurs conclusions.

Tout d'abord, le partitionnement des données de grande dimension reste un grand défi car les données sont de plus en plus volumineuses et précises. Par conséquent, toute analyse doit respecter fidèlement la nature des données et permettre l'extraction d'informations pertinentes et objectives. Il existe une grande variété de méthodes de partitionnement, mais aucune n'est efficace sur tous les types de données et nécessitent par conséquent les interventions des utilisateurs avec des connaissances et réglages non souvent maitrisés. D'où le besoin de trouver des méthodes plus évoluées, plus robustes et autonomes. Comme nous l'avons précisé dans cette partie, trois catégories de méthodes de partitionnement existent : supervisées, semi supervisées et non supervisées. Les méthodes non supervisées sont les plus appropriées pour le partitionnement des données parce qu'elles présentent plus d'avantages que les méthodes supervisées et semi supervisées. En fait, ces méthodes ne nécessitent aucune connaissance *a priori* sur les données à partitionner. Parmi les algorithmes correspondant à ces méthodes, l'AP est le plus élaboré et robuste, en revanche, il a plusieurs inconvénients qui limitent son utilisation. Il est sensible aux choix des valeurs du paramètre de préférence et du facteur d'amortissement. De plus, le critère de recherche de représentants n'est pas pertinent et enfin, son application sur des données de grande taille, telles que des images aériennes issues d'un imageur multispectral ou hyperspectral, reste impossible en raison de sa complexité temporelle de calcul, qui a une relation quadratique avec le nombre de pixels à classer.

Pour remédier aux problèmes de partitionnement, nous développons trois approches hiérarchiques, non supervisées et non paramétriques applicables sur des données de grande taille.

Dans cette partie, nous avons également traité le problème de l'évaluation des algorithmes de partitionnement. C'est une étape essentielle pour mesurer leurs performances. Plusieurs critères d'évaluation sont proposés dans la littérature que nous avons classé en deux catégories, supervisées et non supervisées. Ces critères mesurent la compacité, la séparabilité ou les deux ensembles, ou la connectivité. Nous avons retenu le critère de Levine et Nazif d'après une analyse réalisée au laboratoire dans le cadre de la thèse de Mlle SOLTANI [19]. Ce critère sera donc introduit comme

un critère d’optimisation pour l’estimation du nombre de classes dans les approches de partitionnement développées.

# Partie II

# Approches de partitionnement non supervisées et non paramétriques développées

Cette partie est composée de cinq chapitres. Le *premier chapitre* porte sur le choix de la métrique pour une méthode de partitionnement. Le *deuxième chapitre* présente les deux méthodes adaptatives et non supervisées par AP proposées pour un partitionnement hiérarchique des données de grande taille. Ensuite, le *troisième chapitre* porte sur la méthode non supervisée développée pour la sélection automatique des échantillons d'apprentissage pour les méthodes supervisées. Le *quatrième chapitre* présente la nouvelle méthode de partitionnement portant sur des nouveaux critères d'optimisation liés à la connectivité. Enfin, dans le *cinquième chapitre*, nous évaluons les trois méthodes de partitionnement développées (HUP-OAP, HUP-OAPM-RSM et HUP-DIA) sur trois domaines applicatifs différents : environnement, reconnaissance faciale avec expressions et médecine.

# Chapitre 3 : Choix du critère de similarité

## 3.1 Introduction

La notion de similarité joue un rôle important dans une méthode de partitionnement lors de l'agrégation des individus ou objets dans une classe quelle que soit sa catégorie supervisée, semi-supervisée ou non supervisée. Ainsi, les performances de toutes méthodes dépendent fortement du choix de la métrique ou du critère de similarité à utiliser. Cela explique l'abondance des travaux qui ont été menés pour prouver numériquement cette dépendance [72], [73], [74], [75], [76], [77]. Cette influence a été souvent confirmée numériquement et par comparaison des performances de plusieurs indices de similarité via les résultats de partitionnement de données obtenus avec des approches différentes. Or dans la plupart des méthodes de l'état de l'art portant sur le partitionnement, les conclusions faites sur la pertinence de certains critères de similarité, ne sont pas pris en considération à tort. Doit-on utiliser des indices de similarité qui donnent des classes d'objets homogènes respectant la mesure réelle de l'écart entre les attributs qui les représentent ou des classes biaisées suite à un mauvais choix du critère de similarité ? A titre d'exemple, dans la plupart des études dédiées au partitionnement, la distance Euclidienne ($d_2$) est le critère le plus utilisé comme critère de similarité, mais il ne correspond pas au critère le plus discriminant, si on cherche à former des classes avec rigueur tenant compte de l'information apportée par des capteurs sophistiqués et précis. Plusieurs domaines sont concernés. Citons à titre d'exemple, les capteurs aéroportés multispectraux et hyperspectraux où la richesse d'information est loin d'être exploitée, ce qui conduit à des résultats aberrants de partitionnement [33]. En effet, la caractérisation des individus (pixels) est réalisée suivant plusieurs longueurs d'ondes couvrant un large domaine spectral avec une résolution spatiale de l'ordre de 10 à 60 cm.

Les notions de similarité et de formation des classes homogènes sont des notions fondamentales lors du partitionnement des données et doivent être fortement considérées. Si dans certains domaines la confusion peut être sans conséquence grave immédiatement, dans d'autres comme le domaine médical, une fausse décision peut avoir des conséquences graves.

Dans ce chapitre, nous démontrons théoriquement et numériquement que l'utilisation systématique de la distance $d_2$ dans un processus de partitionnement est non appropriée parce qu'elle ne met pas en évidence objectivement et strictement la similarité (ou la dissimilarité) entre les individus à classer. En effet, elle augmente le degré de similarité artificiellement entre individus, ce qui biaise la formation des classes dans certain cas. Un critère de similarité pertinent doit

comparer rigoureusement les caractéristiques des individus pour pouvoir former des classes compatibles avec les mesures fournies par les capteurs. Aucune manipulation n'est admise pour s'aligner par exemple sur des données erronées d'une VT [33]. Dans le cas où les résultats ne sont pas ceux attendus par l'utilisateur car l'emploi d'un critère de similarité strict met en évidence la différence exacte entre les caractéristiques d'individus à comparer, des explications scientifiques crédibles doivent être recherchées pour mieux comprendre les phénomènes mis en évidence. L'association d'une bonne méthode de partitionnement non supervisée à un critère de similarité strict permet d'aboutir à des résultats fiables ne laissant aucune marge à la confusion et à l'approximation.

Nous prouvons dans ce chapitre que la distance $d_1$ est plus adaptée à la formation des classes homogènes d'individus de manière objective et met en évidence la similarité réelle entre individus sans transformation. Il s'agit de calculer l'erreur exacte entre les vecteurs d'attributs de deux individus et surtout lorsque le vecteur d'attributs correspond à une mesure caractérisant un phénomène physique. Des preuves numériques utilisant une image synthétique sont également présentées pour démontrer encore une fois l'intérêt que présente cette métrique. La méthode de partitionnement utilisé dans le cadre des évaluations est l'AP. Des résultats de partitionnement obtenus suite à l'association des métriques d'ordre élevées ($d_2$, $d_3$, $d_4$, $d_\infty$) et d'autres critères à l'AP sont également présentés.

## 3.2 Travaux associés

Plusieurs critères de similarité sont utilisés dans les problèmes de partitionnement pour évaluer l'homogénéité des individus à travers leurs caractéristiques. Parmi ceux-ci, nous citons les distances $d_1$ et $d_2$, mesure de corrélation, SAM et SID [78], [79], [80], [81]. Ici nous nous intéressons aux critères distances parce que dans de nombreuses études, comme nous l'avons indiqué dans l'introduction, ceux-ci donnent les résultats les plus pertinents [82], [83] pour partitionner les données et par conséquent disqualifient les autres.

Soit $X = \{x_1, x_2, \dots, x_N\}$, l'ensemble des $N$ individus à partitionner où chaque individu $x_i$ est caractérisé par un ensemble $A_i$ de $B$ attributs quantitatifs avec $A_i = (a_{i1}, a_{i2}, \dots, a_{iB})$.

**Définition 3.1 : Distance**

On appelle une distance sur l'ensemble $X$, toute application :

$$d : X \times X \longrightarrow \mathbb{R}^+$$

La distance entre deux individus $x_i$ et $x_j \in X$, notée $d(x_i, x_j)$, doit vérifier les propriétés suivantes :

1) ***Positivité*** **:** $d(x_i, x_j) \geq 0, \forall\, x_i, x_j \in X$
2) ***Symétrie*** **:** $d(x_i, x_j) = d(x_j, x_i), \forall\, x_i, x_j \in X$
3) ***Séparation*** **:** $d(x_i, x_j) = 0 \Leftrightarrow x_i = x_j, \forall\, x_i, x_j \in X$
4) ***Inégalité triangulaire*** **:** $d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j), \forall\, x_i, x_j, x_k \in X$

Soient $x_i$ et $x_j$ deux individus caractérisés respectivement par les vecteurs $(a_{i1}, a_{i2}, \dots, a_{i\mathrm{B}})$ et $(a_{j1}, a_{j2}, \dots, a_{j\mathrm{B}})$.

La distance d'ordre $q$ (distance de Minkowski [84]) entre deux individus $x_i$ et $x_j$ est définie par :

$$d_q(x_i, x_j) = \left( \sum_{k=1}^{\mathrm{B}} \left| a_{ik} - a_{jk} \right|^q \right)^{\frac{1}{q}}, q \geq 1 \tag{3.1}$$

Les distances $d_1$ et $d_2$ entre deux individus $x_i$ et $x_j$ associées respectivement aux normes $L_1$ et $L_2$ sont :

$$d_1(x_i, x_j) = \sum_{k=1}^{\mathrm{B}} \left| a_{ik} - a_{jk} \right| \tag{3.2}$$

$$d_2(x_i, x_j) = \sqrt{\sum_{k=1}^{\mathrm{B}} \left( a_{ik} - a_{jk} \right)^2} \tag{3.3}$$

La distance $d_\propto$ appelée distance de Tchebychev est une variante de la distance de Minkowski où $q = \infty$ (en prenant une limite) :

$$d_\infty(x_i, x_j) = \max_k |a_{ik} - a_{jk}| \tag{3.4}$$

La distance euclidienne ($d_2$) est la métrique la plus utilisée comme indice de similarité pour le partitionnement de données [85], alors que rien ne justifie sa pertinence pour former des classes objectivement homogènes en fonction des caractéristiques mesurées par les capteurs. Ces observations sont aussi confirmés dans [86], où l'algorithme de classification de Ward est généralisé pour être associé à la distance $d_1$ au lieu de $d_2$ afin de classer un ensemble de langues indo-européennes. Pour comparer les résultats obtenus, trois indices d'évaluation ont été associés respectivement aux distances $d_1$ et $d_2$ : Dunn, Silhouette et Connectivité. Suivant ces trois indices d'évaluation, $d_1$ donne les meilleurs résultats par rapport à ceux obtenus par $d_2$. En effet, avec $d_1$ les deux critères Dunn et Silhouette sont maximisés (0.6246 et 0.2571 respectivement) et le critère Connectivité est minimisé (16.52), alors avec $d_2$ les valeurs des trois critères sont moins pertinents par rapport à celles de $d_1$. Elles correspondent respectivement à 0.5557, 0.2129 et 17.10.

Dans [87], une autre étude a également permis l'évaluation des performances des métriques $d_1$ et $d_2$, sur la base de données ORL. Les résultats montrent que la distance $d_1$ donne le meilleur taux de reconnaissance par rapport à $d_2$ avec un apport non négligeable de 6.67% (73.33% contre 66.66%).

Une autre étude menée dans [88], évalue les performances des métriques $d_1$ et $d_2$. Cette fois-ci, elles sont exploitées directement pour comparer les histogrammes des images. En utilisant le rapport entre le nombre d'images pertinentes récupérées et le nombre total d'images dans la collection, les résultats ont montré que $d_1$ présentait un meilleur taux de précision que $d_2$ (0.6 contre 0.4).

La plupart des études évaluant les performances des indices de similarité dans un processus de partitionnement des données donnent l'avantage au critère $d_1$ non seulement par rapport à $d_2$, mais également par rapport à d'autres critères communément utilisés (corrélation, SAM, SID et cosinus).

Dans la section suivante, nous démontrons théoriquement l'objectivité du choix de la métrique $d_1$ pour la construction des matrices de similarité, avant le processus d'agrégation d'individus.

## 3.3 Analyse théorique des critères de similarité

Tenant compte de la diversité des indices de similarité, le choix du critère le plus approprié doit être pris en tenant compte à la fois de la nature des données à partitionner et de la rigueur recherchée lors de la formation des classes. Dans le cas des individus représentés par des variables quantitatives, comme le cas traité ici, il est fait généralement recours aux indices basés sur les distances car les résultats de partitionnement sont plus pertinents. Il reste néanmoins le problème du choix entre $d_1$, $d_2$, …, $d_\infty$. Pour tenir compte de la nature physique des variables représentant les individus à classer, l'indice de similarité entre deux objets doit traduire en toute objectivité la mesure de l'erreur directe entre leur vecteur d'attributs où tous les écarts entre composantes doivent être considérés.

**Définition 3.2.** Toute mesure de similarité entre individus doit quantifier objectivement et réellement leur homogénéité suivant les caractéristiques qui les représentent sans amplification ni atténuation.

**Proposition 3.1.** La distance $d_1$ donne l'écart réel entre les caractéristiques de deux individus, c'est-à-dire, elle mesure la similarité exacte entre individus représentés par leurs attributs quantitatifs (cf. équation (3.2)) sans compensation entre les valeurs positives et négatives.

**Proposition 3.2.** La distance $d_2$ donne une mesure de similarité biaisée par rapport à celle de la distance $d_1$ :

$$\left(\sum_{k=1}^{\mathrm{B}}\left(a_{ik}-a_{jk}\right)^2\right)^{\frac{1}{2}} < \sum_{k=1}^{\mathrm{B}}\left|a_{ik}-a_{jk}\right|, \text{pour } a_{ik}-a_{jk} \neq 0$$

**Preuve 3.2.** La distance $d_2(x_i, x_j)$ augmente artificiellement la similarité entre individus (diminution de la valeur de la distance).

Il faut démontrer que : $\forall\, x_i, x_j \in X, x_i \neq x_j,\ d_2(x_i, x_j) < d_1(x_i, x_j)$.

$$d_2(x_i, x_j) = \left(\sum_{k=1}^{\mathrm{B}}\left(a_{ik}-a_{jk}\right)^2\right)^{\frac{1}{2}} = \left(\left(a_{i1}-a_{j1}\right)^2 + \left(a_{i2}-a_{j2}\right)^2 + \cdots + \left(a_{i\mathrm{B}}-a_{j\mathrm{B}}\right)^2\right)^{\frac{1}{2}}$$

$$= \left(\left|a_{i1}-a_{j1}\right|^2 + \left|a_{i2}-a_{j2}\right|^2 + \cdots + \left|a_{i\mathrm{B}}-a_{j\mathrm{B}}\right|^2\right)^{\frac{1}{2}}$$

$\Longrightarrow d_2(x_i, x_j) < \left(\left|a_{i1} - a_{j1}\right|^2\right)^{\frac{1}{2}} + \left(\left|a_{i2} - a_{j2}\right|^2\right)^{\frac{1}{2}} + \cdots + \left(\left|a_{i\mathrm{B}} - a_{j\mathrm{B}}\right|^2\right)^{\frac{1}{2}}$

$\Longrightarrow d_2(x_i, x_j) < \sqrt{\left|a_{i1} - a_{j1}\right|^2} + \sqrt{\left|a_{i2} - a_{j2}\right|^2} + \cdots + \sqrt{\left|a_{i\mathrm{B}} - a_{j\mathrm{B}}\right|^2}$ (La racine carrée d'une somme est strictement inférieure à la somme des racines carrées)

$\Longrightarrow d_2(x_i, x_j) < \left|a_{i1} - a_{j1}\right| + \left|a_{i2} - a_{j2}\right| + \cdots + \left|a_{i\mathrm{B}} - a_{j\mathrm{B}}\right|$

$\Longrightarrow d_2(x_i, x_j) < \sum_{k=1}^{\mathrm{B}} \left|a_{ik} - a_{jk}\right|$

$\Longrightarrow d_2(x_i, x_j) < d_1(x_i, x_j)$

■

**Proposition 3.3.** En notant $\varepsilon_{ij}^k = \left|a_{ik} - a_{jk}\right|$, l'écart entre l'attribut $k$ des individus $x_i$ et $x_j \in X$, avec $k \in [1, B]$, la distance $d_2$ augmente artificiellement quand $\varepsilon_{ij}^k > 1$ et donne l'inverse du résultat attendu quand $\varepsilon_{ij}^k \in ]0, 1[$.

**Preuve 3.3.** Deux cas à étudier : $\varepsilon_{ij}^k \in ]0, 1[$ et $\varepsilon_{ij}^k > 1$.

- **1ère cas : $\boldsymbol{\varepsilon_{ij}^k \in ]0, 1[}$**

Si $\varepsilon_{ij}^k \in ]0, 1[$ cela signifie que deux individus $x_i$ et $x_j$ sont similaires suivant la composante $k$, alors $\left(\varepsilon_{ij}^k\right)^2 < \varepsilon_{ij}^k$, donc la distance $d_2$ les rapproche davantage. Tandis que la distance $d_1$ conserve l'écart réel $\varepsilon_{ij}^k$ existant entre les individus.

- **2ème cas : $\boldsymbol{\varepsilon_{ij}^k > 1}$**

Si $\varepsilon_{ij}^k > 1$, alors $\left(\varepsilon_{ij}^k\right)^2 > \varepsilon_{ij}^k$. Dans ce cas $\left(\varepsilon_{ij}^k\right)^2$ augmente de manière quadratique et éloigne les individus entre eux dans l'espace de représentation d'une manière artificielle, ce qui crée un déséquilibre dans la proximité des individus. Autrement dit, $d_2(x_i, x_j)$ crée une dissimilarité non objective. Tandis que la distance $d_1$ donne une mesure traduisant l'écart réel cumulé entre composantes de deux vecteurs d'attributs séparant deux individus.

■

**Proposition 3.4.** Soient $q$ et $p$ deux entiers positifs tel que $q \geq p$, alors $d_q \leq d_p$.

Les propositions 3.1, 3.2 et 3.3 indiquent que $d_1$ donne une mesure exacte de l'écart existant entre individus, c'est à dire sans amplification, ni atténuation arbitraire. Par contre l'utilisation de la distance $d_2$ diminue le degré de similarité entre les individus lorsque $\varepsilon_{ij}^k \in ]0,1[$ et l'augmente quand $\varepsilon_{ij}^k > 1$, ce qui se traduit par une diminution ou une augmentation du nombre de classes et surtout lorsque le nombre d'individus à partitionner est important.

**Exemple 3.1.** L'exemple de la Figure 4 illustre ces propositions sur un jeu de données de 6 individus caractérisés chacun par 2 variables dans le cas où $\varepsilon_{ij}^k > 1$, notées respectivement $a_1$ et $a_2$. Le Tableau 6 montre les matrices des distances $d_1$ et $d_2$. Ce tableau met en évidence que la distance $d_2$ augmente le degré de similarité entre les individus d'une manière artificielle non appropriée. Par exemple le couple $(x_1, x_3)$ donne pour $d_1(x_1, x_3)$ une valeur de 3.55 contre 2.69 pour $d_2(x_1, x_3)$ et pour le couple $(x_2, x_5)$, la valeur de $d_1(x_2, x_5)$ est de 10.1 contre 8.07 pour $d_2(x_2, x_5)$. Les différences entre les similarités données par $d_1$ et $d_2$ respectivement pour les couples $(x_1, x_3)$ et $(x_2, x_5)$ sont de 0.86 et 2.03. Ces résultats confirment donc que la distance $d_1$ contrairement à la distance $d_2$ donne les plus grandes valeurs traduisant l'erreur réelle entre les caractéristiques des individus.

| Individus | Attributs | |
|---|---|---|
| | $a_1$ | $a_2$ |
| $x_1$ | 4.3 | 5.9 |
| $x_2$ | 5.7 | 7 |
| $x_3$ | 6.75 | 4.8 |
| $x_4$ | 11.22 | 12.35 |
| $x_5$ | 13.4 | 9.4 |
| $x_6$ | 12.35 | 11.2 |

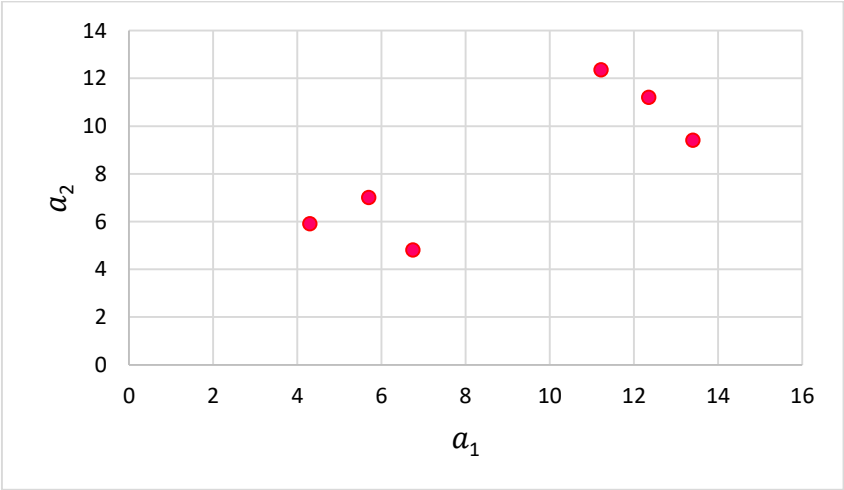



**Figure 4**. Individus caractérisés par deux attributs $a_1$ et $a_2$ ($\varepsilon_{ij}^k > 1$).

**Tableau 6.** Matrice de similarité entre individus de la Figure 4 (distances $d_1$ et $d_2$) dans le cas où $\varepsilon_{ij}^k > 1$.

| $\boldsymbol{d_1}$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|
| $x_1$ | **0** | 2.5 | 3.55 | 13.37 | 12.6 | 13.35 |
| $x_2$ | 2.5 | **0** | 3.25 | 10.87 | 10.1 | 10.85 |
| $x_3$ | 3.55 | 3.25 | **0** | 12.02 | 11.25 | 12 |
| $x_4$ | 13.37 | 10.87 | 12.02 | **0** | 5.13 | 2.28 |
| $x_5$ | 12.6 | 10.1 | 11.25 | 5.13 | **0** | 2.85 |
| $x_6$ | 13.35 | 10.85 | 12 | 2.28 | 2.85 | **0** |

| $\boldsymbol{d_2}$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|
| $x_1$ | **0** | 1.78 | 2.69 | 9.46 | 9.75 | 9.64 |
| $x_2$ | 1.78 | **0** | 2.44 | 7.69 | 8.07 | 7.87 |
| $x_3$ | 2.69 | 2.44 | **0** | 8.77 | 8.09 | 8.50 |
| $x_4$ | 9.46 | 7.69 | 8.77 | **0** | 3.67 | 1.61 |
| $x_5$ | 9.75 | 8.07 | 8.09 | 3.67 | **0** | 2.08 |
| $x_6$ | 9.64 | 7.87 | 8.50 | 1.61 | 2.08 | **0** |

**Exemple 3.2.** Un autre exemple, comme le montre la Figure 5 illustre ces propositions dans le cas où $\varepsilon_{ij}^k \in ]0,1[$, sur un jeu de données de 6 individus caractérisés chacun par 2 variables, notées respectivement $a_1$ et $a_2$. Le Tableau 7 montre les matrices des distances $d_1$ et $d_2$. Ce tableau met en évidence que la distance $d_2$ augmente le degré de similarité entre les individus d'une manière artificielle non appropriée. A titre d'exemples pour le couple d'objets $(x_1, x_3)$; $d_1(x_1, x_3)$ donne une similarité de 0.9 contre 0.64 pour $d_2(x_1, x_3)$. Pour le couple $(x_2, x_5)$, $d_1(x_2, x_5)$ donne une similarité de 6.29 contre 4.72 pour $d_2(x_2, x_5)$. Les différences entre les similarités données par $d_1$ et $d_2$ respectivement pour les couples $(x_1, x_3)$ et $(x_2, x_5)$ sont de 0.26 et 1.57. La distance $d_1$ contrairement à la distance $d_2$ donne les plus grandes valeurs traduisant les différences cumulées entre les caractéristiques de deux individus.

En conclusion, le choix d'un indice de similarité est très important et ne doit en aucun cas transformer artificiellement dans un sens ou dans un autre, l'écart réel entre individus dans l'espace d'observation, sachant que les critères de décision ou d'optimisation des méthodes de partitionnement sont construits sur la base de cet indice.

| Individus | Attributs | |
|---|---|---|
| | $a_1$ | $a_2$ |
| $x_1$ | 4.6 | 6.9 |
| $x_2$ | 4.33 | 6.56 |
| $x_3$ | 4.2 | 6.4 |
| $x_4$ | 2.5 | 2.7 |
| $x_5$ | 2.3 | 2.3 |
| $x_6$ | 2.02 | 2.5 |

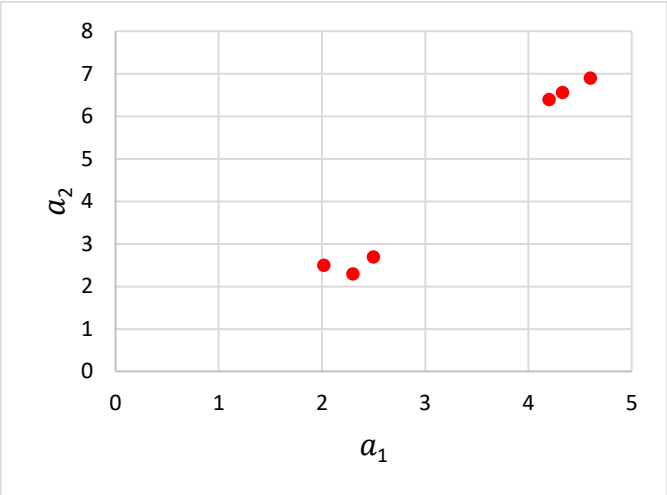



**Figure 5**. Individus caractérisés par deux attributs $a_1$ et $a_2$ ($\varepsilon_{ij}^k \in ]0,1[$).

**Tableau 7.** Matrice de similarité entre individus de la Figure 5 (distances $d_1$ et $d_2$) dans le cas où $\varepsilon_{ij}^k \in ]0,1[$.

| $\boldsymbol{d_1}$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|
| $x_1$ | **0** | 0.61 | 0.9 | 6.3 | 6.9 | 6.98 |
| $x_2$ | 0.61 | **0** | 0.29 | 5.69 | 6.29 | 6.37 |
| $x_3$ | 0.9 | 0.29 | **0** | 5.4 | 6 | 6.08 |
| $x_4$ | 6.3 | 5.69 | 5.4 | **0** | 0.6 | 0.68 |
| $x_5$ | 6.9 | 6.29 | 6 | 0.6 | **0** | 0.48 |
| $x_6$ | 6.98 | 6.37 | 6.08 | 0.68 | 0.48 | **0** |

| $\boldsymbol{d_2}$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|
| $x_1$ | **0** | 0.43 | 0.64 | 4.70 | 5.14 | 5.10 |
| $x_2$ | 0.43 | **0** | 0.21 | 4.27 | 4.72 | 4.67 |
| $x_3$ | 0.64 | 0.21 | **0** | 4.07 | 4.52 | 4.47 |
| $x_4$ | 4.70 | 4.27 | 4.07 | **0** | 0.45 | 0.52 |
| $x_5$ | 5.14 | 4.72 | 4.52 | 0.45 | **0** | 0.34 |
| $x_6$ | 5.10 | 4.67 | 4.47 | 0.52 | 0.34 | **0** |

## 3.4 Validation numérique

Pour montrer la dépendance de la qualité des résultats de partitionnement en fonction du choix de l'indice de similarité ou de la métrique, nous appliquons l'AP dans sa version originale avec différentes métriques qui sont $d_1, d_2, d_3, d_4, d_\infty$, SAM, SID et la mesure de corrélation, sur une image synthétique de petite taille présentée dans la Figure 6. La taille de cette image est limitée à 60×60 pixels (100 bandes spectrales) et qui a été générée à partir d'échantillons d'une image hyperspectrale aérienne réelle acquise par notre plateforme. Le nombre de classes et les échantillons des différentes classes sont connus pour pouvoir évaluer objectivement la pertinence des différentes métriques comparées. Les échantillons de chaque classe sont sélectionnés aléatoirement à partir des données de la VT accompagnant l'image réelle acquise. Cette image est composée de neuf classes qui peuvent être regroupées en trois grandes catégories : eau, substrat et végétation. La catégorie d'eau est composée de 3 classes ($C_1$ : eau profonde, $C_2$ : eau moins

profonde et $C_3$ : eau turbide). La catégorie des substrats est composée de 2 classes ($C_4$ : galets et $C_5$ : sable). La catégorie de végétation est subdivisée en 3 classes (2 classes d'algues vertes $C_6$ et $C_7$ : ulves et Enteromorpha respectivement, et 1 classe d'algues brunes $C_8$ : Fucus) et il y a une classe mixte de substrat et de végétation $C_9$. Le Tableau 8 donne les détails des classes de la VT. La Figure 7 présente les signatures spectrales moyennes des 9 classes et le Tableau 9 précise la longueur d'onde de chaque bande spectrale. Avec cet exemple, nous nous plaçons dans un cadre applicatif réel permettant de savoir quel est l'apport informationnel de l'imagerie hyperspectrale (acquisition coûteuse) mis en évidence via le choix d'un indice de similarité.

Image originale (mode RVB) | 9 sous-classes de la VT | 4 classes principales de la VT

**Figure 6.** Image hyperspectrale synthétique (bandes visualisés : 5, 15 et 25) et images VT.

**Tableau 8.** Détails des classes de la VT de l'image hyperspectrale synthétique de la Figure 6.

| Classes principales et labels | Labels | | Sous-classes | Nombre de pixels |
|---|---|---|---|---|
| Eau | | $C_1$ | Eau profonde | 989 |
| | | $C_2$ | Eau moins profonde | 158 |
| | | $C_3$ | Eau turbide | 177 |
| Substrat | | $C_4$ | Galet | 281 |
| | | $C_5$ | Sable | 786 |
| Végétation | | $C_6$ | Ulves (algue verte) | 255 |
| | | $C_7$ | Enteromorpha (algue verte) | 416 |
| | | $C_8$ | Fucus (algue brune) | 493 |
| Classe mélange | | $C_9$ | Substrat et autres types de végétation | 45 |

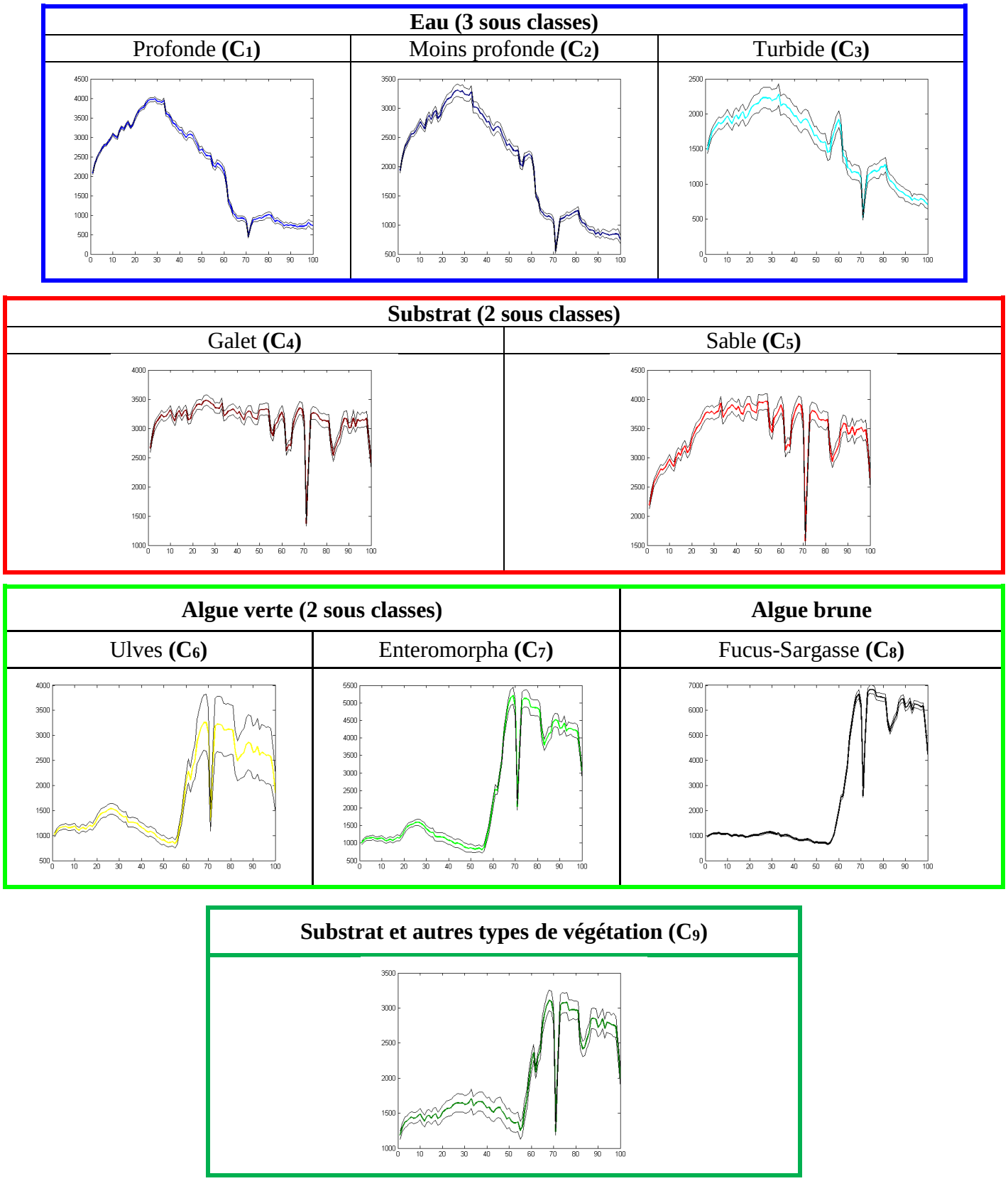



**Figure 7.** Signatures spectrales moyennes ± écart-type des 9 classes de la VT de l'image hyperspectrale synthétique de la Figure 6.

**Tableau 9.** Correspondance bande spectrale/longueur d'onde de l'image hyperspectrale synthétique de la Figure 6.

| Bande spectrale | 1 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Longueur d'onde (nm) | 439.40 | 479.71 | 524.51 | 570.14 | 616.55 | 662.79 | 709.12 | 756.19 | 803.71 | 851.21 | 898.68 |

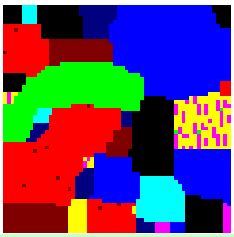
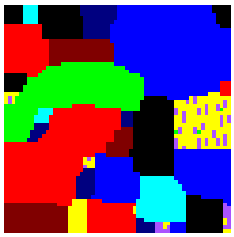
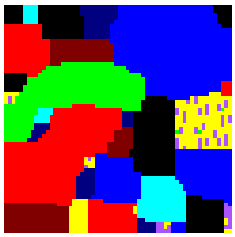
La Figure 8 et la Figure 9 montrent les résultats de partitionnement (Images partitionnées, nombres de classes $N_c$ et le CCR en %), obtenus par l'AP original avec $\lambda = 0.9$, pour la valeur du paramètre de préférence $p$ fixée à la valeur médiane ($p_{med}$) et la valeur minimale ($p_{min}$) de la matrice de similarité respectivement. La Figure 10 montre l'évolution du CCR en fonction des indices de similarité pour $p_{min}$ et $p_{med}$ respectivement. Nous remarquons que l'AP donne le meilleur CCR avec la distance $d_1$ quelle que soit la valeur de $p$ choisie (95.83% pour $p_{min}$ / 85% pour $p_{med}$). Même si le nombre estimé de classes est le même pour $d_1$ que pour $d_2$, nous notons que l'AP est moins performante avec la distance $d_2$, avec des CCR de 94.94% pour $p_{min}$ et 84.22% avec $p_{med}$. Cependant, elle donne de meilleurs résultats que ceux des autres indices. Ce qui confirme l'analyse théorique faite dans la section précédente, que la distance $d_1$ donne la mesure réelle entre les individus sans atténuation ni augmentation.

Alors la distance $d_1$ sera retenue par la suite comme critère de similarité pour les algorithmes de partitionnement non supervisés développés.

| Métriques | $\boldsymbol{d_1}$ | $d_2$ | $d_3$ | $d_4$ |
|---|---|---|---|---|
| Images Partitionnées |  |  |  |  |
| $N_c$ | **13** | 13 | 13 | 13 |
| CCR (%) | **85** | 84.22 | 81.5 | 80.79 |

| Métriques | $d_\infty$ | Corrélation | SAM | SID |
|---|---|---|---|---|
| Images Partitionnées |  |  |  |  |
| $N_c$ | 17 | 9 | 13 | 15 |
| CCR (%) | 59.72 | 79.36 | 75.67 | 71.44 |

**Figure 8.** Résultats de partitionnement de l'image hyperspectrale synthétique de la Figure 6 par l'AP avec $p_{med}$ en fonction du choix de l'indice de similarité.

| Métriques | $\boldsymbol{d_1}$ | $d_2$ | $d_3$ | $d_4$ |
|---|---|---|---|---|
| Images Partitionnées |  |  |  |  |
| $N_c$ | **9** | 9 | 9 | 9 |
| CCR (%) | **95.83** | 94.94 | 94.20 | 94.20 |

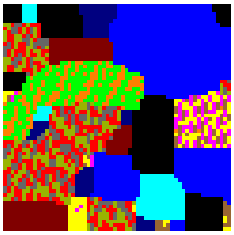
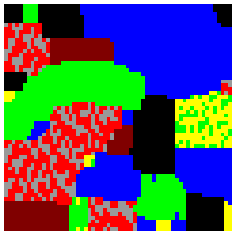
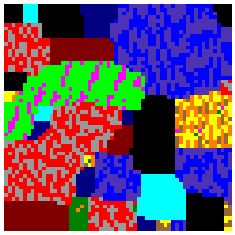
| Métriques | $d_\infty$ | Corrélation | SAM | SID |
|---|---|---|---|---|
| Images Partitionnées |  |  |  |  |
| $N_c$ | 13 | 8 | 12 | 13 |
| CCR (%) | 77.42 | 83.56 | 77 | 74.17 |

**Figure 9.** Résultats de partitionnement de l'image hyperspectrale synthétique de la Figure 6 par l'AP avec $p_{min}$ en fonction du choix de l'indice de similarité.

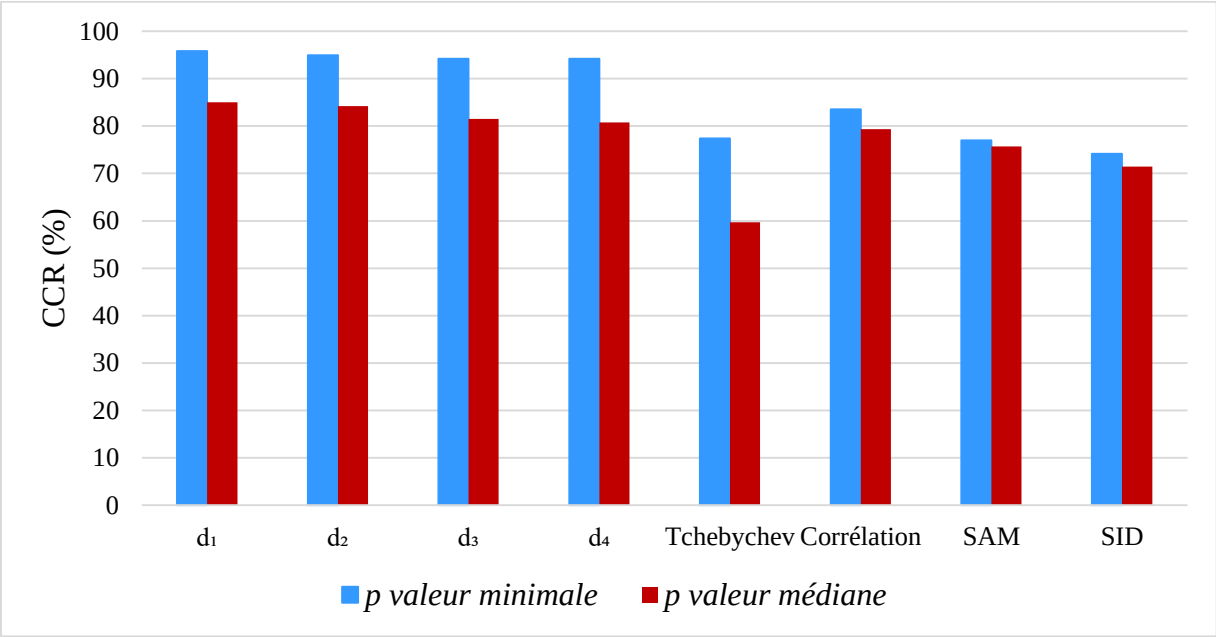



**Figure 10**. CCR de l'AP partitionnant l'image hyperspectrale synthétique de la Figure 6, en fonction des indices de similarité.

## 3.5 Discussion

Le choix du critère de similarité est fondamental dans un problème de partitionnement. En effet, si on cherche à former des classes homogènes, cette homogénéité doit être mesurée sans biais. Parmi les critères de similarités, la distance $d_2$ fait partie des critères les plus utilisés. Or, comme nous venons de le démontrer théoriquement et numériquement, celle-ci affecte la mesure de similarité. Ainsi, cette situation peut conduire à l'agrégation d'individus moins homogènes ou à leur séparation contrairement à la similarité donnée par la distance $d_1$ qui elle peut conduire à plus d'équilibre quant à la comparaison des caractéristiques des attributs d'individus. Il est à rappeler qu'en augmentant l'ordre $q$ de la distance ($d_q$) le degré de similarité entre les individus augmente si la différence pour l'ensemble des caractéristiques d'une classe est inférieure à l'unité. Par conséquent cela augmente la probabilité d'obtenir moins de classes en utilisant une distance avec $q \geq 2$ que la distance avec $q = 1$ et donne un résultat contraire lorsque la différence pour l'ensemble des caractéristiques d'une classe est supérieure à l'unité. En effet, le critère $d_1$ traduit l'erreur totale exacte entre les attributs des individus pris deux à deux et permet donc objectivement de calculer le degré de similarité entre individus sans atténuation, ni amplification, qui aboutit à

des partitions précises, objectives et stables. Ce qui permet de tenir compte de la précision des mesures que les récents capteurs sont capables de fournir aujourd'hui.

En conclusion, la distance $d_1$ utilisée comme indice de similarité (cas de données quantitatives) associée à une méthode de partitionnement non paramétrique et non supervisée conduira à la formation de classes homogènes d'une partition qui traduiront la réalité physique en nombre de classes et en taux de classification à partir de laquelle tout système décisionnel précis doit être construit. Tenant compte des analyses et résultats menés dans ce chapitre, la distance $d_1$ est choisie comme critère de similarité pour les processus de partitionnement des données.

# Chapitre 4 : Méthodes de partitionnement non supervisées et non paramétriques par propagation d'affinité pour des données de grande taille

## 4.1 Introduction

Pour partitionner un ensemble de données, nous rappelons que les méthodes de partitionnement non supervisées présentent de nombreux avantages par rapport aux méthodes supervisées et semi-supervisées : *i*) elles agrègent des individus en classes sans aucune connaissance (ni le nombre de classes à discriminer, ni les échantillons d'apprentissage). Le nombre de classes est estimé objectivement suivant un critère d'optimisation donné ou plusieurs critères d'optimisation ; et *ii*) elles fournissent des résultats plus pertinents, plus proches de la réalité physique, car les critères de décision pour l'agrégation d'objets sont indépendants des données de la VT ou des échantillons d'apprentissage, qui peuvent être biaisés ou simplifiés dans certains cas. Toutes les classes sont localisées, connues ou non, ce qui permet à l'utilisateur de mieux expliquer et analyser les phénomènes auxquels il s'intéresse.

L'état de l'art mené dans la première partie sur les méthodes de partitionnement semi-supervisées et non supervisées a montré la supériorité de l'AP par rapport aux autres méthodes de partitionnement évaluées [89], [90].

Malgré son taux de classification correcte des données, l'application de l'AP sur de données réelles de grande taille, reste impossible à cause de l'espace mémoire et du temps de calcul qu'elle nécessite. Cela rend donc l'algorithme inutilisable dans le cas d'images hyperspectrales de grande dimension spatiale. De plus, les résultats dépendent du choix du paramètre de préférence $p$ et d'un facteur d'amortissement $\lambda$. Finalement, le critère de recherche des exemplaires combinant responsabilité $R$ et disponibilité $A$ est non probant.

Pour étendre cette méthode en ne considérant que les données observées sans aucune connaissance *a priori* introduite par l'utilisateur, nous proposons deux nouvelles méthodes qui permettent le partitionnement de données de grande taille.

Les principaux apports de la première méthode portent sur :

- La prise en compte de la présence d'individus identiques dans l'ensemble des données lors du calcul des matrices de similarité, de responsabilité et de disponibilité ;
- Estimation de la valeur du paramètre de préférence de manière adaptative pour chaque individu ;
- Nouvelle procédure de mise à jour pour estimer les valeurs des critères de responsabilité et de disponibilité ;
- Modification du critère décisionnel utilisé pour identifier les exemplaires et donc estimer le nombre de classes ;
- Découpage en blocs et introduction d'une procédure de partitionnement hiérarchique pour permettre à l'utilisateur d'avoir plusieurs partitions en indiquant la partition optimale.

Quant à l'apport de la seconde méthode, il correspond à l'introduction d'une étape complémentaire à la première méthode permettant la réduction de la taille de la matrice de similarité. Nous avons reformulé le critère de disponibilité en considérant le nombre d'individus identiques. Cette étape permet également la réduction de l'espace mémoire et le temps de calcul.

Dans les sections suivantes nous détaillons les deux méthodes proposées et nous les évaluons sur des images synthétiques et réelles de grande taille, tout en faisant une étude comparative des méthodes semi supervisées et non supervisées de l'état de l'art.

## 4.2 Méthode de partitionnement hiérarchique et non supervisée par AP (HUP-OAP)

Dans cette section, nous présentons la première méthode développée, nommée HUP-OAP, qui est une optimisation de l'AP pour remédier à ses inconvénients. Dans cette méthode nous avons également introduit une procédure de hiérarchisation afin de donner à l'utilisateur la possibilité d'effectuer une analyse plus fine des données. Cette méthode sera décrite en trois étapes, la première concerne la partie optimisation, la deuxième celle du partitionnement par bloc pour pouvoir l'appliquer sur des données de grande taille et la troisième celle de la hiérarchisation.

### 4.2.1 Optimisation de l'AP

Contrairement à l'AP originale, dans la méthode optimisée de l'AP, nommée UP-OAP, tous les paramètres et critères sont calculés de manière adaptative en tenant compte de la présence

d'individus identiques dans l'ensembles de données à partitionner. Enfin, le critère d'identification d'exemplaire de chaque classe est reformulé.

#### *4.2.1.1 Paramètre de préférence*

Dans la version originale de l'AP, le choix du paramètre de préférence $p$ est basé sur la valeur minimale ou médiane de la matrice de similarité entière. Ce choix n'est pas adapté parce que certaines classes peuvent être regroupées même si elles sont éloignées ou séparées des individus alors qu'ils sont dans la même classe. Sachant que les matrices $R$ et $A$ sont non symétriques (chaque individu a ses propres responsabilités et disponibilités), le choix du paramètre $p$ pour chaque individu $x_i$ en prenant également en considération sa similarité vis-à-vis des autres individus, permet de mettre en évidence le comportement de l'individu $x_i$ par rapport aux autres éléments de sa ligne $i$. Par conséquent, le choix de la valeur de $p$ ne doit pas être le même pour tous les individus.

Pour adapter pour chaque individu $x_i$ le paramètre de préférence, $p$, et tenir compte des variations des valeurs de similarité entre les individus, nous associons à chaque individu $x_i$ (ligne $i$ de la matrice de similarité $S$, $S$ calculée sur $X$) le nouveau paramètre de préférence, $\overline{p}_i$, calculé comme suit :

$$\overline{p}_i = \frac{1}{N}\sum_{k=1}^{N} s(x_i, x_k) \tag{4.1}$$

#### *4.2.1.2 Critères de responsabilité et de disponibilité*

Comme déjà mentionné dans le chapitre 2, l'AP est sensible à la présence d'individus identiques dans l'ensemble de données $X$. En effet, l'AP traite les individus identiques comme des éléments différents et calcule leurs responsabilités et leurs disponibilités comme des individus différents.

Supposons qu'il existe des éléments identiques $x_i, x_k$ dans l'ensemble $X$, c'-à-d. $s(x_i, x_k) = 0$, alors on a :

D'après l'équation (2.13),

$$r(x_i, x_k) = s(x_i, x_k) - \max_{k', k' \neq k}\{s(x_i, x_{k'}) + a(x_i, x_{k'})\}$$

$$= -\max_{k', k' \neq k}\{s(x_i, x_{k'}) + a(x_i, x_{k'})\}$$

$\neq r(x_k, x_k)$.

D'après l'équation (2.15),

$$a(x_i, x_k) = \min[0, r(x_k, x_k) + \sum_{i', i' \neq \{i,k\}} \max[0, r(x_{i'}, x_k)]]$$

$$= \min[0, r(x_k, x_k) + \sum_{i', i' \neq k} \max[0, r(x_{i'}, x_k)] - \max[0, r(x_i, x_k)]]$$

$$= \min[0, r(x_k, x_k) + a(x_k, x_k) - \max[0, r(x_i, x_k)]]$$

$$\neq a(x_k, x_k).$$

Comme $r(x_i, x_k) \neq r(x_k, x_k)$ et $a(x_i, x_k) \neq a(x_k, x_k), \forall\, x_i = x_k, i \neq k$, ce qui montre que l'AP traite les éléments identiques comme des éléments différents. Cela signifie que l'AP dans sa version originale ne prend pas en compte la présence des individus identiques dans l'ensemble de données à partitionner et par suite ces derniers influent les résultats de partitionnement.

Pour tenir compte de la présence d'individus identiques dans l'ensemble de données, c'est-à-dire $s(x_i, x_k) = 0$, pour $i \neq k$, deux modifications sont apportées : ($i$) l'attribution de la valeur du paramètre de préférence, $\bar{p}_k$, à tous les éléments nuls de la matrice $S$, au même titre que ceux de sa diagonale et ($ii$) le calcul des éléments de $R$ et $A$, comme suit :

$$r(x_i, x_k)_{i \neq k} = r(x_k, x_k) = \overline{p}_k - \max_{k', k' \neq k} [s(x_k, x_{k'}) + a(x_k, x_{k'})] \tag{4.2}$$

$$a(x_i, x_k)_{i \neq k} = a(x_k, x_k) = \sum_{k', k' \neq k} \max[0, r(x_{k'}, x_k)] \tag{4.3}$$

Pour éviter de choisir le facteur d'amortissement λ lors de la mise à jour de $R$ et $A$ dans les équations (2.17) et (2.18) et pour éviter de donner la priorité à ceux estimés à l'itération $l - 1$ par rapport à ceux estimés à l'itération $l$, une simple opération de lissage est introduite comme suit :

$$\hat{r}(x_i, x_k)_l = [\hat{r}(x_i, x_k)_{l-3} + \hat{r}(x_i, x_k)_{l-2} + \hat{r}(x_i, x_k)_{l-1} + r(x_i, x_k)_l]/4 \tag{4.4}$$

$$\hat{a}(x_i, x_k)_l = [\hat{a}(x_i, x_k)_{l-3} + \hat{a}(x_i, x_k)_{l-2} + \hat{a}(x_i, x_k)_{l-1} + a(x_i, x_k)_l]/4 \tag{4.5}$$

Cette méthode fournit des estimations moins biaisées des matrices $R$ et $A$, contrairement à celles obtenues par l'AP originale et permet la convergence de l'algorithme.

#### *4.2.1.3 Critère de responsabilité pour l'identification des exemplaires*

Dans cette sous-section, nous démontrons que le critère de décision $E^*$ de l'équation (2.19) de l'AP originale, combinant les critères de responsabilité $R$ et de disponibilité $A$ pour identifier les exemplaires n'est pas pertinent et perturbe les résultats de partitionnement. Cela indique que la disponibilité $A$ est moins compatible avec $R$ lors de l'utilisation d'une maximisation du critère $(R + A)$. En fait, la disponibilité $A$ est dans certains cas incohérente avec la responsabilité $R$. Par conséquent, certains exemplaires ne sont pas détectés. L'utilisation du critère de décision $E^*$ peut conduire à l'agrégation d'une classe réellement existante à une autre.

**Proposition 4.1.** Soient $x_i,\ x_j \in X$ deux individus très similaires, c'est-à-dire $s(x_j, x_i) \cong 0$ et $\forall\, x_q \in X,\ s(x_j, x_i) \gg s(x_j, x_q)$, où $x_j$ et $x_q$ sont non similaires, c'est-à-dire $s(x_j, x_q) \ll 0$. Supposons que $r(x_j, x_i) > 0$ et $r(x_j, x_q) > 0$, où $x_j$ sera mieux représenté par $x_i$ que $x_q$, $x_j$ ne sera pas choisi comme exemplaire et $x_i$ n'a été choisi comme exemplaire pour aucun individu, alors $|a(x_j, x_i)| > r(x_j, x_i)$, c'est-à-dire qu'il sera agrégé avec un autre exemplaire, $x_k$, $k \neq i$ et $k \neq j$.

Pour démontrer la Proposition 4.1, nous avons besoin des lemmes suivants :

**Lemme 4.1.** ***Domaines de définition des critères R et A***

1) Pour $i \neq k$: $r(x_i, x_k) \lesseqgtr 0$ et $a(x_i, x_k) \leq 0$.
2) Pour $i = k$: $r(x_k, x_k) < 0$ et $a(x_k, x_k) \geq 0$.

**Preuve 4.1.**

1) Pour $i \neq k$ :

D'après l'équation (2.13) : $r(x_i, x_k) = s(x_i, x_k) - \max\limits_{k', k' \neq k} [s(x_i, x_{k'}) + a(x_i, x_{k'})]$.

Notons $\Gamma = S + A$ et $\tau(x_i, x_j)$ les éléments de la matrice $\Gamma$. Alors $r(x_i, x_k) = s(x_i, x_k) - \max\limits_{k', k' \neq k} [\tau(x_i, x_{k'})]$.

Supposons que $x_j$ est le plus proche voisin de $x_i$ dans Γ, c.-à-d. $\max_{k',k'\neq k}\left[\tau\left(x_i,x_{k'}\right)\right]=\tau(x_i,x_k)\Rightarrow$ $r(x_i,x_k)=s(x_i,x_k)-\tau(x_i,x_k)$.

Trois cas peuvent exister :

- $r(x_i,x_k)>0$ si $s\left(x_i,x_j\right)>\tau\left(x_i,x_j\right)$
- $r(x_i,x_k)=0$ si $s\left(x_i,x_j\right)=\tau\left(x_i,x_j\right)$
- $r(x_i,x_k)<0$ si $s\left(x_i,x_j\right)<\tau\left(x_i,x_j\right)$

D'après l'équation (2.15) : $a(x_i,x_k)=\min\{0,r(x_k,x_k)+\sum_{i',i'\neq\{i,k\}}\max[0,r(x_{i'},x_k)]\}$.

L'analyse de $a(x_i,x_k)$ met en évidence l'existence possible de trois cas :

- $r(x_k,x_k)+\sum_{i',i'\neq\{i,k\}}\max[0,r(x_{i'},x_k)]<0\ \Rightarrow a(x_i,x_k)=r(x_k,x_k)+$ $\sum_{i',i'\neq\{i,k\}}\max[0,r(x_{i'},x_k)]<0$
- $r(x_k,x_k)+\sum_{i',i'\neq\{i,k\}}\max[0,r(x_{i'},x_k)]=0\ \Rightarrow a(x_i,x_k)=0$
- $r(x_k,x_k)+\sum_{i',i'\neq\{i,k\}}\max[0,r(x_{i'},x_k)]>0\ \Rightarrow a(x_i,x_k)=0$

**2)** Pour $i=k$:

D'après l'équation (2.14) : $r(x_k,x_k)=p-\max_{k',k'\neq k}[s(x_k,x_{k'})+a(x_k,x_{k'})]=p-\tau\left(x_k,x_j\right)$.

Si $p=p_{min}$ or $p=p_{med}$, nous avons toujours $p<\tau\left(x_k,x_j\right)\Rightarrow\ r(x_k,x_k)<0$.

D'après l'équation (2.16) : $a(x_k,x_k)=\sum_{k',k'\neq k}\max[0,r(x_{k'},x_k)]\geq 0$.

■

**Lemme 4.2.** Si l'algorithme de propagation d'affinité converge, alors chaque ligne de la matrice de décision $E\ =\ R\ +\ A$ a au plus un élément positif et au moins $N-1$ éléments non positifs.

**Preuve 4.2.** Voir la preuve du Corollaire 1 référence [91].

■

**Lemme 4.3.** ***Conditions pour être un exemplaire***

**1)** Soient $x_i,x_j\in X$, si $r\left(x_j,x_i\right)>0$, alors $x_i$ est un exemplaire candidat de $x_j$.

**2)** $r(x_i, x_i) + a(x_i, x_i) \geq 0$ est une condition nécessaire et suffisante pour que $x_i$ soit un exemplaire.

**3)** Supposons que $\forall\, x_i, x_q \in X, r(x_j, x_i) > 0$ et $r(x_j, x_q) > 0$. Si $r(x_j, x_i) > r(x_j, x_q)$, alors c'est mieux pour $x_j$ d'être représenté par $x_i$ plutôt que $x_q$.

**Preuve 4.3.**

**1)** Supposons que $\forall x_i, x_j \in X, r(x_j, x_i) > 0$.

$$r(x_j, x_i) > 0 \Rightarrow s(x_j, x_i) - \max_{i', i' \neq i}[s(x_j, x_{i'}) + a(x_j, x_{i'})] > 0$$

$$\Rightarrow\ s(x_j, x_i) > \max_{i', i' \neq i}[s(x_j, x_{i'}) + a(x_j, x_{i'})]$$

$$\Rightarrow\ s(x_j, x_i) > s(x_j, x_k) + a(x_j, x_k)$$

$\Rightarrow x_j$ est mieux représenté par $x_i$ que par $x_k$, $i \neq k$

$\Rightarrow x_i$ est un exemplaire candidat.

**2)** Supposons que $\forall\, x_i \in X, r(x_i, x_i) + a(x_i, x_i) \geq 0$, alors d'après le Lemme 4.2, $\forall\, x_k \in X, r(x_i, x_k) + a(x_i, x_k) \leq 0 \Rightarrow E^*(x_i) = argmax_k[r(x_i, x_k) + a(x_i, x_k)] = x_i \Rightarrow x_i$ est un exemplaire.

**3)** Supposons que $\forall\, x_i, x_j, x_q \in X, r(x_j, x_i) > 0$ et $r(x_j, x_q) > 0$, c'-à-d., $x_i$ et $x_q$ deux exemplaires candidats de $x_j$ et $r(x_j, x_i) > r(x_j, x_q)$. Il suffit de démontrer que $s(x_j, x_i) > s(x_j, x_q)$.

Nous avons d'après l'équation (2.13) :

$$r(x_j, x_i) = s(x_j, x_i) - \max_{i', i' \neq i}[s(x_j, x_{i'}) + a(x_j, x_{i'})] = s(x_j, x_i) - \max_{i', i' \neq i}[\tau(x_j, x_{i'})]$$

$$r(x_j, x_q) = s(x_j, x_q) - \max_{q', q' \neq i}[s(x_j, x_{q'}) + a(x_j, x_{q'})] = s(x_j, x_q) - \max_{q', q' \neq q}[\tau(x_j, x_{q'})]$$

- Si $\max_{i', i' \neq i}[\tau(x_j, x_{i'})] = \max_{q', q' \neq q}[\tau(x_j, x_{q'})] = \tau(x_j, x_k)$ :

$$s(x_j, x_i) - \tau(x_j, x_k) > s(x_j, x_q) - \tau(x_j, x_k) \Rightarrow s(x_j, x_i) > s(x_i, x_q)$$

- Si $\max_{i',i' \neq i}[\tau(x_j, x_{i'})] = \tau(x_j, x_q)$ et $\max_{q',q' \neq q}[\tau(x_j, x_{q'})] = \tau(x_j, x_k)$, avec $x_q$ et $x_k$ le premier et le deuxième plus proche voisin de $x_j$ respectivement, c'-à-d., $\tau(x_j, x_q) > \tau(x_j, x_k)$ :

  $s(x_j, x_i) - \tau(x_j, x_q) > s(x_j, x_q) - \tau(x_j, x_k)$

  $\Rightarrow s(x_j, x_i) - s(x_i, x_q) > \tau(x_i, x_q) - \tau(x_j, x_k)$

  $\Rightarrow s(x_j, x_i) - s(x_i, x_q) > \tau(x_i, x_q) - \tau(x_j, x_k)$

  $\Rightarrow s(x_j, x_i) - s(x_i, x_q) > 0$

  $\Rightarrow s(x_j, x_i) > s(x_i, x_q)$

- Si $\max_{i',i' \neq i}[\tau(x_j, x_{i'})] = \tau(x_j, x_k)$ et $\max_{q',q' \neq q}[\tau(x_j, x_{q'})] = \tau(x_j, x_i)$, avec $x_k$ et $x_i$, le premier et le deuxième plus proche voisin de $x_j$ respectivement, c'-à-d, $\tau(x_j, x_k) > \tau(x_j, x_i)$ alors :

  $s(x_j, x_i) - \tau(x_j, x_k) > s(x_j, x_q) - \tau(x_j, x_i)$

  $\Rightarrow s(x_j, x_i) - s(x_i, x_q) > \tau(x_i, x_k) - \tau(x_j, x_i)$

  $\Rightarrow s(x_j, x_i) - s(x_i, x_q) > \tau(x_i, x_k) - \tau(x_j, x_i)$

  $\Rightarrow s(x_j, x_i) - s(x_i, x_q) > 0$

  $\Rightarrow s(x_j, x_i) > s(x_i, x_q)$.

■

**Preuve Proposition 4.1.** Soient $x_i$, $x_j$ et $x_q \in X$. Supposons que $r(x_j, x_i) > 0$ et $r(x_j, x_q) > 0$, avec $r(x_j, x_i) > r(x_j, x_q)$. De plus, supposons que $x_i$ n'a été choisi comme exemplaire pour aucun individu.

$$a(x_j, x_i) = \min\{0, r(x_i, x_i) + \sum_{i',i' \neq \{j,i\}} \max[0, r(x_{i'}, x_i)]\}$$

$\Rightarrow a(x_j, x_i) = \min\{0, r(x_i, x_i) + \sum_{i',i' \neq i} \max[0, r(x_{i'}, x_i)] - \max[0, r(x_j, x_i)]\}$

$\Rightarrow a(x_j, x_i) = \min\{0, r(x_i, x_i) + a(x_i, x_i) - \max[0, r(x_j, x_i)]\}$

Comme $r(x_j, x_i) > 0$, alors : $\max[0, r(x_j, x_i)] = r(x_j, x_i)$,

$\Rightarrow a(x_j, x_i) = \min\{0, r(x_i, x_i) + a(x_i, x_i) - r(x_j, x_i)\}$

$\Rightarrow a(x_j, x_i) \leq r(x_i, x_i) + a(x_i, x_i) - r(x_j, x_i)$

$\Rightarrow a(x_j, x_i) + r(x_j, x_i) \leq r(x_i, x_i) + a(x_i, x_i)$.

Comme $x_i$ n'était pas choisi comme exemplaire, d'après le Lemme 4.3 (2) nous avons :

$r(x_i, x_i) + a(x_i, x_i) < 0 \Rightarrow a(x_j, x_i) + r(x_j, x_i) < 0$.

Comme $a(x_j, x_i) < 0$ (Lemme 4.1) et $r(x_j, x_i) > 0$, nous avons $a(x_j, x_i) + r(x_j, x_i) < 0$, alors $|a(x_j, x_i)| > r(x_j, x_i)$.

■

La Proposition 4.1 montre que lorsque la responsabilité entre deux individus, $x_i$ et $x_j$, est positive et supérieure à toutes les autres responsabilités, mais que l'exemplaire candidat, $x_i$, n'est choisi comme exemplaire pour aucun individu, la disponibilité en valeur absolue dépasse la valeur de la responsabilité et affecte l'individu $x_j$ à une autre classe, même si ces individus ne peuvent pas être agrégés.

La présence de disponibilité dans le critère $E^*(x_i)$ pour la recherche d'exemplaires peut affecter un exemplaire à un individu même s'ils sont très dissemblables. Cela perturbe la décision finale et ne détecte pas correctement les classes réelles présentes.

Dans ces conditions, l'identification d'exemplaires en ne maximisant que $R$, contribue à la formation des classes homogènes représentatives des données observées.

Ceci conduit à modifier le critère de décision $E^*$ pour l'identification des exemplaires en n'utilisant que la responsabilité $R$ :

$$E^*(x_i) = \underset{k}{\operatorname{argmax}} \ [\hat{r}(x_i, x_k)] \tag{4.6}$$

Les étapes de la méthode UP-OAP sont présentées dans l'Algorithme 4.1.

**Algorithme 4.1. UP-OAP**

**Entrée :** Tableau de données ($N$ individus × $B$ attributs) représentant l'ensemble des individus à partitionner

**1.** Calculer la matrice de similarité $S$ de taille $N \times N$
$s(x_i, x_k) = -d_1(x_i, x_k)$, où $d_1$ est la distance associée à la norme $L_1$
**2.** Initialiser : $r(x_i, x_k) = 0$, $a(x_i, x_k) = 0$
**3.** Remplacer les éléments diagonaux de $S$ par la valeur de $\overline{p}_i$
**4.** Calculer toutes les responsabilités compte tenu des disponibilités selon les équations (2.13), (4.2) et (4.4)
**5.** Calculer toutes les disponibilités compte tenu des responsabilités selon les équations (2.15), (4.3) et (4.5)
**6.** Identifier les exemplaires $x_k$ qui maximisent $E^* = \underset{k}{\operatorname{argmax}}\ [\hat{r}(x_i, x_k)]$ (équation (4.6))
**7.** **Si** les exemplaires ne changent pas procéder à l'étape suivante (8)
**Sinon** répéter étapes (4) à (6) jusqu'à convergence
**Fin si**
**8.** Agréger chaque individu à son exemplaire le plus proche et arrêter

**Sorties** : Partition $P$ de $K$ classes et exemplaire $I_j$ de chaque classe $C_j$

### 4.2.2 Partitionnement des données de grande taille par bloc et hiérarchisation

Dans cette section, nous détaillons les principales étapes de la méthode de partitionnement hiérarchique non supervisée des données de grande taille basée sur l'algorithme UP-OAP. Les deux étapes principales de cette méthode HUP-OAP hiérarchique sont respectivement la formation de la première et des autres partitions, en identifiant la plus pertinente, selon un critère d'optimisation. Ces étapes sont décrites ci-dessous.

#### *4.2.2.1 Partitionnement par bloc*

L'application de la méthode UP-OAP sur des données de grande taille, comme les images aériennes hyperspectrales, nécessite un partitionnement par blocs et la fusion des résultats du partitionnement des blocs. Cette sous-section détaille les principales étapes de la formation de la première partition obtenue en appliquant d'abord l'UP-OAP sur chaque bloc, puis suivi d'un processus de fusion des classes de tous les blocs.

Pour pouvoir partitionner tous les pixels d'une image ou un ensemble de données représentées par des attributs quantitatifs, l'image ou l'ensemble de données est divisé en blocs réguliers de même taille, sans chevauchement entre les blocs. Cette procédure est détaillée dans les 3 étapes suivantes et décrite dans l'Algorithme 4.2. Elle est illustrée par le partitionnement d'une image.

Soit $Im$ l'image originale (ou données) à partitionner composée de $N$ pixels, où chaque pixel $x_i \in Im$ est caractérisé par $B$ attributs.

- **Etape 1 :** Division de l'image en blocs

L'image originale, $Im$, est divisée en $N_B$ blocs $B_{ij}$ de taille $Y_1 \times Y_2, i = \{1,2,\dots,M_1\}, j = \{1,2,\dots,M_2\}$, où $M_1$ et $M_2$ sont respectivement le nombre de ligne et de colonne des blocs ($M_1 \times M_2 = N_B$) :

$$Im = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1M_2} \\ B_{21} & B_{22} & \cdots & B_{2M_2} \\ \vdots & \vdots & \vdots & \vdots \\ B_{M_{11}} & B_{M_{12}} & \cdots & B_{M_1M_2} \end{bmatrix}$$

- **Etape 2 :** Application de la méthode UP-OAP (Algorithme 4.1) sur chaque bloc et identification des exemplaires des classes de chaque bloc $B_{ij}$

Soit $P_{ij}$ la partition du bloc $B_{ij}$, $I_{ij}$ l'ensemble des exemplaires de $N_{ij}$ classes de $B_{ij}$ et $I_{ij}^k$ l'exemplaire de la classe $C_{ij}^k, i = \{1,2,\dots,M_1\}, j = \{1,2,\dots,M_2\}$ et $k = \{1,2,\dots,N_{ij}\}$ :

$$I_{ij} = \left\{I_{ij}^1, I_{ij}^2, \dots, I_{ij}^{N_{ij}}\right\}$$

$$P_{ij} = \left\{C_{ij}^1, C_{ij}^2, \dots, C_{ij}^{N_{ij}}\right\}$$

- **Etape 3 :** Formation de la $1^{ère}$ partition

  Fusion des classes des blocs $B_{ij}$ par application de la méthode UP-OAP sur les exemplaires de blocs.

**Définition 4.1.** Le nombre d'exemplaires de pixels, $N_0$, considéré pour le partitionnement est défini par :

$$N_0 = \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} N_{ij} \tag{4.7}$$

où $N_{ij}$ est le nombre de classes obtenues par UP-OAP sur le bloc $B_{ij}$, $M_1$ est le nombre de blocs dans une ligne et $M_2$ est le nombre de blocs dans une colonne.

**Proposition 4.2.** Si $S_0$ est la matrice de similarité de taille $N_0 \times N_0$ calculée sur le nouvel ensemble de données $X_0$, de taille $N_0$ formé par les exemplaires des classes de blocs, alors on a $N_0 \ll N$.

Cette proposition montre que la taille de la matrice de similarité utilisant les exemplaires est plus petite que celle calculée sur l'ensemble de l'image originale, ce qui permet d'appliquer UP-OAP sur des images de très grande taille.

---

**Algorithme 4.2. Partitionnement par bloc**

---

**Entrée :**

- Image originale $Im$ ou Tableau de données ($N$ objets $\times$ $B$ attributs) représentant l'ensemble des individus à partitionner
- Taille des blocs maximale ($Y_1 \times Y_2$) permettant l'application de UP-OAP

**Procédure :**

**1.** Diviser l'image $Im$ ou $X$ en $N_B$ blocs $B_{ij}$, avec $i \in \{1, 2, \dots, M_1\}$ et $j \in \{1, 2, \dots, M_2\}$, où $M_1$ est le nombre des blocs en une ligne et $M_2$ est le nombre des blocs en une colonne

**2.** Appliquer UP-OAP sur chaque bloc
**Pour** $i = 1$ jusqu'à $M_1$ **faire**
**Pour** $j = 1$ jusqu'à $M_2$ **faire**
**Partitionnement du bloc $B_{ij}$ par UP-OAP**
Soit $P_{ij}$ la partition obtenue sur le bloc $B_{ij}$ :
$I_{ij} = \left\{I_{ij}^1, I_{ij}^2, \dots, I_{ij}^{N_{ij}}\right\}$ est l'ensemble des exemplaires et $N_{ij}$ est le nombre des classes de $P_{ij}$
$P_{ij} = \left\{C_{ij}^1, C_{ij}^2, \dots, C_{ij}^{N_{ij}}\right\}$
**Fin pour**
**Fin pour**

**3.** Fusionner les classes de blocs $B_{ij}$ par application de la méthode UP-OAP sur les exemplaires de blocs

**4.** Former la partition $P_1$ : fusion de chaque individu à son exemplaire

**Sortie :** Première Partition $P_1$ et $N_1$ exemplaires

---

#### *4.2.2.2 Hiérarchisation*

Cette étape permet de créer un lien hiérarchique entre les classes des partitions. Ces partitions sont générées en utilisant la méthode UP-OAP à partir d'exemplaires des classes de la première partition, $P_1$. Ainsi, pour obtenir plusieurs partitions de manière hiérarchique, la méthode UP-OAP est appliquée itérativement à chaque niveau, $i$ ($i \geq 2$), aux exemplaires de classes de la partition $P_{i-1}$ correspondant au niveau $i - 1$. Cette opération "exemplaires-partitionnement" est répétée tant

que les résultats du partitionnement ne sont pas stables. A chaque niveau, le nombre de classes de la partition est automatiquement estimé. La partition optimale et le nombre estimé de classes sont donnés par la partition qui maximise le critère de Levine et Nazif [3], noté $LN$. Cette procédure est détaillée dans l'Algorithme 4.3.

En résumé, la méthode développée ici pour partitionner les images de grande taille, nommée Partitionnement hiérarchique non supervisé par propagation d'affinité optimisée (HUP-OAP), est composée des trois algorithmes présentés précédemment : la génération de la première partition (Algorithme 4.2) en utilisant la méthode UP-OAP (Algorithme 4.1) sur l'ensemble des exemplaires des blocs et la hiérarchisation d'un partitionnement (Algorithme 4.3). La méthode HUP-OAP permet la formation de plusieurs partitions de manière hiérarchique tout en fusionnant l'ensemble des classes similaires des blocs. Ses principales étapes pour partitionner des données de grande taille sont résumées dans l'Algorithme 4.4.

**Algorithme 4.3. Hiérarchisation**

**Entrée :** Ensemble de données $X_1$ ($N_1$ exemplaires × $B$ attributs) composé des exemplaires de la première partition $P_1$ de $N_1$ classes $C_j$

1. Appliquer UP-OAP sur l'ensemble de données $X_1$
2. **Répéter** UP-OAP sur le nouvel ensemble $X_i$, $i \geq 2$
   $X_i$ est composé des exemplaires $I_{i-1}^{j}$ de chaque classe $C_j$ de la partition $P_{i-1}$, avec
   $$X_i = \{I_{i-1}^{1}, I_{i-1}^{2}, \dots, I_{i-1}^{N_{i-1}}\}$$
   Formation de la partition $P_i$ : fusionner chaque individu avec son exemplaire
   **Jusqu'à** la stabilité de la partition $P_i$
3. Choisir la partition finale optimale qui maximise le critère $LN$
   $$P_{finale} = \max_i [LN_i]$$

**Sortie :** Les partitions hiérarchiques de l'image, la partition optimale et l'ensemble d'exemplaires de ses classes

**Algorithme 4.4. HUP-OAP**

**Entrée :** Image originale $Im$ ou Tableau de données ($N$ objets $\times B$ attributs) représentant l'ensemble des individus à partitionner

1. Appliquer l'Algorithme 4.2 pour obtenir la première partition $P_1$ et ses exemplaires
2. Former l'ensemble de données $X_1$, composé des exemplaires de $P_1$
3. Appliquer l'Algorithme 4.3 sur l'ensemble des exemplaires $X_1$

**Sortie :** Les partitions hiérarchiques de l'image, la partition optimale et l'ensemble d'exemplaires de ses classes

### 4.2.3 Evaluation numérique

Dans cette section, nous présentons l'évaluation de la méthode proposée sur deux images hyperspectrales. La première est l'image hyperspectrale synthétique de petite taille déjà présentés dans la Figure 6 (page 62) et la seconde est une image réelle de grande taille ou le but applicatif est l'identification des algues marines.

#### *4.2.3.1 Partitionnement de l'image hyperspectrale synthétique*

La Figure 11 montre le résultat du partitionnement optimal (critère $LN$ : 0.25) obtenu au niveau 2 par la méthode HUP-OAP proposée (Algorithme 4.4), où le nombre estimé de classes est de 10. Pour cette évaluation, l'ensemble de 100 caractéristiques correspondant à la signature spectrale de chaque pixel est considéré et l'image est divisée en 16 blocs, où la taille de chaque bloc est de 15×15 pixels, pour couvrir l'ensemble de l'image.

La matrice de confusion du Tableau 10 met en évidence la répartition des pixels des classes de la VT dans les classes formées par la méthode HUP-OAP. La qualité du résultat du partitionnement est évaluée selon le critère CCR. Connaissant l'existence de plusieurs critères d'évaluation et la redondance de certains d'entre eux (KAPPA, CCR, etc.), nous nous sommes limités à celui du CCR qui donne une lecture simple et directe des résultats pour l'utilisateur lorsqu'on dispose d'une VT pour évaluer un algorithme. La valeur du critère optimisant une partition est également donnée à titre indicatif pour chaque partition.

Le CCR obtenu par la méthode proposée est de 96.89%. Ce taux peut être corrigé à 99.91%, si l'on considère l'homogénéité de la classe 10 formée uniquement par un sous-ensemble de pixels de la classe $C_6$ de la VT comme le montre le Tableau 10. Cet exemple illustre parfaitement l'intérêt d'une méthode non supervisée. En effet, cette classe correspond à une algue verte, mais la méthode

proposée la divise clairement en deux sous-classes. Ceci est dû au fait que lors de l'élaboration de la VT, les deux variantes de cette classe n'ont pas été précisées. Les classes 6 et 10 formées par la méthode non supervisée HUP-OAP correspondent donc aux classes d'algues vertes, mais avec des variations. L'utilisateur aura dans ce cas, le choix de conserver ces deux classes pour une analyse plus fine ou tout simplement les fusionner. Par exemple, ici la discrimination des classes intègre une information liée à la profondeur de l'eau (le spectre des algues vertes à travers la colonne d'eau). C'est donc grâce au caractère non supervisé de la méthode proposée qu'il est possible de mettre objectivement en évidence la richesse des informations fournies par l'imagerie hyperspectrale dans le proche infrarouge (NIR) par rapport à celles fournies uniquement dans le domaine visible. La classe 10 formée peut refléter, par exemple, la présence d'algues dans des eaux plus profondes que celle de la classe 6 formée.

Le Tableau 11 montre que pour l'image de la Figure 6, les résultats de partitionnement sont les mêmes quelle que soit la taille des blocs. Nous pouvons remarquer que le résultat du partitionnement de l'image sans division en blocs est identique à celui obtenu par division en blocs. Nous pouvons également constater que plus la taille des blocs est petite, plus le temps CPU et l'espace mémoire sont réduits.

Afin de comparer les performances de la méthode développée, nous avons choisi des méthodes non supervisées (AP Originale et U-OFCM) et des méthodes qui nécessitent un minimum de connaissances *a priori*, c'est-à-dire la connaissance du nombre de classes sans aucun échantillon d'apprentissage (S-OFCM, FCM et *K*-means).

Pour les méthodes semi-supervisées, le nombre de classes a été fixé à 9, afin de correspondre au nombre de classes de la VT et pour les méthodes FCM, U-OFCM et S-OFCM, le paramètre de fuzzification a été fixé à 2. Nous précisons que pour les méthodes *K*-means et FCM, les taux donnés sont la moyenne de cinq résultats fluctuants en raison de leur non-stabilité due au processus d'initialisation des classes. Pour les méthodes de l'état de l'art, nous n'avons pas modifié le critère de similarité, la métrique utilisée est la distance euclidienne ($d_2$).

Le Tableau 12 donne les performances des trois méthodes non supervisées et des trois méthodes semi-supervisées en calculant quatre critères : CCR (%), CCR-SCVT (%) en tenant compte des subdivisions fines des classes de la VT sans confusion, temps CPU (s) et espace mémoire (Mb).

Ces résultats montrent que la méthode développée donne les meilleurs résultats selon trois critères (CCR, CCR-SCVT et temps CPU), en plus de son avantage non supervisé. Par contre, elle nécessite plus d'espace mémoire que les méthodes U-OFCM, S-OFCM, FCM et *K*-means à cause des matrices de responsabilité et de disponibilité, mais beaucoup moins que la méthode AP originale. Nous pouvons également noter que les méthodes classiques semi-supervisées FCM et *K*-means donnent globalement les résultats les moins intéressants selon le critère CCR. De plus, leurs résultats ne sont pas stables d'une exécution à l'autre, malgré l'introduction du nombre de classes.

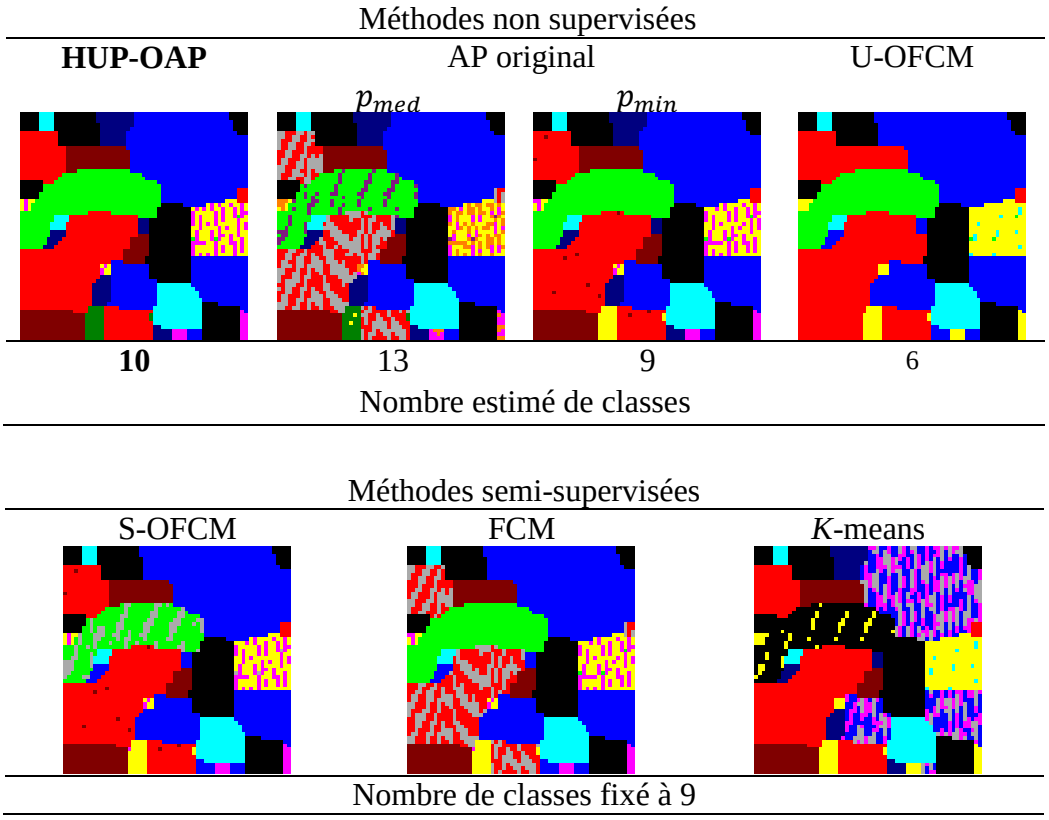



**Figure 11.** Résultats du partitionnement de l'image hyperspectrale synthétique de la Figure 6 par des méthodes non supervisées et semi-supervisées.

**Tableau 10.** Matrice de confusion du résultat du partitionnement de l'image hyperspectrale synthétique de la Figure 6 par HUP-OAP.

| | | Classes formées par HUP-OAP | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | **10** |
| Classes de la VT | $C_1$ | **989** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| | $C_2$ | 0 | **158** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| | $C_3$ | 0 | 0 | **177** | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| | $C_4$ | 0 | 0 | 0 | **281** | 0 | 0 | 0 | 0 | 0 | **0** |
| | $C_5$ | 0 | 0 | 0 | 0 | **786** | 0 | 0 | 0 | 0 | **0** |
| | $C_6$ | 0 | 0 | 0 | 0 | 0 | **143** | 3 | 0 | 0 | **109** |
| | $C_7$ | 0 | 0 | 0 | 0 | 0 | 0 | **416** | 0 | 0 | **0** |
| | $C_8$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **493** | 0 | **0** |
| | $C_9$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **45** | **0** |
| CCR | | 96.89% | | | | | | | | | |
| CCR-SCVT | | **99.91%** subdivision de la classe 6 en 2 sous classes | | | | | | | | | |

**Tableau 11**. Résultats de partitionnement par HUP-OAP de l'image hyperspectrale synthétique de la Figure 6 suivant la taille des blocs.

| Nombre de blocs | (Niveau, $N_c$) | CCR (%) | Temps CPU (s) | Espace Mémoire (Mb) |
|---|---|---|---|---|
| Image complète (60×60 pixels) | (3, **10**) | **96.89** | 49.76 | 302.86 |
| 2 (30×60 pixels) | (2, **10**) | | 20.20 | 153.54 |
| 4 (30×30 pixels) | | | 10.77 | 79.23 |
| 8 (15×30 pixels) | | | 7.08 | 41.93 |
| 16 (15×15 pixels) | | | **3.73** | **23.19** |

**Tableau 12**. Performances de la méthode développée et des cinq autres méthodes comparées sur l'image hyperspectrale synthétique de la Figure 6.

| Méthodes | Non supervisées | | | | Semi-supervisées | | |
|---|---|---|---|---|---|---|---|
| | **HUP-OAP** | AP | | U-OFCM | S-OFCM | FCM (*) | *K*-means (*) |
| | | $p_{med}$ | $p_{min}$ | | | | |
| Nombre de classes | **10 (estimé)** | 13 (estimé) | 9 (estimé) | 6 (estimé) | 9 (fixé) | 9 (fixé) | 9 (fixé) |
| CCR (%) | **96.89** | 83.17 | 94.94 | 86.14 | 86.55 | 83.07 | 72.03 |
| CCR- SCVT (%) | **99.91** | 97.83 | 98.38 | 86.14 | 93.71 | 84.80 | 86.85 |
| Temps CPU (s) | **3.73** | 61.88 | 72.01 | 35.63 | 4.46 | 64.73 | 52.82 |
| Espace Mémoire (Mb) | 23.19 | 296.93 | 296.93 | 3.82 | 3.17 | 2.99 | **2.75** |

(*) Taux moyen de 5 CCR.

#### *4.2.3.2 Partitionnement de l'image hyperspectrale réelle de grande taille*

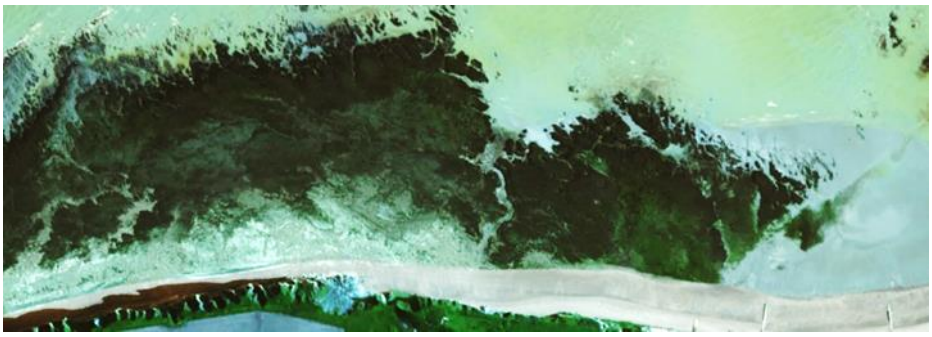
L'objectif de cette application réelle est d'identifier les algues marines et de fournir une cartographie précise du couvert végétal marin. A cette fin, deux principales espèces d'algues (vertes et brunes) doivent être discriminées.

L'image de grande taille (630 × 1800 pixels) de la Figure 12 (a) utilisée pour cette évaluation a été acquise le 27 mai 2013 (partie du littoral français) à l'aide du capteur AISA Eagles intégré à la plateforme d'acquisition aérienne disponible au Laboratoire TSI2M. La résolution spatiale au sol de cette image est de 0.6 m et le nombre de bandes spectrales est de 100, couvrant la gamme spectrale V-NIR de 404.2 nm à 978.5 nm.

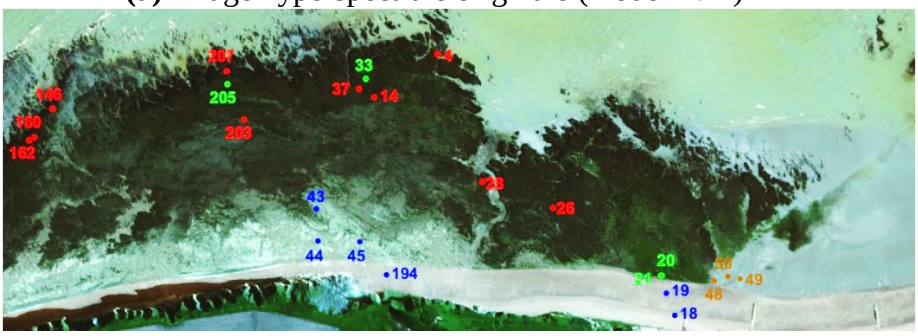
Pour permettre l'évaluation et la validation des résultats de la méthode non supervisée proposée HUP-OAP, nous utilisons des mesures de terrain réalisées en même temps que le relevé aérien. Les mesures spectrales sur le terrain ont été acquises avec un spectroradiomètre couplé à un système GPS. Après cette étape, les points de la VT sont validés [33], où seuls les points au sol ayant une signature spectrale similaire à leurs pixels correspondants dans l'image hyperspectrale aérienne originale ont été retenues.

Cet exemple illustre parfaitement qu'il est impossible en pratique d'élaborer une VT pour tous les pixels d'une image de grande taille. Pour cette raison, nous nous sommes limités à quelques points de relevé sur le terrain, afin d'évaluer et de valider la méthode de partitionnement non supervisée développée.

La Figure 12 (b) montre l'emplacement des mesures de terrain des quatre classes sur l'image hyperspectrale originale. La Figure 13 met en évidence les signatures spectrales de la VT validées et la moyenne ± écart type (en luminance) de ces quatre classes principales : Algues brunes, Algues vertes, Roches+ Galets et Sable. Ces deux dernières classes peuvent être regroupées en une seule classe de Substrat.



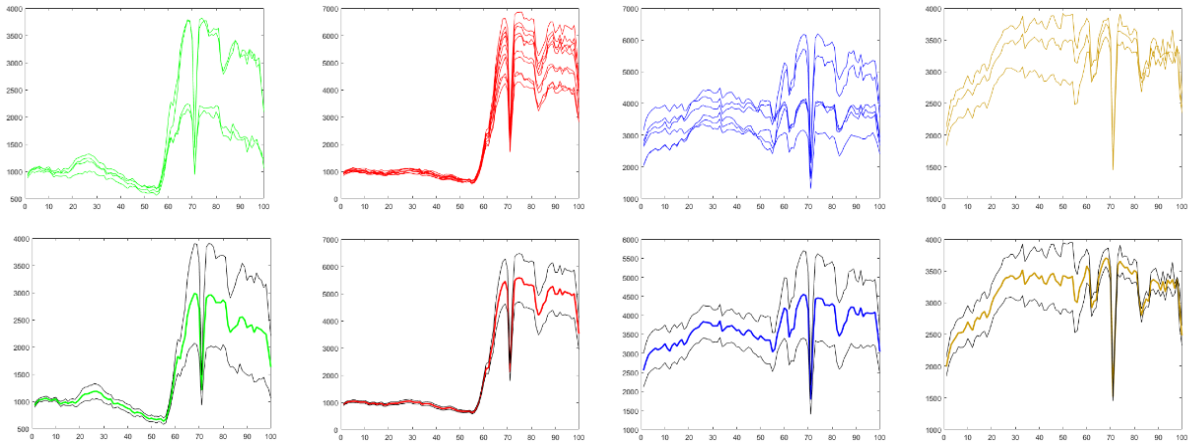
**(a)** Image hyperspectrale originale (mode RVB)




**(b)** Emplacements des points de la VT (mode RVB)

- Algues vertes (4 points : 20, 21, 33, 205)
- Algues brunes (10 points : 4, 14, 26, 28, 37, 146, 160, 162, 203, 207)
- Substrat-Sable (3 points : 48, 49, 50)
- Substrat-Roches+ Galets (6 points : 18, 19, 43, 44, 45, 194)

**Figure 12**. Image hyperspectrale de taille 630 × 1800 pixels (100 bandes) affichée en mode RVB et points de la VT.




(**a**) (**b**) (**c**) (**d**)

**Figure 13.** Signatures spectrales des points de la VT par classe de l'image de la Figure 12. Ligne 1 : (a) Algues vertes (Ulva armoricana) ; (b) Algues brunes (Fucus serratus) ; Substrat ((c) Roches+Galets et (d) Sable) et Ligne 2 : signature spectrale moyenne correspondante ± écart-type de chaque classe.

Pour partitionner cette image hyperspectrale de grande taille (630 × 1800 pixels × 100 bandes spectrales : la taille du tableau de données est de 1 134 000 pixels × 100 attributs) par la méthode HUP-OAP (Algorithme 4.4), la taille choisie de chaque bloc est de 15 × 15 pixels (5040 blocs). La Figure 14 montre le résultat du partitionnement optimal de l'image hyperspectrale de la Figure 12 (a) maximisant le critère $LN$ qui est obtenu au niveau 4. Le nombre estimé de classes pour cette partition est de 5. Le Tableau 13 donne pour chaque niveau de partitionnement le nombre estimé de classes par la méthode HUP-OAP et la valeur du critère d'optimisation $LN$.

Les points de la VT des quatre classes (Algues vertes, Algues brunes, Roches+Galets et Sable) appartiennent aux quatre classes différentes formées. Ce résultat met en évidence une cinquième classe dont la signature spectrale correspond à celle de l'eau. Nous observons sur la Figure 15 que la signature spectrale moyenne ± l'écart-type de chaque classe formée diffère des autres et peut être utilisée comme échantillons d'apprentissage de référence pour complémenter les points de la VT. La partition optimale obtenue au niveau 4 montre que le CCR est de 100%, en vérifiant les positions des 23 points des quatre classes de la VT au sein des classes formées.

La méthode ainsi développée donne, en plus du partitionnement optimal, d'autres partitions qui peuvent contribuer à l'analyse fine et à l'interprétation des données selon les besoins des utilisateurs. Il est également important de souligner que nous avons démontré par les différents tests conduits que le résultat du partitionnement est indépendant du choix de la taille des blocs.

Sur la base des signatures spectrales de référence, le taux de couverture des algues donné par la partition optimale correspond à 44.61% (19% pour les algues brunes et 25.61% pour les algues vertes), comme le montre le Tableau 14.

Le temps de calcul et l'espace mémoire pour le partitionnement de cette image avec la méthode proposée sur un processeur Intel(R) Core (TM) i7-7700 CPU à 3,6 GHz et 16 Go de mémoire sont donnés dans le Tableau 15 pour deux tailles de blocs différentes. On peut voir que le temps de calcul et l’espace mémoire diminuent avec la diminution de la taille des blocs. Le temps indicatif donné ici peut être grandement réduit car le partitionnement des blocs peut être fait en parallèle, sur une machine multiprocesseur.

En comparaison (voir Tableau 16), la méthode non supervisée développée donne de meilleures performances par rapport à la méthode OFCM [48] dans ses versions non supervisée et semi-

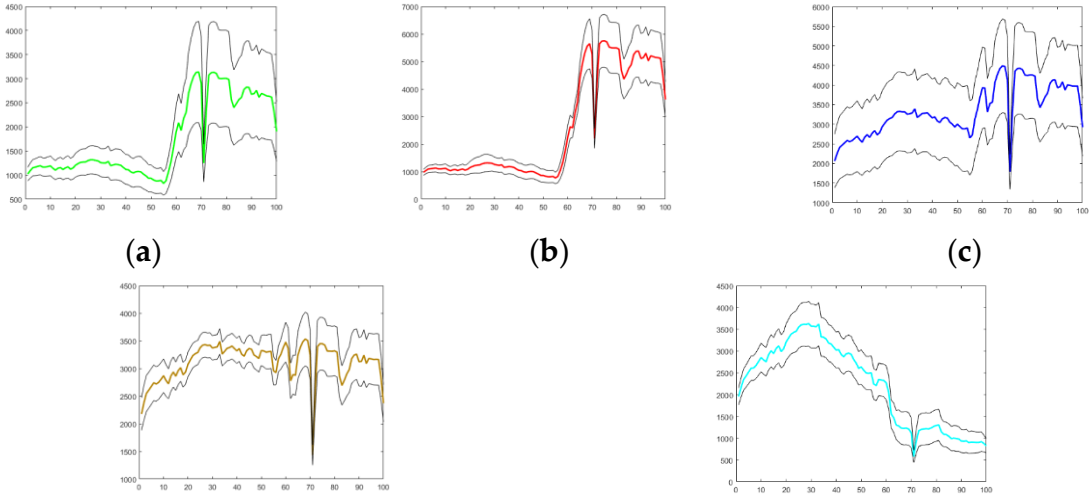
supervisée, désignées respectivement par U-OFCM et S-OFCM et par rapport aux méthodes semi-supervisées *K*-means et FCM. Le nombre de classes pour ces trois dernières méthodes a été fixé à cinq et la métrique utilisée pour calculer la matrice de similarité est la distance euclidienne ($d_2$). Nous rappelons que l'algorithme AP original ne peut pas être appliqué pour partitionner cette image de grande taille.

L'analyse du résultat de U-OFCM montre que le nombre estimé de classes correspond aux classes algues vertes, algues brunes, substrat et eau. Dans ce cas, les points de la VT appartenant aux Roches+Galets et Sable sont agrégés dans la même classe substrat. Cela signifie que les points de la VT 48, 49 et 50 de la classe Sable ne sont pas bien discriminés par rapport à ceux des points de la classe Roches+Galets, ce qui donne un taux de 86.95%. Si l'on ne tient pas compte de la discrimination entre ces deux classes de Substrat, le taux peut donc être de 100%. Cependant, ce résultat est moins précis que celui de la méthode HUP-OAP qui donne plus de détails en divisant la classe substrat en deux sous-classes. Une autre information intéressante est donnée par la partition du niveau 5, où les classes d'algues sont fusionnées dans une classe et les autres (substrats et eau) dans une autre classe. En outre, l'utilisateur peut utiliser les autres partitions de la hiérarchie pour plus de détails. Dans le cas de la méthode S-OFCM, les points de la VT 19 et 44 de la classe Roches+Galets ont été agrégés dans la classe Sable, ce qui donne un CCR de 91.30%.

**Tableau 13.** Nombre estimé de classes par la méthode HUP-OAP et valeurs du critère d'optimisation *LN* pour chaque partition de l'image hyperspectrale réelle de la Figure 12.

| Niveau | Nombre estimé de classes | Valeur du critère *LN* |
|---|---|---|
| 1 | 1162 | 0.029 |
| 2 | 181 | 0.038 |
| 3 | 33 | 0.070 |
| **4** | **5** | **0.170** |
| 5 | 2 | 0.140 |


**Figure 14.** Résultat du partitionnement optimal obtenu par la méthode HUP-OAP appliquée à l'image hyperspectrale de la Figure 12 (5 classes localisées, *LN* = 0.17 au niveau 4).



(**a**) (**b**) (**c**) (**d**) (**e**)

**Figure 15**. Signature spectrale moyenne ± écart-type de chaque classe de la Figure 14 obtenue par HUP-OAP sur l'image hyperspctrale réelle de la Figure 12 : (a) Algues vertes ; (b) Algues brunes ; (c) Roches+Galets ; (d) Sable ; (e) Eau.

**Tableau 14.** Taux de couverture de chaque classe de la Figure 14 obtenue au niveau 4 par HUP-OAP.

| Classes | Nombre de pixels | Taux de couverture (%) |
|---|---|---|
| Algue verte | 290452 | 25.61 |
| Algue Brune | 215509 | 19.00 |
| Roches+Pierres | 180802 | 15.94 |
| Sable | 170617 | 15.05 |
| Eau | 276620 | 24.39 |

**Tableau 15.** Temps CPU et espace mémoire de l'image hyperspctrale réelle partitionnée de la Figure 12 par HUP−OAP.

| Nombre de blocs | Temps CPU (s) | Espace mémoire (Mb) |
|---|---|---|
| 200 (63×90) | 6874.25 | 148850 |
| 5040 (15×15) | 3126.51 | 9472.8 |

**Tableau 16.** Performances de la méthode développée HUP-OAP, U-OFCM, S-OFCM, FCM et *K*−means sur l'image hyperspctrale réelle de la Figure 12.

| Méthodes | Nombre de classes | CCR (%) |
|---|---|---|
| **HUP-OAP** | **5 (estimé)** | **100** |
| U-OFCM | 4 (estimé) | 86.85 |
| S-OFCM | 5 (fixé) | 91.30 |
| FCM (*) | 5 (fixé) | 79.12 |
| *K*-means (*) | 5 (fixé) | 78.25 |

(*) Taux moyen de 5 CCR.

### 4.2.4 Conclusion

Dans cette section, nous avons développé une nouvelle méthode de partitionnement basée sur une optimisation de l'AP en introduisant une étape de découpage en blocs pour pouvoir partitionner des données de grande taille et une hiérarchisation pour permettre à l'utilisateur d'avoir plusieurs partitions. Les évaluations de la méthode développée sur des images hyperspectrales synthétiques et réelles montrent que les résultats sont pertinents sans aucune intervention de l'utilisateur final. Cependant, le temps de calcul et l'espace mémoire sont important, à cause des matrices de similarité, de responsabilité et de disponibilité. De plus, dans l'étape d'hiérarchisation, il n'est pas tenu compte du nombre d'individus dans chaque classe formée, ce qui influe sur le résultat de partitionnement.

Pour remédier à ces inconvénients, nous développons dans la section suivante une nouvelle méthode en effectuant des transformations appropriées pour garantir la stabilité des résultats en présence des individus dupliqués.

## 4.3 Méthode HUP-OAP avec réduction de la taille de la matrice de similarité

Dans cette section, nous présentons une nouvelle méthode basée sur le principe de celle de l'HUP-OAP décrite précédemment, en considérant toutes les configurations possibles des grands ensembles de données et en effectuant les transformations appropriées sur chaque ensemble afin de réduire la taille de la matrice de similarité (Reduced Similarity Matrix : RSM). Cette méthode nommée HUP-OAPM-RSM permet également de réduire le temps de calcul et la complexité et ainsi garantir la stabilité des résultats en présence ou en l'absence des individus identiques. L'apport de cette méthode réside dans la reformulation du critère de disponibilité. Dans cette reformulation nous tenons compte du nombre d'individus identiques ou associés dans l'ensemble de données afin de réduire la taille des matrices de similarité, de responsabilité et de disponibilité.

Il est à noter que la réduction de la taille de la matrice de similarité contribue à la réduction du temps de calcul et de l'espace mémoire.

Dans cette section nous démontrons d'abord l'importance de la reformulation du critère de disponibilité. Ensuite, nous décrivons la procédure de transformation des données permettant la réduction de la taille de la matrice de similarité dans le processus de partitionnement. Enfin, nous précisons les étapes de l'algorithme correspondant à la méthode HUP-OAP-RSM et les résultats de son évaluation.

### 4.3.1 Réformulation du critère de disponibilité

Afin de mettre en évidence l'importance de la reformulation du critère de disponibilité, nous démontrons tout d'abord que les résultats de l'AP original changent en présence ou en absence d'individus identiques comme prouvé numériquement dans la Section 2.3 du chapitre 2, c'est-à-dire que les critères d'optimisation initiaux de l'AP ne prennent pas en compte le nombre d'individus identiques s'ils ne sont représentés que par leur exemplaire dans l'ensemble de données à partitionner.

Soit chaque individu, $x_i \in X$, caractérisé par le vecteur d'attributs $A_i = (a_{i1}, a_{i2}, \ldots, a_{iB})$, où $B$ désigne le nombre d'attributs. Supposons qu'il y a des individus identiques et des individus non dupliqués dans $X$ et il peut être écrit comme suit :

$X = X_1 \cup X_2$, avec $X_1 = \{x_1, x_2, \ldots, x_{N_1}\}$ est l'ensemble de $N_1$ individus non-dupliqués, c'-à-d. $\forall\, x_i \in X_1, H(x_i) = 1$ et $X_2 = \{x_{N_1+1}, x_{N_1+2}, \ldots, x_k, \ldots, x_N\}$ est l'ensemble de $N_2$ individus dupliqués, où $H(x_j)$ est le nombre des individus identiques représentés par $x_j$, avec $H(x_j) \geq 2$.

Soit $C_l$ l'ensemble des classes des individus dupliqués de $X_2$ tel que :

$$X_2 = \bigcup_{l=1}^{N_c} C_l \text{ et } \bigcap_{l=1}^{N_c} C_l = \emptyset$$

Avec $N_c < N_2$, où $N_c$ est le nombre de classes des individus dupliqués dans $X_2$.

Supposons que chaque classe $C_l$ est représentée par un de ses éléments, qui peut être noté $z_l$ et $X_3$ est l'ensemble des exemplaires avec $X_3 = \{z_{N_1+1}, z_{N_1+2}, \ldots, z_{N_1+l}, \ldots, z_{N_1+N_c}\}$ et $X_3 \subset X_2$.

En considérant $X_1$ l'ensemble des individus non dupliqués et $X_3$ ceux des exemplaires de classes $C_l$, on reconstruit un nouvel ensemble de données à partitionner, noté comme $X_0$, avec $X_0 = X_1 \cup X_3$ :

$$X_0 = \{z_1, z_2, \dots, z_{N_1}, z_{N_1+1}, \dots, z_{N_1+N_c}\}$$

où $z_i = x_i$, pour $i = 1, 2, \dots, N_1$ et $card(X_0) = N_1 + N_c = N_0$ avec $N_0 < N_1 + N_2$.

**Proposition 4.3.** Soit $X$ l'ensemble de données de taille $N \times B$ à partitionner, où $B$ désigne le nombre de caractéristiques. Soit $X_0$ le nouvel ensemble de données de taille réduite $N_0 \times B$, composé d'exemplaires d'individus identiques et d'individus non dupliqués ($X_0 \subset X$). Considérons $P_X$ et $P_{X_0}$ les partitions obtenues en appliquant AP sur $X$ et $X_0$, respectivement, alors $P_X \neq P_{X_0}$.

**Preuve 4.3.** Pour démontrer que $P_X \neq P_{X_0}$, il suffit de démontrer que les valeurs des responsabilités et des disponibilités changent dans les cas de $X$ et $X_0$.

Pour le calcul de disponibilité trois cas sont à considérer :

***cas 1***: $X = X_1$ avec $X_2 = \emptyset$ et $N_1 = N$;

***cas 2***: $X = X_1 \cup X_2$ ;

***cas 3***: $X = X_2$ avec $X_1 = \emptyset$ et $N_2 = N$.

- **Cas 1 : $X = X_1$ avec $X_2 = \emptyset$ et $N_1 = N$**

Ce cas suppose que l'ensemble de données à partitionner n'est pas composé d'individus dupliqués, c'est-à-dire, $\forall\, x_i \in X_1, H(x_i) = 1$.

Le critère de disponibilité est identique aux équations (2.15) et (2.16). Alors, dans ce cas :

$$\boldsymbol{P_X = P_{X_0}}.$$

- **Cas 2 : $X = X_1 \cup X_2$**

Ce cas suppose que l'ensemble de données est composé d'individus non dupliqués et d'individus identiques, c'est-à-dire, $\forall\, x_i \in X_1, H(x_i) = 1$ et $\forall\, x_j \in X_2, H(x_j) \geq 2$, respectivement. Dans ce cas, cinq sous-cas sont à considérer pour le calcul de la disponibilité :

**a) $x_i, x_k \in X_1$**

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{\substack{x_j \in X_1 \\ j \neq \{i,k\}}} \max[0, r(x_j, x_k)] + \sum_{x_l \in X_2} \max[0, r(x_l, x_k)]\}$$

$$= \min\{0, r(x_k, x_k) + \sum_{\substack{x_j \in X_1 \\ j \neq \{i,k\}}} \max[0, r(x_j, x_k)] + \sum_{x_q \in C_1} \max[0, r(x_q, x_k)] + \sum_{x_\mathfrak{z} \in C_2} \max[0, r(x_\mathfrak{z}, x_k)] + \cdots$$

$$+ \sum_{x_\sigma \in C_{N_c}} \max[0, r(x_\sigma, x_k)]\}$$

$$\begin{aligned} a(x_i, x_k) = \min\{0, & r(x_k, x_k) \\ & + \sum_{\substack{x_j \in X_1 \\ j \neq \{i,k\}}} \max[0, r(x_j, x_k)] + \boldsymbol{H(x_q)} \times \max[0, r(x_q, x_k)] + \boldsymbol{H(x_\mathfrak{z})} \\ & \times \max[0, r(x_\mathfrak{z}, x_k)] + \cdots + \boldsymbol{H(x_\sigma)} \times \max[0, r(x_\sigma, x_k)]\} \end{aligned} \tag{4.8}$$

$$\begin{aligned} a(x_k, x_k) &= \sum_{\substack{x_j \in X_1 \\ j \neq k}} \max[0, r(x_j, x_k)] + \sum_{x_l \in X_2} \max[0, r(x_l, x_k)] \\ &= \sum_{\substack{x_j \in X_1 \\ j \neq k}} \max[0, r(x_j, x_k)] + \sum_{x_q \in C_1} \max[0, r(x_q, x_k)] + \sum_{x_\mathfrak{z} \in C_2} \max[0, r(x_\mathfrak{z}, x_k)] + \cdots \\ &\quad + \sum_{x_\sigma \in C_{N_c}} \max[0, r(x_\sigma, x_k)] \end{aligned}$$

$$\begin{aligned} a(x_k, x_k) = & \sum_{\substack{x_j \in X_1 \\ j \neq k}} \max[0, r(x_j, x_k)] + \boldsymbol{H(x_q)} \times \max[0, r(x_q, x_k)] + \boldsymbol{H(x_\mathfrak{z})} \\ & \times \max[0, r(x_\mathfrak{z}, x_k)] + \cdots + \boldsymbol{H(x_\sigma)} \times \max[0, r(x_\sigma, x_k)] \end{aligned} \tag{4.9}$$

Si on applique AP sur $X_0$, on a $H(x_q) = H(x_z) = H(x_o) = 1$, notons $A_0$ et $R_0$ les nouvelles matrices de responsabilité et de disponibilité et $r_0\ (x_i, x_k)$ et $a_0\ (x_i, x_k)$ leurs éléments respectifs.

$$a_0(x_i, x_k) = \min\{0, r_0(x_k, x_k) + \sum_{\substack{x_j \in X_1 \\ j \neq \{i,k\}}} \max[0, r_0(x_j, x_k)] +$$

$$\max[0, r_0(x_q, x_k)] + \max[0, r_0(x_\mathfrak{z}, x_k)] + \cdots + \max[0, r_0(x_\sigma, x_k)]\} < a(x_i, x_k).$$

$a_0(x_k, x_k) = \sum_{\substack{x_j \in X_1 \\ j \neq k}} \max[0, r_0(x_j, x_k)] + \max[0, r_0(x_q, x_k)] + \max[0, r_0(x_ʒ, x_k)] + \cdots +$

$\max[0, r_0(x_{\mathcal{O}}, x_k)] < a(x_k, x_k)$.

Donc $\boldsymbol{P_X \neq P_{X_0}}$.

**b) $\boldsymbol{x_i \in X_2}$ et $\boldsymbol{x_k \in X_1}$**

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{\substack{x_j \in X_1 \\ j \neq k}} \max[0, r(x_j, x_k)] + \sum_{\substack{x_l \in X_2 \\ l \neq i}} \max[0, r(x_l, x_k)]\}$$

Supposons que $x_i \in C_1$, alors $\forall\, x_q \in C_1$, $H(x_i) = H(x_q)$.

$$\begin{aligned} a(x_i, x_k) = \min\{\, 0, r(x_k, x_k) &+ \sum_{\substack{x_j \in X_1 \\ j \neq k}} \max[0, r(x_j, x_k)] \\ &+ \sum_{\substack{x_q \in C_1 \\ q \neq i}} \max[0, r(x_q, x_k)] + \sum_{x_ʒ \in C_2} \max[0, r(x_ʒ, x_k)] + \cdots \\ &+ \sum_{x_{\mathcal{O}} \in C_{N_c}} \max[0, r(x_{\mathcal{O}}, x_k)]\} \end{aligned}$$

$$\begin{aligned} a(x_i, x_k) = \min\{0, r(x_k, x_k) &+ \sum_{\substack{x_j \in X_1 \\ j \neq k}} \max[0, r(x_j, x_k)] + \boldsymbol{(H(x_i) - 1)} \\ &\times \max[0, r(x_i, x_k)] + \boldsymbol{H(x_ʒ)} \times \max[0, r(x_ʒ, x_k)] + \cdots \\ &+ \boldsymbol{H(x_{\mathcal{O}})} \times \max[0, r(x_{\mathcal{O}}, x_k)]\} \end{aligned} \tag{4.10}$$

$a(x_k, x_k)$ est identique à l'équation (4.9).

Si on applique AP sur $X_0$, on a $H(x_q) = H(x_z) = H(x_o) = 1$, par conséquent,

$a_0(x_i, x_k) = \min\{0, r_0(x_k, x_k) + \sum_{\substack{x_j \in X_1 \\ j \neq k}} \max[0, r_0(x_j, x_k)] + \max[0, r_0(x_ʒ, x_k)] + \cdots +$

$\max[0, r_0(x_{\mathcal{O}}, x_k)]\} \;<\; a(x_i, x_k)$ et $a_0(x_k, x_k) < a(x_k, x_k)$.

Donc $\boldsymbol{P_X \neq P_{X_0}}$.

**c) $x_i \in X_1$ et $x_k \in X_2$**

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{\substack{x_j \in X_1 \\ j \neq i}} \max[0, r(x_j, x_k)] + \sum_{\substack{x_l \in X_2 \\ l \neq k}} \max[0, r(x_l, x_k)]\}$$

Supposons que $x_k \in C_2$, alors $\forall\, x_\mathfrak{z} \in C_2$, $H(x_k) = H(x_\mathfrak{z})$.

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{\substack{x_j \in X_1 \\ j \neq i}} \max[0, r(x_j, x_k)] + \sum_{x_q \in C_1} \max[0, r(x_q, x_k)]$$

$$+ \sum_{\substack{x_\mathfrak{z} \in C_2 \\ \mathfrak{z} \neq k}} \max[0, r(x_\mathfrak{z}, x_k)] + \cdots + \sum_{x_\sigma \in C_{N_c}} \max[0, r(x_\sigma, x_k)]\}$$

$$\begin{aligned} a(x_i, x_k) = \min\{0, r(x_k, x_k) &+ \sum_{\substack{x_j \in X_1 \\ j \neq i}} \max[0, r(x_j, x_k)] + \boldsymbol{H(x_q)} \times \max[0, r(x_q, x_k)] \\ &+ (\boldsymbol{H(x_\mathfrak{z}) - 1}) \times \max[0, r(x_\mathfrak{z}, x_k)] + \cdots + \boldsymbol{H(x_\sigma)} \\ &\times \max[0, r(x_\sigma, x_k)]\} \end{aligned} \tag{4.11}$$

$$a(x_k, x_k) = \sum_{x_j \in X_1} \max[0, r(x_j, x_k)] + \sum_{\substack{x_l \in X_2 \\ l \neq k}} \max[0, r(x_l, x_k)]$$

$$= \sum_{x_j \in X_1} \max[0, r(x_j, x_k)] + \sum_{x_q \in C_1} \max[0, r(x_q, x_k)] + \sum_{\substack{x_\mathfrak{z} \in C_2 \\ \mathfrak{z} \neq k}} \max[0, r(x_\mathfrak{z}, x_k)] + \cdots$$

$$+ \sum_{x_\sigma \in C_{N_c}} \max[0, r(x_\sigma, x_k)]$$

$$\begin{aligned} a(x_k, x_k) = \sum_{x_j \in X_1} &\max[0, r(x_j, x_k)] + \boldsymbol{H(x_q)} \times \max[0, r(x_q, x_k)] + (\boldsymbol{H(x_\mathfrak{z}) - 1}) \\ &\times \max[0, r(x_\mathfrak{z}, x_k)] + \cdots + \boldsymbol{H(x_\sigma)} \times \max[0, r(x_\sigma, x_k)] \end{aligned} \tag{4.12}$$

Si on applique AP sur $X_0$, on a $H(x_q) = H(x_\mathfrak{z}) = H(x_\sigma) = 1$, par conséquent

$$a_0(x_i,x_k) = \min\{0, r_0(x_k,x_k) + \sum_{\substack{x_j \in X_1 \\ j \neq k}} \max\left[0, r_0\left(x_j,x_k\right)\right] + \cdots + \max[0, r_0(x_{\sigma'},x_k)]\} <$$

$$a(x_i,x_k).$$

$$a_0(x_k,x_k) = \sum_{x_j \in X_1} \max\left[0, r_0\left(x_j,x_k\right)\right] + \max\left[0, r_0\left(x_q,x_k\right)\right] + \cdots + \max[0, r_0(x_{\sigma'},x_k)]$$

$$< a(x_k,x_k).$$

Donc $\boldsymbol{P_X} \neq \boldsymbol{P_{X_0}}$.

**d) $\boldsymbol{x_i, x_k} \in \boldsymbol{X_2}$ avec $\boldsymbol{x_i} \in \boldsymbol{C_l}$ et $\boldsymbol{x_k} \in \boldsymbol{C_f}$**

$$a(x_i,x_k) = \min\{0, r(x_k,x_k) + \sum_{x_j \in X_1} \max\left[0, r\left(x_j,x_k\right)\right] + \sum_{\substack{x_l \in X_2 \\ l \neq \{i,k\}}} \max[0, r(x_l,x_k)]\}$$

Supposons que $x_i \in C_1$ et $x_k \in C_2$, alors $\forall\, x_q \in C_1$, $H\left(x_q\right) = H(x_i)$ et $r\left(x_q,x_k\right) = r(x_i,x_k)$, et $\forall\, x_\mathfrak{z} \in C_2$, $H(x_k) = H\left(x_\mathfrak{z}\right)$.

$$a(x_i,x_k) = \min\{0, r(x_k,x_k) + \sum_{x_j \in X_1} \max\left[0, r\left(x_j,x_k\right)\right] + \sum_{\substack{x_q \in C_1 \\ q \neq i}} \max\left[0, r\left(x_q,x_k\right)\right]$$

$$+ \sum_{\substack{x_\mathfrak{z} \in C_2 \\ \mathfrak{z} \neq k}} \max\left[0, r\left(x_\mathfrak{z},x_k\right)\right] + \cdots + \sum_{x_{\sigma'} \in C_{N_c}} \max[0, r(x_{\sigma'},x_k)]\}$$

$$\begin{aligned} a(x_i,x_k) = \min\{0, r(x_k,x_k) &+ \sum_{x_j \in X_1} \max\left[0, r\left(x_j,x_k\right)\right] + (\boldsymbol{H(x_i) - 1}) \\ &\times \max[0, r(x_i,x_k)] + \left(\boldsymbol{H\left(x_\mathfrak{z}\right) - 1}\right) \times \max\left[0, r\left(x_\mathfrak{z},x_k\right)\right] + \cdots + \boldsymbol{H(x_{\sigma'})} \\ &\times \max[0, r(x_{\sigma'},x_k)]\} \end{aligned} \tag{4.13}$$

$a(x_k,x_k)$ est identique à l'équation (4.12).

Si on applique AP sur $X_0$, on a $H\left(x_q\right) = H\left(x_\mathfrak{z}\right) = H(x_{\sigma'}) = 1$, par conséquent

$$a_0(x_i,x_k) = \min\{0, r_0(x_k,x_k) + \sum_{x_j \in X_1} \max[0, r_0(x_j,x_k)] + \cdots + \max[0, r_0(x_{\sigma'},x_k)]\} < a(x_i,x_k)$$

et $a_0(x_k, x_k) < a(x_k, x_k)$.

Donc $\boldsymbol{P_X} \neq \boldsymbol{P_{X_0}}$.

**e) $\boldsymbol{x_i, x_k} \in \boldsymbol{X_2}$ avec $\boldsymbol{x_i, x_k} \in \boldsymbol{C_l}$**

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{x_j \in X_1} \max[0, r(x_j, x_k)] + \sum_{\substack{x_l \in X_2 \\ l \neq \{i,k\}}} \max[0, r(x_l, x_k)]\}$$

Supposons que $x_i, x_k \in C_1$, alors $\forall\, x_q \in C_1$, $H(x_q) = H(x_i) = H(x_k)$.

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{x_j \in X_1} \max[0, r(x_j, x_k)] + \sum_{\substack{x_q \in C_1 \\ q \neq \{i,k\}}} \max[0, r(x_q, x_k)]$$
$$+ \sum_{x_\mathfrak{z} \in C_2} \max[0, r(x_\mathfrak{z}, x_k)] + \cdots + \sum_{x_\mathcal{o} \in C_{N_C}} \max[0, r(x_\mathcal{o}, x_k)]\}$$

$$\begin{aligned} a(x_i, x_k) = {} & \min\{0, r(x_k, x_k) + \sum_{x_j \in X_1} \max[0, r(x_j, x_k)] + (\boldsymbol{H(x_i) - 2}) \\ & \times \max[0, r(x_i, x_k)] + \boldsymbol{H(x_\mathfrak{z})} \times \max[0, r(x_\mathfrak{z}, x_k)] + \cdots \\ & + \boldsymbol{H(x_\mathcal{o})} \times \max[0, r(x_\mathcal{o}, x_k)]\} \end{aligned} \tag{4.14}$$

$$a(x_k, x_k) = \sum_{x_j \in X_1} \max[0, r(x_j, x_k)] + \sum_{\substack{x_l \in X_2 \\ l \neq k}} \max[0, r(x_l, x_k)]$$

$$a(x_k, x_k) = \sum_{x_j \in X_1} \max[0, r(x_j, x_k)] + \sum_{\substack{x_q \in C_1 \\ q \neq k}} \max[0, r(x_q, x_k)] + \sum_{x_\mathfrak{z} \in C_2} \max[0, r(x_\mathfrak{z}, x_k)] + \cdots$$
$$+ \sum_{x_\mathcal{o} \in C_{N_C}} \max[0, r(x_\mathcal{o}, x_k)]$$

$$\begin{aligned} a(x_k, x_k) = {} & \sum_{x_j \in X_1} \max[0, r(x_j, x_k)] + (\boldsymbol{H(x_i) - 1}) \times \max[0, r(x_i, x_k)] + \boldsymbol{H(x_\mathfrak{z})} \\ & \times \max[0, r(x_\mathfrak{z}, x_k)] \quad + \cdots + \boldsymbol{H(x_\mathcal{o})} \times \max[0, r(x_\mathcal{o}, x_k)] \end{aligned} \tag{4.15}$$

Si on applique AP sur $X_0$, on a $H(x_i) = H(x_\mathfrak{z}) = H(x_\mathfrak{o}) = 1$, par conséquent

$a_0(x_i, x_k) = \min\{0, r_0(x_k, x_k) + \sum_{x_j \in X_1} \max[0, r_0(x_j, x_k)] + \max[0, r_0(x_\mathfrak{z}, x_k)] + \cdots +$

$\max[0, r_0(x_\mathfrak{o}, x_k)]\} < a(x_i, x_k)$.

$a_0(x_k, x_k) = \sum_{x_j \in X_1} \max[0, r_0(x_j, x_k)] + \max[0, r_0(x_\mathfrak{z}, x_k)] + \cdots + \max[0, r_0(x_\mathfrak{o}, x_k)] <$

$a(x_k, x_k)$.

Donc $\boldsymbol{P_X \neq P_{X_0}}$.

- **Cas 3 : $\boldsymbol{X = X_2}$ avec $\boldsymbol{X_1 = \emptyset}$ et $\boldsymbol{N_2 = N}$**

Ce cas suppose que l'ensemble de données est composé uniquement d'individus identiques, c'est-à-dire, $\forall\, x_i \in X$, $H(x_i) \geq 2$. Alors deux sous-cas doivent être considérés :

**a) $\boldsymbol{x_i, x_k \in X_2}$ avec $\boldsymbol{x_i \in C_l}$ et $\boldsymbol{x_k \in C_f}$**

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{\substack{x_l \in X_2 \\ l \neq \{i,k\}}} \max[0, r(x_l, x_k)]\}$$

Supposons que $x_i \in C_1$ et $x_k \in C_2$, alors $\forall\, x_q \in C_1$, $H(x_q) = H(x_i)$ et $r(x_q, x_k) = r(x_i, x_k)$, et

$\forall\, x_\mathfrak{z} \in C_2$, $H(x_k) = H(x_\mathfrak{z})$.

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{\substack{x_q \in C_1 \\ q \neq i}} \max[0, r(x_q, x_k)] + \sum_{\substack{x_\mathfrak{z} \in C_2 \\ \mathfrak{z} \neq k}} \max[0, r(x_\mathfrak{z}, x_k)] + \cdots$$

$$+ \sum_{x_\mathfrak{o} \in C_{N_c}} \max[0, r(x_\mathfrak{o}, x_k)]\}$$

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + (\boldsymbol{H(x_i) - 1}) \times \max[0, r(x_i, x_k)] + (\boldsymbol{H(x_\mathfrak{z}) - 1}) \times \max[0, r(x_\mathfrak{z}, x_k)] + \cdots + \boldsymbol{H(x_\mathfrak{o})} \times \max[0, r(x_\mathfrak{o}, x_k)]\} \tag{4.16}$$

$$a(x_k, x_k) = \sum_{\substack{x_l \in X_2 \\ l \neq k}} \max[0, r(x_l, x_k)]$$

$$= \sum_{x_q \in C_1} \max[0, r(x_q, x_k)] + \sum_{\substack{x_\mathfrak{z} \in C_2 \\ \mathfrak{z} \neq k}} \max[0, r(x_\mathfrak{z}, x_k)] + \cdots + \sum_{x_\mathfrak{o} \in C_{N_c}} \max[0, r(x_\mathfrak{o}, x_k)]$$

$$a(x_k, x_k) = \boldsymbol{H(x_q)} \times \max[0, r(x_q, x_k)] + (\boldsymbol{H(x_ʒ) - 1}) \times \max[0, r(x_ʒ, x_k)] + \cdots + \boldsymbol{H(x_ơ)} \times \max[0, r(x_ơ, x_k)] \quad (4.17)$$

Si on applique AP sur $X_0$, on a $H(x_i) = H(x_q) = H(x_ʒ) = H(x_ơ) = 1$, par conséquent

$a_0(x_i, x_k) = \min\{0, r_0(x_k, x_k) + \cdots + \max[0, r_0(x_ơ, x_k)]\} < a(x_i, x_k)$.

$a_0(x_k, x_k) = \max[0, r_0(x_q, x_k)] + \cdots + \max[0, r_0(x_ơ, x_k)] < a(x_k, x_k)$.

Donc $\boldsymbol{P_X} \neq \boldsymbol{P_{X_0}}$.

**b) $\boldsymbol{x_i, x_k} \in \boldsymbol{X_2}$ avec $\boldsymbol{x_i, x_k} \in \boldsymbol{C_l}$**

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{\substack{x_l \in X_2 \\ l \neq \{i,k\}}} \max[0, r(x_l, x_k)]\}$$

Supposons que $x_i, x_k \in C_1$, alors $\forall\, x_q \in C_1$, $H(x_q) = H(x_i) = H(x_k)$.

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{\substack{x_q \in C_1 \\ q \neq \{i,k\}}} \max[0, r(x_q, x_k)] + \sum_{x_ʒ \in C_2} \max[0, r(x_ʒ, x_k)] + \cdots + \sum_{x_ơ \in C_{N_c}} \max[0, r(x_ơ, x_k)]\}$$

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + (\boldsymbol{H(x_i) - 2}) \times \max[0, r(x_i, x_k)] + \boldsymbol{H(x_ʒ)} \times \max[0, r(x_ʒ, x_k)] + \cdots + \boldsymbol{H(x_ơ)} \times \max[0, r(x_ơ, x_k)]\} \quad (4.18)$$

$$a(x_k, x_k) = \sum_{\substack{x_l \in X_2 \\ l \neq k}} \max[0, r(x_l, x_k)]$$

$$a(x_k, x_k) = \sum_{\substack{x_q \in C_1 \\ q \neq k}} \max[0, r(x_q, x_k)] + \sum_{x_ʒ \in C_2} \max[0, r(x_ʒ, x_k)] + \cdots + \sum_{x_ơ \in C_{N_c}} \max[0, r(x_ơ, x_k)]$$

$$a(x_k, x_k) = (\boldsymbol{H(x_i) - 1}) \times \max[0, r(x_i, x_k)] + \boldsymbol{H(x_ʒ)} \times \max[0, r(x_ʒ, x_k)] + \cdots + \boldsymbol{H(x_ơ)} \times \max[0, r(x_ơ, x_k)] \quad (4.19)$$

Si on applique AP sur $X_0$, on a $H(x_i) = H(x_q) = H(x_ʒ) = H(x_{ơ}) = 1$, par conséquent

$a_0(x_i, x_k) = \min\{0, r_0(x_k, x_k) + \max[0, r_0(x_ʒ, x_k)] + \cdots + \max[0, r_0(x_{ơ}, x_k)]\} < a(x_i, x_k)$.

$a_0(x_k, x_k) = \max[0, r_0(x_ʒ, x_k)] + \cdots + \max[0, r_0(x_{ơ}, x_k)] < a(x_k, x_k)$.

Donc $\boldsymbol{P_X \neq P_{X_0}}$.

Comme la matrice de disponibilité n'est pas symétrique, les mêmes étapes doivent être suivies pour la démonstration de $a(x_k, x_i)$.

■

Suivant la Proposition 4.3, la présence d'individus identiques affecte les résultats de l'AP, pour résoudre ce problème nous donnons une nouvelle définition comme suit :

**Définition 4.2.** (Individus associés) Un ensemble d'individus est dit associé à un exemplaire $I_i$ s'il appartient à la même classe $C_i$ représentée par $I_i$ obtenu par UP-OAP. Ainsi, chaque classe d'individus associés est donc considérée comme étant composée d'individus dupliqués représentés par un exemplaire.

Pour réduire la taille de la matrice de similarité, c'est-à-dire le nombre d'individus à partitionner, nous introduisons la notion d'association. Cela signifie que chaque ensemble d'individus associés sera représenté par un exemplaire et le nombre de ses membres, noté $H(I_i)$. Dans ce cas, l'algorithme UP-OAP ne sera appliqué que sur les exemplaires des individus associés, c'est-à-dire sur les données originales transformées en respectant le principe du calcul des critères de responsabilité et de disponibilité employés. Avec cette transformation, nous devons nous assurer que le résultat du partitionnement des données originales reste le même que celui du partitionnement des données transformées. Pour répondre à cette condition, le critère de disponibilité doit être reformulé. Par conséquent, le problème abordé ici est comment calculer le critère de disponibilité en prenant en compte un nouvel ensemble de données $X_0$ de taille $N_0$, composé d'individus non dupliqués et d'exemplaires d'individus associés ou dupliqués au lieu de l'ensemble des $N$ individus ($N_0 < N$). L'objectif est d'obtenir les mêmes résultats en réduisant ou non la taille de la matrice de similarité. La reformulation des critères utilisés dans l'algorithme AP

pour identifier les exemplaires est fondamentale pour partitionner de grands ensembles de données. Dans ce cas, la responsabilité sera modifiée automatiquement via le critère de disponibilité.

**Corollaire 4.1.** Si l'ensemble de données $X$ à partitionner contient des individus dupliqués, alors l'application de la méthode UP-OAP sur l'ensemble réduit $X_0$, formé d'exemplaires d'individus dupliqués et d'individus non dupliqués nécessite la réécriture du critère de disponibilité en prenant en compte $H(I_j)$ selon les équations suivantes :

- ***Reformulation de la disponibilité dans le cas 1* : $\boldsymbol{X = X_1}$ avec $\boldsymbol{X_2 = \emptyset}$ et $\boldsymbol{N_1 = N}$**

Le critère de disponibilité est reformulé selon les équations (2.15) et (2.16).

- ***Reformulation de la disponibilité dans le cas 2* : $\boldsymbol{X = X_1 \cup X_2}$**

**a) $\boldsymbol{x_i, x_k \in X_1}$**

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{\substack{x_j \in X_1 \\ j \neq \{i,k\}}} \max\left[0, r\left(x_j, x_k\right)\right] + \sum_{l=1}^{N_c} \boldsymbol{H(x_l)} \times \max[0, r(x_l, x_k)]\} \tag{4.20}$$

$$a(x_k, x_k) = \sum_{\substack{x_j \in X_1 \\ j \neq k}} \max\left[0, r\left(x_j, x_k\right)\right] + \sum_{l=1}^{N_c} \boldsymbol{H(x_l)} \times \max[0, r(x_l, x_k)]\} \tag{4.21}$$

**b) $\boldsymbol{x_i \in X_2}$ et $\boldsymbol{x_k \in X_1}$**

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{\substack{x_j \in X_1 \\ j \neq k}} \max\left[0, r\left(x_j, x_k\right)\right] + (\boldsymbol{H(x_i) - 1}) \times \max[0, r(x_i, x_k)] + \sum_{\substack{l=1 \\ l \neq i}}^{N_c} \boldsymbol{H(x_l)} \times \max[0, r(x_l, x_k)]\} \tag{4.22}$$

$a(x_k, x_k)$ est identique à l'équation (4.21).

**c) $\boldsymbol{x_i \in X_1}$ et $\boldsymbol{x_k \in X_2}$**

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{\substack{x_j \in X_1 \\ j \neq i}} \max[0, r(x_j, x_k)] + \sum_{\substack{l=1 \\ l \neq k}}^{N_c} \boldsymbol{H(x_l)} \times \max[0, r(x_l, x_k)]\} \tag{4.23}$$

$$a(x_k, x_k) = \sum_{x_j \in X_1} \max[0, r(x_j, x_k)] + \sum_{\substack{l=1 \\ l \neq k}}^{N_c} \boldsymbol{H(x_l)} \times \max[0, r(x_l, x_k)]\} \tag{4.24}$$

**d) $\boldsymbol{x_i, x_k \in X_2}$ avec $\boldsymbol{x_i \in C_l}$ et $\boldsymbol{x_k \in C_f}$**

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + \sum_{x_j \in X_1} \max[0, r(x_j, x_k)] + (\boldsymbol{H(x_i) - 1}) \times \max[0, r(x_i, x_k)] + \sum_{\substack{l=1 \\ l \neq \{i,k\}}}^{N_c} \boldsymbol{H(x_l)} \times \max[0, r(x_l, x_k)]\} \tag{4.25}$$

$a(x_k, x_k)$ est identique à l'équation (4.24).

**e) $\boldsymbol{x_i, x_k \in X_2}$ avec $\boldsymbol{x_i, x_k \in C_l}$**

$$a(x_i, x_k) = a(x_k, x_k) = \sum_{x_j \in X_1} \max[0, r(x_j, x_k)] + \sum_{\substack{l=1 \\ l \neq k}}^{N_c} \boldsymbol{H(x_l)} \times \max[0, r(x_l, x_k)]\} \tag{4.26}$$

– ***Reformulation de la disponibilité dans le cas 3 :*** $\boldsymbol{X = X_2}$ **avec** $\boldsymbol{X_1 = \emptyset}$ **et** $\boldsymbol{N_2 = N}$

**a) $\boldsymbol{x_i, x_k \in X_2}$ avec $\boldsymbol{x_i \in C_l}$ et $\boldsymbol{x_k \in C_f}$**

$$a(x_i, x_k) = \min\{0, r(x_k, x_k) + (\boldsymbol{H(x_i) - 1}) \times \max[0, r(x_i, x_k)] + \sum_{\substack{l=1 \\ l \neq \{i,k\}}}^{N_c} \boldsymbol{H(x_l)} \times \max[0, r(x_l, x_k)]\} \tag{4.27}$$

$$a(x_k, x_k) = \sum_{\substack{l=1 \\ l \neq k}}^{N_c} H(x_l) \times \max[0, r(x_l, x_k)]\} \tag{4.28}$$

**b) $\boldsymbol{x_i, x_k \in X_2}$ avec $\boldsymbol{x_i, x_k \in C_l}$**

$$a(x_i, x_k) = a(x_k, x_k) = \sum_{\substack{l=1 \\ l \neq k}}^{N_c} \boldsymbol{H(x_l)} \times \max[0, r(x_l, x_k)]\} \tag{4.29}$$

**Preuve Corollaire 4.1.** Il est facile de démontrer cette proposition par la preuve de la Proposition 4.2.

■

La nouvelle méthode de partitionnement prenant en considération la modification du critère de disponibilité nommée UP-OAPM est décrite dans l'Algorithme 4.5.

### 4.3.2 Transformation des données réduisant la taille la matrice de similarité

Pour pouvoir partitionner des ensembles de données de grande taille tout en prenant en considération le nombre d'objets dupliqués, nous exploitons les résultats de la reformulation de la matrice de disponibilité présentés dans 4.3.1. Cette reformulation sera utilisée pour transformer les données, ce qui permet de réduire la taille de la matrice de similarité en garantissant que le résultat final du partitionnement est identique à celui qui peut être obtenu avec une matrice de similarité complète.

Cette opération est effectuée en deux étapes. Elle consiste tout d'abord à subdiviser l'image en blocs, puis à appliquer UP-OAPM sur chaque bloc en prenant en considération la reformulation de la disponibilité (Algorithme 4.5 détaillé ci-dessous). Pour chacune des classes formées dans un bloc, nous lui associons un exemplaire qui remplacera tous les pixels qui la composent et aussi son effectif. Cette stratégie, appliquée à tous les blocs, permet la formation d'un nouvel ensemble de données (données transformées) à partitionner limité aux exemplaires des classes des blocs.

Dans cette sous-section, nous détaillons l'étape de transformation des données.

**Algorithme 4.5. UP-OAPM (avec modification du critère de disponibilité)**

**Entrée :** Tableau de données ($N$ objets $\times$ $B$ attributs) représentant l'ensemble des individus à partitionner

**Etape 0 :**

1. Calculer l'histogramme $H(x_i)$ des individus $x_i$
2. Former le nouvel ensemble de données $X_0$ de taille $N_0$ ($N_0 < N$), composé d'individus non dupliqués et d'exemplaires des individus associés

**Etape 1 :**

3. Calculer la matrice de similarité $S_0$ de taille $N_0 \times N_0$
   $s_0(x_i, x_k) = -d_1(x_i, x_k)$, où $d_1$ est la distance associée à la norme $L_1$
4. Initialiser : $r(x_i, x_k) = 0$, $a(x_i, x_k) = 0$
5. Remplacer les éléments diagonaux de $S_0$ par la valeur de $\overline{p}_i$ en utilisant l'équation (4.2)
6. Calculer toutes les responsabilités compte tenu des disponibilités selon les équations (2.13), (4.2) et (4.4)
7. Calculer toutes les disponibilités compte tenu des responsabilités selon les cas (équations (4.20) - (4.29) ) et (4.5)
8. Identifier les exemplaires $x_k$ qui maximisent $E^* = \underset{k}{\operatorname{argmax}}\ [\hat{r}(x_i, x_k)]$ (équation (4.6))
9. **Si** les exemplaires ne changent pas procéder à l'étape suivant (10)
   **Sinon** répéter étapes (6) à (8) jusqu'à convergence
   **Fin si**
10. Agréger chaque individu à son exemplaire le plus proche et arrêter

**Sorties :** Partition $P$ de $K$ classes, exemplaire $I_j$ de chaque classe $C_j$, et $H(C_j)$ le nombre de pixels dans chaque classe $C_j$

Soit $Im$ l'image originale (ou l'ensemble $X$) à partitionner composée de $N$ pixels, où chaque pixel $x_i \in Im$ ($\in X$) est caractérisé par $B$ attributs. Soit $Im_C$ l'image transformée où chaque pixel est caractérisé par les mêmes $B$ attributs. Nous illustrons cette opération en prenant l'exemple d'une image.

Considérez comment nous transformons l'image $Im$ avec une matrice de similarité de grande taille $N \times N$, en image $Im_C$ avec une matrice de similarité de taille réduite $N_0 \times N_0$. Cette transformation est détaillée dans les quatre étapes suivantes (voir Algorithme 4.6) :

- **Etape 1 :** Découpage de l'image en blocs
- **Etape 2 :** Partitionnement de chaque bloc par l'Algorithme 4.5.
- **Etape 3 :** Transformation des blocs

Chaque bloc $B_{ij}$ se transforme en un nouveau bloc $B_{ij}^{C}$ en exploitant les exemplaires de la partition $P_{ij}$ de bloc $B_{ij}$. Chaque pixel appartenant à la classe $C_{ij}^{k}$ du bloc $B_{ij}$ est remplacé par le vecteur de son exemplaire représentatif $I_{ij}^{k}$.

- **Etape 4 :** Transformation de l'image originale

L'image transformée $Im_C$ à partir de laquelle le partitionnement final sera effectué correspond à celle obtenue à partir de l'ensemble des blocs transformés $B_{ij}^{C}$ comme suit :

$$Im_C = \begin{bmatrix} B_{11}^{C} & B_{12}^{C} & \cdots & B_{1M_2}^{C} \\ B_{21}^{C} & B_{22}^{C} & \cdots & B_{2M_2}^{C} \\ \vdots & \vdots & \vdots & \vdots \\ B_{M_{12}}^{C} & B_{M_{12}}^{C} & \cdots & B_{M_1M_2}^{C} \end{bmatrix}$$

**Proposition 4.4.** Si $Im$ l'image originale à partitionner et $Im_C$ l'image transformée obtenue par la procédure décrite ci-dessus, en agrégeant les individus similaires de $Im$ suivant la métrique $L_1$-norme, alors $Im_C$ est similaire à $Im$. Dans ce cas, soit $E[Im - Im_C]$ la moyenne de l'erreur entre $Im$ et $Im_C$, alors $E[Im - Im_C] \approx 0$.

**Preuve 4.4.** Pour montrer que la moyenne de l'erreur entre $Im$ et $Im_C$ est très faible, il faut démontrer que la moyenne de l'erreur entre chaque bloc et sa transformée est très faible.

Soit $B_{ij}$ un bloc de $Im$ et $B_{ij}^{c}$ sa transformée, avec $I_{ij} = \left\{I_{ij}^{1}, I_{ij}^{2}, \dots, I_{ij}^{N_{ij}}\right\}$ est l'ensemble des exemplaires et $N_{ij}$ est le nombre de classes $C_{ij}^{k}$ de $P_{ij}$, avec $P_{ij} = \left\{C_{ij}^{1}, C_{ij}^{2}, \dots, C_{ij}^{N_{ij}}\right\}$, où $I_{ij}(x_k) = \max_{l}[r(x_k, x_l)]$.

Comme démontré dans la Section 4.2, chaque individu du bloc possède un exemplaire qui lui est similaire selon le critère d'optimisation basé sur la norme $L_1$. Etant donné que ce critère donne des classes homogènes et fines, alors nous avons $\forall\, x_{ij} \in C_{ij}^{k}$, $I_{ij}^{k} - x_{ij} \approx 0$. Donc $B_{ij} - B_{ij}^{c} = \varepsilon_{ij}^{k} \approx 0$.

Alors $E\left[B_{ij} - B_{ij}^{c}\right] \approx 0$ et $E[Im - Im_C] \approx 0$.

■

**Proposition 4.5.** Soit $S$ la matrice de similarité de taille $N \times N$ calculée sur l'image originale $Im$ et $S_0$ la matrice de similarité de taille $N_0 \times N_0$ calculée sur le nouvel ensemble de données $X_0$ de

taille $N_0$ formé par les exemplaires des individus identiques et les exemplaires des individus dupliqués de l'image transformée $Im_C$. On a $N_0 \ll N$.

La Proposition 4.5 montre que la taille de la matrice de similarité en utilisant l'image transformée est inférieure à celle calculée sur l'image originale, ce qui permet de partitionner des images ou données de très grande taille.

L'Algorithme 4.6 permet la transformation des données en utilisant l'algorithme de partitionnement 4.5 qui prend en considération la reformulation de la disponibilité. L'algorithme 4.6 est détaillé ci-dessous.

---

**Algorithme 4.6. Transformation de l'image originale ou Tableau de données**

---

**Entrée :**

- Image originale $Im$, ou Tableau de données ($N$ objets $\times B$ attributs) représentant l'ensemble des individus à partitionner
- Taille des blocs maximale ($Y_1 \times Y_2$) permettant l'application d'UP-OAPM

**Procédure :**

**1.** Diviser l'image $Im$ en $N_B$ blocs $B_{ij}$, avec $i \in \{1, 2, \dots, M_1\}$ et $j \in \{1, 2, \dots, M_2\}$, où $M_1$ est le nombre des blocs en une ligne et $M_2$ est le nombre des blocs en une colonne

**2.** Transformer les blocs

**Pour** $i = 1$ jusqu'à $M_1$ **faire**

**Pour** $j = 1$ jusqu'à $M_2$ **faire**

**(a) Partitionner le bloc $B_{ij}$ par UP-OAPM**

Soit $P_{ij}$ la partition obtenue sur le bloc $B_{ij}$ :

$I_{ij} = \left\{I_{ij}^1, I_{ij}^2, \dots, I_{ij}^{N_{ij}}\right\}$ est l'ensemble des exemplaires et $N_{ij}$ est le nombre des classes de $P_{ij}$

$P_{ij} = \left\{C_{ij}^1, C_{ij}^2, \dots, C_{ij}^{N_{ij}}\right\}$

**(b) Transformer le bloc $B_{ij}$ :** $B_{ij}^C$

Chaque pixel appartenant à la classes $C_{ij}^k$ est remplacé par la signature spectrale de son exemplaire $I_{ij}^k$

**Fin pour**

**Fin pour**

**3.** Transformer l'image $Im$ à partir des blocs $B_{ij}^C$

**Sortie :** Image transformée $Im_C$

---

Avec cette opération, chaque bloc de l'image originale est transformé indépendamment pour construire des données transformées à partir desquelles le partitionnement va être réalisé. Le

partitionnement d'ensembles de données de grande taille est devenu donc possible en garantissant le même résultat de l'image originale et de sa version transformée.

### 4.3.3 Partitionnement des données transformées et hiérarchisation

Le partitionnement de l'image transformée nécessite le calcul du critère de disponibilité et la connaissance du nombre de pixels associés à chaque exemplaire, comme expliqué au paragraphe 4.3.1.

Le partitionnement des données transformées par l'Algorithme 4.5 (UP-OAPM) donne une première partition notée $P_1$.

Pour obtenir la partition finale et estimer le nombre de classes de l'image originale, la méthode UP-OAPM est appliquée itérativement à chaque niveau, $i$, à l'image transformée, obtenue à partir de la partition $P_{i-1}$ de niveau $i-1$. Cette opération "transformation-partitionnement" identique à celle utilisée dans la méthode HUP-OAP donne un partitionnement hiérarchique.

Cette stratégie de partitionnement est parfaitement adaptée, parce que le nombre d'individus associés après transformation est intégré dans le calcul du critère de disponibilité, comme décrit précédemment.

La méthode développée ici pour le partitionnement des images de grande taille, nommée HUP-OAPM-RSM, est composée des trois algorithmes présentés précédemment : la transformation de l'image originale à partitionner (Algorithme 4.6) en utilisant la méthode UP-OAPM (Algorithme 4.5) et le partitionnement hiérarchique (Algorithme 4.7). Les principales étapes de cette méthode sont résumées dans l'Algorithme 4.8.

Le partitionnement final obtenu à partir de l'image originale après diverses associations et étapes de fusion est indépendant du choix de la taille de bloc comme démontré dans la Proposition 4.6.

**Proposition 4.6.** En subdivisant l'image en blocs de taille plus ou moins grande, la partition optimale de l'image transformée est la même, c'est-à-dire, soit $B_{ij}$ une première subdivision de $Im$ en $N_B$ blocs de taille $Y_1 \times Y_2$ et $B'_{kl}$ une autre subdivision de $Im$ en $N'_B$ blocs de taille $Q_1 \times Q_2$, tels que $Y_1 \neq Q_1$ et $Y_2 \neq Q_2$.

Dans ce cas, si la partition optimale obtenue en appliquant la méthode HUP-OAPM-RSM sur l'image transformée à l'aide des blocs $B_{ij}$ est $P_y$, et la partition optimale obtenue en appliquant la même méthode sur l'image transformée à l'aide des blocs $B'_{kl}$ est $P'_z$, alors nous avons $P_y = P'_z$.

**Preuve 4.6.** Soit $Im$ l'image originale à partitionner, $Im_C$ son image transformée obtenue en utilisant les blocs $B_{ij}$ et $Im'_C$ son image transformée obtenue en utilisant les blocs $B'_{kl}$.

Supposons que la partition optimale obtenue en appliquant la méthode HUP-OAPM-RSM sur les images transformées $Im_C$ et $Im'_C$ sont $P_y$ et $P'_z$, respectivement.

Nous avons prouvé dans la Proposition 4.4 que l'image originale à partitionner et l'image transformée obtenue par la procédure décrite, en agrégeant les individus similaires de $Im$ suivant la métrique $L_1$-norm, alors l'image originale et sa transformée sont similaires.

Par conséquent, $Im \cong Im_C$ et $Im \cong Im'_C$, on a donc $P_y = P'_z$.

■

**Algorithme 4.7. Partitionnement Hiérarchique des données transformées**

**Entrée :** Tableau de données correspondant à l'image transformée ou un Tableau de données transformé

**Etape préliminaire :**

- Calculer la fréquence d'apparition $H(x_i)$ de chaque individu $x_i$
- Former le nouvel ensemble de données $X_0$ de taille $N_0$ ($N_0 < N$), composé d'individus non associés et d'exemplaires des individus associés
- Initialiser : $X_1 = X_0$

**Procédure** :

1. Appliquer l'Etape 1 de UP-OAPM sur l'ensemble de données formé $X_1$
2. **Répéter** UP-OAPM sur le nouvel ensemble $X_i$, $i \geq 2$

   $X_i$ est composé des exemplaires $I_{i-1}^j$ de chaque classe $C_j$ de la partition $P_{i-1}$, avec

   $$X_i = \{I_{i-1}^1, I_{i-1}^2, \dots, I_{i-1}^{N_{i-1}}\}$$

   Former la partition $P_i$ : fusionner chaque individu avec son exemplaire

   Enregistrer la nouvelle partition $P_i$ obtenue à partir de $N_i$ classes, du critère $LN_i$, de l'exemplaire $I_i^j$ de chaque classe $C_j$ et $H(I_i^j)$ le nombre de pixels dans chaque classe $C_j$

   **Jusqu'à** la stabilité de la partition $P_i$
3. Choisir la partition finale ($P_{finale}$) qui maximise le critère $LN$

   $$P_{finale} = \max_i [LN_i]$$

**Sorties :** Les partitions hiérarchiques de l'image, la partition optimale et l'ensemble d'exemplaires de ses classes

**Algorithme 4.8. HUP-OAPM-RSM**

**Entrée :** Image originale $Im$, ou Tableau de données ($N$ objets $\times B$ attributs) représentant l'ensemble des individus à partitionner

1. Application de l'Algorithme 4.6 pour obtenir l'image transformée $Im_C$
2. Application de l'Algorithme 4.7 sur l'image transformée $Im_C$

**Sorties :** Les partitions hiérarchiques de l'image, la partition optimale et l'ensemble d'exemplaires de ses classes

### 4.3.4 Evaluation numérique

La méthode proposée HUP-OAPM-RSM a été évaluée sur les mêmes images synthétique et réelles présentées dans la Figure 6 et la Figure 12. Nous montrons que l'approche développée de transformation d'une image pour réduire la taille de la matrice de similarité et la méthode de partitionnement sont pertinentes, sur la base des trois critères que nous nous sommes fixés : amélioration du CCR et réduction de l'espace mémoire et du temps de calcul.

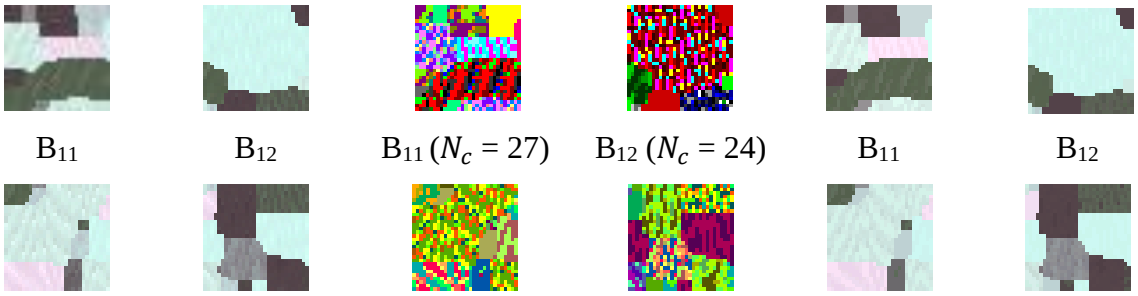
#### *4.3.4.1 Partitionnement de l'image synthétique*

La pertinence de cette nouvelle approche de réduction de la taille de la matrice de similarité par transformation des données est démontrée à travers l'image hyperspectrale synthétique présentée à la Figure 6.

La Figure 16 montre les résultats respectifs correspondant aux quatre étapes de la procédure de transformation de l'image originale, c'est-à-dire diviser en blocs (Figure 16 (a)), partitionner les blocs (Figure 16 (b)), transformer les blocs (Figure 16 (c)) et obtenir l'image transformée finale en fusionnant les blocs transformés (Figure 16 (d)). Ce résultat montre que l'image transformée est pratiquement identique à l'originale pour des images codées sur 16 bits. En effet, lors du calcul de l'erreur entre les bandes originales et les bandes reconstruites, la valeur moyenne de l'erreur absolue est très faible ($E\ [|\ Im - Im_C\ |] = 15.25$). Cette erreur est encore plus petite si l'erreur absolue n'est pas prise en compte ($E\ [Im - Im_C] = 0.85$).

Cet exemple montre que la taille de la matrice de similarité de l'image originale (avant transformation) est de 3 600 × 3 600, alors que celle de l'image transformée n'est que de 107 × 107.

La taille de la matrice de similarité, à partir de laquelle le partitionnement sera effectué, est donc significativement réduite, ainsi que pour les matrices de responsabilité et de disponibilité utilisées dans la méthode HUP-OAPM-RSM, où la taille de chacune est de 3 600 × 3 600 avant transformation.



$B_{11}$ $B_{12}$ $B_{11}$ ($N_c$ = 27) $B_{12}$ ($N_c$ = 24) $B_{11}$ $B_{12}$

$B_{21}$ $B_{22}$ $B_{21}$ ($N_c$ = 25) $B_{22}$ ($N_c$ = 31) $B_{21}$ $B_{22}$

**(a)** Division de l'image originale en 4 blocs

**(b)** Résultats du partitionnement des blocs ($N_c$ : nombre de classes par bloc)

**(c)** Blocs transformés

Bandes visualisées (5, 15, 25) Bandes visualisées (30, 70, 90)

**(d)** Image transformée

**Figure 16.** Transformation de l'image originale de la Figure 6 : étape par étape.

Pour démontrer numériquement l'absence d'influence de la taille du bloc sur les résultats de partitionnement de l'image transformée, nous utilisons comme exemples 4 et 8 blocs. La Figure 17 montre que le résultat du partitionnement de l'image est identique avant et après transformation. On observe que la partition optimale est obtenue au niveau 2 pour l'image transformée, alors qu'elle est obtenue au niveau 3 pour l'image originale, avec la même valeur du critère d'optimisation et le même nombre de classes.

Le nombre estimé de classes est de neuf et le CCR est de 100%. A titre de comparaison, les performances données par les méthodes *K*-means, FCM, S-OFCM, U-OFCM et AP original avec $p$ égal aux valeurs minimale ($p_{\min}$) et médiane ($p_{\text{med}}$) de la matrice de similarité sur l'image originale sont données dans Tableau 17. Les résultats de partitionnement de la méthode proposée confirment sa supériorité aux méthodes de l'état de l'art comparées.

Lors de cette évaluation, nous avons mis également en évidence, l'amélioration de la qualité d'estimation de la mise à jour des matrices de responsabilité et de disponibilité obtenue par la méthode HUP-OAPM-RSM, par rapport à l'AP originale. Comme le montre le Tableau 18, cette amélioration est évaluée en calculant l'erreur absolue moyenne de mise à jour et la variance de cette erreur. Bien que le premier critère soit strictement calculé (pas de compensation entre les valeurs d'erreur positives et négatives), ses valeurs sont pratiquement égales à zéro pour l'approche proposée par rapport à celles de l'AP originale. La variance de l'erreur de mise à jour est également très faible.

La méthode HUP-OAPM-RSM proposée fournit de meilleurs résultats, améliorant la qualité de partitionnement de l'AP originale, avec une réduction du temps de calcul (0.094 s contre 61.88 s de temps CPU) et de l'espace mémoire (11.40 Mb contre 296.93 Mb), sur un processeur Intel(R) Core(TM) i7-7700 avec 3,6 GHz et 16 Go de mémoire.

À l'aide de cet exemple, nous montrons que l'approche développée de transformation d'une image pour réduire la taille de la matrice de similarité contribue globalement à l'amélioration des résultats de partitionnement : possibilité de partitionnement d'images de grande taille, amélioration du CCR et réduction de l'espace mémoire et le temps de calcul.

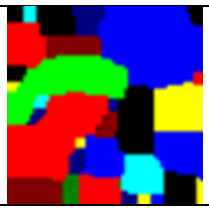
| | | |
|---|---|---|
| Résultat de partitionnement de l'image originale<br><br>Niveau 3, nombre estimé de classes 9 ($LN$ = 0.26) |  | Résultat de partitionnement de deux images transformées (4 et 8 blocs)<br>Niveau 2, nombre estimé de classes 9 ($LN$ = 0.26) |

**Figure 17.** Résultats de partitionnement par HUP-OAPM-RSM de l'image hyperspectrale synthétique de la Figure 6 et de ses deux images transformées utilisant 4 et 8 blocs.

**Tableau 17.** Comparaison des performances des méthodes de partitionnement sur l'image hypespectrale synthétique de la Figure 6.

| Méthodes | Non supervisées | | | | Semi-supervisées | | |
|---|---|---|---|---|---|---|---|
| | **HUP-OAPM-RSM** | AP | | U-OFCM | S-OFCM | FCM (*) | *K*-means (*) |
| | | $p_{med}$ | $p_{min}$ | | | | |
| Nombre de classes | **9 (estimé)** | 13 (estimé) | 9 (estimé) | 6 (estimé) | 9 (fixé) | 9 (fixé) | 9 (fixé) |
| CCR (%) | **100** | 83.17 | 94.94 | 86.14 | 86.55 | 83.07 | 72.03 |
| CCR-SCVT (%) | **100** | 97.83 | 98.38 | 86.14 | 93.71 | 84.80 | 86.85 |
| Temps CPU (s) | **0.094** | 61.88 | 72.01 | 35.63 | 4.46 | 64.73 | 52.82 |
| Espace Mémoire (Mb) | 11.40 | 296.93 | 296.93 | 3.82 | 3.17 | 2.99 | **2.75** |

(*) Taux moyen de 5 CCR.

**Tableau 18.** Moyenne et variance de l'erreur de mise à jour des matrices $R$ et $A$ calculées par AP originale et HUP-OAPM-RSM sur l'image hypespectrale synthétique de la Figure 6.

| Critères | $E[\lvert\hat{R}-R\rvert]$ | $var[\lvert\hat{R}-R\rvert]$ | $E[\lvert\hat{A}-A\rvert]$ | $var[\lvert\hat{A}-A\rvert]$ |
|---|---|---|---|---|
| AP originale ($p_{med}$) | 39.16 | 18.28 | 15.02 | 78.41 |
| HUP-OAPM-RSM | **0.08** | **0.4** | **0.008** | **0.001** |

#### *4.3.4.2 Partitionnement de l'image réelle de grande taille*

Pour partitionner cette image hyperspectrale (630 × 1800 pixels), elle a d'abord été divisée en 5040 blocs de 15 × 15 pixels chacun. L'image a été transformée selon la procédure décrite dans la section précédente et est représentée sur la Figure 18 (a). La taille de la matrice de similarité de cette image ne prenant en compte que les exemplaires des classes formées dans les différents blocs est de 11 542 × 11 542 contre 1 134 000 × 1 134 000 pour l'image originale.

La Figure 18 (b) montre le résultat de la partition optimale de l'image hyperspectrale de la Figure 18 (a) obtenue au niveau 7 en maximisant le critère *LN*. Pour cette partition, le nombre estimé de classes est de quatre et reste stable pour le niveau 8. Le Tableau 19 présente le nombre estimé de classes par la méthode HUP-OAPM-RSM pour chaque niveau de partitionnement et la valeur du critère d'optimisation *LN* pour chaque partition. Ce partitionnement a nécessité un total de 1 430.16 s de temps CPU sur un processeur Intel(R) Core(TM) i7-7700 à 3.6 GHz et 16 Go de mémoire.

Les points de la VT des trois classes (algues vertes, algues brunes et classe de substrat) appartiennent à trois classes différentes. Ce résultat met en évidence une quatrième classe "eau".

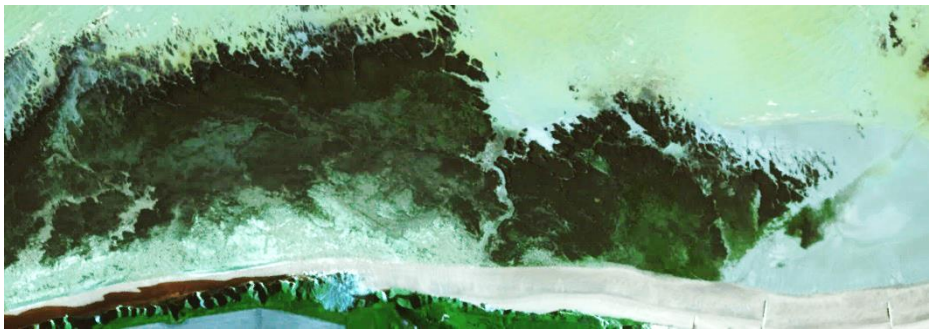
Nous observons sur la Figure 19 que la signature spectrale moyenne de chaque classe formée diffère des autres.

En plus de la partition optimale obtenue, d'autres partitions sont données à chaque niveau qui peuvent contribuer à l'interprétation fine des données selon les besoins des utilisateurs. Par exemple, la partition obtenue au niveau 6, où les classes d'algues vertes et d'algues brunes sont chacune divisées en deux.

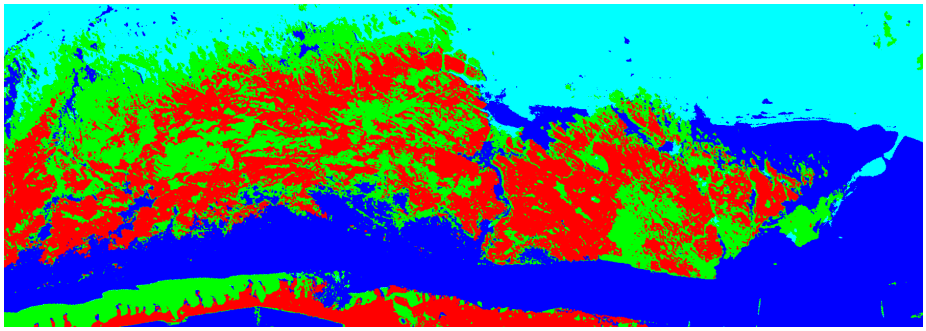
La partition optimale du niveau 7 montre que le taux d'identification est de 100%, en vérifiant les positions des 23 points des trois classes de vérité terrain au sein des classes formées.

Le taux de couverture des algues dans cette image correspond à 47.12% (23.74% pour les algues brunes et 23.38% pour les algues vertes), comme le montre le Tableau 20.

La méthode développée non supervisée appliquée sur l'image transformée donne de meilleurs résultats par rapport aux méthodes semi-supervisées et non supervisées de l'état de l'art appliquées sur l'image originale (voir Tableau 21). Le nombre de classes pour les méthodes semi-supervisées a été fixé à quatre et la métrique utilisée pour calculer la matrice de similarité est la distance euclidienne ($d_2$).



**(a)** Image hyperspectrale transformée affichée en mode RVB



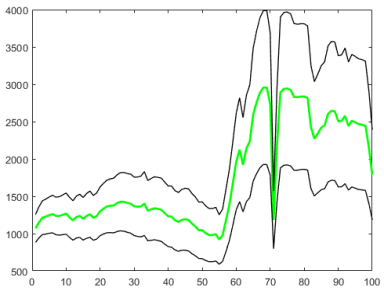
**(b)** Résultat du partitionnement optimal

- Algues vertes
- Algues brunes
- Substrat
- Eau

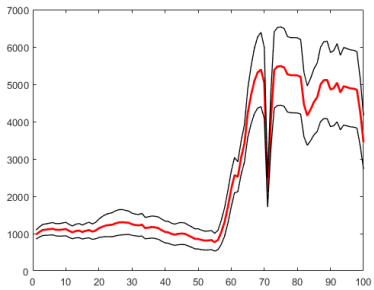
**Figure 18.** Image hyperspectrale transformée 630 × 1800 pixels (bandes visualisées : 5, 27, 48) et son résultat de partitionnement optimal obtenu par la méthode HUP-OAPM-RSM (4 classes, *LN* = 0.18 au niveau 7).

**Tableau 19.** Nombre estimé de classes par la méthode HUP-OAPM-RSM sur l'image hyperspctrale réelle de la Figure 18 par niveau de partitionnement et valeurs du critère d'optimisation *LN* pour chaque partition.

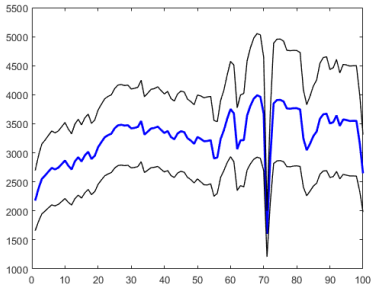
| Niveau | Nombre estimé de classes | Valeur du critère *LN* |
|---|---|---|
| 1 | 1502 | 0.025 |
| 2 | 246 | 0.034 |
| 3 | 101 | 0.041 |
| 4 | 45 | 0.050 |
| 5 | 16 | 0.090 |
| 6 | 6 | 0.150 |
| **7** | **4** | **0.180** |
| 8 | 4 | 0.180 |



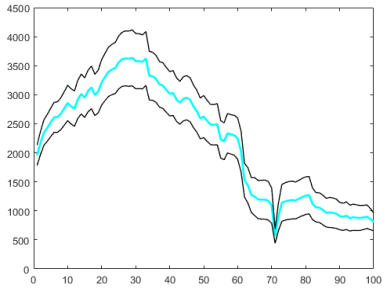
**(a)** Classe Algues vertes



**(b)** Classe Algues brunes



**(c)** Classe Substrat



**(d)** Classe Eau

**Figure 19.** Signature spectrale moyenne ± écart-type de chaque classe obtenue par HAUP-OAP-RSM sur l'image hyperspectrale réelle de la Figure 18 (partition optimale : niveau 7, 4 classes).

**Tableau 20.** Taux de couverture de chaque classe obtenue au niveau 7 par HUP-OAPM-RSM sur l'image hyperspectrale réelle de la Figure 18.

| Classes | Nombre de pixels | Taux de couverture (%) |
|---|---|---|
| Algues vertes | 265092 | 23.38 |
| Algues Brunes | 269184 | 23.74 |
| Substrat | 333425 | 29.40 |
| Eau | 266299 | 23.48 |

**Tableau 21.** Performance de la méthode développée HUP-OAPM-RMS, U-OFCM, S-OFCM, FCM et *K*-means sur l'image hypespectrale réelle de la Figure 18.

| Méthodes | Nombre de classes | CCR (%) |
|---|---|---|
| **HUP-OAPM-RSM** | **4 (estimé)** | **100** |
| U-OFCM | 4 (estimé) | 86.85 |
| S-OFCM | 4 (fixé) | 91.30 |
| FCM (*) | 4 (fixé) | 95.65 |
| *K*-means (*) | 4 (fixé) | 95.65 |

(*) Taux moyen de 5 CCR.

## 4.4 Discussion

Nous avons présenté dans ce chapitre deux nouvelles méthodes adaptatives non supervisées de partitionnement hiérarchique basées sur l'AP adaptées à des ensembles de données de grande taille. Le résultat du partitionnement est déterminé à partir de l'image observée seule sans aucune connaissance *a priori*. La première méthode proposée HUP-OAP permet de calculer d'une manière adaptative la responsabilité et la disponibilité en prenant en compte la présence des individus identiques dans l'ensemble de données. Cependant, elle ne prend pas en considération le nombre d'individus identiques lors de la hiérarchisation, ce qui affecte légèrement les résultats de partitionnement dans certains cas. De plus, elle nécessite un temps de calcul et un espace mémoire importants. Par contre, la seconde méthode tient compte du nombre d'individus identiques dans l'ensemble de données ou les effectifs des classes d'un partitionnement. De plus, elle réduit la taille de la matrice de similarité en transformant les données à partitionner, ce qui permet de diminuer considérablement le temps de calcul et l'espace mémoire.

De plus, ces deux méthodes adaptent la valeur du paramètre de préférence à chaque individu. La procédure de mise à jour des responsabilités et des disponibilités est modifiée sans introduire le facteur d'amortissement. Ces deux méthodes hiérarchiques adaptatives et optimisées permettent de sélectionner la meilleure partition suivant un critère objectif. L'introduction de la procédure de découpage en blocs des données contribue efficacement à partitionner des images de grande taille et peut être appliquée de manière parallèle, ce qui contribue aussi à la réduction du temps de calcul.

Les évaluations des deux méthodes sur des données synthétiques et sur des images hyperspectrales réelles de grande taille montrent que les résultats sont pertinents. Nous démontrons que leurs applications à des images de grande taille donnent le même résultat quel que soit le choix de taille de bloc. Cela prouve qu'elles peuvent être généralisées et, par conséquent, appliquées universellement à un large éventail d'applications sans aucune intervention de l'utilisateur.

Les méthodes proposées répondent donc aux exigences de partitionnement des images de grande taille fournies par les capteurs hyperspectraux modernes. Ainsi, elles peuvent donner à l'utilisateur la possibilité de mieux interpréter les données en fournissant plusieurs partitions hiérarchiques.

# Chapitre 5 : Méthode non supervisée et autonome pour la sélection des échantillons d'apprentissage

## 5.1 Introduction

Les techniques d'apprentissage automatique et d'apprentissage profond ont récemment suscité un grand intérêt dans de nombreux domaines d'application et diverses méthodes ont été publiées, telles que la machine à vecteurs de support (SVM) [31], les K-plus proches voisins (KNN) [92], Les réseaux de neurones artificiels (ANN) [93], les réseaux de neurones convolutifs (CNN) [94], les réseaux de neurones récurrents (RNN) [95], les réseaux de neurones à pointes basés sur la curiosité (CBSNN) [96] et le réseau récurrent Cuckoo Search for Elman (CSERN) [97]. Ces méthodes ont été appliquées dans différents domaines tels que la médecine [98], [99], [100], [101], [102], [103], la photographie numérique [104], [105], l'environnement [15], [106], [107] et la détection de la fraude [108], [109].

Toutes ces méthodes supervisées nécessitent des échantillons d'apprentissage pour effectuer le processus de partitionnement. Par conséquent, leurs performances sont fortement liées à la pertinence des échantillons d'apprentissage sélectionnés. La difficulté d'obtenir des échantillons d'apprentissage fiables a toujours été l'un des principaux facteurs empêchant ces méthodes d'atteindre une précision élevée. Ainsi, leur sélection est une phase critique de toutes les méthodes de partitionnement supervisées. En effet, l'efficacité et la fiabilité de ces méthodes sont significativement affectées lorsque ces échantillons ne sont pas correctement sélectionnés. L'utilisation d'échantillons d'apprentissage biaisés ou simplifiés conduit systématiquement à un partitionnement non rigoureux des données [110], [111], [33]. Ce problème doit être abordé de manière rigoureuse ; sinon, des résultats de partitionnement non fiables ou même faux conduiront automatiquement à des résultats qui ne sont pas proches de la réalité. Enfin, les méthodes supervisées ne peuvent pas être simplement utilisées s'il est difficile d'obtenir des échantillons d'apprentissage.

Pour remédier aux inconvénients des méthodes de partitionnement supervisées, lorsque les échantillons d'apprentissage sont indisponibles, peu fiables ou insuffisants [23], une méthode préliminaire non supervisée et autonome peut être introduite pour la sélection objective de ces échantillons [33]. Le développement de cette méthode répond à des besoins réels exprimés par plusieurs utilisateurs.

En fait, les méthodes non supervisées telles que définies dans le chapitre 2 de la partie I peuvent répondre à ce besoin car elles ne nécessitent pas de connaître le nombre de classes, les échantillons d'apprentissage associés (données étiquetées), ou toute autre connaissance *a priori*. Le nombre de classes est estimé objectivement en fonction d'un critère d'optimisation donné ou de plusieurs critères d'optimisation. Il est plus approprié d'explorer l'analyse des données et de découvrir ainsi un nouveau phénomène qui peut être ajouté comme une nouvelle classe d'apprentissage ou une observation importante pouvant contribuer à l'explication d'un phénomène nouveau. Dans ce cas, l'utilisateur a la possibilité de sélectionner les classes qui l'intéressent après le processus de partitionnement et non avant. Par conséquent, cette catégorie conduit à une analyse fine et détaillée des données observées et ne limite pas l'étude aux classes connues. Pour ces raisons, cette catégorie de méthodes est nécessaire comme étape préliminaire à une méthode supervisée, pour faciliter l'accès aux échantillons d'apprentissage et ainsi répondre aux besoins réels générés par certains domaines applicatifs, tels que l'observation de la Terre, où les zones à étudier sont très grandes et/ou difficiles d'accès. En effet, cette étape permet d'obtenir des solutions précises, objectives et cohérentes qui reflètent le contenu informationnel réel des images indépendamment des données de la VT ou des échantillons d'apprentissage qui peuvent être biaisés ou/et simplifiés ou inexistants.

Afin d'apporter une solution au problème de la sélection des échantillons d'apprentissage pour les méthodes de partitionnement supervisées, nous proposons dans ce chapitre une nouvelle méthode non supervisée, autonome et objective. Ainsi, la méthode de partitionnement hiérarchique et non supervisée HUP-OAP est d'abord appliquée sur les données disponibles, puis trois alternatives sont proposées pour la sélection des échantillons d'apprentissage, basées sur la moyenne et l'écart-type des distances entre les individus d'une classe et leur exemplaire. Les trois ensembles obtenus sont appelés "Homogénéité forte d'échantillons d'apprentissage : SH-TS", "Homogénéité modérément forte d'échantillons d'apprentissage : MSH-TS" et "Homogénéité d'échantillons d'apprentissage : H-TS".

Pour évaluer la pertinence de cette méthode non supervisée et autonome, nous choisissons de l'associer comme étape préliminaire à trois méthodes supervisées bien connues nécessitant une étape de sélection d'échantillons d'apprentissage, qui sont SVM, KNN et ANN [31], [92], [93]. Nous présentons les résultats obtenus sur trois bases de données : la première est constituée de données IRIS, largement utilisées pour l'évaluation des algorithmes de partitionnement (Figure 21

page 130), la seconde est l'image hyperspectrale synthétique de la Figure 2 (page 49) déjà utilisée de taille 64 × 64 pixels × 54 bandes spectrales et la troisième est l'image hyperspectrale réelle de grande taille de la Figure 12 (page 87). Dans le cas de l'évaluation utilisant les données IRIS, les résultats disponibles de certaines méthodes d'apprentissage profond de l'état de l'art sont également donnés et comparés.

L'utilisation de ces bases de données est très représentative du problème posé dans ce chapitre, qui concerne *la correction des VT biaisées ou simplifiées* ou *la fourniture de données de la VT dans le cas où elles ne sont pas disponibles ou pas suffisantes*. En plus de l'intérêt que des données de la VT fiables peuvent offrir à l'utilisateur, elles sont également essentielles pour l'évaluation objective et la validation des résultats de partitionnement [33], afin de ne pas disqualifier à tort un algorithme de partitionnement quelle que soit sa catégorie, c'est-à-dire supervisé, semi-supervisé ou non supervisé [112].

Le reste de ce chapitre décrit la méthode de sélection des échantillons d'apprentissage proposée et présente les résultats de son évaluation.

## 5.2 Approche proposée

Nous présentons dans cette section la méthode non supervisée et autonome proposée pour la validation et la sélection des échantillons d'apprentissage requis pour l'étape d'apprentissage d'une méthode supervisée. L'intérêt principal de cette approche est de fournir à l'utilisateur des échantillons d'apprentissage de toutes les classes présentes dans l'ensemble de données. Ensuite, c'est à l'utilisateur de sélectionner les classes qui l'intéressent en fonction de ses besoins afin de disposer des échantillons d'apprentissage correspondants de manière autonome et objective. Cette approche est plus appropriée et objective pour une interprétation complète et non biaisée des données.

Dans un premier temps, nous présentons les principaux travaux de l'état de l'art relatifs aux méthodes de partitionnement supervisées, puis nous décrivons la méthode proposée pour la sélection des échantillons d'apprentissage.

### 5.2.1 Travaux associés

Ces dernières années, les méthodes supervisées ont attiré une grande attention et plusieurs méthodes ont été développées. Ces méthodes plutôt paramétriques peuvent être divisées en deux

catégories principales : les algorithmes d'apprentissage automatique et d'apprentissage profond. Parmi les méthodes d'apprentissage automatique, les méthodes SVM [31], KNN [92] et ANN [93] sont les plus utilisées. Le SVM a suscité un grand intérêt au cours de la dernière décennie et a été appliqué à divers domaines applicatifs. Il est basé sur une théorie d'apprentissage statistique et sur le principe de la minimisation du risque structurel. Cette méthode vise à identifier, à partir d'échantillons d'apprentissage, l'emplacement des frontières de décision, également connues sous le nom d'hyperplans qui génèrent une séparation optimale des classes [113], [114]. La performance de cette méthode dépend de trois paramètres internes qui sont la fonction noyau, le paramètre de marge douce et le paramètre d'optimisation, et évidemment de la qualité des échantillons d'apprentissage. L'algorithme KNN est l'un des algorithmes d'apprentissage automatique les plus simples. L'idée de l'algorithme KNN consiste à assigner un individu à la classe d'apprentissage, dont le nombre d'objets est la majorité dans un espace limité par un nombre $K$ de plus proches voisins. Le paramètre $K$ indiquant le nombre de plus proches voisins est fixé par l'utilisateur. L'algorithme ANN est un modèle mathématique prédictif non linéaire inspiré de la structure neurologique du cerveau humain. La structure d'un ANN est une régression non linéaire des variables cibles qui est construite sur les variables de décision. Les noyaux d'une structure ANN sont : (1) les couches cachées qui sont composées d'un certain nombre de nœuds cachés et (2) les fonctions d'activation qui traitent et extraient des informations explicites entre les caractéristiques et les variables cibles.

Pour les approches d'apprentissage profond, plusieurs méthodes ont été proposées, qui exploitent la transformation non linéaire des données à travers plusieurs couches. Parmi elles, la méthode des réseaux neuronaux convolutifs (CNN) est la plus connue. La méthode CNN contient de nombreuses couches, notamment des couches convolutionnelles, des couches de sous-échantillonnage et des couches d'activation. L'objectif d'un CNN est d'affecter un individu à la distribution de probabilité ou à l'étiquette de sortie en fonction de la caractéristique représentative extraite.

Plusieurs autres méthodes supervisées ont été proposées dans la littérature, par exemple dans [96], les auteurs proposent un réseau de neurones à pointes (SNN : Spiking Neural Network) basé sur la curiosité (CB). Cette méthode nommée CBSNN permet de réduire les contraintes d'application en temps réel du SNN. Dans le CBSNN, le réseau est d'abord entraîné avec des

principes de plasticité biologiquement plausibles, afin d'obtenir des estimations de nouveauté pour tous les échantillons en une seule étape. Deuxièmement, le CBSNN apprend de manière répétée les échantillons dont les estimations de nouveauté sont supérieures au seuil de nouveauté et met activement à jour les estimations de nouveauté des échantillons en fonction des résultats de l'apprentissage en 5 étapes. Troisièmement, CBSNN réapprend tous les échantillons en une seule étape pour éviter le sur-ajustement des nouveaux échantillons et l'oubli des échantillons appris. Enfin, les deux dernières étapes sont répétées jusqu'à ce que le réseau converge. Cette méthode nécessite des échantillons d'apprentissage et quatre paramètres sont à fixer par l'utilisateur et leur choix affecte les performances de l'algorithme.

Dans [97], les auteurs ont proposé un nouvel algorithme de recherche métaheuristique appelé Cuckoo Search (CS) pour surmonter les inconvénients de RNN. Ces inconvénients sont la vitesse de convergence lente et l'impossibilité de trouver le minimum global de la fonction d'erreur, car la descente du gradient peut être bloquée dans des minima locaux. Cette méthode basée sur le comportement de l'oiseau coucou pour entraîner le réseau récurrent d'Elman (ERN) et le réseau récurrent d'Elman à rétropropagation (BPERN) nécessite le réglage de cinq paramètres par l'utilisateur. Dans [115], une méthode SuperTML est proposée, pour traiter le problème de classification sur des données tabulaires. Tout d'abord, les caractéristiques sont projetées dans un encastrement bidimensionnel sous forme d'image, puis cette image est introduite dans des modèles CNN bidimensionnels pour la classification. Cette dernière méthode traite automatiquement les données catégorielles et les valeurs manquantes dans les données tabulaires, sans avoir besoin de les prétraiter en valeurs numériques. Cette méthode est également paramétrique puisque plusieurs paramètres doivent être fixés par l'utilisateur.

Les performances de toutes les méthodes de partitionnement supervisée avec ou sans apprentissage profond dépendent fortement de la qualité et du nombre d'échantillons d'apprentissage sélectionnés pour les entraîner, avec la difficulté supplémentaire d'ajuster les paramètres internes. Dans la plupart des cas, les utilisateurs de ces méthodes supposent que toutes les étiquettes associées aux modèles d'apprentissage sont correctes, sans aucune évaluation préalable. Malheureusement, en pratique, la connaissance des échantillons étiquetés des classes requises par les méthodes supervisées n'est pas toujours accessible, en particulier pour les images aériennes hyperspectrales comportant de grandes surfaces paysagères. De plus, comme indiqué

dans [32], [33], [38], ces connaissances *a priori* sont figées et ne permettent pas la découverte des nouvelles classes pertinentes. Dans ce cas, l'introduction de ces connaissances comme données d'entrée peut être considérée comme une contrainte et ne reflète pas souvent la réalité des données observées. En outre, tous ces algorithmes nécessitent le réglage subjectif d'un ou plusieurs paramètres. Ainsi, le réglage des paramètres doit être effectué avec des échantillons d'apprentissage fiables. En conclusion, la fiabilité des échantillons d'apprentissage sélectionnés est d'une importance fondamentale pour un apprentissage adéquat et un réglage approprié des paramètres internes de la méthode de partitionnement.

Compte tenu de ces nombreux avantages qui répondent parfaitement aux problèmes réels du partitionnement des données, une méthode comme HUP-OAP est plus adaptée pour plusieurs raisons. Premièrement, elle partitionne les données sans aucune connaissance *a priori*, non paramétrique et stable. De plus, elle est applicable à des données de grande taille de manière parallèle quel que soit le type de données représentées par des attributs quantitatifs.

La méthode de sélection d'échantillons d'apprentissage proposée est réalisée en quatre étapes en partitionnant de manière parallèle, l'ensemble de données quantitatives de grande taille ; répondant ainsi aux besoins de l'utilisateur dans un temps court. *La première étape* divise l'ensemble de données en blocs en choisissant une taille identique, sans chevauchement entre eux. *La deuxième étape* consiste à détecter en parallèle toutes les classes de chaque bloc de données par UP-OAP, afin de découvrir toutes les classes présentes dans chaque bloc et leurs exemplaires. *La troisième étape* fusionne les classes des blocs en utilisant leurs exemplaires sélectionnés par HUP-OAP pour fournir la partition optimale finale. *La dernière étape* valide ou sélectionne les échantillons d'apprentissage de manière autonome à partir de la partition optimale obtenue. L'organigramme de la Figure 20 résume les quatre principales étapes de cette méthode.

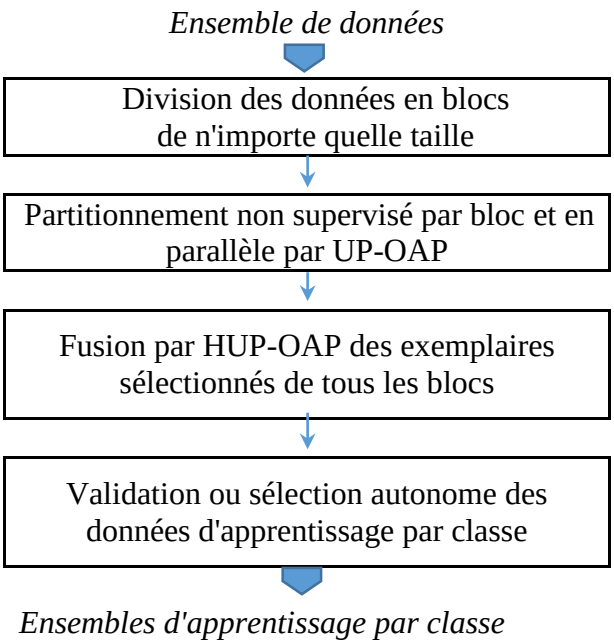



**Figure 20.** Organigramme de la méthode non supervisée et autonome de validation ou de sélection des échantillons d'apprentissage.

### 5.2.2 Sélection des échantillons d'apprentissage

Soit $P_{opt}$ la partition optimale obtenue par HUP-OAP, qui est formée de $M$ classes $C_i$ et qui sera utilisée pour la sélection des ensembles d'apprentissage :

$$\boldsymbol{P_{opt} = \{C_1, C_2, \dots, C_M\}}$$

Afin de donner à l'utilisateur, la possibilité de sélectionner les échantillons d'apprentissage en fonction des contraintes de l'application et du degré de précision requis, nous proposons trois possibilités pour la sélection des échantillons d'apprentissage de chaque classe existante. Le choix sera en fonction du degré d'homogénéité souhaitée des échantillons. Ainsi, trois niveaux d'homogénéité, selon les besoins de l'utilisateur, sont définis comme suit :

– ***Echantillons d'apprentissage par Homogénéité Forte (SH-TS) :***

Dans ce cas, les échantillons d'apprentissage de la classe $C_i$ sont dits fortement homogènes ou purs si :

$$d_{m_i} - \frac{d_{\sigma_i}}{2} \leq d(x_j, E_i) \leq d_{m_i} + \frac{d_{\sigma_i}}{2} \tag{5.1}$$

où $d_{m_i}$ et $d_{\sigma_i}$ sont respectivement, la moyenne et l'écart-type des distances $L_1$-norme, $d_1$, entre les individus $x_j$ de la classe $C_i$ et son exemplaire $E_i$.

Dans ce cas, le nombre sélectionné d'échantillons d'apprentissage par classe $C_i$ est noté $N_{1i}$ et le nombre total d'échantillons sélectionnés est :

$$\boldsymbol{N_1 = \sum_{i=1}^{M} N_{1i}}$$

– ***Echantillons d'apprentissage par Homogénéité Modérément Forte (MSH-TS) :***

Les échantillons d'apprentissage sélectionnés d'une classe $C_i$ doivent vérifier dans ce cas la condition suivante :

$$d_{m_i} - d_{\sigma_i} \leq d(x_j, E_i) \leq d_{m_i} + d_{\sigma_i} \tag{5.2}$$

Dans cette condition, le nombre d'échantillons d'apprentissage par classe $C_i$ est noté $N_{2i}$ et le nombre total d'échantillons sélectionnés est :

$$\boldsymbol{N_2 = \sum_{i=1}^{M} N_{2i}}$$

– ***Echantillons d'apprentissage par Homogénéité (H-TS) :***

La sélection des échantillons d'apprentissage dans ce cas est étendue à tous les individus de chaque classe $C_i$ de $P_{opt}$, le résultat de partitionnement optimal obtenu par HUP-OAP. Par conséquent, le nombre d'échantillons d'apprentissage sélectionnés suivant le critère d'homogénéité par classe est le total d'individus disponibles $M_i$, $o$ù $M_i = card(C_i)$.

En conclusion, le nombre d'échantillons d'apprentissage sélectionnés augmente en fonction de l'élargissement du degré d'homogénéité, c'est-à-dire $N_{1i} < N_{2i} < M_i$ et $N_1 < N_2 < N$.

## 5.3 Evaluation numérique

Pour montrer la pertinence de la méthode de sélection des échantillons d'apprentissage proposée, appelée TS-HUP-OAP, les résultats du partitionnement avec et sans son introduction

comme étape préliminaire dans le processus d'apprentissage de trois méthodes supervisées sont comparés. Trois bases de données sont utilisées pour prouver ses performances : les données IRIS, l'image hyperspectrale synthétique de la Figure 2 déjà utilisée de taille 64 × 64 pixels × 54 bandes spectrales et l'image hyperspectrale réelle de grande taille de la Figure 12. Nous rappelons que les VT accompagnant ces bases de données ne sont utilisées que pour l'évaluation de l'approche non supervisée proposée. Nous soulignons que le choix de ces trois bases de données nous donne la possibilité de vérifier et commenter objectivement les VT qui les accompagnent. Ces VT posent des problèmes distincts dont nous montrons la résolution par la méthode proposée qui sont : *i*) la vérification de la validité des VT et leur correction quand elles sont biaisées et *ii*) la fourniture de plus d'échantillons pour les compléter quand elles ne sont pas suffisantes.

Trois méthodes supervisées sont choisies pour l'évaluation de TS-HUP-OAP, à savoir SVM, KNN et ANN. Ce choix est guidé par la disponibilité des algorithmes. Comme ces trois méthodes sont paramétriques, nous avons fait varier les paramètres de chacune d'entre elles pour montrer l'impact de leur choix sur les performances de partitionnement et pour obtenir la combinaison optimale des paramètres qui donne les meilleures performances. Le Tableau 22 résume les paramètres d'entrée de chaque méthode à définir par l'utilisateur et les valeurs testées.

**Tableau 22.** Paramètres d'entrée des méthodes supervisées SVM, KNN et ANN.

<table>
<tr><td>Méthodes paramétriques supervisées</td><td colspan="6">Paramètres d'entrée</td></tr>
<tr><td rowspan="2">SVM</td><td colspan="4">Fonction noyau</td><td>Paramètre de marge douce $C$</td><td>Précision de l'optimisation $\alpha$</td></tr>
<tr><td>Linéaire</td><td>Polynomiale</td><td>RBF</td><td>Sigmoïde</td><td>$C \in [0.1, 100]$<br>4 valeurs testées :<br>0.1, 1, 10 et 100</td><td>$\alpha \in [0.0001, 10]$<br>5 valeurs testées :<br>0.0001, 0.1, 0.98, 1 et 10</td></tr>
<tr><td>KNN</td><td colspan="6">Nombre de voisins les plus proches $K$</td></tr>
<tr><td rowspan="2">ANN</td><td colspan="5">Taux d'apprentissage $\eta$</td><td>Nombre de neurones dans la couche cachée $NH$</td></tr>
<tr><td colspan="5">$\eta \in [0.1, 0.9]$</td><td>$NH \in [1, \zeta]$<br>$\zeta$ : paramètre dépend du nombre d'éléments et de la taille de chaque base de données</td></tr>
</table>

Pour SVM, nous avons d'abord testé quatre types de fonctions noyau : Linéaire, Polynomiale, RBF et Sigmoïde avec $C$ et $\alpha$ fixés. Ensuite, nous avons fait varier $C$ entre 0.1 et 100 tout en fixant le type de fonction noyau et $\alpha$. Enfin, nous avons fait varier $\alpha$ entre 0.0001 et 10 tout en fixant le type de fonction noyau et $C$. Les valeurs de $C$ choisies sont 0.1, 1, 10 et 100 et celles de $\alpha$ sont 0.0001, 0.1, 0.98, 1 et 10.

Pour KNN, nous avons fait varier le nombre de voisins les plus proches $K$ entre 1 et $n$, où $n$ dépend du nombre d'individus dans chaque base de données utilisée.

Pour ANN, d'abord $HN$ a été fixé et le taux d'apprentissage $\eta$ est varié entre 0.1 et 0.9. Ensuite, $\eta$ a été fixé et le nombre de neurones $NH$ est varié entre 1 et $\zeta$.

Pour les évaluations suivantes utilisant les algorithmes supervisés ci-dessus, les individus de chaque classe sélectionnée par HUP-OAP sont divisés en deux groupes, en particulier si le nombre total d'échantillons sélectionnés est $N_1$ ou $N_2$. Le premier, $N_j$ ($j$ = 1 ou 2) est utilisé dans l'étape d'apprentissage et le second, qui correspond au nombre d'individus restants, $N - N_j$, est utilisé pour l'étape d'identification. En effet, dans cette évaluation, nous n'utilisons pas les $N$ individus pour l'étape d'apprentissage, car il ne restera plus d'échantillons pour l'étape d'identification.

Pour chaque méthode, quatre ensembles d'apprentissage sont testés, qui sont les suivants :

- **SH-TS** : Homogénéité forte d'échantillons d'apprentissage.
- **MSH-TS** : Homogénéité modérément forte d'échantillons d'apprentissage.
- **CGT-TS** : VT corrigée en appliquant HUP-OAP où 30% d'individus de chaque classe sont sélectionnés aléatoirement comme échantillons d'apprentissage.
- **OGT-TS** : VT originale biaisée qui accompagne les données où 30% d'individus de chaque classe sont choisis au hasard comme échantillons d'apprentissage.

### 5.3.1 Évaluation sur les données IRIS

Les données Iris [116] sont formées de 3 classes de la VT de 50 individus chacune. Ces classes sont : Iris Setosa (classe $C_1$), Iris Versicolor (classe $C_2$) et Iris Verginica (classe $C_3$), comme le montre la Figure 21. Quatre caractéristiques mesurées en centimètres, représentant la largeur du pétale (PW), la longueur du pétale (PL), la largeur du sépale (SW) et la longueur du sépale (SL) caractérisent chaque type de fleur.


**Figure 21.** Les 3 espèces de fleurs IRIS.

#### *5.3.1.1 Validation de la VT*

Afin de valider la VT originale des données IRIS, nous appliquons d'abord la méthode non supervisée HUP-OAP pour partitionner les 150 individus sans les diviser en blocs en raison de leur petite taille. Pour confirmer la fiabilité de ce résultat de partitionnement, la distance $L_1$-norm entre les individus et chaque barycentre des classes de la VT originale est utilisée.

L'application de la méthode HUP-OAP à cette base de données donne une partition optimale au niveau 3 de la hiérarchie, c'est-à-dire la partition trois. Le nombre de classes de cette partition est correctement estimé à trois. En revanche, ce résultat de partitionnement montre qu'un individu de $C_2$ de la VT est affecté à $C_3$ de la partition optimale et neuf individus de $C_3$ de la VT sont affectés à $C_2$ de la partition optimale. Pour vérifier la fiabilité de ce résultat, nous montrons dans le Tableau 23, la distance $L_1$-norme entre les individus associés par HUP-OAP à $C_2$ et $C_3$ et le barycentre des classes de la VT originale $C_2$ et $C_3$ respectivement. Ce tableau montre que suivant le critère de distance $L_1$-norme, l'individu 87 de la classe de la VT originale $C_2$, devrait plutôt être assigné à la classe de la VT originale $C_3$, puisqu'il est plus proche de $C_3$ que de $C_2$. Ce critère de similarité objectif et pertinent confirme le résultat obtenu par HUP-OAP. La même observation peut être faite pour les neuf individus de $C_3$ de la VT affectés par la méthode HUP-OAP à $C_2$. Sur la base de ces observations objectives, la VT originale peut être considérée comme biaisée et doit donc être corrigée. Après ces corrections, le nombre d'individus dans les classes de la VT corrigées $C_1$, $C_2$ et $C_3$ est respectivement de 50, 58 et 42. Cette VT ou GT corrigée est nommée « CGT ».

Le partitionnement par la méthode HUP-OAP met en évidence que la VT originale ne peut être considéré comme une référence absolue et peut le corriger. Ainsi, cette VT illustre parfaitement l'un des problèmes posés dans ce chapitre, celui d'une VT biaisée. Les sources d'erreurs dans cette VT, qui représentent 6.66%, peuvent être multiples : soit l'extraction de certains attributs est biaisée, soit les caractéristiques utilisées ne sont pas assez discriminantes, soit tout simplement, il y a une erreur dans l'étiquetage des échantillons.

En conclusion, grâce à la méthode de partitionnement non supervisée HUP-OAP utilisant les caractéristiques originales mesurées, la VT originale des données IRIS est objectivement corrigée. Dans ce cas, nous pouvons supposer que la VT corrigée à partir duquel la sélection des échantillons d'apprentissage sera effectuée est cohérent et fiable.

**Tableau 23.** Évaluation de la partition obtenue par HUP-OAP sur les données originales de la VT d'Iris en utilisant le critère de distance $L_1$-norme.

| Individu étiqueté $x_i$ | Classe de la VT originale | $d_1(x_i, \bar{G}_2)$ | $d_1(x_i, \bar{G}_3)$ | Assignation à la vraie classe de la VT par HUP-OAP |
|---|---|---|---|---|
| 87 | **C2** | 1.70 | **1.61** | **C3** |
| 102 | **C3** | **1.62** | 1.64 | **C2** |
| 107 | | **1.92** | 3.54 | |
| 114 | | **1.92** | 1.94 | |
| 120 | | **1.55** | 2.44 | |
| 122 | | **1.68** | 1.84 | |
| 127 | | **1.31** | 1.54 | |
| 134 | | **1.40** | 1.44 | |
| 139 | | **1.31** | 1.59 | |
| 143 | | **1.62** | 1.64 | |

#### *5.3.1.2 Sélection des échantillons d'apprentissage*

Dans cette évaluation, la méthode des échantillons d'apprentissage TS-HUP-OAP est appliquée aux 150 individus disponibles avant d'utiliser les algorithmes supervisés SVM, KNN et ANN. Nous ne montrons que les résultats pour chacun des deux ensembles d'apprentissage sélectionnés dont le nombre d'individus est respectivement $N_1$ et $N_2$.

Le Tableau 24 résume les résultats de la sélection des échantillons d'apprentissage obtenus par la méthode proposée. Le nombre d'individus de chaque classe et leurs échantillons d'apprentissage correspondants sont indiqués. Il est également indiqué les nombres d'échantillons des trois ensembles d'apprentissage potentiels ($N_1$, $N_2$ et $N$) détectés qui sont respectivement de 69, 106 et 150 objets.

**Tableau 24.** Nombre d'échantillons d'apprentissage obtenus par la méthode proposée sur les données IRIS.

| Classes $C_i$ formées par HUP-OAP | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|
| Nombre d'individus dans chaque classe | 50 | 58 | 42 |
| $N_{1i}$ | 20 | 27 | 22 |
| $N_{2i}$ | 33 | 43 | 30 |
| $N_1$ | 69 | | |
| $N_2$ | 106 | | |

#### *5.3.1.3 Évaluation des ensembles d'apprentissage sélectionnés*

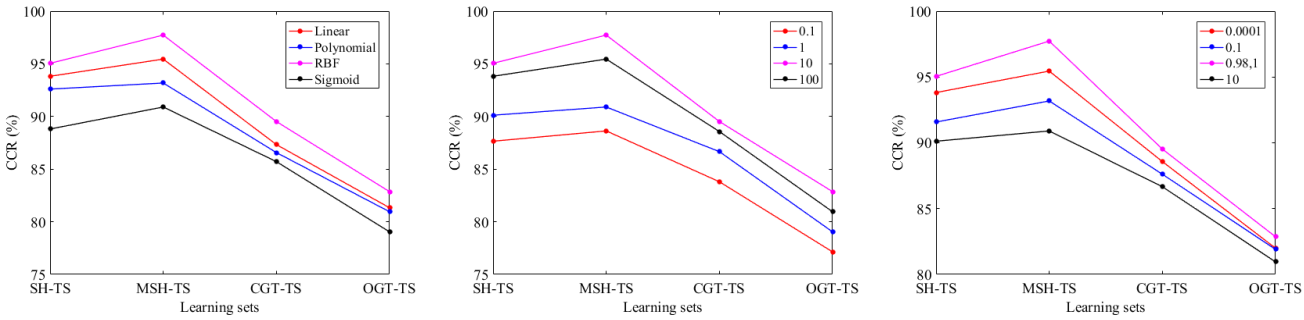
Dans ces évaluations, l'impact d'une VT originale biaisée est également inclus. Pour cela, nous ajoutons deux évaluations sans la sélection des ensembles d'apprentissage par la méthode proposée : *i*) la première prend en compte les classes de la VT biaisée originale accompagnant les données IRIS, où nous avons nommé l'ensemble d'apprentissage dans ce cas comme OGT-TS et *ii*) la seconde utilise les données de la VT corrigée par HUP-OAP, où nous avons nommé l'ensemble d'apprentissage dans ce cas comme CGT-TS.

- *Résultats SVM avec et sans sélection préliminaire d'échantillons d'apprentissage*

L'application de SVM sur les données IRIS en utilisant les ensembles d'apprentissage SH-TS, MH-TS, CGT-TS et OGT-TS respectivement donne les résultats montrés dans la Figure 22. La Figure 22 (a) montre les performances de SVM obtenus avec des fonctions à noyau Linéaire, Polynomial, RBF et Sigmoïde en termes de CCR en fixant $C$ à 10 et $\alpha$ à 0.98. Ces résultats confirment que le choix non optimisé des paramètres d'entrée du SVM a un impact négatif sur ses performances.

En fixant empiriquement les paramètres d'entrée de la méthode SVM, les Figure 22(a), Figure **22**(b) et 22(c) montrent que les meilleurs résultats sont obtenus avec la fonction noyau RBF, $C = 10$ et $\alpha = 0.98$ quels que soient les échantillons d'apprentissage sélectionnés. Nous observons également que l'introduction de la sélection des échantillons d'apprentissage par TS-HUP-OAP et leur utilisation dans le SVM améliore considérablement ses performances. 97.73% est la meilleure performance obtenue par la fonction noyau RBF en utilisant l'ensemble d'apprentissage MSH-TS contre 95.05% pour SH-TS. Ces résultats montrent également que le CCR utilisant les données de la VT biaisée, c'est-à-dire l'ensemble d'apprentissage OGT-TS (82.86%), est inférieur au CCR

utilisant les données de la VT corrigée, c'est-à-dire CGT-TS (89.52%). Cela correspond à une amélioration non négligeable de 6.66%. De plus, les résultats par SVM sans la sélection des échantillons d'apprentissage sont les plus mauvais en utilisant l'ensemble d'apprentissage OGT-TS sélectionné aléatoirement à partir des données OGT quel que soit l'ensemble des paramètres utilisés. Nous notons que pour les ensembles d'apprentissage SH-TS, MSH-TS et CGT-TS les taux sont calculés en utilisant les données de la VT corrigée comme référence et pour l'ensemble d'apprentissage OGT-TS les taux sont calculés en utilisant les données de la VT originale. La meilleure performance du SVM avec MSH-TS (97.73%) fournit une amélioration importante par rapport aux résultats obtenus avec CGT-TS (89.52%) ou OGT-TS (82.86%). Pour les performances de CGT-TS et de OGT-TS, 30% des échantillons d'apprentissage sont choisis aléatoirement dans chaque classe. Par rapport à 97.73% cela correspond donc à une amélioration significative de 8.21% et 14.87% respectivement.




**(a)** En variant la fonction noyau ($C = 10$, $\alpha = 0.98$)

**(b)** En variant $C$ (Kernel RBF, $\alpha = 0.98$)

**(c)** En variant $\alpha$ (Kernel RBF, $C = 10$)

**Figure 22.** CCR du SVM en faisant varier ses trois paramètres d'entrée pour chaque ensemble d'apprentissage des données IRIS : SH-TS, MSH-TS, CGT-TS et OGT-TS.

- *Résultats KNN avec et sans sélection préliminaire d'échantillons d'apprentissage*

La performance de la méthode KNN en fonction du paramètre d'entrée $K$ est étudiée en utilisant les quatre ensembles d'apprentissage comme le montre la Figure 23. Les résultats de partitionnement sont donnés pour les valeurs de $K$ variant de 1 à 35. Le meilleur résultat est obtenu avec $K$ égal à 7, 9, 11 et 13, en utilisant l'ensemble d'apprentissage MSH-TS avec un CCR de 97.73%. Les meilleurs CCR avec CGT-TS (93.33%) et OGT-TS (88.57%) sont respectivement obtenus avec $K$ égal à 5, 11, 13, 17 et avec $K$ égal à 5. Pour les performances de CGT-TS et de OGT-TS, les 30% des échantillons d'apprentissage sont sélectionnés de manière aléatoire dans

chaque classe. Par conséquent, la sélection des échantillons d'apprentissage par la méthode développée donne les meilleurs résultats par rapport aux ensembles d'apprentissage CGT-TS et OGT-TS quelle que soit la valeur de *K*. Cela représente une amélioration de 4.4% et 9.16% respectivement. Par ailleurs, nous observons également que les performances de KNN sont faibles quelle que soit la valeur de *K*, lorsque OGT-TS est utilisé et que le choix des échantillons d'apprentissage est aléatoire. Ces résultats confirment l'intérêt d'utiliser la méthode de sélection des échantillons comme une étape préliminaire à l'apprentissage des méthodes supervisées.

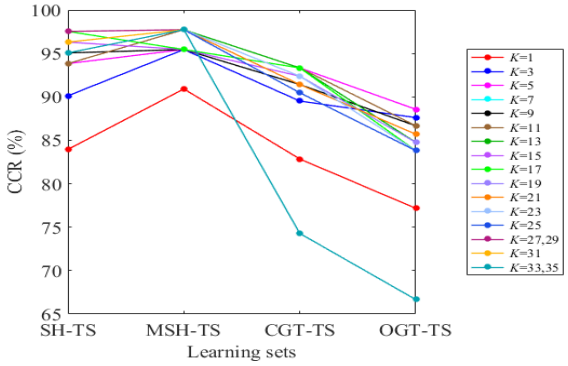



**Figure 23**. CCR obtenus par KNN en faisant varier le paramètre *K* pour chaque ensemble d'apprentissage des données IRIS.

- *Résultats ANN avec et sans sélection préliminaire d'échantillons d'apprentissage*

Dans cette évaluation, l'intérêt de la sélection des échantillons d'apprentissage par la méthode proposée à travers la méthode ANN supervisée et paramétrique est discuté. La sensibilité des résultats du partitionnement en fonction du choix de deux de ses paramètres d'entrée ($\eta$, $HN$) est également analysée.

La Figure 24 (a) montre le CCR en fonction du taux d'apprentissage, $\eta$, en le faisant varier dans l'intervalle [0.1, 0.9] et pour $HN$ fixé à 3 par défaut. Les résultats sont donnés pour chaque ensemble d'apprentissage SH-TS, MS-TH, CGT-TS et OGT-TS. En fixant $HN$ à 3, le meilleur résultat est obtenu pour MSH-TS lorsque $\eta = 0.2$. Nous pouvons voir qu'en faisant varier le paramètre du taux d'apprentissage, le CCR varie et le résultat optimal de chaque ensemble d'apprentissage est obtenu pour différentes valeurs de $\eta$. Par exemple, en utilisant les ensembles d'apprentissage SH-TS et MSH-TS, le CCR optimal est obtenu lorsque $\eta = 0.2$, avec des valeurs

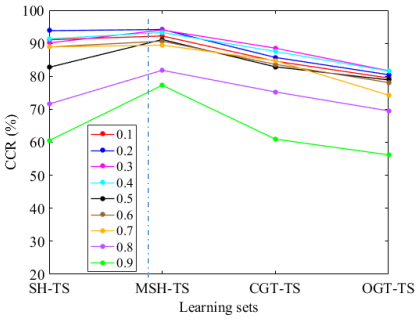
de 93.83% et 94.18% respectivement. Par conséquent, en utilisant les ensembles d'apprentissage CGT-TS ou OGT-TS sans les échantillons sélectionnés par TS-HUP-OAP, le CCR optimal est obtenu pour $\eta$ = 0.3, avec des valeurs de 88.52% et 81.67% respectivement. Les performances de CGT-TS et OGT-TS, sélectionnés aléatoirement par rapport à SH-TS et MSH-TS sont moins bonnes. Par exemple, par rapport à SH-TS, il y a une perte de -5.31% et - 12.51% respectivement.

Dans la Figure 24 (b), les CCR sont obtenus en faisant varier le paramètre $HN$ dans l'intervalle [1, 10] et en fixant $\eta$ à 0.3 lorsque les ensembles d'apprentissage SH-TS et MS-TS sont utilisés et à 0.2 dans le cas de CGT-TS et OGT-TS. Ces résultats prouvent une fois de plus l'intérêt de la sélection des échantillons d'apprentissage par la méthode proposée comme étape préliminaire du processus d'apprentissage.

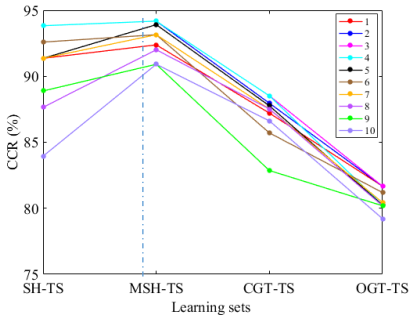
La Figure 24 (b) montre également que le meilleur résultat est obtenu pour $HN$ = 2, 3 et 4 en utilisant l'ensemble d'apprentissage MSH-TS avec un CCR de 94.18%.

Dans la Figure 24 (a) et la Figure 24 (b), la méthode développée donne les meilleurs résultats contre les ensembles d'apprentissage CGT-TS et OGT-TS quelle que soit la valeur de $\eta$ et $HN$.

L'association de l'étape de sélection des échantillons avec la méthode ANN est également confirmée avec une amélioration de 5.66% et 12.51% (94.18% au lieu de 81.67%) respectivement.




**(a)** En variant $\eta$ ($HN = 3$)




**(b)** En variant $HN$ ($\eta = 0.2$ pour SH-TS ou MSH-TS et $\eta = 0.3$ pour CGT-TS ou OGT-TS

**Figure 24.** CCR obtenus par ANN en fonction de ses deux paramètres d'entrée suivant les quatre ensembles d'apprentissage des données IRIS.

En conclusion, les résultats résumés dans le Tableau 25 et la Figure 25 prouvent la pertinence de la méthode de sélection des échantillons d'apprentissage proposée pour l'application des méthodes supervisées, comme les algorithmes SVM, KNN et ANN. En effet, par exemple, le taux obtenu quelle que soit la méthode utilisant l'ensemble CGT-TS sans la procédure de sélection est inférieur à ceux obtenus avec celle-ci, quel que soit le nombre d'échantillons utilisés ($N_1$ ou $N_2$). Ainsi, l'application de l'algorithme TS-HUP-OAP proposé représente une amélioration pertinente des performances. Le Tableau 25 montre également que SVM donne les meilleurs résultats contre KNN et ANN quel que soit l'ensemble d'apprentissage utilisé. Il convient de noter que ces meilleurs résultats des trois méthodes supervisées et paramétriques ont été obtenus après une étape préliminaire de réglage empirique de leurs paramètres d'entrée.

Afin de confirmer l'amélioration des performances des méthodes supervisées par la sélection d'échantillons d'apprentissage par la méthode proposée, nous avons étendu l'évaluation à une autre méthode supervisée et à trois méthodes d'apprentissage profond de l'état de l'art, dont les performances sur les données IRIS sont disponibles. Le Tableau 26 montre les performances de ces méthodes de l'état de l'art choisies, de la méthode non supervisée développée et du TS-HUP-OAP par rapport à SVM, KNN et ANN en utilisant l'ensemble d'apprentissage MSH-TS. Nous pouvons remarquer que la méthode non supervisée et autonome HUP-OAP donne un taux de 100% sans aucun paramétrage et aucune étape d'apprentissage.

La méthode supervisée présentée dans [117] donne le résultat le plus faible (84%). Nous notons que pour les méthodes d'apprentissage profond, tous les paramètres d'entrée sont fixés par les auteurs sans aucune explication sur la façon dont ils ont été choisis. Parmi ces méthodes d'apprentissage profond, le CRN donne un taux (94%), qui reste inférieur à celui obtenu par les algorithmes SVM et KNN (97.73%) après avoir introduit la méthode préliminaire de sélection des échantillons d'apprentissage et fixé empiriquement leurs paramètres d'entrée. Sans cette étape préliminaire, leur taux baisse à 82.86% et 88.57% respectivement, ce qui les place en dessous de certaines des méthodes du Tableau 26.

Si nous supposons que tous les échantillons de chaque classe obtenue par la méthode de partitionnement HUP-OAP formeront les échantillons d'apprentissage, alors ces échantillons sont valides à 100%.

**Tableau 25.** CCR (%) des méthodes KNN, SVM et ANN en fonction de la sélection automatique des échantillons d'apprentissage sur les données IRIS avec le réglage optimal de leurs paramètres d'entrée.

| Méthodes | SH-TS | **MSH-TS** | CGT-TS | OGT-TS |
|---|---|---|---|---|
| SVM | 95.05 | **97.73** | 89.52 | 82.86 |
| KNN | 97.53 | **97.73** | 93.33 | 88.57 |
| ANN | 93.83 | **94.18** | 88.52 | 81.67 |

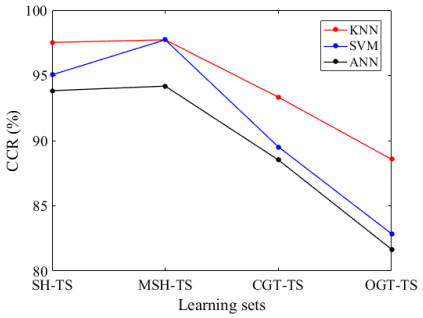



**Figure 25.** CCR obtenus par les méthodes KNN, SVM et ANN en fonction des ensembles d'apprentissage sur les données IRIS avec le réglage optimal de leurs paramètres d'entrée.

**Tableau 26.** Performances des méthodes supervisées avec sélection d'échantillons d'apprentissage, des méthodes d'apprentissage profond et de la méthode non supervisée sur les données IRIS.

| Méthodes / références / années de publication | Nombre ou % de pixels de chaque classe pour l'apprentissage | Nombre ou % de pixels de chaque classe pour l'identification | CCR (%) |
|---|---|---|---|
| **Méthodes supervisées** | | | |
| SVM / [31] / 1995 | MSH-TS ($N_2$) | $N - N_2$ | 97.73 |
| KNN / [92] / 1991 | | | 97.73 |
| ANN / [93] / 1995 | | | 94.18 |
| LE-LSTM : Long Short-Term Memory based on Laplacian Eigenmap / [117] / 2020 | 67% | 33% | 84 |
| **Méthodes d'apprentissage profond de l'état de l'art** | | | |
| CRN : Chemical reaction networks / [118] / 2020 | 100% | 100% | 94 |
| CNN : with superTLM / [115] / 2019 | 80% | 20% | 93.33 |
| CNN : with xgboost GB / [119] / 2016 | | | 93.33 |
| **Méthode non supervisée** | | | |
| HUP-OAP / [112] / 2021 | -------- | ------- | **100** |

### 5.3.2 Évaluation sur l'image hyperspectrale synthétique

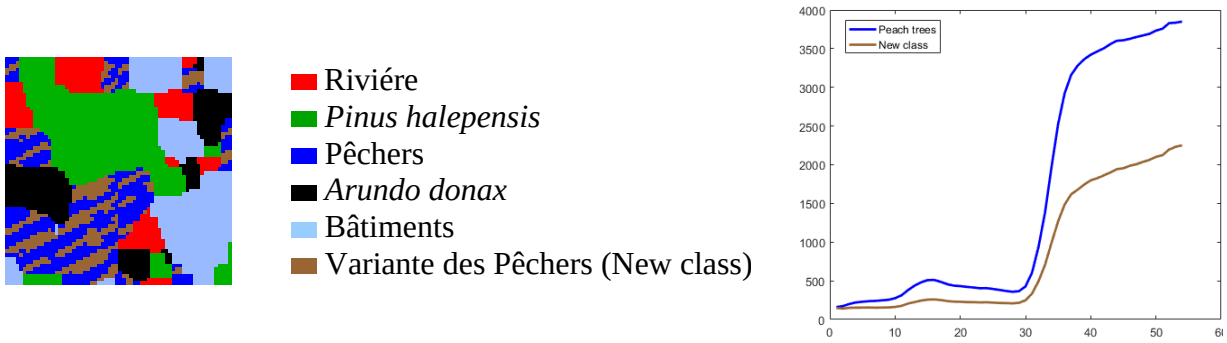
Nous évaluons maintenant l'approche développée sur l'image hyperspectrale synthétique de petite taille 64 × 64 pixels et 54 bandes spectrales déjà présentée dans la Figure 2 du chapitre 2. Pour rappeler, la VT de cette image est formée de 5 classes principales : Rivière, *Pinus halepensis*, Pêchers, *Arundo donax* et Bâtiments. Le problème applicatif visé ici est la détection de la végétation invasive. Nous soulignons que cette VT n'est utilisée que pour évaluer et valider le caractère autonome de l'approche proposée.

#### *5.3.2.1 Validation des données de la VT*

Afin de valider ou de sélectionner les échantillons d'apprentissage, cette image hyperspectrale est d'abord partitionnée par la méthode non supervisée HUP-OAP (voir Figure 26 (a)). Cette partition optimale ($P_2$) est obtenue au niveau 2, montre que la classe Pêchers de la VT est divisée en deux classes. Ainsi le nombre estimé de classes est de 6 au lieu de 5. La Figure 26 (b) montre la pertinence de cette division en se référant aux signatures spectrales moyennes de ces deux classes formées. Ceci est dû au fait que ces deux variantes de la classe des Pêchers n'ont pas été spécifiées lors de l'élaboration de la VT. Ainsi, la méthode proposée met en valeur les informations fournies par l'imagerie hyperspectrale dans le proche infrarouge. Dans ce cas, le CCR corrigé en tenant compte de cette division est de 100%. Au contraire, si l'on considère la VT simplifiée avec 5 classes, le CCR n'est que de 88.31%, alors que l'algorithme met en évidence une information objective et indiscutable. Cet exemple confirme encore une fois l'intérêt d'une méthode non supervisée pour corriger les données de la VT simplifiée en exploitant le pouvoir discriminant de l'imagerie hyperspectrale grâce à son domaine spectral V-NIR.

#### *5.3.2.2 Sélection des échantillons d'apprentissage*

La Figure 27 et le Tableau 27 montrent les échantillons d'apprentissage obtenus par la méthode TS-HUP-OAP proposée. Le nombre possible d'échantillons des trois ensembles d'apprentissage, $N_1$, $N_2$ et $N$ est respectivement de 1515, 3033 et 4096.




**(a)** **(b)**

**Figure 26.** Résultats du partitionnement de l'image hyperspectrale synthétique de la Figure 2 par HUP-OAP, (a) : image partitionnée et (b) : signatures spectrales moyennes (en abscisse le nombre de bandes spectrales et en ordonnée la réflectance) des deux sous-classes de pêchers (Pêchers et New classe de Pêchers).

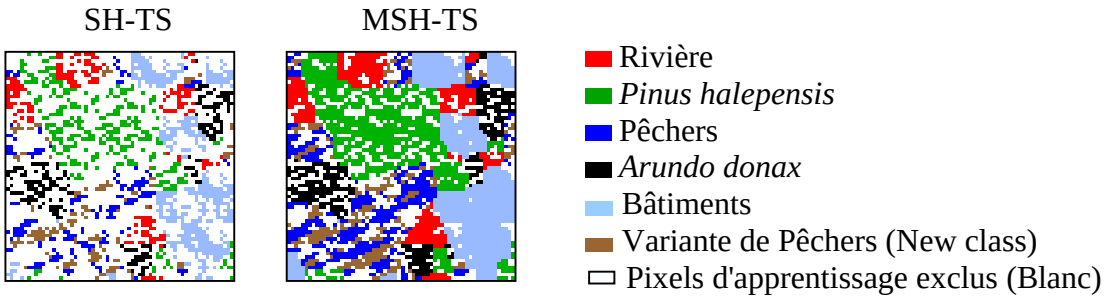



**Figure 27.** Echantillons d'apprentissage sélectionnés de l'image hyperspectrale synthétique de la Figure 2 par la méthode proposée.

**Tableau 27.** Nombre d'échantillons d'apprentissage par classe de l'image hyperspectrale synthétique de la Figure 2 obtenue par la méthode proposée.

| Classes $C_i$ formées par HUP-OAP | Rivière | *Pinus halepensis* | Pêchers | Variante de Pêchers (New class) | *Arundo donax* | Batiments |
|---|---|---|---|---|---|---|
| | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ |
| Nombre d'individus dans chaque classe | 452 | 1068 | 716 | 473 | 500 | 887 |
| $N_{1i}$ | 199 | 327 | 196 | 155 | 198 | 440 |
| $N_{2i}$ | 372 | 755 | 463 | 285 | 328 | 830 |
| $N_1$ | 1515 | | | | | |
| $N_2$ | 3033 | | | | | |

#### *5.3.2.3 Évaluation des ensembles d'apprentissage sélectionnés*

Pour les mêmes raisons que celles mentionnées précédemment, seuls les deux ensembles d'apprentissage de données de la VT validée avec les échantillons $N_1$ ou $N_2$ sont utilisés et les échantillons restants de chacun sont utilisés pour l'étape d'identification.

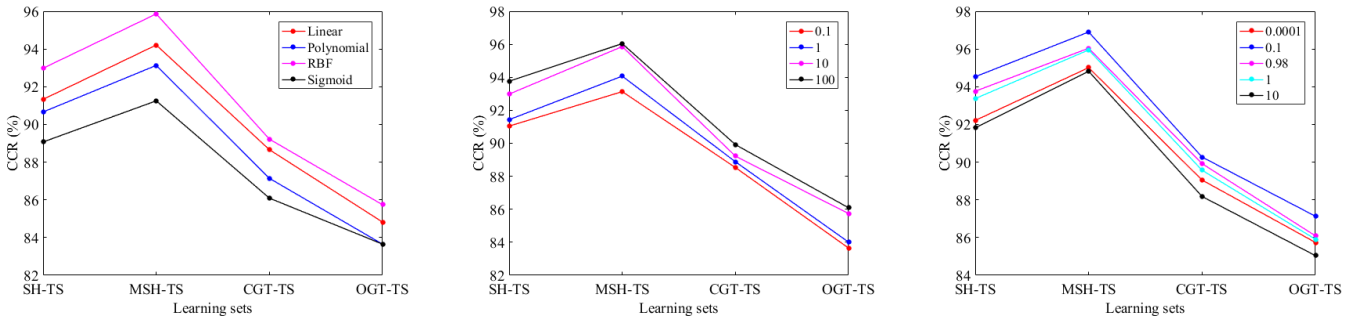
- *Résultats SVM avec et sans sélection d'échantillons d'apprentissage préliminaire*

L'application du SVM sur cette image hyperspectrale synthétique en utilisant les 4 ensembles d'apprentissage donne les résultats montrés dans la Figure 28. La Figure 28 (a) montre le CCR du SVM obtenu avec les fonctions à noyau Linéaire, Polynomial, RBF et Sigmoïde avec les quatre ensembles d'apprentissage en fixant $C$ à 10 et $\alpha$ à 0.98. Nous remarquons que le résultat optimal est obtenu avec la fonction noyau RBF quel que soit l'ensemble d'apprentissage utilisé. Pour cette raison, la fonction RBF est utilisée dans les tests présentés dans la Figure 28 (b) et Figure 28 (c).

Comme nous pouvons le voir (Figure 28), effectuer une sélection des échantillons d'apprentissage par TS-HUP-OAP avant l'application du SVM permet de fournir des performances significativement meilleures que la sélection aléatoire. En particulier, la meilleure performance est obtenue par la fonction noyau RBF en utilisant l'ensemble d'apprentissage MSH-TS (96.90%) avec $C$ = 100 et $\alpha$ = 0.1 (voir Figure 28 (c)). Ces résultats montrent également que le CCR utilisant les données de la VT biaisée, c'est-à-dire avec l'ensemble d'apprentissage OGT-TS (87.13%), est inférieur au CCR utilisant les données GT corrigées, c'est-à-dire avec l'ensemble d'apprentissage CGT-TS (90.27%) avec la fonction noyau RBF, $C$ = 10 et $\alpha$ = 0.1.

De plus, ces résultats confirment également que l'ensemble d'apprentissage OGT-TS sélectionné aléatoirement à partir des données de la VT originale, quels que soient les ensembles de paramètres utilisés, donne les plus mauvais résultats.

Pour la méthode SVM, l'introduction du MSH-TS obtenu par la méthode proposée dans la phase d'apprentissage donne les meilleures performances et apporte ainsi une amélioration de 9.77% (96.9% au lieu de 87.13%) et 6.63% (96.9% au lieu de 90.27%) en considérant les ensembles d'apprentissage OGT-TS et CGT-TS tirés au hasard respectivement.




**(a)** En variant la fonction noyau ($C = 10, \alpha = 0.98$)

**(b)** En varaint $C$ (RBF, $\alpha = 0.98$)

**(c)** En variant $\alpha$ (RBF, $C = 100$)

**Figure 28.** CCR obtenus par SVM en faisant varier ses trois paramètres d'entrée en fonction des quatre ensembles d'apprentissage de l'image hyperspectrale synthétique de la Figure 2.

- *Résultats KNN avec et sans sélection d'échantillons d'apprentissage préliminaire*

Pour KNN, les mêmes tests que pour les données IRIS ont été effectués avec les quatre ensembles d'apprentissage, mais cette fois en faisant varier le nombre de *K* voisins entre 1 et 535. La Figure 29 montre la variation du CCR en fonction de *K* pour les quatre ensembles d'apprentissage. Ces résultats montrent que le meilleur CCR est obtenu avec *K* égal à 5 en utilisant l'ensemble d'apprentissage MSH-TS (95.02%). Par conséquent, l'algorithme KNN associé à la méthode de sélection des échantillons d'apprentissage donne les meilleurs résultats quelle que soit la valeur de *K* par rapport à KNN utilisant uniquement CGT-TS où le meilleur CCR est de 88.64% pour *K* = 9 et OGT-TS où le CCR est de 84.81% pour *K* = 33.

Avec la méthode KNN, les conclusions sont les mêmes que celles données par la méthode SVM. L'introduction de MSH-TS obtenu par la méthode proposée dans le processus d'apprentissage donne la meilleure performance (95.02%) et fournit ainsi une amélioration de 10.21% et 6.38%, en considérant 30% des échantillons d'apprentissage choisis aléatoirement sur OGT et CGT respectivement. Les taux donnés sont calculés en prenant comme référence la VT corrigée.

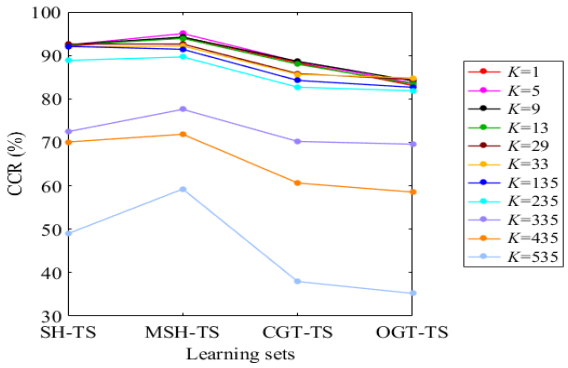
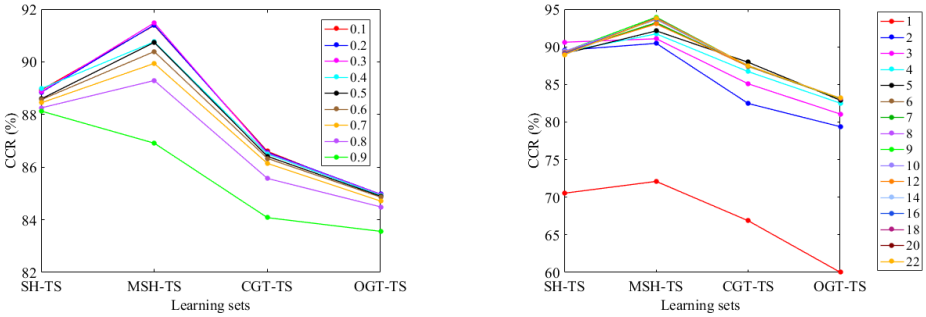



**Figure 29**. CCR obtenus par KNN en faisant varier le paramètre $K$ suivant les quatre ensembles d'apprentissage de l'image hyperspectrale synthétique de la Figure 2.

- *Résultats ANN avec et sans sélection d'échantillons d'apprentissage préliminaire*

Les résultats de l'évaluation de l'algorithme ANN associé aux quatre échantillons d'apprentissage sélectionnés sont présentés à la Figure 30. Le taux d'apprentissage $\eta$ est varié entre 0.1 et 0.9, et le nombre de neurones $HN$ entre 1 et 22. La Figure 30 (a) montre la variation du CCR en fonction de $\eta$, en fixant $HN$ à 18. Ces résultats soulignent que le CCR change en variant l'ensemble d'apprentissage et que le résultat optimal de chaque ensemble d'apprentissage est obtenu pour différentes valeurs de $\eta$. Par exemple, en utilisant l'ensemble d'apprentissage SH-TS, le CCR optimal (88.98%) est obtenu pour $\eta$ = 0.4, au contraire avec MSH-TS le CCR optimal (91.48%) est obtenu pour $\eta$ = 0.3. Enfin, en utilisant les ensembles d'apprentissage CGT-TS et OGT-TS, le CCR optimal est de 86.56% et 84.97% respectivement pour $\eta$ = 0.2.

La Figure 30 (b) montre la variation du CCR en fonction de $HN$ et en fixant $\eta$ à 0.4, 0.3 et 0.2 lors de l'utilisation des ensembles d'apprentissage SH-TS, MSH-TS et CGT-TS ou OGT-TS respectivement. Cette figure montre que le meilleur résultat est obtenu pour $HN$ = 9 en utilisant l'ensemble d'apprentissage MSH-TS avec un CCR de 93.95%. Ces résultats confirment également que l'introduction de l'étape de sélection des échantillons d'apprentissage donne également les meilleurs résultats contre les ensembles d'apprentissage CGT-TS et OGT-TS quelle que soit la valeur de $\eta$ et $HN$. Enfin, nous remarquons que la sélection des échantillons d'apprentissage par homogénéité modérément forte donne les meilleurs résultats dans tous les cas.




**(a)** En variant $\eta$ ($HN = 18$)

**(b)** En variant $HN$ ($\eta = 0.4$ pour SH-TS, $\eta = 0.3$ pour MSH-TS, and $\eta = 0.2$ pour CGT-TS ou OGT-TS)

**Figure 30.** CCR obtenus par ANN en faisant varier ses deux paramètres d'entrée suivant les quatre ensembles d'apprentissage de l'image hyperspectrale synthétique de la Figure 2.

Les résultats résumés dans le Tableau 28 et la Figure 31 confirment également l'amélioration des performances des méthodes supervisées testées sur l'image hyperspectrale synthétique en introduisant la méthode de sélection des échantillons d'apprentissage. En effet, par exemple le CCR obtenu par SVM sans la procédure de sélection (90.27%) est inférieur à ceux obtenus avec celle-ci, quel que soit le nombre d'échantillons utilisés ($N_1$ ou $N_2$), qui sont respectivement de 94.53% et 96.90%. Ainsi, l'application de l'algorithme TS-HUP-OAP proposé représente une amélioration des performances de 4.26% et 6.63% respectivement. La même affirmation peut être faite pour les résultats obtenus en utilisant les algorithmes KNN et ANN. Ce tableau montre également que SVM donne les meilleurs résultats par rapport au KNN et ANN, quel que soit l'ensemble d'apprentissage utilisé.

En considérant le meilleur CCR dans le cas de l'ensemble MSH-TS, les améliorations par rapport aux meilleurs résultats obtenus par un tirage aléatoire d'échantillons de CGT ou OGT sont respectivement de 6.63% et 9.77%. Ces performances sont obtenues avec l'ensemble optimal des paramètres de SVM.

**Tableau 28.** CCR (%) des méthodes KNN, SVM et ANN en fonction de la sélection automatique des échantillons d'apprentissage sur l'image hyperspectrale synthétique de la Figure 2 avec le réglage optimal de leurs paramètres d'entrée.

| Méthodes | SH-TS | **MSH-TS** | CGT-TS | OGT-TS |
|---|---|---|---|---|
| SVM | 94.53 | **96.90** | 90.27 | 87.13 |
| KNN | 92.63 | **95.02** | 88.64 | 84.81 |
| ANN | 90.61 | **93.95** | 87.94 | 84.97 |

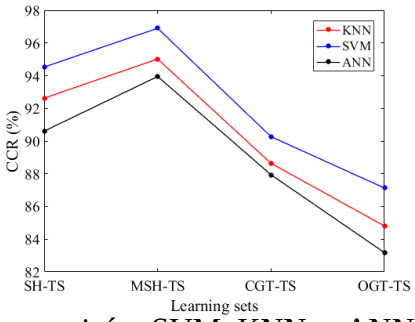



**Figure 31.** CCR des méthodes supervisées SVM, KNN et ANN en fonction des ensembles d'apprentissage sur l'image hyperspectrale synthétique de la Figure 2 avec l'ensemble optimal de paramètres.

### 5.3.3 Evaluation sur l'image hyperspectrale aérienne réelle de grande taille

Le but de cette évaluation est de montrer sur une grande image réelle l'importance de la méthode autonome pour fournir des échantillons de la VT et d'apprentissage dans le cas où il est impossible de les obtenir pratiquement par l'utilisateur. L'objectif est ensuite d'identifier les principales classes existantes dans cette image réelle et de fournir une cartographie précise de leur taux de couverture à utiliser pour le processus d'apprentissage. L'image utilisée ici est celle de la Figure 12.

La partition optimale de cette image est obtenue au niveau hiérarchique 4, où le nombre estimé de classes est de 5. Sur la base de cette partition optimale, le taux de couverture du sol de chaque classe détectée correspond à 19% pour les algues brunes, 25.61% pour les algues vertes, 15.94% pour Roches+Galets, 15.05% pour le sable et 24.39% pour l'eau, comme le montre le Tableau 29. Ce tableau montre également une augmentation significative des échantillons d'apprentissage de chaque classe de la VT par la méthode proposée. Elle remplit donc parfaitement son rôle dans cette application où l'accès à de multiples échantillons d'apprentissage est à la fois difficile et coûteux.

**Tableau 29.** Taux de couverture au sol de chaque classe obtenue au niveau 4 par HUP-OAP sur l'image hyperspectrale réelle de la Figure 12.

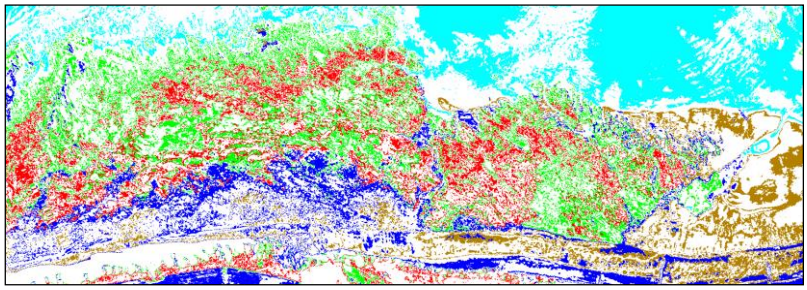
| Classes | Nombre de points collectés au sol par l'opérateur | Augmentation des échantillons de la VT par la méthode proposée | Taux de couverture au sol (%) |
|---|---|---|---|
| Algues vertes | 4 | 290 452 | 25.61 |
| Algues brunes | 10 | 215 509 | 19.00 |
| Roches+Galets | 6 | 180 802 | 15.94 |
| Sable | 3 | 170 617 | 15.05 |
| Eau | 0 | 276 620 | 24.39 |

#### 5.3.3.1 Sélection des échantillons d'apprentissage

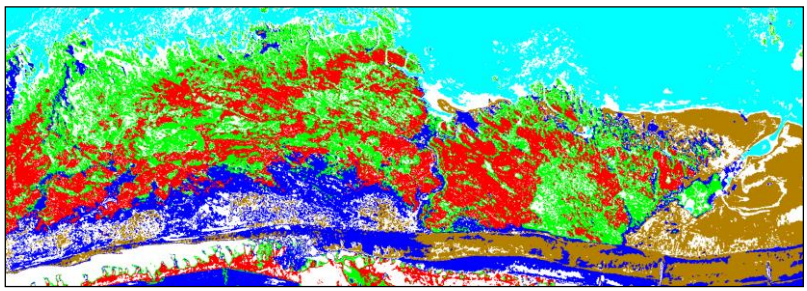
Le Tableau 30 montre les nombres d'échantillons d'apprentissage obtenus par la méthode TS-HUP-OAP proposée. Le nombre possible d'échantillons d'apprentissage $N_1$, $N_2$ et $N$ est respectivement de 489 885, 868 803 et 1 134 000. La Figure 32 illustre les échantillons appartenant aux deux ensembles d'apprentissage sélectionnés SH-TS et MSH-TS avec $N_1$ et $N_2$ pixels respectivement, où tous les pixels exclus de chaque classe de la phase d'apprentissage sont étiquetés en blanc.

**Tableau 30.** Nombre d'échantillons d'apprentissage par classe de l'image hyperspectrale réelle de la Figure 12 obtenue par la méthode proposée.

| Classes $C_i$ formées par HUP-OAP | Algues brunes | Algues vertes | Roches et galets | Sable | Eau |
|---|---|---|---|---|---|
| | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ |
| Nombre d'individus dans chaque classe | 215 509 | 290 452 | 180 802 | 170 617 | 276 620 |
| $N_{1i}$ | 95 315 | 114 045 | 85 402 | 59 803 | 135 320 |
| $N_{2i}$ | 186 633 | 201 639 | 146 773 | 126 991 | 206 767 |
| $N_1$ | 489 885 | | | | |
| $N_2$ | 868 803 | | | | |



SH-TS



MSH-TS

Algues vertes — Algues brunes
Eau — Roches+Galets
Pixels d'apprentissage exclus — Sable

**Figure 32.** Echantillons d'apprentissage sélectionnés de l'image hyperspectrale réelle par la méthode proposée.

#### *5.3.3.2 Évaluation des ensembles d'apprentissage sélectionnés*

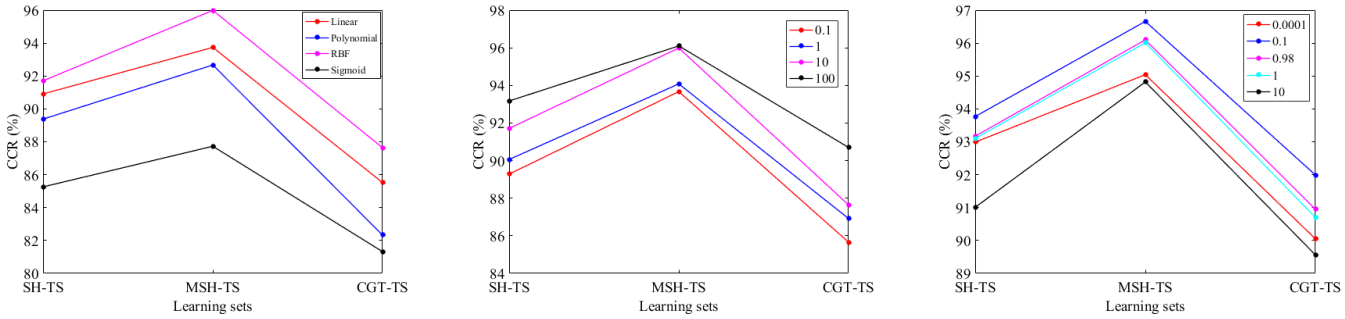
Pour les mêmes raisons que précédemment, seuls les deux ensembles d'apprentissage avec les échantillons $N_1$ ou $N_2$ sont utilisés dans l'étape d'apprentissage et les échantillons restants de chacun sont utilisés pour l'étape d'identification.

Nous notons que dans ces expérimentations, la VT originale n'est formée que de quelques pixels déjà validés ; nous n'avons donc pas besoin de la corriger. Cependant, comme ce nombre de pixels est très faible pour effectuer les deux phases d'apprentissage et d'identification, nous considérons dans ce cas, la partition optimale obtenue par la méthode HUP-OAP comme une VT de référence, que nous avons nommée CGT. Ainsi, CGT-TS sera générée en sélectionnant aléatoirement 30% des pixels de chaque classe pour l'étape d'apprentissage à partir de la VT obtenue par HUP-OAP.

- *Résultats SVM avec et sans sélection d'échantillons d'apprentissage préliminaire*

L'application du SVM sur cette image hyperspectrale réelle en utilisant les trois ensembles d'apprentissage donne les résultats présentés dans la Figure 33. La Figure 33 (a) montre le CCR du SVM obtenu avec les fonctions à noyau Linéaire, Polynomial, RBF et Sigmoïde en fixant $C$ à 10 et $\alpha$ à 0.98. Le résultat optimal est également obtenu avec la fonction noyau RBF quel que soit l'ensemble d'apprentissage utilisé. Le meilleur CCR (95.99%) est obtenu avec la fonction RBF et l'ensemble d'apprentissage MSH-TS. Par conséquent, la fonction RBF est utilisée dans les tests présentés dans la Figure 33 (b) et la Figure 33 (c) en faisant varier $C$ et $\alpha$ respectivement. D'après la Figure 33 (b), le meilleur CCR (96.10%) est obtenu lorsque $C$ = 100 et $\alpha$ = 0.98 et d'après la Figure 33 (c), le meilleur CCR (96.66%) est obtenu lorsque $C$ = 100 et $\alpha$ = 0.1.

Ces résultats d'évaluation confirment comme les tests précédents sur les données Iris et l'image hyperspectrale synthétique que la sélection des échantillons d'apprentissage par TS-HUP-OAP et leur utilisation dans le processus d'apprentissage du SVM donne des performances significativement meilleures que la sélection aléatoire sur l'ensemble des données de la VT originale et corrigée, en particulier, lorsque l'ensemble d'apprentissage MSH-TS est associé à la fonction noyau RBF, $C$ = 100 et $\alpha$ = 0.1.




**(a)** En variant la fonction noyau ($C = 10, \alpha = 0.98$)

**(b)** En variant $C$ (RBF, $\alpha = 0.98$)

**(c)** En variant $\alpha$ (RBF, $C = 100$)

**Figure 33.** CCR obtenus par SVM en faisant varier ses trois paramètres d'entrée selon les trois ensembles d'apprentissage de l'image hyperspectrale réelle de la Figure 12.

- *Résultats KNN avec et sans sélection d'échantillons d'apprentissage préliminaire*

Pour KNN, les mêmes tests que précédemment ont été effectués avec les 3 ensembles d'apprentissage, mais cette fois-ci en faisant varier le nombre de $K$ voisins entre 1 et 1035. La Figure 34 montre la variation du CCR en fonction de $K$. Le meilleur CCR (91.12%) est obtenu avec $K$ égal à 55 en utilisant l'ensemble d'apprentissage MSH-TS.

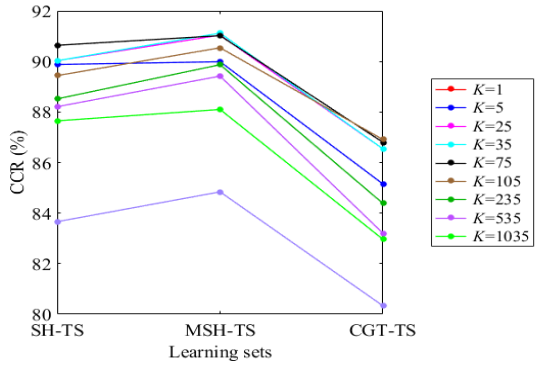


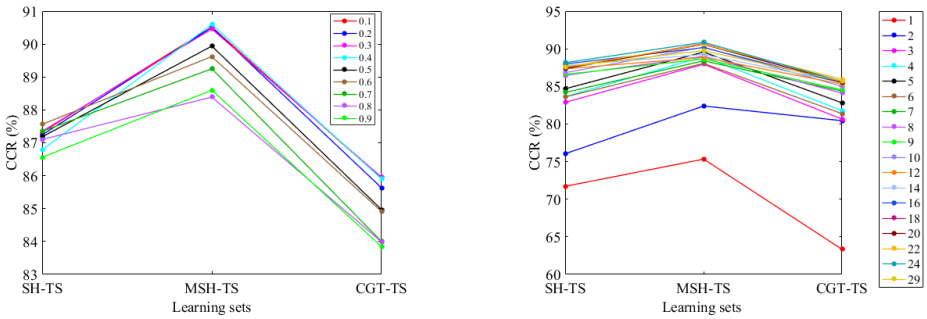

**Figure 34**. CCR obtenus par KNN en faisant varier le paramètre $K$ suivant les ensembles d'apprentissage SH-TS, MSH-TS et CGT-TS de l'image hyperspectrale réelle de la Figure 12.

- *Résultats ANN avec et sans sélection d'échantillons d'apprentissage préliminaire*

Dans ces évaluations avec les trois ensembles d'apprentissage, les performances de l'algorithme ANN sont évaluées en faisant varier ses deux paramètres d'entrée. Premièrement, le nombre de neurones $HN$ est fixé à 22 et le taux d'apprentissage $\eta$ varie entre 0.1 et 0.9. Deuxièmement, le paramètre $\eta$ est fixé à 0.6, 0.4 et 0.3 en faisant varier le paramètre $HN$ entre 1 et 29. Dans ces deux cas, les variations du CCR sont données dans la Figure 35 (a) et la Figure 35

(b) respectivement. Ces résultats confirment à nouveau la forte influence des paramètres $\eta$ et $HN$ sur la qualité des résultats ANN. Par exemple, dans la Figure 35 (a), en fixant $NH$ à 22, le CCR optimal utilisant l'ensemble d'apprentissage SH-TS (87.56%) ou MSH-TS (90.60%) ou CGT-TS (85.95%) est obtenu pour $\eta$= 0.6, $\eta$ = 0.4 et $\eta$ = 0.3 respectivement.

Dans l'exemple de la Figure 35 (b), la variation du CCR en fonction de $HN$ est calculée en fixant $\eta$ à 0.6, 0.4 et 0.3 en utilisant les ensembles d'apprentissage SH-TS, MS-TH et CGT-TS respectivement. Le meilleur résultat avec un CCR de 90.87% est obtenu pour $HN$ = 24 et $\eta$ = 0.6 en utilisant l'ensemble d'apprentissage MSH-TS. De plus, les échantillons d'apprentissage sélectionnés par la méthode proposée donnent les meilleurs résultats par rapport à l'ensemble d'apprentissage sélectionné aléatoirement dans CGT quelle que soit la valeur de $\eta$ et $HN$.




**(a)** En variant $\eta$ ($NH$ = 22)

**(b)** En variant $HN$ ($\eta = 0.6$ pour SH-TS, $\eta = 0.4$ pour MSH-TS et $\eta = 0.3$ pour CGT-TS)

**Figure 35.** CCR obtenus par ANN en faisant varier ses deux paramètres d'entrée selon les ensembles d'apprentissage SH-TS, MSH-TS et CGT-TS de l'image hyperspectrale réelle de la Figure 12.

Les résultats résumés dans le Tableau 31 et la Figure 36 confirment comme les résultats précédents des données IRIS et de l'image hyperspectrale synthétique, la pertinence de la méthode de sélection des échantillons d'apprentissage proposée lors de l'utilisation de méthodes supervisées, comme les algorithmes SVM, KNN et ANN. Nous précisons que les résultats de partitionnement sont obtenus en essayant d'optimiser empiriquement le choix des paramètres d'entrée de chaque algorithme. En effet, par exemple le CCR obtenu par SVM sans la procédure de sélection (91.99%) est inférieur à ceux obtenus avec celle-ci, quel que soit le nombre d'échantillons utilisés ($N_1$ ou $N_2$), qui est respectivement de 93.77% et 96.66%. Ainsi, l'application de l'algorithme TS-HUP-OAP proposé représente une amélioration des performances de 1.78% et 4.67% respectivement. La même remarque peut être faite pour les résultats obtenus en utilisant les algorithmes KNN et ANN.

Ce tableau montre également que le SVM donne les meilleurs résultats par rapport au KNN et ANN, quel que soit l'ensemble d'apprentissage utilisé.

Pour résumer l'ensemble des évaluations de la méthode de sélection d'apprentissage proposée effectuées sur les données IRIS et des images hyperspectrales, nous présentons dans le Tableau 32, le Tableau 33 et le Tableau 34 les principaux résultats des méthodes SVM, KNN et ANN en optimisant empiriquement leurs paramètres d'entrée.

A titre indicatif, le Tableau 35 montre le temps CPU obtenu en appliquant HUP-OAP aux trois bases de données, avec et sans la parallélisation par partitionnement de blocs. La machine utilisée est un processeur Intel(R) Core (TM) i7-7700 avec 3.6 GHz et 16 Go de mémoire. Comme cette machine ne possède que quatre cœurs, les blocs sont donc traités par paquets de 4, c'est-à-dire que chaque paquet de 4 blocs est traité séparément en même temps. La parallélisation de partitionnement permet de réduire considérablement le temps de calcul.

**Tableau 31.** CCR en % des méthodes de partitionnement supervisées SVM, KNN et ANN sur l'image hyperspectrale réelle de la Figure 12 en fonction de la sélection des échantillons d'apprentissage avec le réglage optimal de leurs paramètres.

| Méthodes | SH-TS | **MSH-TS** | CGT-TS |
|---|---|---|---|
| SVM | 93.77 | **96.66** | 91.99 |
| KNN | 90.64 | **91.12** | 86.91 |
| ANN | 88.15 | **90.87** | 85.95 |

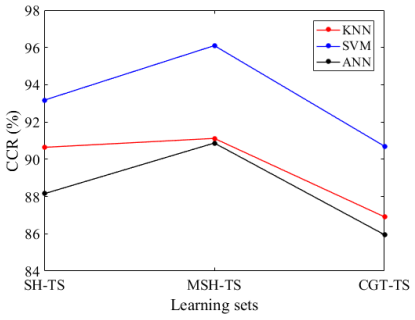



**Figure 36.** CCR des méthodes supervisées SVM, KNN et ANN suivant les ensembles d'apprentissage SH-TS, MSH-TS et CGT-TS sur l'image hyperspectrale réelle de la Figure 12 avec le réglage optimal de leurs paramètres.

**Tableau 32.** Les paramètres SVM donnant les meilleures performances sur les 3 bases de données (IRIS, Images de la Figure 2 et de la Figure 12).

| Bases de données | SH-TS | | | **MSH-TS** | | | CGT-TS | | | OGT-TS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Noyau | $C$ | $\alpha$ | **Noyau** | $\boldsymbol{C}$ | $\boldsymbol{\alpha}$ | Noyau | $C$ | $\alpha$ | Noyau | $C$ | $\alpha$ |
| IRIS | RBF | 10 | 0.98 | **RBF** | **10** | **0.98** | RBF | 10 | 0.98 | RBF | 10 | 0.98 |
| Image synthétique | | 100 | 0.1 | | **100** | **0.1** | | 100 | 0.1 | | 100 | 0.1 |
| Image hyperspectrale réelle | | 100 | 0.1 | | **100** | **0.1** | | 100 | 0.1 | - | - | - |

**Tableau 33**. Le paramètre *K* de KNN fournit la meilleure performance sur les 3 bases de données (IRIS, Images de la Figure 2 et de la Figure 12).

| Bases de données | SH-TS | **MSH-TS** | CGT-TS | OGT-TS |
|---|---|---|---|---|
| IRIS | 17 | **7** | 5 | 5 |
| Image synthétique | 29 | **5** | 9 | 33 |
| Image hyperspectrale réelle | 75 | **55** | 105 | - |

**Tableau 34**. Les paramètres ANN donnant les meilleures performances sur les 3 bases de données (IRIS, Images de la Figure 2 et de la Figure 12).

| Bases de données | SH-TS | | **MSH-TS** | | CGT-TS | | OGT-TS | |
|---|---|---|---|---|---|---|---|---|
| | $\eta$ | $HN$ | $\boldsymbol{\eta}$ | $\boldsymbol{HN}$ | $\eta$ | $HN$ | $\eta$ | $HN$ |
| IRIS | 0.2 | 2 | **0.2** | **2** | 0.3 | 3 | 0.3 | 1 |
| Image synthétique | 0.4 | 3 | **0.3** | **9** | 0.2 | 5 | 0.2 | 14 |
| Image hyperspectrale réelle | 0.6 | 24 | **0.4** | **24** | 0.3 | 22 | - | - |

**Tableau 35.** Temps CPU (s) par HUP-OAP avec et sans parallélisation sur les trois bases de données (IRIS, Images de la Figure 2 et de la Figure 12).

| Bases de données | Nombre de blocs | HUP-OAP sans parallélisation | HUP-OAP avec parallélisation | Réduction du temps en % |
|---|---|---|---|---|
| IRIS | 1 (150 individus) | 0.14 | ----- | ----- |
| | 10 (taille du bloc : 15 individus) | 0.10 | 0.034 | 34 |
| Image synthétique | 1 (64×64 pixels) | 88.17 | ----- | ----- |
| | 16 (taille du bloc : 16×16 pixels) | 3.37 | 0.84 | 25 |
| Image hyperspectrale réelle | 200 (taille du bloc : 63×90 pixels) | 4 932.67 | 1 233.16 | 25 |
| | 1260 (taille du bloc : 30×30 pixels) | 2 389.03 | 595.25 | 25 |

## 5.4 Discussion

Dans ce chapitre, nous avons proposé une méthode non supervisée et autonome pertinente qui apporte des réponses concrètes et pratiques aux difficultés que l'utilisateur peut rencontrer lors de la mise en œuvre d'une méthode de partitionnement supervisée avec ou sans apprentissage profond. En effet, les problèmes concernant les échantillons d'apprentissage auxquels la méthode proposée peut apporter une solution adaptée sont : i) les échantillons d'apprentissage existent mais sont biaisés et/ou insuffisants et ii) les données de la VT sont difficilement accessibles pour des raisons de sécurité ou dans le cas des données de grande taille.

L'objectif de la méthode proposée est d'améliorer les performances des méthodes nécessitant une étape d'apprentissage en fournissant aux utilisateurs des ensembles d'apprentissage faciles à sélectionner en fonction de leur degré d'homogénéité, qui répondent aux précisions requises. Cette méthode est donc réalisée en optimisant un critère objectif sans intervention de l'utilisateur. Elle donne à l'utilisateur la possibilité de découvrir toutes les classes présentes dans une base de données et de sélectionner les classes répondant à ses objectifs en toute connaissance des autres classes existantes qui peuvent apporter un complément d'information sur les phénomènes à analyser et à interpréter. Elle offre donc la possibilité de corriger les VT biaisées ou de générer des VT en cas d'absence de ceux-ci suivant des critères d'optimisation indépendant de tout utilisateur et ainsi éviter toute subjectivité.

Les trois bases de données (données IRIS, une image hyperspectrale synthétique et une image hyperspectrale réelle) sur lesquelles notre méthode est objectivement évaluée illustrent parfaitement les exemples de problèmes qui peuvent être rencontrés lors de la sélection des échantillons d'apprentissage. Les évaluations de la méthode proposée sur ces bases de données illustrent la pertinence des ensembles d'apprentissage créés. En effet, l'application de trois algorithmes supervisés (SVM, KNN et ANN) en introduisant des ensembles d'apprentissage sélectionnés montre que les performances de ces méthodes sont meilleures en utilisant des échantillons d'apprentissage sélectionnés de manière non supervisée qu'en utilisant des échantillons sélectionnés de manière aléatoire. Nous avons montré également que cette méthode non supervisée et autonome donne la possibilité de corriger des données de la VT biaisée ou simplifiée. Elle permet même d'obtenir des échantillons d'apprentissage fiables dans le cas de l'absence de données de la VT ou de leur insuffisance. Ceci illustre parfaitement l'intérêt pratique de cette méthode et met en

évidence le fait que chaque donnée de la VT fournie par l'utilisateur ne doit pas automatiquement être considérée comme une référence absolue comme nous l'avons prouvé avec les trois exemples traités ici.

Pour conclure ces résultats d'évaluation concordants et convaincants, nous soulignons que l'introduction de la méthode autonome non supervisée proposée pour fournir des échantillons d'apprentissage de la VT, leur validation et leur augmentation améliorent les performances des méthodes supervisées avec et sans apprentissage profond.

En perspective, il reste à évaluer le temps de calcul en parallélisant au maximum le partitionnement des blocs, même si ce temps reste dépendant de la puissance de calcul disponible de chaque machine.

# Chapitre 6 : Méthode de partitionnement non supervisée et non paramétrique par association directe et indirecte pour des données de grande taille

## 6.1 Introduction

Les méthodes développées dans le chapitre 4 utilisent des critères d'optimisation plus élaborés, cependant, elles ne tiennent pas compte de la connectivité spatiale entre individus ce qui peut contribuer à certaines ambiguïtés lors de la formation des classes. Nous proposons dans ce chapitre une nouvelle méthode de partitionnement également non supervisée et non paramétrique en définissant de nouveaux critères faisant appel à des notions d'association entre individus ou objets lors du processus d'agrégation. Dans cette méthode, les nouveaux critères permettent d'évaluer le degré de connectivité entre les individus via le lien direct et indirect entre eux, que nous nommons respectivement « association directe » et « association indirecte ». Cette méthode est aussi applicable sur des données de grande taille sans aucune intervention de l'utilisateur est nommée HUP-DIA (partitionnement hiérarchique non supervisée par associations directe et indirecte).

Ce travail apporte les contributions suivantes :

- Introduction des nouveaux critères d'optimisation (associations directe et indirecte) basant sur la mesure de distance entre les individus et description de la procédure de la formation des classes ;
- Introduction d'une approche de partitionnement par blocs pour les données de grande taille et proposition d'une procédure de fusion des classes obtenues sur les blocs ;
- Création d'un partitionnement hiérarchique où le nombre de classes est estimé pour chaque partition afin de permettre une analyse plus fine des données suivant le besoin de l'utilisateur.

Ce chapitre est structuré en quatre sections. Dans la Section 6.2, nous présentons les travaux associés aux approches de partitionnement utilisant la connectivité. Ensuite, dans la Section 6.3 nous décrivons et détaillons la méthode proposée. Puis, dans la Section 6.4 nous évaluons cette méthode sur les images présentées dans le chapitre 4. Finalement, la Section 6.5 conclut les travaux menés du chapitre.

## 6.2 Travaux associés

Dans la littérature, les approches de partitionnement utilisant la notion de connectivité s'appuient généralement sur la théorie des graphes. Elles consistent à créer un graphe à partir de l'ensemble de données à partitionner et d'exploiter les différentes propriétés de ce graphe pour former des classes. L'idée générale consiste à modéliser les individus par les sommets de ce graphe et les caractéristiques (distance, similarité, etc.) reliant ces individus par des arêtes.

Le dénominateur commun de ces méthodes est leur caractère paramétrique et semi-supervisé, qui met un frein à leur utilisation lorsque le nombre réel de classes des données à partitionner est inconnu ou avec un choix empirique des valeurs de certains seuils ou paramètres, qui influent fortement les résultats de partitionnement.

Par exemple, parmi les méthodes paramétriques, nous citons celle proposée par Hartuv et Shamir [120]. L'algorithme de partitionnement, qui s'appelle HCS (Highly Connected Subgraphs), fait appel à la notion de composante connexe. Dans le graphe seuil de similarité, ils identifient les classes par les sous-graphes fortement connexes dont l'arête-connectivité excède la moitié du nombre de sommets. Ces sous-graphes sont déterminés en utilisant des algorithmes de recherche des ensembles minimaux d'arêtes dont la suppression déconnecte le graphe initial. Dans le même ordre d'idée, dans [121], un algorithme de partitionnement hiérarchique basé sur le principe de connectivité de graphe flou est proposé. L'algorithme présenté applique la théorie des ensembles flous à la méthode de partitionnement hiérarchique afin de découvrir des classes connexes. Dans [122], un nouvel algorithme appelé SNGC (Shard Neighbors Graph Clustering) introduit à la fois les voisins partagés et la connectivité entre les sommets d'un graphe pour former les classes. Dans [123], un modèle de bloc stochastique sous-espace pour explorer les structures des classes dans les graphes attribués est proposé. Le point clé est de voir à la fois la structure topologique et les informations sur les attributs comme les facteurs latents pour conduire la formation de classes dans le nouveau modèle génératif proposé. Plus précisément, les attributs pertinents sont appris de manière itérative pour chaque classe, puis utilisés comme informations précieuses à intégrer dans le modèle de bloc stochastique. Dans le même contexte, dans [124], la classification de sous-espaces dans un graphe attribué à plusieurs valeurs est étudiée et un algorithme SCMAG (Subspace Clustering in Multi-valued Attributed Graph) pour la détection de communauté est proposé. Cet algorithme utilise une approche de classification de sous-espaces à base de cellules et identifie les

cellules avec une connectivité dense dans les sous-espaces. « La marche aléatoire » avec redémarrage est utilisée pour mesurer la connectivité structurelle et la similitude des attributs. Un schéma d'indexation est conçu pour prendre en charge le calcul efficace de la connectivité cellulaire à partir de scores de marche aléatoires. Également une nouvelle stratégie de combinaison de cellules sur les dimensions des attributs catégoriels et un nouveau mécanisme pour gérer les attributs à valeurs multiples sont introduits.

Dans [125], un nouvel algorithme génétique multi-objectif (MOGA-OCD) conçu pour identifier les communautés dans les réseaux sociaux qui se chevauchent est proposé, en utilisant des mesures liées à la connectivité du réseau. Cet algorithme utilise un codage de type phénotype basé sur les informations de bord, et une nouvelle fonction de mise en forme centrée sur l'optimisation de deux objectifs classiques en problème de détection de communauté : le premier est utilisé pour maximiser la connectivité interne des communautés, tandis que le second est utilisé pour minimiser les connexions externes au reste du graphe.

Dans [126], une nouvelle méthode de partitionnement hiérarchique fractionnée et fusionnée est proposée dans laquelle un arbre couvrant minimum (MST) et un graphe basé sur MST sont utilisés pour guider le processus de division et de fusion. Dans le processus de fractionnement, les sommets avec des degrés élevés dans le graphe basé sur MST sont sélectionnés comme prototypes initiaux, et le *K*-means est utilisé pour fractionner le jeu de données. Dans le processus de fusion, les paires de sous-groupes sont filtrées et seules les paires voisines sont prises en compte pour la fusion. La méthode proposée nécessite la connaissance du nombre de classes.

Dans [127], les auteurs améliorent la connectivité des graphes en ajoutant une étape de projection efficace à la méthode SSC-OMP (Sparse Subspace Clustering by Orthogonal Matching Pursuit). Avec cette étape de projection, il est possible d'établir une garantie théorique que la condition de préservation du sous-espace conduit directement au résultat exact de partitionnement, qui comble l'écart.

Il existe de nombreuses méthodes de partitionnement basées sur la densité pour préserver la connectivité des classes [128], [129], [130], parmi lesquelles le partitionnement spatial basée sur la densité des applications avec bruit (DBSCAN) [128] est une méthode typique. DBSCAN recherche les zones de densité atteignables dans l'espace d'entités et regroupe les points de données par accessibilité de densité. Cependant, le nombre de classes est sensible à la taille d'un quartier et

au seuil de densité. Trois méthodes ont été proposée visant sur l'amélioration de DBSCAN. Une première méthode intègre la connectivité spatiale des pixels pour partitionner les images couleurs [131]. Une deuxième vise à améliorer la vitesse d'exécution de DBSCAN en garantissant les mêmes résultats [132] et la troisième [133], améliore l'efficacité de DBSCAN en réduisant le nombre global de requêtes de voisinage. Les travaux conduits dans [132] traitent ce problème en examinant les stratégies d'interrogation de région les plus pertinentes pour DBSCAN. Toutes ces méthodes sont paramétriques, où plusieurs paramètres doivent être fixés empiriquement par l'utilisateur et leurs choix influent fortement les résultats de partitionnement.

Dans [134], l'algorithme de partitionnement RElative COre MErge (RECOME) est proposé. Le cœur de RECOME est une nouvelle mesure de densité, qui est la densité relative du *K* noyau voisin le plus proche (RNNKD). RECOME identifie les objets core avec l'unité RNNKD et partitionne les objets non-core en groupes d'atomes en suivant successivement des relations de voisinage de densité plus élevée vers les objets core. Les objets de base et leurs classes d'atomes correspondant sont ensuite fusionnés via des chemins α-accessible sur un graphe KNN. Le nombre de classes calculé par RECOME est une fonction d'étape du paramètre α avec discontinuité de saut sur une petite collection de valeurs. C'est une méthode paramétrique.

D'autres méthodes faisant appel à la connectivité ont été proposées qui sont aussi paramétriques comme dans [135], [136] et [137].

On trouve également l'introduction du critère de la connectivité dans l'algorithme *K*-means pour pouvoir former des classes spatialement liées [138]. Le maintien de la connectivité des classes nécessite la spécification de la relation d'adjacence dans l'énoncé du problème, en utilisant le cadre des graphes pondérés par les bords. La dernière méthode que nous pouvons citer est celle proposée dans [139]. Cette approche construit la matrice de similarité de nœuds d'un graphe sur la base d'une nouvelle métrique de similitude de nœuds, puis applique l'algorithme *K*-means à cette matrice. Ces deux approches sont semi-supervisés.

La plupart des méthodes de partitionnement mentionnées ci-dessus tentent de comprendre les points de données à classifier sous un angle local et statique. Une fois qu'un groupe de paramètres est donné, un résultat de partitionnement unique et fixe sera obtenu. Cependant, il est difficile voire impossible de définir automatiquement les paramètres appropriés pour ces algorithmes de

partitionnement. Dans ce cas, la plupart des algorithmes existants ne parviendront pas à obtenir tous les résultats souhaités.

Dans ce chapitre, nous développons une nouvelle méthode de partitionnement déterministe, non supervisée et non paramétrique en tenant compte des propriétés d'association tout en introduisant des nouvelles définitions. En premier lieu, nous décrivons le principe de la méthode et les étapes de sa mise en œuvre. Puis nous l'évaluons numériquement sur deux bases de données.

## 6.3 Méthode proposée

Avant de présenter la nouvelle méthode de partitionnement appelée UP-DIA, nous précisons d'abord quelques définitions, puis, nous décrivons ses principales étapes en y intégrant sa version hiérarchisée.

### 6.3.1 Notations et Définitions

Dans cette sous-section, nous définissons formellement certains termes et précisons les notations utilisées.

Soit $X = \{x_1, x_2, \dots, x_N\}$ l'ensemble de données à partitionner, où chaque objet $x_i$ est caractérisé par un vecteur d'attributs $A_i = (a_{1i}, a_{2i}, \dots, a_{Bi})$, avec $B$ le nombre d'attributs.

**Définition 6.1 : Partition préliminaire**

La partition préliminaire de l'ensemble de données $X$ est formée à partir des individus strictement identiques. Plus précisément, cela consiste à regrouper automatiquement les individus strictement identiques et à remplacer chaque groupe formé dont les objets sont strictement identiques au sens de la distance associée à la norme $L_1$ par un seul exemplaire.

**Définition 6.2 : Association directe**

Soient $x_i, x_k \in X$, on dit que $x_i$ est associé directement à $x_k$, notée $x_i \sim_{\mathrm{d}} x_k$, si $d(x_i, x_k) = \min_{j, j \neq i}\{d(x_i, x_j)\}$.

On note $Ass_d$ la matrice des associations directes, dont les éléments sont $ass_d(x_i, x_k)$. Ces éléments mettent en évidence la présence ou l'absence d'un lien direct entre individus. La matrice d'association directe est donc définie comme suit :

$$ass_d(x_i, x_k) = \begin{cases} 1, si\ x_i \sim_{\mathrm{d}} x_k \\ 0, sinon \end{cases} \tag{6.1}$$

La matrice d'association directe n'est pas symétrique, c'à-d. si $x_i \sim_{\mathrm{d}} x_k \nRightarrow x_k \sim_{\mathrm{d}} x_i$

L'association directe donne des informations sur les relations locales entre les individus dans un espace de représentation.

**Définition 6.3 : Association indirecte**

Soient $x_i, x_k \in X$, on dit que $x_i$ est associé indirectement à $x_k$, notée $x_i \sim_{\mathrm{id}} x_k$, s'il existe $x_j \in X$, tel que $x_i \sim_{\mathrm{d}} x_j$ et $x_j \sim_{\mathrm{d}} x_k$.

On note $Ass_{id}$ la matrice des associations indirectes, dont les éléments sont $ass_{id}(x_i, x_k)$. Ceux-ci traduisent la présence ou l'absence d'un lien indirect entre un individu et ses associés indirects. La matrice d'association indirecte peut être définie comme suit :

$$ass_{id}(x_i, x_k) = \begin{cases} 1, si\ x_i \sim_{\mathrm{d}} x_j et x_j \sim_{\mathrm{d}} x_k \\ 0, sinon \end{cases} \tag{6.2}$$

**Propriété d'associativité** : La relation d'association indirecte est non symétrique et transitive. Alors, la matrice d'association indirecte n'est pas symétrique.

**Preuve.**

**Transitivité :** Il faut démontrer que si $x_i \sim_{\mathrm{id}} x_j$ et $x_j \sim_{\mathrm{id}} x_k$ alors $x_i \sim_{\mathrm{id}} x_k$.

$x_i \sim_{\mathrm{id}} x_j \Rightarrow \exists\ x_l \in X$, tel que $x_i \sim_{\mathrm{d}} x_l$ et $x_l \sim_{\mathrm{d}} x_j$.

$x_j \sim_{\mathrm{id}} x_k \Rightarrow \exists\ x_m \in X$, tel que $x_j \sim_{\mathrm{d}} x_m$ et $x_m \sim_{\mathrm{d}} x_k$.

Comme $x_l \sim_{\mathrm{d}} x_j$ et $x_j \sim_{\mathrm{d}} x_m$, alors $x_l \sim_{\mathrm{id}} x_m . x_i \sim_{\mathrm{d}} x_l \sim_{\mathrm{id}} x_m \sim_{\mathrm{d}} x_k \Rightarrow x_i \sim_{\mathrm{id}} x_k$.

■

**Définition 6.4 : Classe principale**

Une classe $C_i^p$ est une classe principale si elle est formée à partir des associations directes, autrement dit, chacun de ses éléments a au moins une association directe. Dans ce cas

$$C_i^p = \{x_j \in X : x_i \sim_d x_j\} \tag{6.3}$$

Ces classes sont les plus évidentes, parce qu'elles représentent les liens directs entre les individus.

**Définition 6.5 : Classe secondaire**

Une classe $C_i^s$ est une classe secondaire si elle est formée à partir des associations indirectes, c'est-à-dire :

$$C_i^s = \{x_j \in X : x_i \sim_{id} x_j\} \tag{6.4}$$

Autrement, si $C_i^p$ et $C_j^p$ sont deux classes principales tel que $x_i \in C_i^p$, $x_j \in C_j^p$ et il existe $x_k \in C_i^p \cap C_j^p$ alors la classe secondaire est définie par :

$$C_i^s = C_i^p \cup C_j^p \tag{6.5}$$

**Définition 6.6 : Partition finale**

La partition finale est formée à partir des classes finales qui sont construites à partir des matrices d'associations indirectes, si $ass_{id}(x_i, x_k) = 1$ et $ass_{id}(x_i, x_j) = 1$, alors $x_i, x_j, x_k$ appartiennent à la même classe. Autrement, la partition finale est l'union des classes secondaires. Formellement,

$$P_f = \{C_1, C_2, \dots, C_i, \dots, C_{N_c}\} \tag{6.6}$$

où $C_i = C_k^s \cup C_j^s \mid \exists\, x_l \in C_k^s \cap C_j^s$

**Définition 6.7 : Degrés d'association directe et indirecte d'un individu**

– Le ***degré d'association directe*** d'un individu $x_i$, noté $\deg_d(x_i)$ est le nombre d'individus associés directement à l'individu $x_i$ :

$$\deg_d(x_i) = \left|\{x_j : x_i \sim_{\mathrm{d}} x_j\}\right| = \sum_{j=1}^{N} ass_d(x_j, x_i) \tag{6.7}$$

– Le ***degré d'association indirecte*** d'un individu $x_i$, noté $\deg_{id}(x_i)$ est le nombre d'individus associés indirectement à l'individu $x_i$ :

$$\deg_{id}(x_i) = \left|\{x_j : x_i \sim_{\mathrm{id}} x_j\}\right| = \sum_{j=1}^{N} ass_{id}(x_j, x_i) \tag{6.8}$$

**Définition 6.8** : **Centre d'une classe**

Le centre de la classe $C_i$, noté $\overline{C}_i$, est le vecteur moyen de chaque attribut caractérisant les individus de la classe $C_i$, avec :

$$\overline{C}_i = \frac{1}{N_{c_i}} \sum_{j=1}^{N_{c_i}} x_j \tag{6.9}$$

où $N_{c_i}$ est le nombre d'individus dans la classe $C_i$.

**Définition 6.9: Exemplaire d'une classe**

– L'exemplaire de la classe $C_i$ est l'individu qui possède le degré direct maximal :

$$E(C_i) = \max_k \{\deg_d(x_k) : x_k \in C_i\} \tag{6.10}$$

– Si $\forall x_j, x_k \in C_i$ possèdent le même degré d'association directe maximal, $\deg_d(x_j) = \deg_d(x_k)$, alors l'exemplaire de la classe $C_i$ est l'individu qui possède le degré indirect maximal :

$$E(C_i) = \max_k \{\deg_{id}(x_k) : x_k \in C_i\} \tag{6.11}$$

– Si $\forall x_j, x_k \in C_i$ possèdent le même degré d'association indirecte maximal, $\deg_{id}(x_j) = \deg_{id}(x_k)$, alors l'individu le plus proche du centre de la classe $C_i$ sera choisi comme exemplaire. Formellement, si $\deg_{id}(x_j) = \deg_{id}(x_k)$, alors :

$$E(C_i) = \min\{d(x_j, G_i), d(x_k, G_i)\} \tag{6.12}$$

### 6.3.2 Approche hiérarchique et non supervsée par association directe et indirecte

Dans cette section nous décrivons les étapes de la méthode proposée qui sont : l'algorithme UP-DIA, la formation de la première partition et le partitionnement hiérarchique.

#### *6.3.2.1 Algorithme de partitionnement par associations directe et indirecte (UP-DIA)*

Tout d'abord, une partition préliminaire est formée pour réduire la taille de la matrice de similarité. Cela consiste à regrouper automatiquement les individus strictement identiques et à remplacer chaque groupe formé par un seul exemplaire. Le critère d'agrégation utilisé est la distance associée à la norme $L_1$. Soit $X_0$ le nouvel ensemble de données de taille $N_0$ avec $N_0 < N$.

Durant la première itération, l'algorithme commence à calculer pour chaque individu $x_i$ toutes ces associations directes suivant l'équation (6.1), c'est-à-dire en cherchant la distance minimale sur chaque ligne de la matrice de similarité $S_0$ calculée sur l'ensemble $X_0$, pour former les classes principales, $C_k^p$, $k \in \{1,2,\dots,N_1\}$, où $N_1$ est le nombre de classes préliminaires formées suivant l'équation (6.3).

Lors de la deuxième itération, les classes secondaires sont formées à partir des associations indirectes suivant l'équation (6.2). Ces classes sont formées par les classes principales, c'est-à-dire s'il existe des individus communs entres les classes principales, ces classes seront agrégées.

Formellement, s'il existe $x_k \in C_i^p \cap C_j^p$, alors $C_i^s = C_i^p \cup C_j^p$. A ce stade la partition finale est formée à partir d'agrégation des classes secondaires s'il existe des objets communs entre elles.

L'exemplaire de chaque classe formée de la partition finale est cherché à partir des degrés directes et indirectes de chaque individu. Si deux ou plusieurs individus sont en conflit pour représenter une classe, c'est-à-dire possédant le même degré maximal, alors l'exemplaire entre eux est l'individu qui est le plus proche du centre de la classe $C_k$.

Formellement, s'il existe $x_i, x_j \in C_k$, tel que $deg(x_i) = \deg(x_j)$, alors $E(C_k) = min\{d_1(x_i, \overline{C}_k), d_1(x_j, \overline{C}_k)\}$.

Les étapes de UP-DIA sont présentés dans l'Algorithme 6.1.

**Algorithme 6.1. UP-DIA**

**Entrée :** Tableau de données ($N$ objets $\times$ $B$ attributs) représentant l'ensemble des individus à partitionner

1. Former un nouvel ensemble de données $X_0$ de taille $N_0$ ($N_0 < N$), composé d'individus non dupliqués et exemplaires des individus dupliqués
2. Calculer la matrice de similarité $S_0$ de taille $N_0 \times N_0$
   $s_0(x_i, x_k) = -d_1(x_i, x_k)$, où $d_1$ est la distance associée au norme $L_1$
3. Calculer l'association directe suivant l'équation (6.1)
4. Calculer l'association indirecte suivant l'équation (6.2)
5. Former les classes principales suivant l'équation (6.3)
6. Former les classes secondaires suivant l'équation (6.4)
7. Rechercher la partition finale $P_f$ suivant l'équation (6.6)
8. Calculer les degrés d'association directe et indirecte de chaque individu suivant les équations (6.7) et (6.8) respectivement
9. Calculer les centres des classes $C_i$ de la partition finale $P_f$ suivant l'équation (6.9)
10. Rechercher l'exemplaire de chaque classe de la partition finale suivant les équations (6.10), (6.11) et (6.12)

**Sortie :** Partition $P$ de $N_c$ classes et exemplaire de chaque classe $C_i$

#### *6.3.2.2 Algorithme de partitionnement hiérarchique pour les ensembles de données de grande taille (HUP-DIA)*

Afin de pouvoir appliquer la méthode HUP-DIA proposée à des ensembles de données de grande taille, telles que des images aériennes hyperspectrales, nous utilisons ici la même approche de la méthode HUP-OAP, mais cette fois ci en appliquant la méthode UP-DIA.

Cette opération consiste tout d'abord à subdiviser l'image en blocs puis à appliquer la méthode UP-DIA à chaque bloc, puis fusionner les classes obtenues sur chaque bloc par ré-application de UP-DIA.

Pour obtenir la partition finale de l'image originale, la méthode UP-DIA n'est donc appliquée qu'aux exemplaires des classes identifiées dans les différents blocs. Pour l'obtention d'un partitionnement hiérarchique, la procédure est identique à celle utilisée dans l'Algorithme 4.3 (cf. page 80).

## 6.4 Evaluation numérique

Dans cette section, nous évaluons la méthode proposée HUP-DIA sur les mêmes images hyperspectrales synthétique et réelle déjà utilisées dans le chapitre 4 et comparons ses performances à celles des méthodes semi-supervisées et non supervisées déjà évaluées dans la Section 4.2. A savoir, les 2 méthodes proposées HUP-AOP et HUP-OAPM-RSM et les cinq autres algorithmes semi-supervisés et non supervisés de l'état de l'art.

Pour les méthodes semi-supervisées, le nombre de classes est celui de la VT fournie avec les données. Le paramètre de fuzzification pour FCM, S-OFCM et U-OFCM est fixé à 2. Pour l'AP originale, le paramètre de préférence est fixé à la valeur minimale ($p_{min}$) puis à la valeur médiane ($p_{med}$) de la matrice de similarité et le facteur d'amortissement $\lambda$ est fixé à 0.9.

La métrique utilisée pour le calcul de la matrice de similarité est la distance euclidienne pour les méthodes de l'état de l'art et la distance $d_1$ associée au norme $L_1$ pour les méthodes proposées.

### 6.4.1 Partitionnement de l'image synthétique

La Figure 37 montre le résultat du partitionnement optimal (critère $LN$ : 0.26) de l'image présentée dans la Figure 6 obtenu au niveau 2 par la méthode HUP-DIA proposée, où le nombre estimé de classes est de 9. Pour cette évaluation, l'ensemble de 100 caractéristiques correspondant à la signature spectrale de chaque pixel est considéré et l'image est divisée en 16 blocs, où la taille

de chaque bloc est de 15×15 pixels, pour couvrir l'ensemble de l'image. Le CCR obtenu par la méthode proposée est de 100%.

Le Tableau 36 donne les performances des trois méthodes non supervisées et des trois méthodes semi-supervisées en calculant quatre critères : CCR (%), CCR-SCVT (%), temps CPU (s) et espace mémoire (Mb).

Ces résultats montrent que la méthode développée donne les meilleurs résultats selon les deux critères CCR et le temps CPU, en plus de son avantage non supervisé. Nous pouvons également noter que les méthodes semi-supervisées FCM et *K*-means donnent globalement les résultats les moins intéressants selon le critère CCR. De plus, leurs résultats ne sont pas stables d'une exécution à l'autre, malgré l'introduction du nombre de classes. Le

Tableau 37 montre les performances de trois méthodes développées suivant le temps de calcul et l'espace mémoire. Nous remarquons que la méthode HUP-DIA nécessite moins d'espace mémoire et de temps CPU.

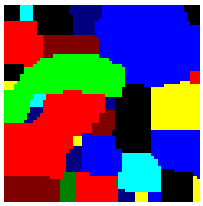


**Figure 37.** Résultats de partitionnement par HUP-DIA de l'image hyperspectrale synthétique de la Figure 6 (nombre de classes 9, *LN* = 0.26).

**Tableau 36.** Comparaison des performances des méthodes de partitionnement sur l'image hyperspectrale synthétique de la Figure 6.

| Méthodes | Non supervisées | | | | Semi-supervisées | | |
|---|---|---|---|---|---|---|---|
| | **HUP-ADI** | AP | | U-OFCM | S-OFCM | FCM (*) | *K*-means (*) |
| | | $p_{med}$ | $p_{min}$ | | | | |
| Nombre de classes | **9** | 13 | 9 | 6 | 9 | 9 | 9 |
| CCR (%) | **100** | 83.17 | 94.94 | 86.14 | 86.55 | 83.07 | 72.03 |
| CCR-SCVT (%) | **100** | 97.83 | 98.38 | 86.14 | 93.71 | 84.80 | 86.85 |
| Temps CPU (s) | **0.59** | 61.88 | 72.01 | 35.63 | 4.46 | 64.73 | 52.82 |
| Espace Mémoire (MB) | 4.47 | 296.93 | 296.93 | 3.82 | 3.17 | 2.99 | **2.75** |

(*) Taux moyen de 5 CCR.

**Tableau 37.** Comparaison de trois méthodes de partitionnement développées sur l'image hyperspectrale synthétique de la Figure 6.

| Méthodes | CCR (%) | Nombre de blocs | Temps CPU (s) | Espace Mémoire (Mb) |
|---|---|---|---|---|
| HUP-OAP | 100 | 16 (15×15 pixels) | 3.73 | 23.19 |
| HUP-OAPM-RSM | | | 1.99 | 9.76 |
| **HUP-DIA** | | | **0.59** | **4.47** |

### 6.4.2 Partitionnement de l'image réelle de grande taille

Pour partitionner cette image hyperspectrale (630 × 1800 pixels), elle a été divisée d'abord en 5040 blocs d'une taille de 15 × 15 pixels chacun. La taille de la matrice de similarité pour la formation de la première partition en ne prenant en compte que les exemplaires des classes formées dans les différents blocs sont de 19 445 × 19 445 contre 1 134 000 × 1 134 000 pour l'image originale. Le Tableau 38 donne le nombre estimé de classes par la méthode HUP-DIA pour chaque niveau de partitionnement et la valeur du critère d'optimisation *LN* pour chaque partition. La Figure 38 montre le résultat de partitionnement optimal de l'image hyperspectrale de la Figure 12 (a) obtenu au niveau 6 en maximisant le critère *LN*. Pour cette partition, le nombre estimé de classes est de 4. Ce partitionnement a nécessité au total 442.87 seconde de temps CPU. La Figure 39 montre la signature spectrale moyenne de chaque classe formée. En vérifiant les positions des 23 points des trois classes de la VT au sein des classes formées, le taux d'identification est de 100%. Les pixels des classes de la partition optimale peuvent donc être également utilisés comme échantillons d'apprentissage. En plus de la partition optimale, les autres partitions peuvent être exploitées pour contribuer à une analyse plus fine des données en fonction des besoins des utilisateurs.

Le taux de couverture des algues dans cette image correspond à 48.5% (28.63% pour les algues brunes et 19.87% pour les algues vertes) comme indiqué dans le Tableau 39.

La méthode développée non supervisée donne de meilleurs résultats par rapport aux méthodes de l'état de l'art appliquées sur l'image originale (voir Tableau 40). Le Tableau 41 montre les performances de trois méthodes développées en termes de temps CPU et l'espace mémoire. Nous remarquons d'après ce tableau que la méthode HUP-DIA a besoin le moins de temps CPU et d'espace mémoire par rapport à HUP-OAP et HUP-OAPM-RSM.

**Tableau 38.** Nombre estimé de classes par la méthode HUP-DIA de l'image hyperspectrale réelle de la Figure 12 et la valeur du critère d'optimisation *LN* pour chaque partition.

| Niveau | Nombre estimé de classes | Valeur du critère *LN* |
|---|---|---|
| 1 | 2596 | 0.016 |
| 2 | 630 | 0.044 |
| 3 | 148 | 0.049 |
| 4 | 44 | 0.055 |
| 5 | 13 | 0.120 |
| **6** | **4** | **0.160** |
| 7 | 2 | 0.140 |
| 8 | 1 | 0.100 |

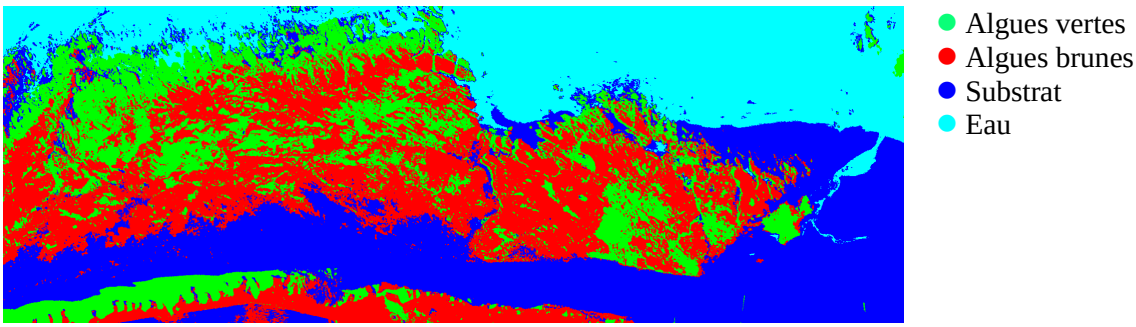
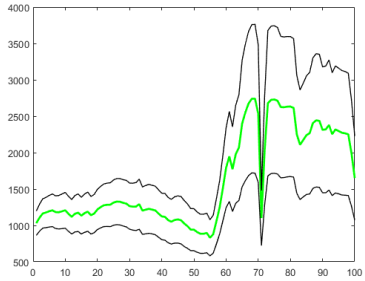
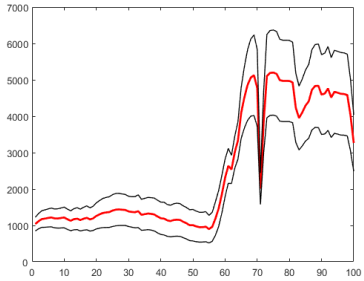


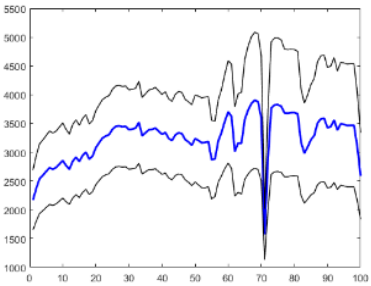

**Figure 38.** Résultat de partitionnement optimal par la méthode HUP-DIA de l'image hypersepctrale réelle de la Figure 12 (4 classes, *LN* = 0.16 au niveau 6).



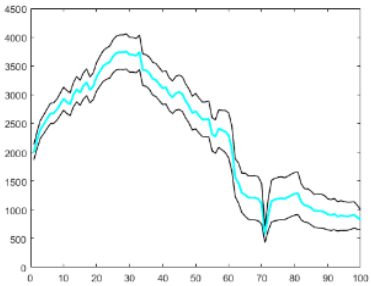
**(a)** Classe Algues vertes



**(b)** Classe Algues brunes



**(c)** Classe Substrat



**(d)** Classe Eau

**Figure 39.** Signature spectrale moyenne ± écart-type de chaque classe obtenue par HAUP-DIA (partition optimale : niveau 6, 4 classes) sur l'image hypersepctrale réelle de la Figure 12.

**Tableau 39.** Taux de couverture de chaque classe obtenue au niveau 6 par HUP-DIA sur l'image hypersepctrale réelle de la Figure 12.

| Classes | Nombre de pixels | Taux de couverture (%) |
|---|---|---|
| Algues vertes | 225 366 | 19.87 |
| Algues Brunes | 324 645 | 28.63 |
| Substrat | 251 232 | 22.15 |
| Eau | 332 757 | 29.35 |

**Tableau 40.** Performance de la méthode développée HUP-DIA, U-OFCM, S-OFCM, FCM et *K*-means sur l'image hypersepctrale réelle de la Figure 12.

| Méthodes | Nombre de classes | CCR (%) |
|---|---|---|
| **HUP-DIA** | **4 (estimé)** | **100** |
| U-OFCM | 4 (estimé) | 86.85 |
| S-OFCM | 4 (fixé) | 91.30 |
| FCM (*) | 4 (fixé) | 95.65 |
| *K*-means (*) | 4 (fixé) | 95.65 |

(*) Taux moyen de 5 CCR.

**Tableau 41.** Comparaison des performances des méthodes développées sur l'image hypersepctrale réelle de la Figure 12.

| Méthodes | CCR (%) | Nombre de blocs | Temps CPU (s) | Espace Mémoire (Mb) |
|---|---|---|---|---|
| HUP-OAP | 100 | 5040 (15×15 pixels) | 3126.51 | 9472.8 |
| HUP-OAPM-RSM | | | 1430.16 | 5913.5 |
| **HUP-DIA** | | | **442.87** | **4847.9** |

## 6.5 Discussion

Dans ce chapitre, une méthode de partitionnement hiérarchique, non paramétrique et non supervisée adaptée aux ensembles de données de grands taille, tels que l'imagerie aérienne hyperspectrale est proposée. Cette méthode utilise les associations directes et indirectes entre les individus pour former des classes. L'idée principale présentée ici est la localisation de points similaires et leur regroupement suivant leurs associations directes et indirectes.

Le résultat du partitionnement est fait sur la seule base de l'image observée sans aucune connaissance *a priori*. Pour sélectionner la meilleure partition, la méthode hiérarchique (HUP-DIA) est appliquée uniquement sur les exemplaires des classes obtenues pour chaque niveau hiérarchique. L'introduction de la procédure de découpage en blocs contribue efficacement à

partitionner les images de grande taille et peut être appliquée de manière parallèle. Cependant, nous notons que les résultats sont légèrement sensibles au choix de la taille des blocs mais avec des résultats cohérents.

Les évaluations de la méthode développée sur de données synthétique et une image hyperspectrale réelle de grande taille montrent que les résultats sont pertinents. Cela prouve qu'il peut être généralisé et donc appliqué objectivement à un large éventail d'applications sans aucune intervention de l'utilisateur.

# Chapitre 7 : Application des méthodes développées à trois domaines : l'environnement, la reconnaissance faciale avec expressions et la médecine

## 7.1 Introduction

Dans ce chapitre, nous étudions et comparons les performances des approches développées sur plusieurs bases de données de types différentes pour prouver leur caractère général et leur application à un large éventail de problèmes sans l'intervention de l'utilisateur. Trois domaines applicatifs différent sont utilisés : environnement, reconnaissance faciale avec expressions et médecine.

Pour le domaine environnemental, deux bases de données sont utilisées qui sont une image hyperspectrale synthétique de petite taille et une image hyperspectrale réelle de grande taille. Le but de cette application est la détection des plantes invasives. Pour la reconnaissance de visages et l'identification des expressions faciales trois bases de données bien connus sont utilisées : JAFFE, YALE et ORL. Finalement, pour le domaine médical deux bases de données sont utilisées : IRM et Mélanomes.

Nous rappelons que pour l'ensemble des évaluations sur les données environnementales, la métrique utilisée pour les méthodes développées est $d_1$ (distance associée au norme $L_1$). Pour les bases de données d'identification faciales et médicales, les travaux antérieurs [140], [141] utilisaient les caractéristiques SIFT [142] pour l'appariement d'images et l'apprentissage par catégories (voir Annexe 1). Chaque caractéristique est décrite par un vecteur 128D. Nous suivons la procédure décrite par Lowe pour compter le nombre de correspondances de caractéristiques significatives en comparant l'image $x_i$ avec l'image $x_k$ (notée $CN(x_i, x_k)$) : pour chaque caractéristique locale de l'image $x_i$, les caractéristiques les plus proches et les secondes les plus proches sont recherchées dans l'image $x_k$. La correspondance est considérée comme significative si le rapport de distance entre le plus proche et le deuxième plus proche voisin est supérieur à un seuil $\zeta$ choisi entre 0.5 et 1. Ensuite, la matrice de similarité est définie comme étant le nombre de correspondances significatives normalisées par la soustraction de moyennes sur les deux dimensions [140] :

$$s(x_i, x_k) = CN(x_i, x_k) - \frac{1}{N}\sum_{j=1}^{N} CN(x_i, x_j) - \frac{1}{N}\sum_{j=1}^{N} CN(x_j, x_k) \quad (7.1)$$

Ce chapitre est structuré en trois sections. La Section 7.2 porte sur l'application des trois méthodes de partitionnement développées sur le domaine environnemental. La Section 7.3 présente l'application sur la reconnaissance faciale avec expressions. En fin, la Section 7.4 présente l'application sur le domaine médical.

## 7.2 Application à l'environnement-Détection de plantes invasives

Cette application concerne l'identification de plantes invasives et non-invasives d'une zone de la région de Murcia (sud est de l'Espagne). La disponibilité de relevés sur cette zone nous a permis de valider et de comparer les approches non supervisées développées dans le cadre de cette thèse.

Deux images hyperspectrales sont utilisées pour montrer les performances des méthodes développées. La première est une image synthétique de petite taille et la seconde est une image réelle de grande taille où le but est la détéction des plantes invasives. Les résultats sont comparés suivant 4 critères : CCR, CCR-SCVT lorsque les classes d'une VT sont subdivisées, le temps CPU et l'espace mémoire.

### 7.2.1 Application à une image hyperspectrale synthétique

L'image utilisée dans ces évaluations est l'image présentée dans la Figure 2 de la Partie I, qui est formée de 5 classes principale : Rivière, *Pinus halepensis*, Pêchers, *Arundo donax* et Bâtiments.

La Figure 40 synthétise les résultats optimaux obtenus par HUP-OAP, HUP-OAPM-RSM et HUP-DIA. Nous remarquons que le nombre estimé de classes est de 6, 6 et 7 obtenu au niveau 2 par HUP-OAP et au niveau 4 par HUP-OAPM-RSM et HUP-DIA respectivement. Les trois méthodes développées divisent la classe de la VT ''Pêchers'' en plusieurs sous-classes. Pour vérifier et valider cette division, nous montrons dans la Figure 41, les signatures spectrales moyennes des sous-classes obtenues par les trois méthodes. Cette figure montre que les signatures spectrales moyennes des classes obtenues sont distinctes et que grâce à la nature non supervisée des méthodes proposées qu'il est possible de mettre objectivement en évidence la richesse des informations fournies par l'imagerie hyperspectrale dans le proche infrarouge (NIR) par rapport à

celles fournies uniquement dans le domaine visible. Le CCR obtenu par les méthodes proposées est de 88.45%, 90.84% et 87.36% respectivement si on tient compte de la non précision de la VT originale. Ces taux peuvent être corrigé à 100% si l'on considère l'homogénéité des sous-classes de Pêchers formées comme le montre la Figure 41.

Ces résultats montrent que la méthode développée HUP-DIA donne les meilleurs résultats selon deux critères (temps CPU et espace mémoire). D'autre part, HUP-OAP nécessite plus d'espace mémoire et de temps de calcul en raison des matrices de responsabilité et de disponibilité calculées sur l'image complète.

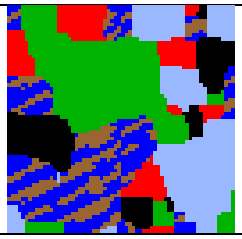
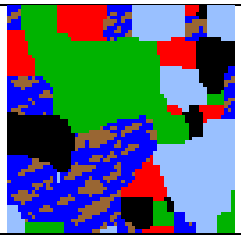
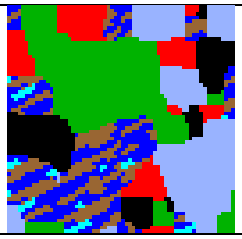
| Méthodes | HUP-OAP | HUP-OAPM-RSM | HUP-DIA |
|---|---|---|---|
| Images Partitionnées |  |  |  |
| Niveau/ Nombre estimé de classes | 2 / 6 | 4 / **6** | 4 / 7 |
| CCR (%) | 88.45 | **90.84** | 87.36 |
| CCR-SCVT (%) | **100** | **100** | **100** |
| Temps CPU (s) | 88.17 | 4.61 | **1.54** |
| Espace Mémoire (MB) | 384.50 | 36.44 | **36.33** |

**Figure 40.** Performances des méthodes développées sur l'image hyperspectrale synthétique de la Figure 2.

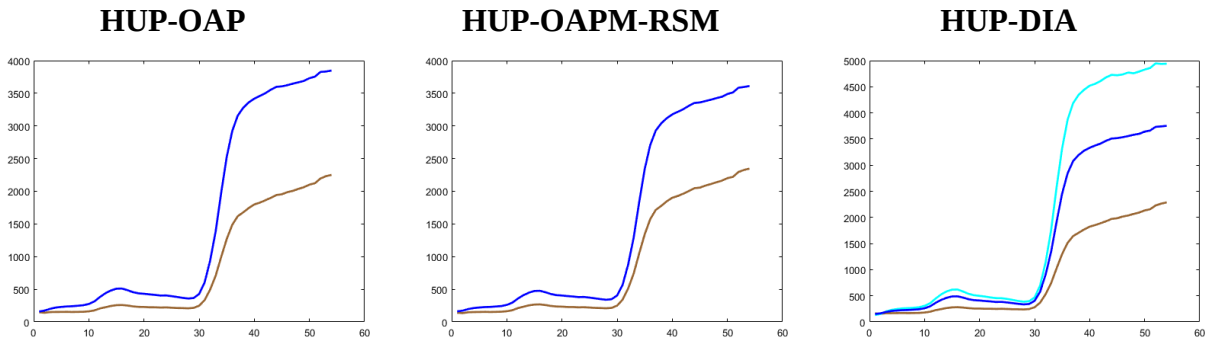


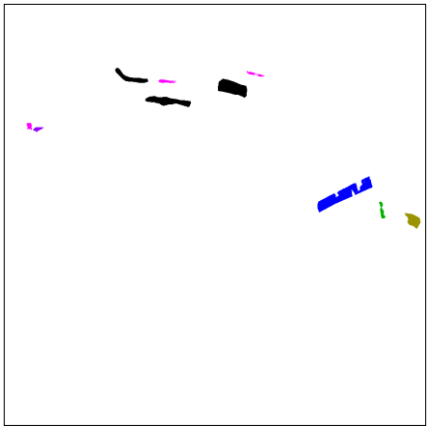

**Figure 41.** Signatures spectrales moyennes de la classe Pêchers subdivisée en sous-classes par les trois méthodes proposées de l'image hyperspectrale synthétique de la Figure 2.

### 7.2.2 Application à une image réelle hyperspectrale

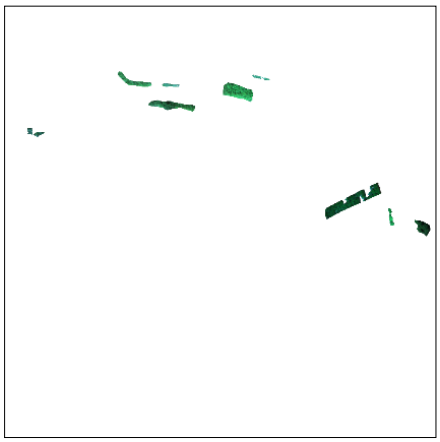
#### *7.2.2.1 Présentation de la base de données*

L'image utilisée est une zone de l'image hyperspectrale Cieza-Région de Murcia, Espagne- (8075×9748 pixels) qui contient le maximum de données de la VT. La taille de la zone choisie est de 1000×1000 pixels. Six classes sont présentes dans cette zone dont trois correspondent à des plantes invasives (Phragmites australis, Tamarix et *Arundo donax*), comme illustré sur la Figure 42. Le Tableau 42 donne les détails des classes de la VT et la Figure 43 montre les signatures spectrales moyennes ± écart types de ces classes.

Zone de l'image hyperspectrale
(Bandes visualisés : 25, 17, 8)



Masques des classes de la VT



Zones de l'image hyperspectrale correspondant aux masques

**Figure 42.** Image hyperspectrale Cieza acquise par le capteur AISA Eagle et zones d'intérêt à partitionner.

**Tableau 42**. Détails des classes de la VT de l'image hyperspectrale réelle de la Figure 42.

| Labels | | Classes | Nombre de pixels |
|---|---|---|---|
| | 1 | *Pinus halepensis* | 274 |
| | 2 | Pêchers | 3115 |
| | 3 | *Arundo donax* | 4305 |
| | 4 | Phragmites australis | 556 |
| | 5 | Tamarix | 162 |
| | 6 | Ulmus minor | 795 |
| Nombre de pixels total | | | 9207 |

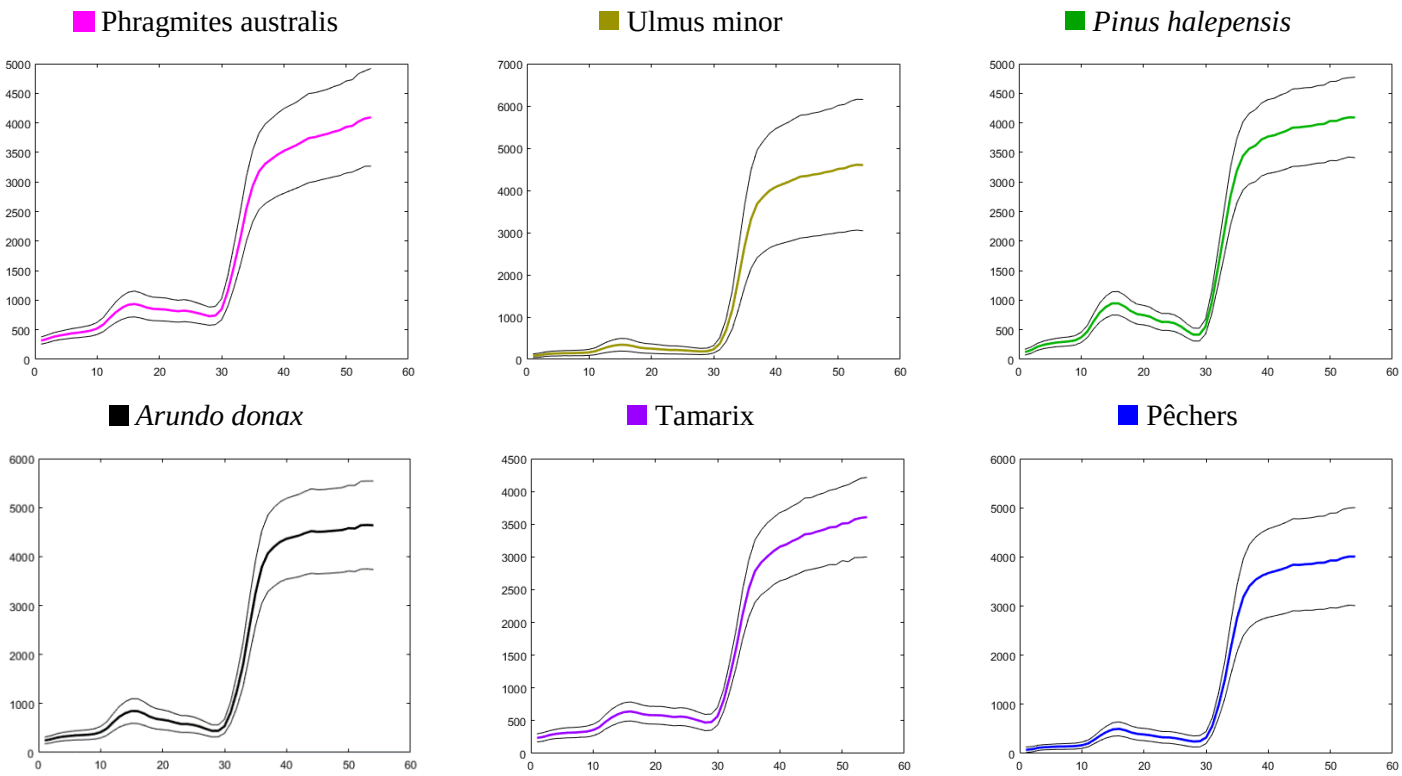



**Figure 43.** Signatures spectrales moyennes ± écart types des 6 classes de la VT de l'image hyperspectrale réelle de la Figure 42.

#### *7.2.2.2 Résultats de partitionnement*

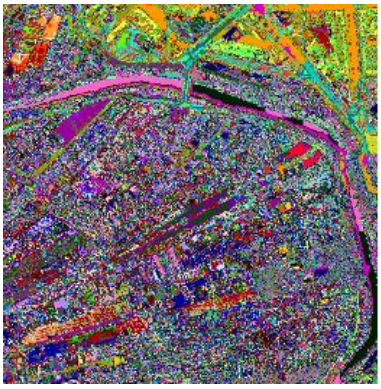
Pour évaluer les trois méthodes, HUP-OAP, HUP-OAPM-RSM et HUP-DIA, nous divisons cette image en 2500 blocs chacun de taille 20×20 pixels.

#### 7.2.2.2.1 Résultats obtenus par HUP-OAP

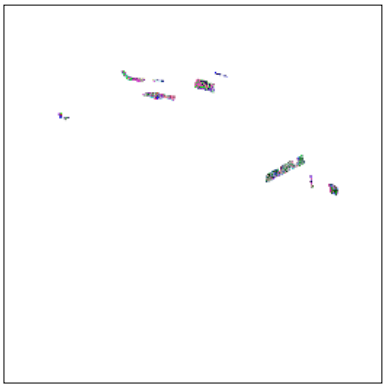
La Figure 44 montre le résultat de partitionnement hiérarchique obtenu par HUP-OAP. La première colonne donne le résultat de partitionnement sur l'image compléte et la deuxième colonne montre le résultat correspondant seulement aux zones d'intêrets. La partition optimale maximisant le critère *LN* est obtenue au niveau 3 où le nombre estimé de classes est de 10. Le CCR obtenu par la méthode proposée est de 73.22%. Ce taux peut être corrigé à 100%, si l'on considère que les nouvelles classes formées correspondant uniquement à des subdivisons des classes de la VT (Phragmites australis, Ulmus minor, *Arundo donax* et Pêchers) comme le montre le Tableau 43 et la Figure 45.

Image Partitionnée | Zones d'intêret

Niveau 1

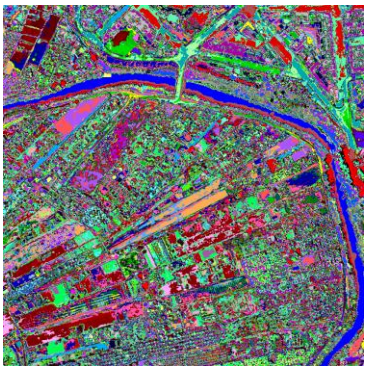


$N_c$= 336, $LN$ = 0.29

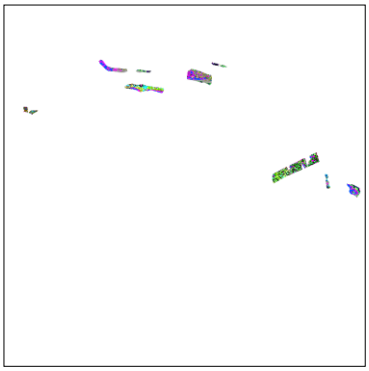


$N_c$= 99, $LN$ = 0.39

Niveau 2

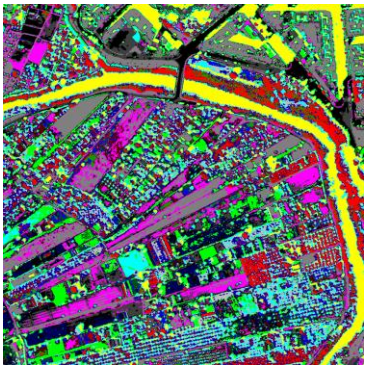


$N_c$= 84, $LN$ = 0.32

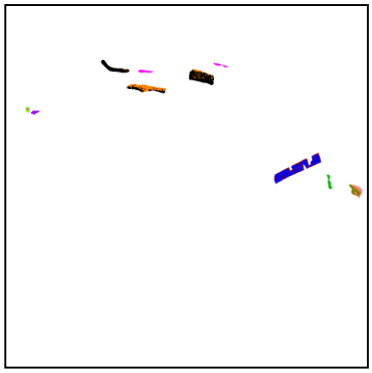


$N_c$= 34, $LN$ = 0.44

Niveau 3

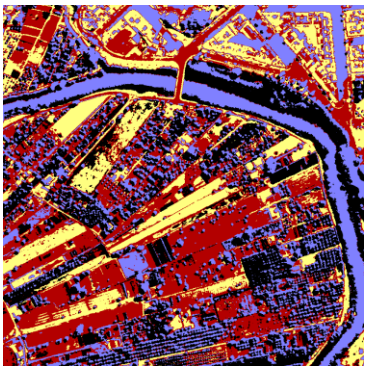


**$N_c$= 16, $LN$ = 0.75**

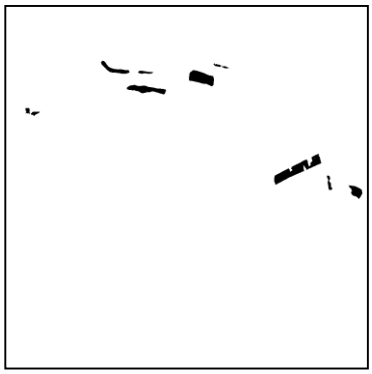


**$N_c$= 10, $LN$ = 0.65**

Niveau 4



$N_c$= 4, $LN$ = 0.68



$N_c$= 1, $LN$ = 0.65

**Figure 44**. Résultat de partitionnement hiérarchique par niveau obtenu par la méthode HUP-OAP sur l'image complète de la Figure 42 et les zones d'intérêt, $N_c$ : nombre estimé de classes, $LN$ : valeur du critère d'optimisation.

**Tableau 43.** Matrice de confusion correspondant à la partition optimale de l'image hyperspectrale réelle de la Figure 42 obtenue par HUP-OAP au niveau 3, où le nombre estimé de classes est 10.

| | | Classes obtenues par HUP-OAP | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| Classes de la VT | ■ Phragmites australis | **403** | 0 | 0 | 0 | 0 | 0 | 153 | 0 | 0 | 0 |
| | ■ Ulmus minor | 0 | **443** | 0 | 0 | 0 | 0 | 0 | 352 | 0 | 0 |
| | ■ Pinus halepensis | 0 | 0 | **274** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ■ Arundo donax | 0 | 0 | 0 | **3217** | 0 | 0 | 0 | 0 | 1088 | 0 |
| | ■ Tamarix | 0 | 0 | 0 | 0 | **162** | 0 | 0 | 0 | 0 | 0 |
| | ■ Pêchers | 0 | 0 | 0 | 0 | 0 | **2243** | 0 | 0 | 0 | 872 |
| | CCR | 73.22% | | | | | | | | | |
| | CCR-SCVT | 100% | | | | | | | | | |

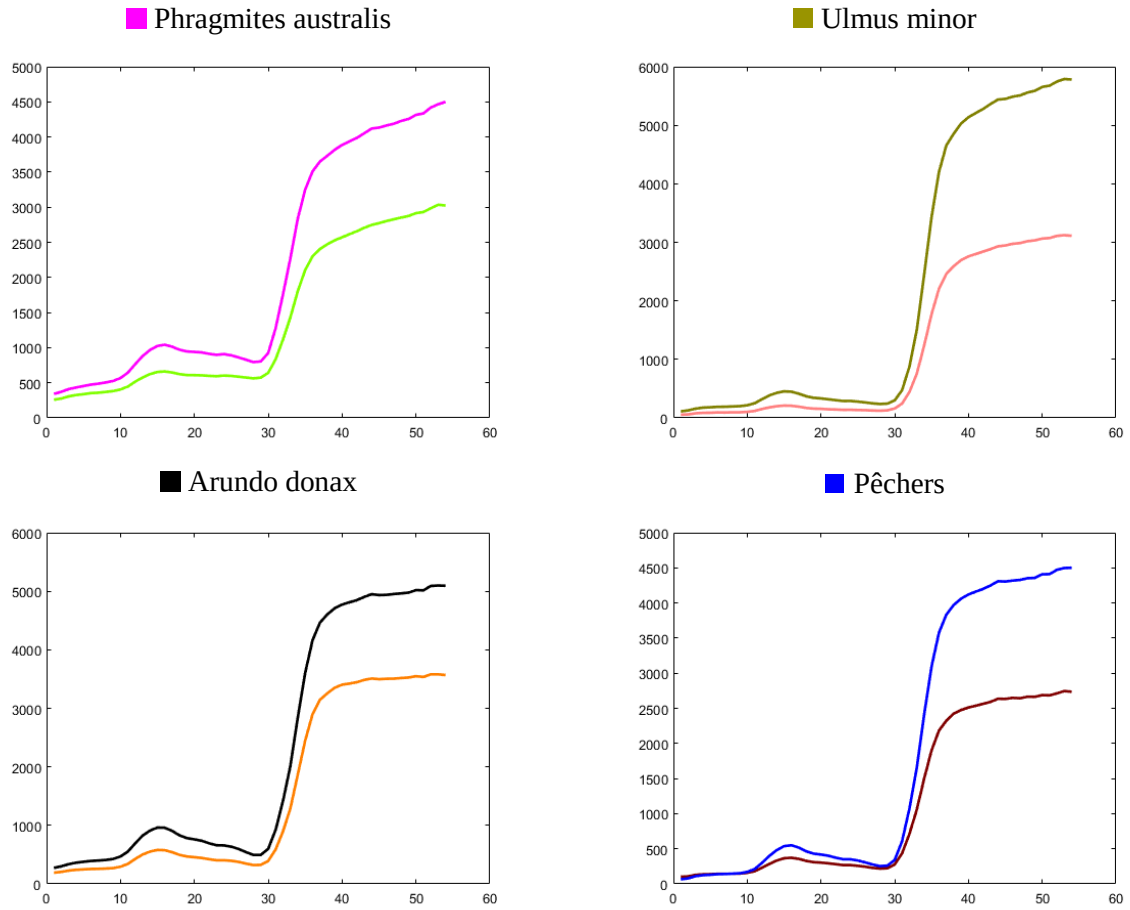


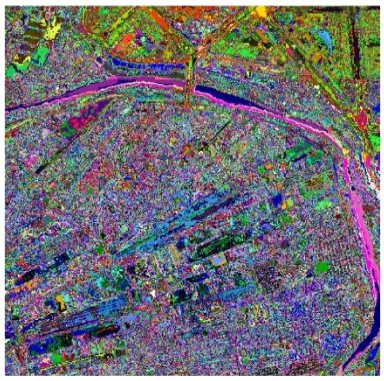

**Figure 45.** Signatures spectrales ± écart type des classes de la VT subdivisées de l'image hyperspectrale réelle de la Figure 42.

#### 7.2.2.2.2 Résultats obtenus par HUP-OAPM-RSM

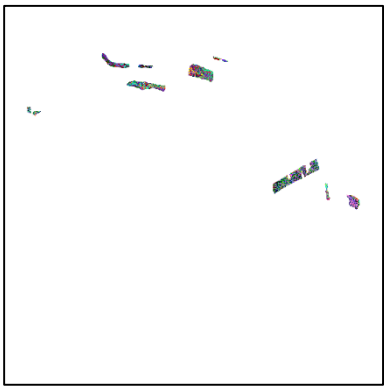
Le résultat de partitionnement hiérarchqiue obtenu par HUP-OAPM-RSM est présenté dans la Figure 46. La première colonne donne le résultat de partitionnement sur l'image compléte et la deuxième colonne montre le résultat correspondant seulement aux zones d'intêrets. La partition optimale maximisant le critère *LN* est obtenue au niveau 4 où le nombre de classes est estimé à 9. Le CCR obtenu par la méthode proposée est de 82.07%. Ce taux peut être aussi corrigé à 100%, car les nouvelles classes formées par la méthode correspondent uniquement à des subdivisions des classes de la VT comme le montre le Tableau 44 et la Figure 47. Ces subdivisions sont cohérentes car elles prennent en considérartion l'information apportée par le capteur hyperspectrale.

Image Partitionnée | Zones d'intêret

Niveau 1

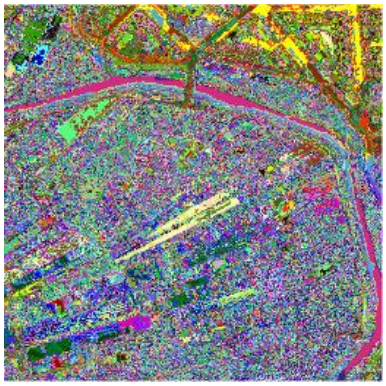


$N_c$= 649, $LN$ = 0.24

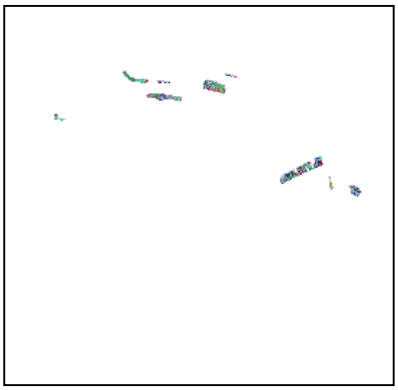


$N_c$ =173, $LN$ = 0.30

Niveau 2

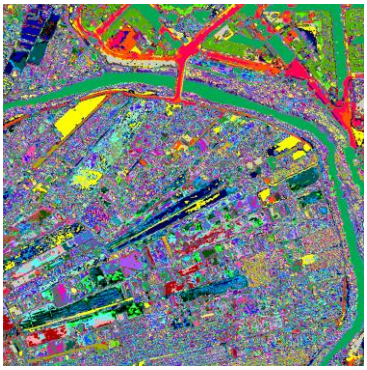


$N_c$= 352, $LN$ = 0.28

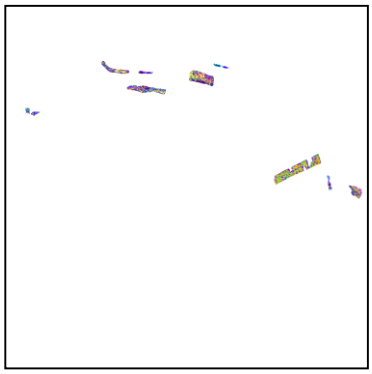


$N_c$= 116, $LN$ = 0.35

Niveau 3

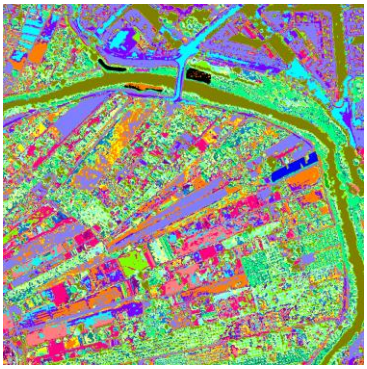


$N_c$= 65, $LN$ = 0.44

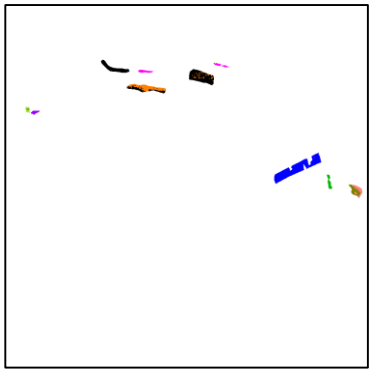


$N_c$= 33, $LN$ = 0.45

Niveau 4

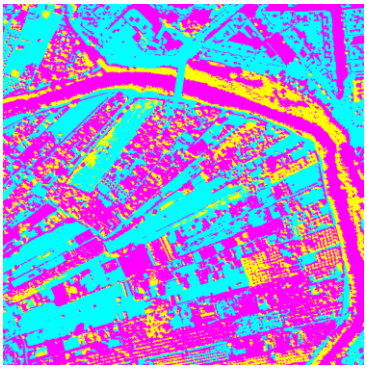


**$N_c$= 16, $LN$ = 0.75**

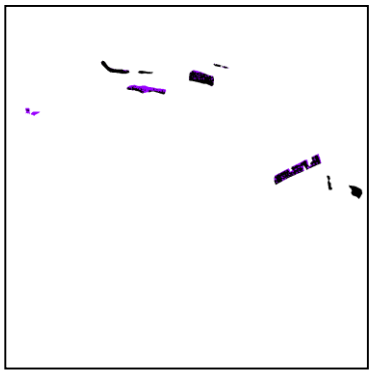


**$N_c$= 9, $LN$ = 0.66**

Niveau 5



$N_c$= 3, $LN$ = 0.65



$N_c$= 2, $LN$ = 0.63

**Figure 46**. Résultat de partitionnement hiérarchique par niveau obtenu par la méthode HUP-OAPM-RSM sur l'image complète de la Figure 42 et les zones d'intérêt, $N_c$ : nombre estimé de classes, $LN$ : valeur du critère d'optimisation.

**Tableau 44.** Matrice de confusion correspondant à la partition optimale de l'image hyperspectrale réelle de la Figure 42 obtenue par HUP-OAPM-RSM au niveau 4, où le nombre estimé de classes est 9.

| | | Classes obtenues par HUP-OAPM-RSM | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | |
| Classes de la VT | Phragmites australis | **415** | 0 | 0 | 0 | 0 | 0 | 141 | 0 | 0 |
| | Ulmus minor | 0 | **453** | 0 | 0 | 0 | 0 | 0 | 342 | 0 |
| | *Pinus halepensis* | 0 | 0 | **274** | 0 | 0 | 0 | 0 | 0 | 0 |
| | *Arundo donax* | 0 | 0 | 0 | **3138** | 0 | 0 | 0 | 0 | 1167 |
| | Tamarix | 0 | 0 | 0 | 0 | **162** | 0 | 0 | 0 | 0 |
| | Pêchers | 0 | 0 | 0 | 0 | 0 | **3115** | 0 | 0 | 0 |
| | CCR | 82.07% | | | | | | | | |
| | CCR-SCVT | 100% | | | | | | | | |

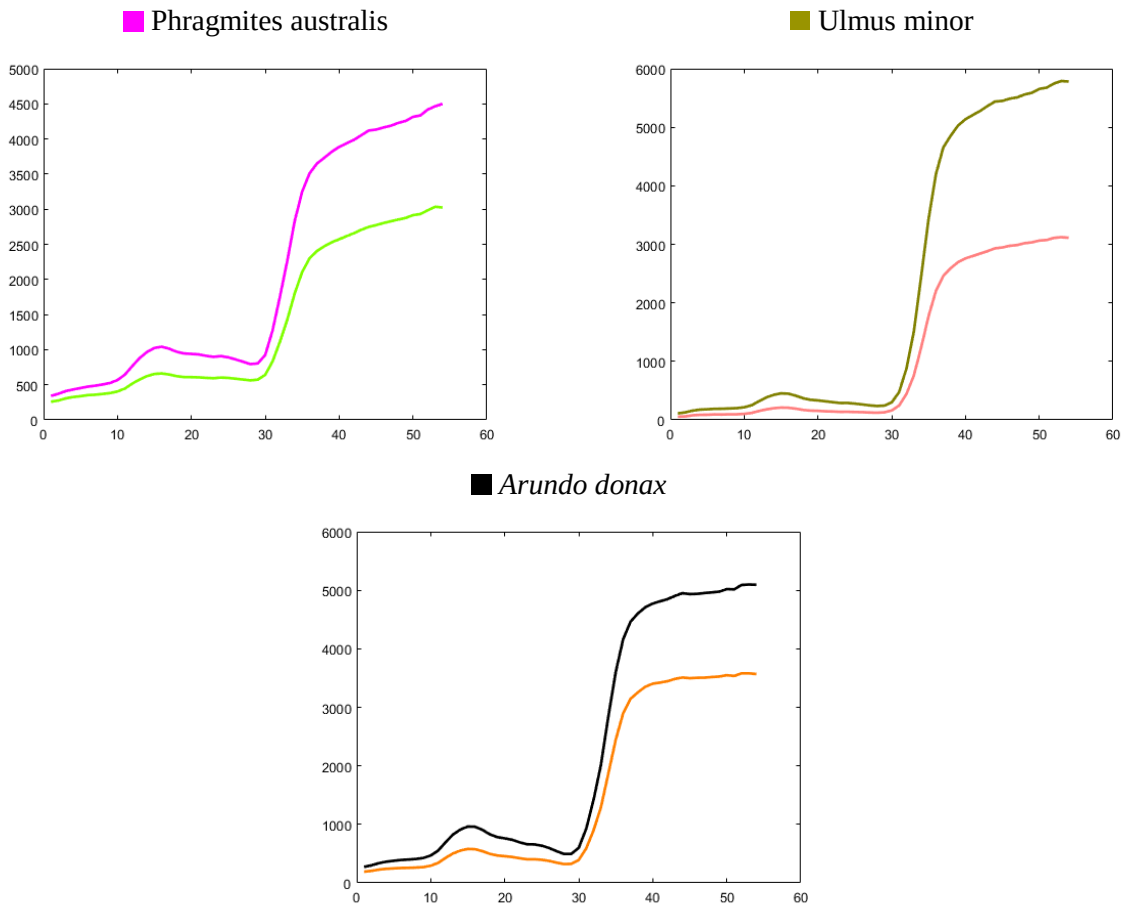


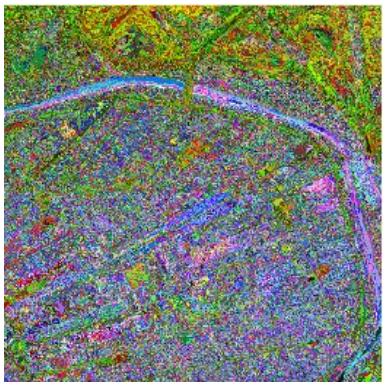

**Figure 47.** Signatures spectrales ± écart type des classes de la VT divisées de l'image hyperspectrale réelle de la Figure 42.

#### 7.2.2.2.3 Résultats obtenus par HUP-DIA

La Figure 48 montre le résultat par niveau du partitionnement hiérérchique obtenu par HUP-DIA. La deuxième colonne visualise le résultat correspondant seulement aux zones d'intêrets. La partition optimale maximisant le critère *LN* est obtenue au niveau 4 où le nombre estimé de classes est 9. Le CCR obtenu par la méthode proposée est de 90.31%. Ce taux peut être également corrigé à 100%, si l'on tient compte de la subdivision des classes de la VT comme le montre le Tableau 45 et la Figure 49.

Images Partitionnées | Zones d'intêret

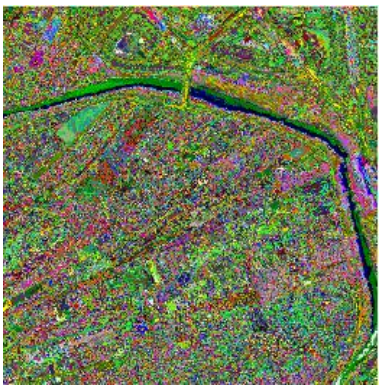
Niveau 1



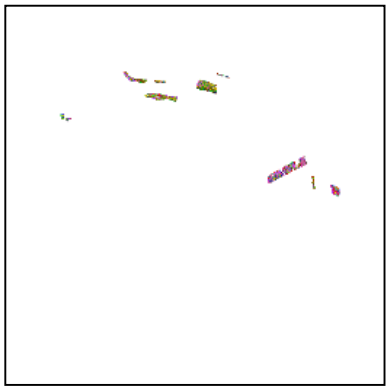
$N_c$= 1539, $LN$ = 0.18

$N_c$= 239, $LN$ = 0.29

Niveau 2

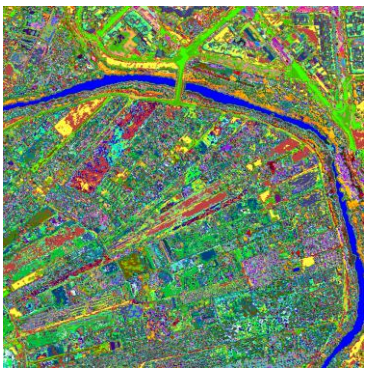


$N_c$= 589, $LN$ = 0.26

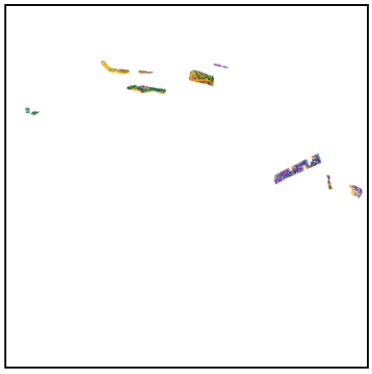


$N_c$= 94, $LN$ = 0.2

Niveau 3

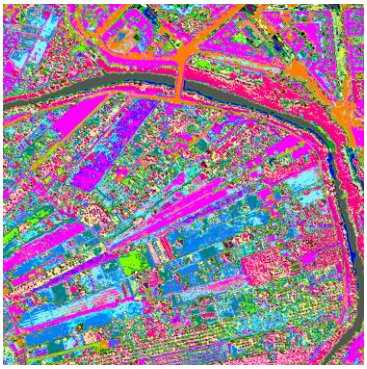


$N_c$= 131, $LN$ = 0.30

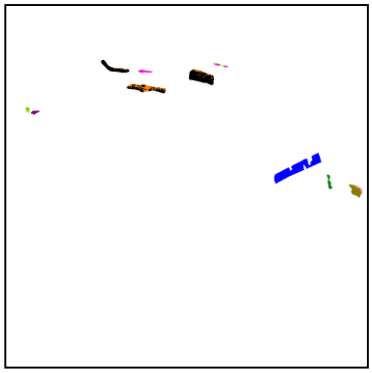


$N_c$= 36, $LN$ = 0.40

Niveau 4

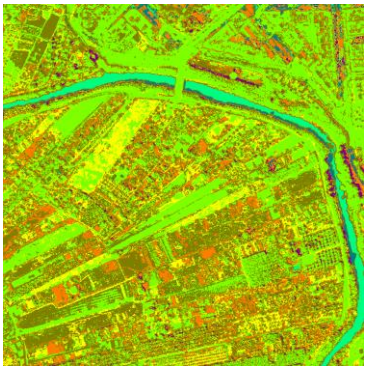


**$N_c$= 33, $LN$ = 0.69**

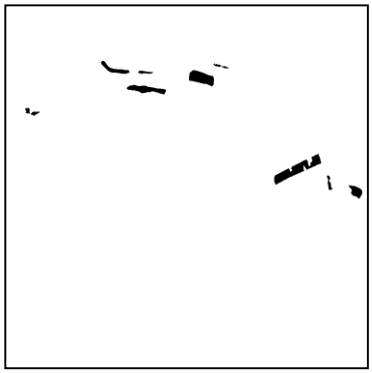


**$N_c$= 9, $LN$ = 0.67**

Niveau 5



$N_c$= 7, $LN$ = 0.66



$N_c$= 1, $LN$ = 0.65

**Figure 48**. Résultat de partitionnement hiérarchique par niveau obtenu par la méthode HUP-DIA sur l'image complète de la Figure 42 et les zones d'intérêt, $N_c$ : nombre estimé de classes, $LN$ : valeur du critère d'optimisation.

**Tableau 45.** Matrice de confusion correspondant à la partition optimale de l'image hyperspectrale réelle de la Figure 42 obtenue par HUP-DIA au niveau 4, où le nombre estimé de classes est 9.

| | | Classes obtenues par HUP-DIA | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | |
| Classes de la VT | Phragmites australis | **355** | 0 | 0 | 0 | 0 | 0 | 201 | 0 | 0 |
| | Ulmus minor | 0 | **740** | 0 | 0 | 0 | 0 | 0 | 55 | 0 |
| | *Pinus halepensis* | 0 | 0 | **274** | 0 | 0 | 0 | 0 | 0 | 0 |
| | *Arundo donax* | 0 | 0 | 0 | **3669** | 0 | 0 | 0 | 0 | 636 |
| | Tamarix | 0 | 0 | 0 | 0 | **162** | 0 | 0 | 0 | 0 |
| | Pêchers | 0 | 0 | 0 | 0 | 0 | **3115** | 0 | 0 | 0 |
| | CCR | 90.31% | | | | | | | | |
| | CCR-SCVT | 100% | | | | | | | | |

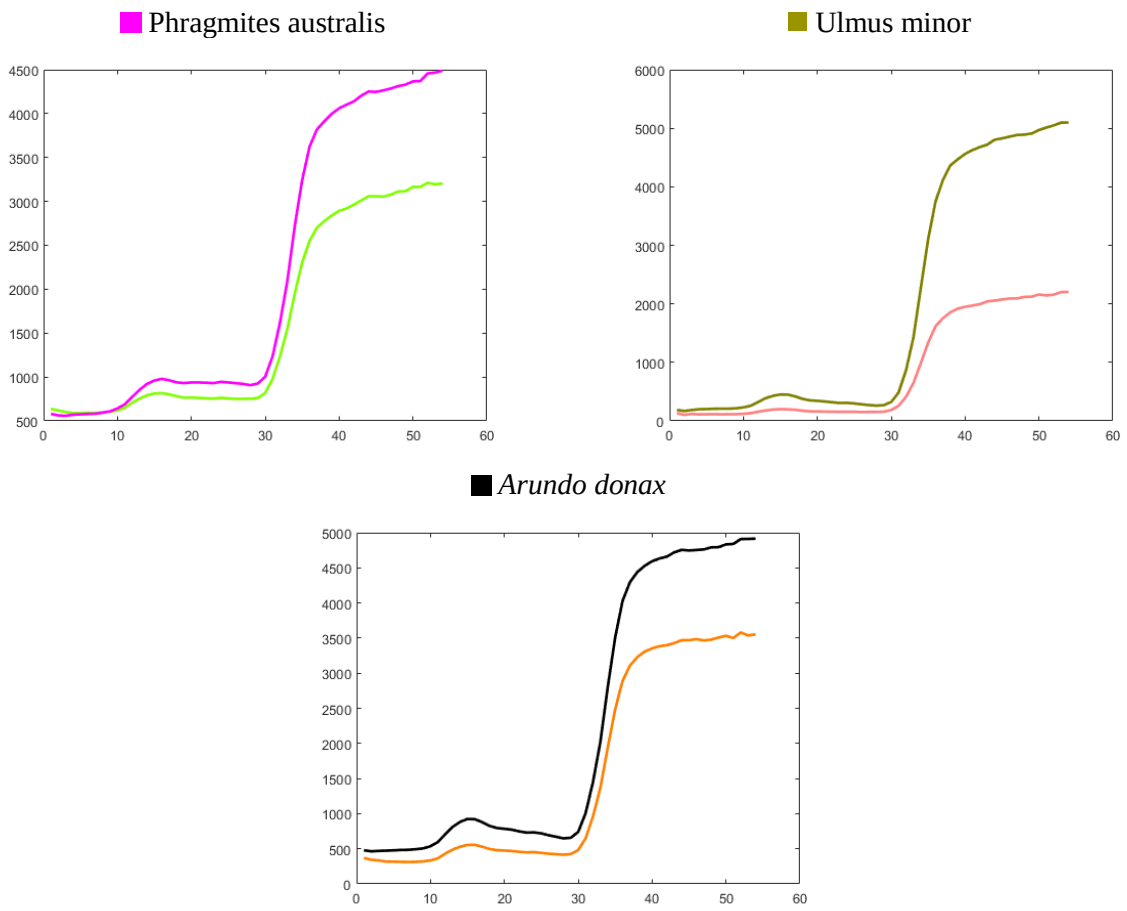



**Figure 49.** Signatures spectrales ± écart type des classes de la VT divisées de l'image hyperspectrale réelle de la Figure 42.

Le Tableau 46 donne les performances des trois méthodes développées HUP-OAP, HUP-OAPM-RSM et HUP-DIA. Ce tableau montre que la méthode HUP-DIA nécessite moins de temps de calcul et d'espace mémoire.

**Tableau 46.** Synthèse des performances des trois méthodes développées pour le partitionnement de l'image hyperspectrale réelle de la Figure 42.

| Méthodes | Nombre de blocs | (Niveau, $N_c$) | CCR (%) | CCR-SCVT (%) | Temps CPU (s) | Espace Mémoire (Mb) |
|---|---|---|---|---|---|---|
| HUP-OAP | 2500 ou (20×20 pixels) | (3, 10) | 73.22 | **100** | 3 426.70 | 9210.9 |
| HUP-OAPM-RSM | | (4, 9) | 82.07 | | 1 172.11 | 2351.8 |
| **HUP-DIA** | | (5, 9) | 90.31 | | **444.78** | **2327.01** |

## 7.3 Application à la reconnaissance faciale avec expressions

Pour cette évaluation, nous avons utilisées trois bases de données des visages bien connus : JAFFE [143], YALE [144] et ORL [145].

Pour cette base de données, les paramètres pour l'extraction des caractéristiques SIFT sont fixés comme suit :

- Nombre d'octaves = 3
- Nombre d'échelles = 3
- Facteur d'échelle ($\sigma$) = 0.5
- Seuil $\zeta = 0.6$ pour JAFFE

### 7.3.1 Application à la base de données JAFFE

La base de données JAFFE contient 213 images de 7 expressions faciales posées par 10 femmes japonaises (cf. Annexe 2). Les sept expressions faciales comprennent six expressions faciales de base, à savoir : bonheur, tristesse, surprise, colère, dégoût, peur et une expression neutre. Ces expressions sont notées comme le montre le Tableau 47.

L'objectif du partitionnement est d'identifier et de regrouper pour chaque femme ces sept expressions faciales chacune dans une classe.

Le nombre total de classes est alors de 70 et il existe trois à quatre représentants par personne pour chaque expression.

**Tableau 47.** Les 7 expressions faciales de JAFFE.

| Expressions | bonheur | tristesse | surprise | colère | dégoût | peur | neutre |
|---|---|---|---|---|---|---|---|
| | HA | SA | SU | AN | DI | FE | NE |

#### *7.3.1.1 Résultats de partitionnement par HUP-OAP*

L'application de HUP-OAP sur cette base de données donne les résultats résumés dans le Tableau 48. La partition optimale qui maximise le critère *LN* est obtenue au niveau 1, où 68 classes sont obtenues au lieu de 70 supposées être des vraies classes. Il n'y a pas de confusion entre les visages des femmes, ce qui représente un CCR de 100%, contre 98.6% donné dans [59] par la méthode MEAP. Le Tableau 49 montre la matrice de confusion et le CCR des expressions faciales des résultats obtenus par l'approche proposée. Le CCR moyen des expressions faciales est de

95.35% si on considère que le classement des expressions de référence (VT) fourni est correct. Par contre, si on considère à la fois la ressemblance visuelle et la similitude des attributs extraits comme le montre la Figure 50, le CCR moyen des expressions faciales peut-être corrigé à 100%. En fait, ces exemples montrent que la partition de la base de données d'origine est probablement erronée, car l'expression du visage marquée comme « triste » (SA) ressemble plus à une expression « bonheur ». Ce qui pose le problème de la subjectivité de l'analyse lors de la construction de ces échantillons d'apprentissage [33].

**Tableau 48.** Partitionnement des données JAFFE par HUP-OAP : nombre estimé de classes ($N_c$), valeur de $LN$ et temps CPU par partition.

| (Niveau, $N_c$) | **(1, 68)** | (2, 22) | (3, 5) |
|---|---|---|---|
| $LN$ | **0.25** | 0.13 | 0.07 |
| Temps CPU (s) | **0.31** | 0.34 | 0.341 |

**Tableau 49.** Matrice de confusion et CCR par HUP-OAP sur la base de données JAFFE au niveau 1.

| KA(23) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | **3** | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | **3** | **0** | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | **4** | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | **4** | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | **3** | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | **3** | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | **3** |
| | **86.96%** | | | | | | |

| KL(22) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | **3** | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | **4** | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | **1** | **2** | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | **3** | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | **3** | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | **3** | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | **3** |
| | **94.45%** | | | | | | |

| KM(22) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | **3** | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | **2** | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | **3** | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | **4** | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | **3** | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | **4** | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | **3** |
| | **100%** | | | | | | |

| KR(20) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | **3** | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | **3** | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | **3** | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | **2** | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | **3** | 0 | 0 |
| SA | 0 | 0 | 0 | **1** | 0 | **2** | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | **3** |
| | **95%** | | | | | | |

| MK(21) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | **3** | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | **3** | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | **3** | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | **3** | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | **3** | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | **3** | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | **3** |
| | **100%** | | | | | | |

| NA(21) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | **3** | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | **0** | **3** | **0** | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | **3** | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | **3** | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | **3** | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | **3** | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | **3** |
| | **100%** | | | | | | |

| NM(20) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | **3** | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | **2** | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | **3** | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | **2** | **1** | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | **3** | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | **3** | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | **3** |
| | **95%** | | | | | | |

| TM(21) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | **3** | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | **3** | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | **3** | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | **3** | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | **3** | **0** | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | **3** | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | **3** |
| | **85.71%** | | | | | | |

| UY(21) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | **3** | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | **3** | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | **3** | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | **3** | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | **3** | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | **3** | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | **3** |
| | **100%** | | | | | | |

| YM(22) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | **3** | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | **3** | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | **1** | **3** | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | **3** | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | **3** | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | **3** | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | **3** |
| | **95.45%** | | | | | | |

**95.35%**

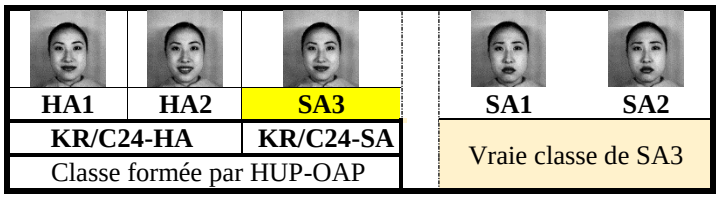



**Figure 50.** Classe obtenue par la méthode HUP-OAP avec confusion entre les expressions HA et SA3 de la femme KR.

### *7.3.1.2 Résultats de partitionnement par HUP-OAPM-RSM*

L'application de HUP-OAPM-RSM sur cette base de données donne les résultats résumés dans le Tableau 50. La partition optimale qui maximise le critère *LN* est obtenue au niveau 1. Le nombre de classes est estimé à 68 au lieu de 70 supposé être le véritable nombre de classes. Il n'y a pas de confusion entre les visages des femmes, ce qui représente un CCR de 100%. Le Tableau 51 montre la matrice de confusion et le CCR des expressions faciales des résultats obtenus par l'approche proposée. Le CCR moyen des expressions faciales est de 95.81% si on considère que le classement des expressions fournies comme référence. Par contre, si on considère à la fois la ressemblance visuelle et la similitude des attributs extraits comme le montre la Figure 51, le CCR moyen des expressions faciales peut être considéré comme 100%. Cet exemple montre comme précédemment dit, que la partition de la base de données d'origine est probablement erronée, car les expressions du visage marquées comme « Colère » (AN) et « Dégout » (DI) se ressemblent fortement.

**Tableau 50**. Partitionnement des données JAFFE par HUP-OAPM-RSM : nombre estimé de classes ($N_c$), valeur de $LN$ et temps CPU par partition.

| (Niveau, $N_c$) | **(1, 68)** | (2, 25) | (3, 6) |
|---|---|---|---|
| $LN$ | **0.26** | 0.14 | 0.08 |
| Temps CPU (s) | **0.30** | 0.33 | 0.331 |

**Tableau 51.** Matrice de confusion et CCR par HUP-OAPM-RSM sur la base de données JAFFE au niveau 1.

| KA(23) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 4 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 4 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | **86.96%** | | | | | | |

| KL(22) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 4 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | **100%** | | | | | | |

| KM(22) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 4 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 4 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | **100%** | | | | | | |

| KR(20) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 1 | 0 | 2 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | **95%** | | | | | | |

| MK(21) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | **100%** | | | | | | |

| NA(21) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | **100%** | | | | | | |

| NM(20) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 2 | 1 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | **95%** | | | | | | |

| TM(21) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | **85.71%** | | | | | | |

| UY(21) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | **3** | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | **3** | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | **3** | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | **3** | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | **3** | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | **3** | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | **3** |
| | **100%** | | | | | | |

| YM(22) | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | **3** | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | **3** | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | **1** | **3** | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | **3** | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | **3** | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | **3** | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | **3** |
| | **95.45%** | | | | | | |

**95.81%**


**Figure 51.** Classe obtenue par la méthode HUP-OAPM-RSM avec confusion entres les expressions AN et DI de la femme KA.

#### *7.3.1.3 Résultats de partitionnement par HUP-DIA*

L'application de HUP-DIA donne les résultats résumés dans le Tableau 52. 68 classes sont obtenues au niveau 1 au lieu de 70 supposées être des vraies classes. Le taux de reconnaissance des visages est de 100% et celui des expressions est de 96.76%. Le Tableau 53 montre la matrice de confusion et le CCR des résultats obtenus par l'approche proposée

**Tableau 52**. Partitionnement des données JAFFE par HUP-DIA : nombre estimé de classes ($N_c$), valeur de *LN* et temps CPU par partition.

| (Niveau, $N_c$) | **(1, 68)** | (2, 27) | (3, 8) |
|---|---|---|---|
| *LN* | **0.27** | 0.15 | 0.10 |
| Temps CPU (s) | **0.26** | 0.262 | 0.2621 |

**Tableau 53.** Matrice de confusion et CCR par HUP-DIA sur JAFFE au niveau 1.

| KA | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 4 | 0 | 0 | 0 | 0 |
| HA | 1 | 0 | 0 | 3 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | 82.6% | | | | | | |

| KL | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 4 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | 100% | | | | | | |

| KM | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 4 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 4 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | 100% | | | | | | |

| KR | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 1 | 0 | 2 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | 95% | | | | | | |

| MK | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | 100% | | | | | | |

| NA | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | 100% | | | | | | |

| NM | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | 90% | | | | | | |

| TM | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | 100% | | | | | | |

| UY | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | 100% | | | | | | |

| YM | AN | DI | FE | HA | NE | SA | SU |
|---|---|---|---|---|---|---|---|
| AN | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| DI | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| FE | 0 | 0 | 4 | 0 | 0 | 0 | 0 |
| HA | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| NE | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| SA | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| SU | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | 100% | | | | | | |

96.76%

### 7.3.2 Application à la base de données YALE

La base de données des visages Yale contient 165 images frontales de 15 sujets. Il y a 11 images par sujet, une pour chacune des expressions ou configurations faciales suivantes : lumière

centrale, avec lunettes, heureux, lumière gauche, sans lunettes, normal, lumière droite, triste, somnolent, surpris et clin d'œil. Notons que cette base est plus difficile que la base de données JAFFE car elle montre de plus grandes variations d'éclairage (éclairage non uniforme). Les 11 configurations sont résumées dans le Tableau 54.

Pour cette base de données nous avons étudié l'influence du choix du paramètre $\zeta$ lors de la construction de la matrice de similarité.

**Tableau 54**. Les onze configurations de la base de données YALE.

| Configurations | lunette | sans lunette | heureux | lumière centrale | lumière gauche | lumière droite | triste | somnolent | surpris | clin d'œil | normal |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | LU | SL | HE | LC | LG | LD | TR | SO | SU | CO | NO |

#### *7.3.2.1 Résultats de partitionnement par HUP-OAP*

Pour analyser l'influence de seuil $\zeta$ lors de la construction de la matrice de similarité, nous avons exécuté HUP-OAP en faisant varier le paramètre de similitude $\zeta$ de 0.5 à 1 avec un pas de 0.1. Les valeurs de CCR et de *LN* rapportées en fonction de $\zeta$ représentées dans le Tableau 55, montrent que sur l'ensemble de données Yale, la valeur de $\zeta$ la plus appropriée est 1 où la méthode HUP-OAP proposée a généré les meilleurs résultats avec un CCR de 96.96%.

La Figure 52 montre une classe obtenue par HUP-OAP au niveau 1 où il y a une confusion entre les individus de différentes classes et la Figure 53 montre les individus de la vraie classe de l'individu (1) et le nombre de correspondance entre l'individu (1) avec tous les autres pour vérifier l'origine de cette confusion. Nous remarquons que le nombre de correspondance est maximal entre l'individu (1) et (4) et aussi entre les individus de la classe formée par HUP-OAP à laquelle il est agrégé. C'est pour cette raison, il n'a pas été regroupé avec les individus de sa vraie classe. De plus d'après la Figure 54 qui représente la correspondance entre l'individu (1) et trois autres individus (2-9-15), nous constatons que juste le fond est détecté et pas les expressions des individus, ce qui confirme que les caractéristiques détectées ne sont pas discriminantes et sont donc la source du mauvais partitionnement.

**Tableau 55.** Résultats d'estimation du nombre de classes ($N_c$) sur les données YALE, le CCR-SCVT et l'indice *LN* obtenus par HUP-OAP au niveau 1 en fonction de $\zeta$.

| $\zeta$ | HUP-OAP | | |
|---|---|---|---|
| | $N_c$ | CCR-SCVT (%) | *LN* |
| **1** | **36** | **96.96** | **0.39** |
| 0.9 | 35 | 92.68 | 0.35 |
| 0.8 | 40 | 91.52 | 0.36 |
| 0.7 | 38 | 90.91 | 0.24 |
| 0.6 | 37 | 89.70 | 0.22 |
| 0.5 | 40 | 87.88 | 0.20 |

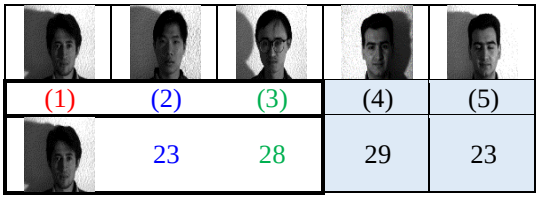



**Figure 52.** Exemple d'une classe obtenue par HUP-OAP avec $\zeta = 1$ : confusion des individus (1), (2) et (3) des classes $C_3$, $C_4$ et $C_8$ avec ceux des individus (4) et (5) de la classe $C_{15}$ et nombre de mises en correspondance.

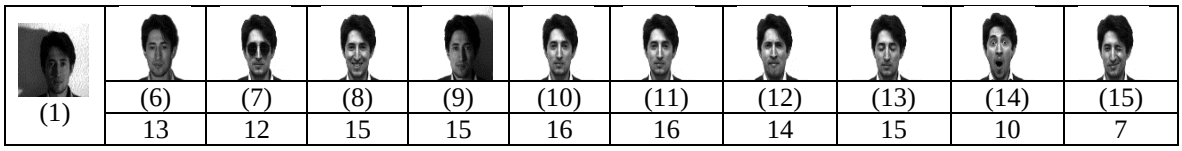



**Figure 53.** La vraie classe d'appartenance de l'individu (1) d'après les données de la VT et nombre de correspondances

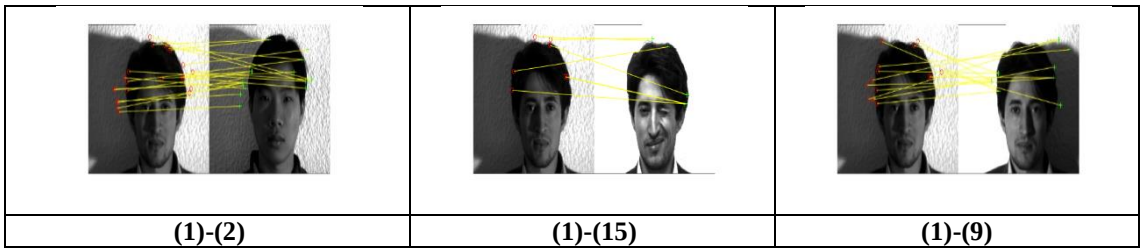



**Figure 54.** Exemple de mises en correspondance entre l'individu (1) et les individus (2), (15) et (9).

### *7.3.2.2 Résultats de partitionnement par HUP-OAPM-RSM et HUP-DIA*

Comme le résultat optimal par HUP-OAP est obtenu pour $\zeta = 1$, alors pour l'application de HUP-OAPM-RSM et HUP-DIA nous fixons $\zeta$ à 1. Les partitions optimales obtenues par HUP-OAPM-RSM et HUP-DIA correspondent au niveau 1 où le nombre de classes est estimé à 36 et à 35 respectivement. Pour les deux méthodes le CCR-SCVT correspondant est de 97.58%. Les mêmes remarques faites avant sur les résultats obtenus par HUP-OAP peuvent être faites sur ces 2 méthodes.

### 7.3.3 Application à la base de données ORL

Cette base de données de visages a été créée au laboratoire AT&T. Elle contient 40 personnes de sexe différent. Les images sont de taille 112×92 pixels. 10 vues différentes de chaque sujet ont été collectées. Ces vues présentent différentes poses et expressions faciales (expression neutre, sourire et yeux fermés) et des occlusions partielles par des lunettes, sous des conditions de luminosité variables.

L'objectif ici est de regrouper les 40 personnes dans 40 classes différentes où chaque classe contient l'ensemble des 10 individus indépendamment de la vue.

#### *7.3.3.1 Résultats de partitionnement par HUP-OAP*

Comme pour le partitionnement de la base YALE, le Tableau 56 montre les CCR-SCVT obtenus par HUP-OAP en fonction de la valeur de $\zeta$. Les valeurs de CCR-SCVT et *LN* montrent que sur l'ensemble de données ORL, la valeur de $\zeta$ la plus appropriée est 1 donnant un CCR de 99.25%.

La Figure 55 montre une classe obtenue par HUP-OAP au niveau 1 où il y a une confusion entre les individus des différentes classes (marquées par la couleur jaune) et la Figure 56 montre les individus de la vraie classe de l'individu (79) et le nombre de correspondances entre l'individu (79) avec tous les autres, pour vérifier l'origine de cette confusion. Nous remarquons que le nombre de correspondance est maximal entre l'individu (79) et (98) et aussi entre les individus de la classe formée par HUP-OAP à laquelle il est agrégé. C'est pour cette raison, il n'a pas été regroupé avec les individus de sa vraie classe.

**Tableau 56.** Résultats d'estimation du nombre de classes ($N_c$) de la base ORL par HUP-OAP en fonction de $\zeta$ : le CCR-SCVT et l'indice *LN*.

| $\zeta$ | HUP-OAP | | |
|---|---|---|---|
| | $N_c$ | CCR-SCVT (%) | *LN* |
| **1** | **117** | **98.75** | **0.43** |
| 0.9 | 125 | 98.25 | 0.40 |
| 0.8 | 126 | 98 | 0.38 |
| 0.7 | 131 | 97.75 | 0.37 |
| 0.6 | 122 | 97.5 | 0.32 |
| 0.5 | 121 | 96.25 | 0.34 |

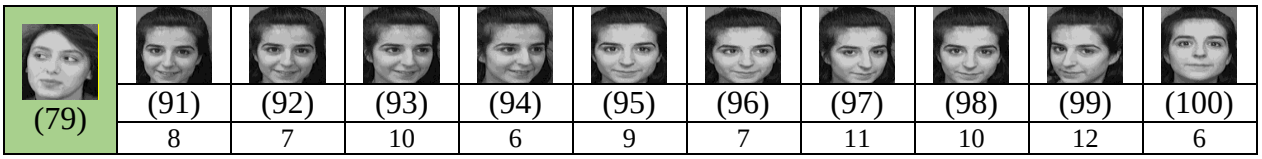



**Figure 55.** Exemple d'un individu (79) mal classé par HUP-OAP et nombre de correspondances.

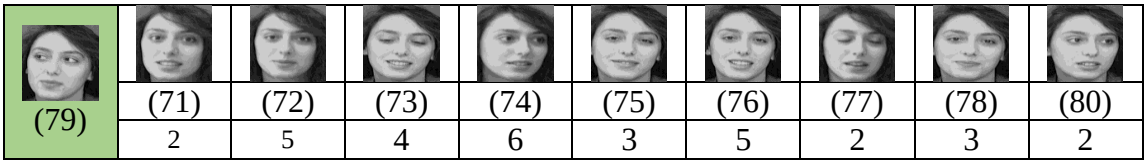



**Figure 56.** La vraie classe d'appartenance de l'individu (79) et nombre de correspondances.

#### *7.3.3.2 Résultats de partitionnement par HUP-OAPM-RSM et HUP-DIA*

Etant donné que la partition optimale est obtenue par HUP-OAP pour $\zeta = 1$, alors pour l'application de HUP-OAPM-RSM et HUP-DIA nous fixons $\zeta$ à 1. Les partitions optimales par HUP-OAPM-RSM et HUP-DIA sont obtenues au niveau 1 où le nombre de classes est estimé à 116 et à 122 respectivement. Le CCR-SCVT est de 99.25% et 99.50% respectivement. Les remarques faites sur les résultats obtenus par HUP-OAP sont pratiquement identiques à celles de ces 2 méthodes.

Pour résumer toutes les évaluations effectuées sur les données d'expressions faciales, nous présentons dans le Tableau 57, les performances des méthodes HUP-OAP, HUP-OAPM-RSM et HUP-DIA. Ce tableau montre que la méthode HUP-DIA est la plus rapide, a le moins besoin d'espace mémoire et donne des CCR légèrement supérieurs en se référant aux données des VT.

**Tableau 57**. Performances des trois méthodes développées pour la reconnaissance faciale avec expressions.

| Bases de données | Critères | HUP-OAP | HUP-OAPM-RSM | HUP-DIA |
|---|---|---|---|---|
| JAFFE | $N_c$ | 68 | 68 | **68** |
| | CCR (%) | 95.31 | 95.77 | **96.24** |
| | Temps CPU (s) | 0.31 | 0.30 | **0.26** |
| | Espace Mémoire (Mb) | 1.17 | 1.15 | **1.14** |
| | | | | |
| YALE | $N_c$ | 36 | 36 | **35** |
| | CCR-SCVT (%) | 96.96 | **97.58** | **97.58** |
| | Temps CPU (s) | 0.46 | 0.45 | **0.24** |
| | Espace Mémoire (Mb) | 0.64 | 0.64 | **0.63** |
| | | | | |
| ORL | $N_c$ | 117 | 116 | **122** |
| | CCR-SCVT (%) | 98.75 | 99.25 | **99.5** |
| | Temps CPU (s) | 0.69 | 0.68 | **0.52** |
| | Espace Mémoire (Mb) | 3.92 | 3.91 | **3.90** |

## 7.4 Applications médicales

Pour cette évaluation, nous avons utilisées deux bases de données médicales qui sont : images IRM et images mélanomes, qui sont présentés dans l'Annexe 3.

Pour ces bases de données, les paramètres pour l'extraction des caractéristiques SIFT sont fixés comme suit :

- Nombre d'octaves = 4
- Nombre d'échelles = 5
- Facteur d'échelle ($\sigma$) = 0.5
- Seuil $\zeta = 1$

### 7.4.1 Application à la détection de tumeur cérébrale par IRM

Cette base de données contient 394 images du cerveau contenant 4 classes qui sont : glioma_tumor, Meningiorma_tumor, No_tumor et Pituitary_tumor, où chaque classe contient au moins 5 sous classes.

Deux expérimentations ont été faites : la première est l'application de HUP-OAP, HUP-OAPM-RSM et HUP-DIA sur les images originales directement et la deuxième en éliminant le crâne. Les résultats obtenus par les trois méthodes sont résumés dans le Tableau 58, le Tableau 59 et le Tableau 60 respectivement. Ces résultats montrent que le critère *LN* est optimale au niveau 1 avec ou sans la présence du crâne où le nombre de classes est estimé à 35, 37 et 40 respectivement avec crâne et à 62, 64 et 66 sans crâne respectivement. Nous remarquons qu'en éliminant le crâne les résultats sont meilleurs qu'avec crâne pour les trois méthodes développées.

**Tableau 58.** Résultats de partitionnement par HUP-OAP pour la détection de tumeur cérébrale par IRM.

| Niveau | Avec crâne | | | Sans crâne | | |
|---|---|---|---|---|---|---|
| | CCR-SCVT (%) | *LN* | $N_c$ | CCR-SCVT (%) | *LN* | $N_c$ |
| 1 | 92.85 | 0.33 | 35 | **96.36** | **0.36** | **62** |
| 2 | 91.64 | 0.31 | 10 | 95.62 | 0.34 | 15 |
| 3 | 61.89 | 0.19 | 3 | 89.34 | 0.27 | 5 |

**Tableau 59.** Résultats de partitionnement par HUP-OAPM-RSM pour la détection de tumeur cérébrale par IRM.

| Niveau | Avec crâne | | | Sans crâne | | |
|---|---|---|---|---|---|---|
| | CCR-SCVT (%) | $LN$ | $N_c$ | CCR-SCVT (%) | $LN$ | $N_c$ |
| 1 | 93.15 | 0.34 | 37 | **97.21** | **0.37** | **64** |
| 2 | 92.60 | 0.32 | 13 | 96.10 | 0.35 | 18 |
| 3 | 78.12 | 0.25 | 7 | 86.29 | 0.30 | 8 |
| 4 | 63.20 | 0.20 | 5 | 81.21 | 0.29 | 6 |

**Tableau 60.** Résultats de partitionnement par HUP-DIA pour détection de tumeur cérébrale par IRM.

| Niveau | Avec crâne | | | Sans crâne | | |
|---|---|---|---|---|---|---|
| | CCR (%) | $LN$ | $N_c$ | CCR (%) | $LN$ | $N_c$ |
| 1 | 91.37 | 0.36 | 40 | **97.46** | **0.39** | **66** |
| 2 | 88.83 | 0.29 | 15 | 96.70 | 0.36 | 18 |
| 3 | 76.11 | 0.24 | 6 | 93.40 | 0.30 | 6 |

### 7.4.2 Application à la détection de mélanomes

Une base de données d'images de lésions a été sélectionnée. Elle est composée d'images de la base de données de l'ISIC [146], sélectionnées par diagnostique (nævi sains ou mélanomes malins), où les lésions sont totalement présentes. Le nombre d'images sélectionné est 98, dont 48 images correspondent aux mélanomes malins et 50 images correspondent aux nævus sains.

Le Tableau 61 montre les résultats obtenus par les méthodes développées par niveau de partition. Nous ne pouvons pas calculer le CCR sur cette base de données car les détails de la VT ne sont pas fournis.

**Tableau 61.** Résultats par niveau de partitionnements des trois méthodes développées pour la détection de mélanomes.

| Niveau | HUP-OAP | | HUP-OAPM-RSM | | HUP-DIA | |
|---|---|---|---|---|---|---|
| | $N_c$ | $LN$ | $N_c$ | $LN$ | $N_c$ | $LN$ |
| 1 | 19 | 0.07 | 21 | 0.09 | **22** | **0.10** |
| 2 | 5 | 0.05 | 10 | 0.08 | 6 | 0.07 |
| 3 | 2 | 0.03 | 7 | 0.06 | 2 | 0.04 |
| 4 | - | - | 2 | 0.03 | 1 | 0.01 |

## 7.5 Discussion

Les trois approches non supervisées et hiérarchiques développées ont été appliquées et évaluées sur trois domaines applicatifs différents (environnement, reconnaissance faciale avec expressions et médecine). Les résultats obtenus montrent la robustesse, la pertinence et le caractère général de ces trois méthodes développées. La méthode HUP-DIA est la plus rapide et nécessite moins d'espace mémoire, en deuxième position vient la méthode HUP-OAPM-RSM puis la méthode HUP-OAP.

# Conclusion générale

Dans cette thèse, nous avons proposé trois approches de partitionnement de données hiérarchiques et une méthode automatique de sélection des échantillons d'apprentissage.

Les méthodes de partitionnement développées sont hiérarchiques, non supervisées et non paramétriques applicable sur des données de grande taille. Ces méthodes présentes neuf avantages principaux qui peuvent être objectivement énumérés : 1) aucune connaissance *a priori* n'est requise, notamment la non-introduction d'échantillons d'apprentissage, pour donner à l'utilisateur la possibilité de détecter et de localiser des classes connues et de nouvelles classes dites "classes de découverte" ; 2) stabilité des résultats grâce à leur caractère déterministe ; 3) la sélection de l'exemplaire de chaque classe et l'affectation d'un pixel (ou d'un individu) à une classe se font de manière très élaborée selon des critères d'optimisation ; 4) temps de calcul très faible avec le traitement par blocs, contrairement aux méthodes comparées ; 5) applicable aux données ou images quelle que soit leur taille avec la possibilité de paralléliser le partitionnement des blocs ; 6) possibilité d'élaborer plusieurs partitions hiérarchiques en indiquant la plus pertinente selon un critère objectif ; 7) possibilité de sélectionner objectivement les échantillons des classes dans un système d'apprentissage afin de pouvoir les détecter par la suite ; 8) possibilité de valider et de corriger les échantillons d'apprentissage et enfin, 9) applicable à plusieurs domaines sans contraintes d'apprentissage.

Les approches développées ont été évaluées sur trois domaines applicatifs différents (environnement, reconnaissance faciale avec expressions et médecine). Ces évaluations montrent que les résultats sont pertinents sans aucune intervention de l'utilisateur final. Les taux de classification correcte (CCR) obtenus par les méthodes proposées sont meilleurs que ceux des méthodes semi-supervisées comme le S-OFCM, le *K*-means et le FCM. Elles surpassent également les méthodes non supervisées comparées telles que U-OFCM et la méthode AP originale. Les différentes évaluations ont montré également que la méthode HUP-DIA donne un résultat de partitionnement plus fin avec un nombre estimé de classes supérieur à ceux de la méthode HUP-OAP et de la méthode HUP-OAPM-RSM. Elle est aussi plus performante suivant les critères temps de calcul et espace mémoire, suivie par la méthode HUP-OAPM-RSM.

Quant à la méthode de sélection des échantillons d'apprentissage proposée, elle permet de fournir une solution au problème de la non-disponibilité des échantillons d'apprentissage, que le nombre de classes soit connu ou non. Elle permet également de valider et de corriger les vérités de terrain biaisées ou simplifiées existantes lorsqu'elles sont disponibles. Tout d'abord, le partitionnement hiérarchique et non supervisé est appliqué sur les données disponibles, puis trois alternatives sont proposées pour la sélection des échantillons d'apprentissage, basées sur la moyenne et l'écart-type des distances entre les objets d'une classe et leur exemplaire. L'introduction d'un partitionnement non supervisé et non paramétrique comme une étape préliminaire pour les méthodes supervisées contribue fortement à l'amélioration de leurs performances.

Les évaluations de la méthode de sélection des échantillons d'apprentissage proposée sur trois bases de données illustrent la pertinence des ensembles d'apprentissage créés. En effet, l'application de trois algorithmes supervisés (SVM, KNN et ANN) en introduisant des ensembles d'apprentissage sélectionnés de manière non supervisée montre que les performances de ces méthodes sont meilleures qu'en utilisant des échantillons sélectionnés de manière aléatoire.

L'architecture des méthodes développées permet l'exécution parallèle des différentes étapes, réduisant ainsi le temps de calcul.

Quelques perspectives de ces travaux concernent tout d'abord l'évaluation du temps de calcul en parallélisant au maximum le partitionnement des blocs, même si ce temps reste dépendant de la puissance de calcul disponible de chaque machine. Ensuite, une étape de validation finale est à mener sur d'autres bases de données accompagnées de la VT, sans reproche, en nombre significatif.

# Annexes

## Annexe 1 Transformation de caractéristiques visuelles invariante à l'échelle (SIFT)

La transformation de caractéristiques visuelles invariante à l'échelle [142] est un algorithme utilisé pour détecter et identifier les éléments similaires entre différentes images numériques.

L'étape fondamentale de la méthode consiste à calculer ce que l'on appelle les « descripteurs SIFT » des images à étudier. Il s'agit d'informations numériques dérivées de l'analyse locale d'une image et qui caractérisent le contenu visuel de cette image de la façon la plus indépendante possible de l'échelle (zoom et résolution du capteur), du cadrage, de l'angle d'observation et de l'exposition (luminosité).

- **Définition : Caractéristiques d'une image**

Désignent des zones d'intérêt de l'image, qui peuvent correspondre à des contours, des points ou des régions d'intérêts. A chaque attribut ou caractéristique détecté, est associé un vecteur, appelé descripteur qui décrit la zone concernée. Ces descripteurs doivent :

- se retrouver dans les images représentant la même scène malgré les différences géométriques et photométriques ;
- être suffisamment unique et non-ambiguë au sein d'une image ;
- correspondre à une zone suffisamment petite décrite selon son voisinage seulement.

La méthode proposée comprend deux parties :

1) **Détection :** création de l'espace des échelles, calcul de la différence de gaussiennes et localisation des points d'intérêt.
2) **Description :** assignation d'orientation et création des descripteurs.

## ➢ Détection

La première étape de l'algorithme est la détection des points d'intérêt, dits *points-clés*. Un point-clé ($x$, $y$ et $\sigma$) est défini d'une part, par ses coordonnées sur l'image $(x, y)$, et d'autre part, par son facteur d'échelle caractéristique ($\sigma$). Il s'agit d'une zone d'intérêt circulaire, le rayon de la zone étant proportionnel au facteur d'échelle.

- **Création de l'espace d'échelle**

Cela consiste à flouter l'image $Im$ et à réduire sa taille plusieurs fois. Le flou est créé en appliquant un filtre gaussien de variance $\sigma$, notée $G$, qui va estomper les détails de l'image de rayon inférieur à $\sigma$. Soit :

$$L(x, y, \sigma) = G(x, y, \sigma) * Im(x, y)$$

avec * l'opérateur de convolution.

- **Calcul de la différence de gaussiennes (DoG)**

La détection des objets de dimension approximativement égale à $\sigma$ se fait en étudiant l'image appelée différence de gaussiennes définie comme suit :

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

où $k$ est un paramètre fixe de l'algorithme qui dépend de la finesse de la discrétisation de l'espace des échelles voulue.

Dans cette image seuls les objets observables dans des facteurs d'échelle qui varient entre $\sigma$ et $k\sigma$ sont préservés.

- **Localisation des points d'intérêt**

Les points d'intérêt font partie des extrema locaux de l'ensemble des DoG. De ce fait, un point-clé candidat $(x, y, \sigma)$ est défini comme un point où un extremum du DoG est atteint par rapport à ses voisins immédiats. Pour les trouver, chaque pixel est comparé à ses 8 voisins, mais également à ses 9 voisins dans les DoG au-dessus et en dessous.

L'étape de détection d'extremums produit en général un grand nombre de points-clés candidats, dont certains sont instables. De ce fait, des traitements supplémentaires sont appliqués, pour un objectif double : d'une part, reconverger la position des points pour améliorer la précision sur $x, y$ et $\sigma$ ; d'autre part, éliminer les points de faible contraste ou situés sur des arêtes de contour à faible courbure et donc susceptibles de « glisser » facilement.

- **Amélioration de la précision par interpolation des coordonnées**

Cette interpolation s'obtient par un développement de Taylor à l'ordre 2 de la fonction DoG, $D(x, y, \sigma)$, en prenant comme origine les coordonnées du point-clé candidat. Ce développement s'écrit comme suit :

$$D(\mathrm{x}) = D + \frac{\partial D^T}{\partial \mathrm{x}}\mathrm{x} + \frac{1}{2}\mathrm{x}^T \frac{\partial^2 D}{\partial \mathrm{x}^2}\mathrm{x}$$

où $D$ et ses dérivées sont évaluées au point-clé candidat $\mathrm{x} = (x, y, \sigma)^T$. Les dérivées sont estimées par différences finies à partir des points voisins connus de façon exacte. La position précise de l'extremum $\hat{\mathrm{x}}$ est déterminée en résolvant l'équation annulant la dérivée de cette fonction par rapport à x ; on trouve ainsi :

$$\hat{\mathrm{x}} = \frac{\partial^2 D^{-1}}{\partial \mathrm{x}^2}\frac{\partial D}{\partial \mathrm{x}}$$

Un $\hat{\mathrm{x}}$ supérieur à 0.5 dans l'une des trois dimensions signifie que le point considéré est plus proche d'un des voisins dans l'espace des échelles discret. Dans ce cas, le point-clé candidat est mis à jour et l'interpolation est réalisée à partir des nouvelles coordonnées. Sinon, le delta est ajouté au point candidat initial qui gagne ainsi en précision.

- **Élimination des points-clés de faible contraste**

La valeur de $D(\mathrm{x})$ aux coordonnées précises $\hat{\mathrm{x}}$ du point-clé peut être calculée à partir du développement de Taylor de cette fonction et constitue donc un extremum local. Un seuillage absolu sur cette valeur permet d'éliminer les points instables, à faible contraste.

- **Élimination des points situés sur les arêtes**

Les points situés sur les arêtes (ou contours) doivent être éliminés car la fonction DoG y prend des valeurs élevées, ce qui peut donner naissance à des extremums locaux instables, très sensibles au bruit.

Un point candidat à éliminer, si l'on considère les deux directions principales à sa position, est caractérisé par le fait que sa courbure principale le long du contour sur lequel il est positionné est très élevée par rapport à sa courbure dans la direction orthogonale.

La courbure principale est représentée par les valeurs propres de la matrice Hessienne *MH* :

$$MH = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

Les dérivées doivent être évaluées aux coordonnées du point d'intérêt $(x, y, \sigma)$ dans l'espace des échelles. Les valeurs propres de *MH* sont proportionnelles aux courbures principales de $D$, dont seul le rapport $\rho$ est intéressant. La trace de *MH* représente la somme de ces valeurs, le déterminant son produit. Par conséquent, en adoptant un seuil $r_{th}$ sur le ratio des courbures ($r_{th}$ =10), un point-clé candidat va être retenu, si :

$$\frac{tr(MH)^2}{\det(MH)} = \frac{(\rho + 1)^2}{\rho} < \frac{(r_{th} + 1)^2}{r_{th}}$$

Lorsque ce critère n'est pas vérifié, le point est considéré comme localisé le long d'une arête et il est par conséquent rejeté.

- **Description**
  - **Assignation d'orientation**

L'étape d'assignation d'orientation consiste à attribuer à chaque point-clé une ou plusieurs orientations déterminées localement sur l'image à partir de la direction des gradients dans un voisinage autour du point. Cette étape est essentielle pour garantir l'invariance de ceux-ci à la rotation.

Pour un point-clé donné $(x_0, y_0, \sigma_0)$, le calcul s'effectue sur $L(x, y, \sigma_0)$, à savoir le gradient de la pyramide dont le paramètre est le plus proche du facteur d'échelle du point. De cette façon, le calcul est également invariant à l'échelle. À chaque position dans un voisinage du point-clé, on estime le

gradient par différences finies symétriques, puis sa norme ou son amplitude $am(x, y)$ et son orientation $\theta(x, y)$ :

$$am(x, y) = \sqrt{\left(L(x+1, y) - L(x-1, y)\right)^2 + \left(L(x, y+1) - L(x, y-1)\right)^2}$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right)$$

Un histogramme des orientations sur le voisinage est réalisé avec 36 intervalles, couvrant chacun 10 degrés d'angle. Chaque voisin est doublement pondéré dans le calcul de l'histogramme, d'une part, par son amplitude $am(x, y)$; d'autre part, par une fenêtre circulaire gaussienne de paramètre égal à 1.5 fois le facteur d'échelle $\sigma_0$ du point-clé. Les pics dans cet histogramme correspondent aux orientations dominantes. À l'issue de cette étape, un point-clé est donc défini par quatre paramètres $(x, y, \sigma, \theta)$.

- **Descripteur de point-clé**

L'étape de calcul des vecteurs descripteurs traduit numériquement les points-clés définis.

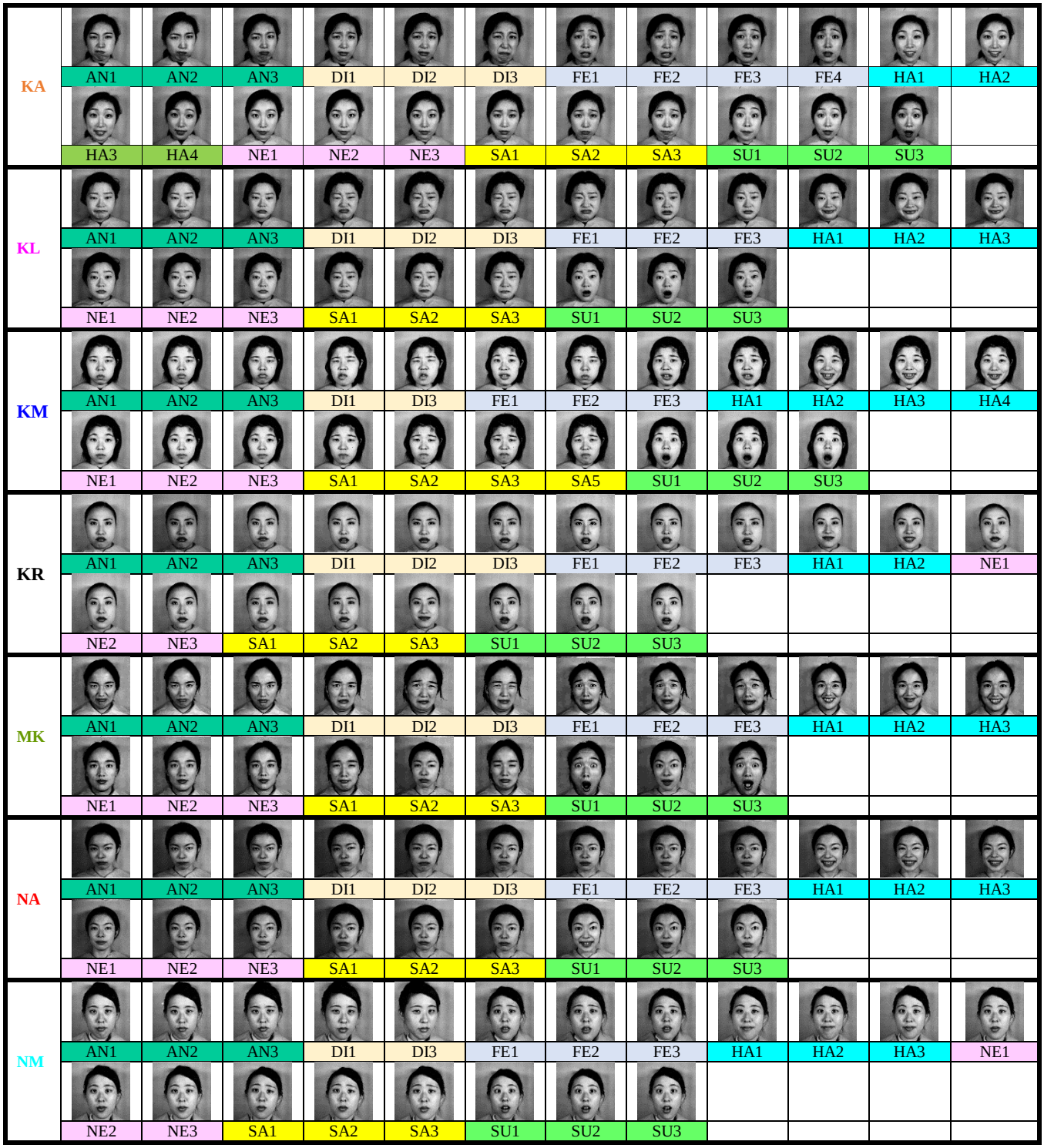
Autour d'un point-clé, on commence par modifier le système de coordonnées local pour garantir l'invariance à la rotation, en utilisant une rotation d'angle égal à l'orientation du point-clé, mais de sens opposé. On considère ensuite, toujours autour du point-clé, une région de 16 × 16 pixels, subdivisée en 4 × 4 zones de 4 × 4 pixels chacune. Sur chaque zone est calculé un histogramme des orientations comportant 8 intervalles. En chaque point de la zone, l'orientation et l'amplitude du gradient sont calculés comme précédemment. L'orientation détermine l'intervalle à incrémenter dans l'histogramme, ce qui se fait avec une double pondération – par l'amplitude et par une fenêtre gaussienne centrée sur le point-clé, de paramètre égal à 1.5 fois le facteur d'échelle du point-clé.

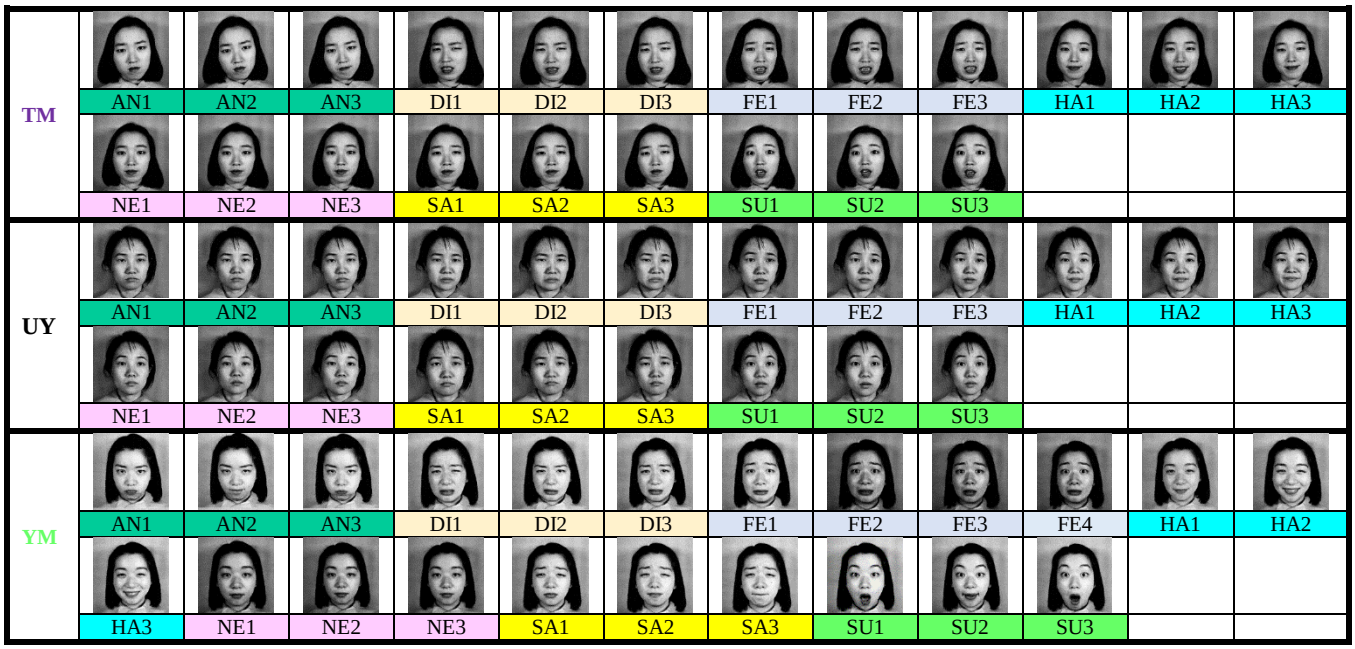
Ensuite, les 16 histogrammes à 8 intervalles chacun sont concaténés et normalisés. Dans le but de diminuer la sensibilité du descripteur aux changements de luminosité, les valeurs sont plafonnées à 0.2 et l'histogramme est de nouveau normalisé, pour finalement fournir le descripteur SIFT du point-clé, de dimension 128.

# Annexe 2 Bases de données expressions faciales

- **2.1 Base de données JAFFE**

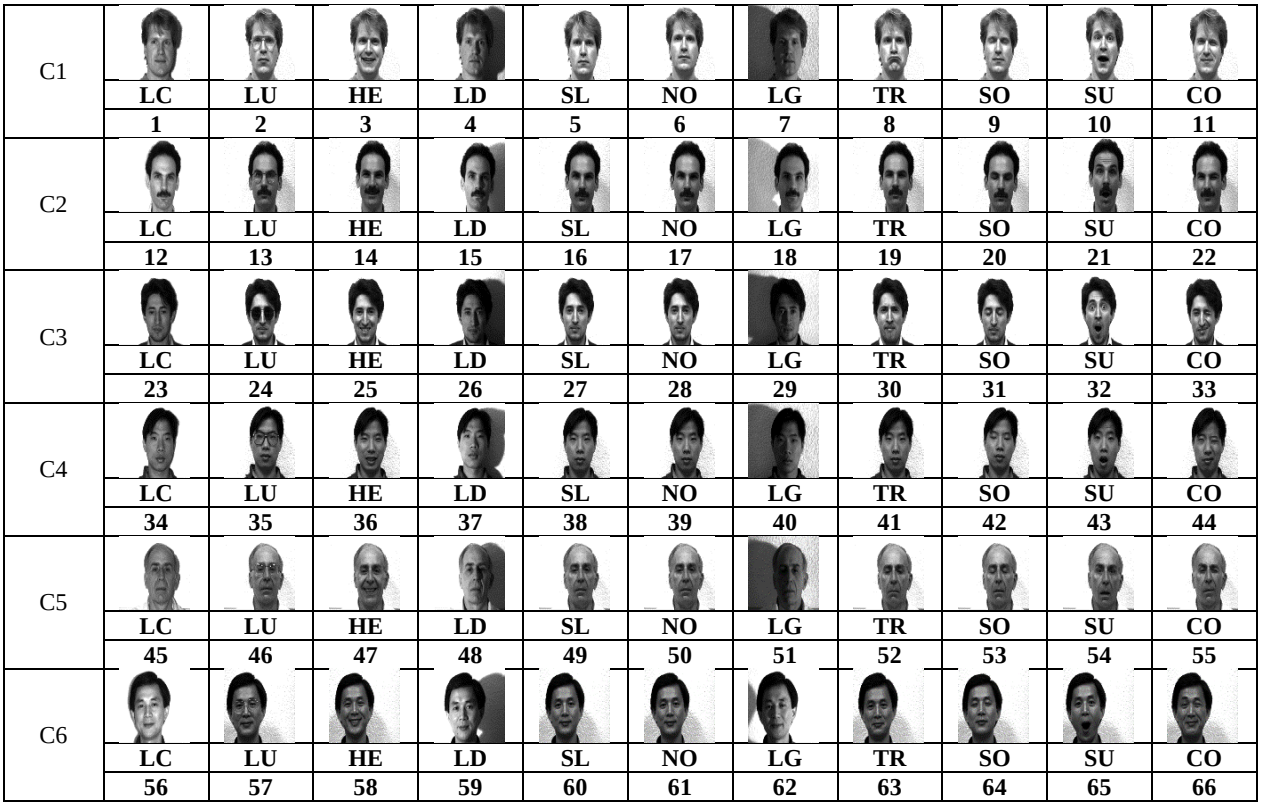
213 images monochromes : 10 classes de visages de femmes avec 7 expressions différentes.






- **2.2 Base de données YALE**

165 images monochromes : 15 visages d'hommes différents avec 11 expositions différentes.

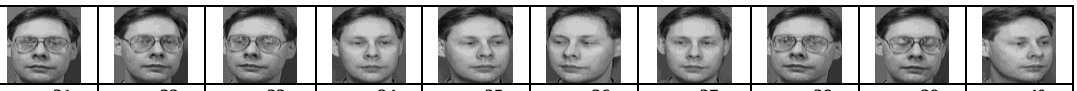
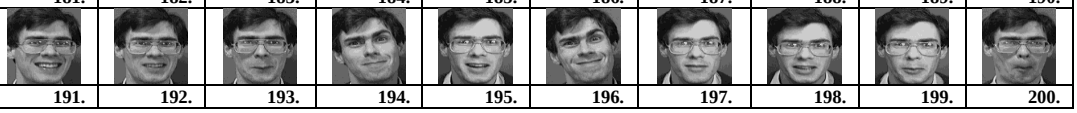


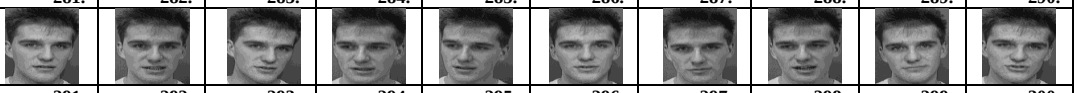
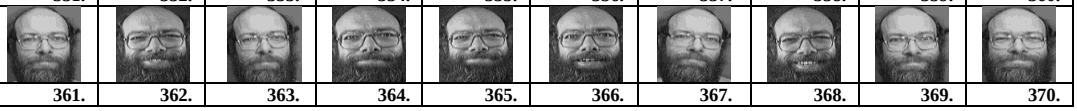
| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C7 | | | | | | | | | | | |
| | **LC** | **LU** | **HE** | **LD** | **SL** | **NO** | **LG** | **TR** | **SO** | **SU** | **CO** |
| | **67** | **68** | **69** | **70** | **71** | **72** | **73** | **74** | **75** | **76** | **77** |
| C8 | | | | | | | | | | | |
| | **LC** | **LU** | **HE** | **LD** | **SL** | **NO** | **LG** | **TR** | **SO** | **SU** | **CO** |
| | **78** | **79** | **80** | **81** | **82** | **83** | **84** | **85** | **86** | **87** | **88** |
| C9 | | | | | | | | | | | |
| | **LC** | **LU** | **HE** | **LD** | **SL** | **NO** | **LG** | **TR** | **SO** | **SU** | **CO** |
| | **89** | **90** | **91** | **92** | **93** | **94** | **95** | **96** | **97** | **98** | **99** |
| C10 | | | | | | | | | | | |
| | **LC** | **LU** | **HE** | **LD** | **SL** | **NO** | **LG** | **TR** | **SO** | **SU** | **CO** |
| | **100** | **101** | **102** | **103** | **104** | **105** | **106** | **107** | **108** | **109** | **110** |
| C11 | | | | | | | | | | | |
| | **LC** | **LU** | **HE** | **LD** | **SL** | **NO** | **LG** | **TR** | **SO** | **SU** | **CO** |
| | **111** | **112** | **113** | **114** | **115** | **116** | **117** | **118** | **119** | **120** | **121** |
| C12 | | | | | | | | | | | |
| | **LC** | **LU** | **HE** | **LD** | **SL** | **NO** | **LG** | **TR** | **SO** | **SU** | **CO** |
| | **122** | **123** | **124** | **125** | **126** | **127** | **128** | **129** | **130** | **131** | **132** |
| C13 | | | | | | | | | | | |
| | **LC** | **LU** | **HE** | **LD** | **SL** | **NO** | **LG** | **TR** | **SO** | **SU** | **CO** |
| | **133** | **134** | **135** | **136** | **137** | **138** | **139** | **140** | **141** | **142** | **143** |
| C14 | | | | | | | | | | | |
| | **LC** | **LU** | **HE** | **LD** | **SL** | **NO** | **LG** | **TR** | **SO** | **SU** | **CO** |
| | **144** | **145** | **146** | **147** | **148** | **149** | **150** | **151** | **152** | **153** | **154** |
| C15 | | | | | | | | | | | |
| | **LC** | **LU** | **HE** | **LD** | **SL** | **NO** | **LG** | **TR** | **SO** | **SU** | **CO** |
| | **155** | **156** | **157** | **158** | **159** | **160** | **161** | **162** | **163** | **164** | **165** |

- **2.3 Base de données ORL**

400 images monochromes : 40 visages d'hommes ou de femmes avec 10 variantes chacun.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| C1 | | | | | | | | | | |
| | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | **7.** | **8.** | **9.** | **10.** |
| C2 | | | | | | | | | | |
| | **11.** | **12.** | **13.** | **14.** | **15.** | **16.** | **17.** | **18.** | **19.** | **20.** |
| C3 | | | | | | | | | | |
| | **21.** | **22.** | **23.** | **24.** | **25.** | **26.** | **27.** | **28.** | **29.** | **30.** |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| C4 |  | | | | | | | | | |
| | **31.** | **32.** | **33.** | **34.** | **35.** | **36.** | **37.** | **38.** | **39.** | **40.** |
| C5 |  | | | | | | | | | |
| | **41.** | **42.** | **43.** | **44.** | **45.** | **46.** | **47.** | **48.** | **49.** | **50.** |
| C6 |  | | | | | | | | | |
| | **51.** | **52.** | **53.** | **54.** | **55.** | **56.** | **57.** | **58.** | **59.** | **60.** |
| C7 |  | | | | | | | | | |
| | **61.** | **62.** | **63.** | **64.** | **65.** | **66.** | **67.** | **68.** | **69.** | **70.** |
| C8 |  | | | | | | | | | |
| | **71.** | **72.** | **73.** | **74.** | **75.** | **76.** | **77.** | **78.** | **79.** | **80.** |
| C9 |  | | | | | | | | | |
| | **81.** | **82.** | **83.** | **84.** | **85.** | **86.** | **87.** | **88.** | **89.** | **90.** |
| C10 |  | | | | | | | | | |
| | **91.** | **92.** | **93.** | **94.** | **95.** | **96.** | **97.** | **98.** | **99.** | **100.** |
| C11 |  | | | | | | | | | |
| | **101.** | **102.** | **103.** | **104.** | **105.** | **106.** | **107.** | **108.** | **109.** | **110.** |
| C12 |  | | | | | | | | | |
| | **111.** | **112.** | **113.** | **114.** | **115.** | **116.** | **117.** | **118.** | **119.** | **120.** |
| C13 |  | | | | | | | | | |
| | **121.** | **122.** | **123.** | **124.** | **125.** | **126.** | **127.** | **128.** | **129.** | **130.** |
| C14 |  | | | | | | | | | |
| | **131.** | **132.** | **133.** | **134.** | **135.** | **136.** | **137.** | **138.** | **139.** | **140.** |
| C15 |  | | | | | | | | | |
| | **141.** | **142.** | **143.** | **144.** | **145.** | **146.** | **147.** | **148.** | **149.** | **150.** |
| C16 |  | | | | | | | | | |
| | **151.** | **152.** | **153.** | **154.** | **155.** | **156.** | **157.** | **158.** | **159.** | **160.** |
| C17 |  | | | | | | | | | |
| | **161.** | **162.** | **163.** | **164.** | **165.** | **166.** | **167.** | **168.** | **169.** | **170.** |
| C18 |  | | | | | | | | | |
| | **171.** | **172.** | **173.** | **174.** | **175.** | **176.** | **177.** | **178.** | **179.** | **180.** |
| C19 |  | | | | | | | | | |
| | **181.** | **182.** | **183.** | **184.** | **185.** | **186.** | **187.** | **188.** | **189.** | **190.** |
| C20 |  | | | | | | | | | |
| | **191.** | **192.** | **193.** | **194.** | **195.** | **196.** | **197.** | **198.** | **199.** | **200.** |

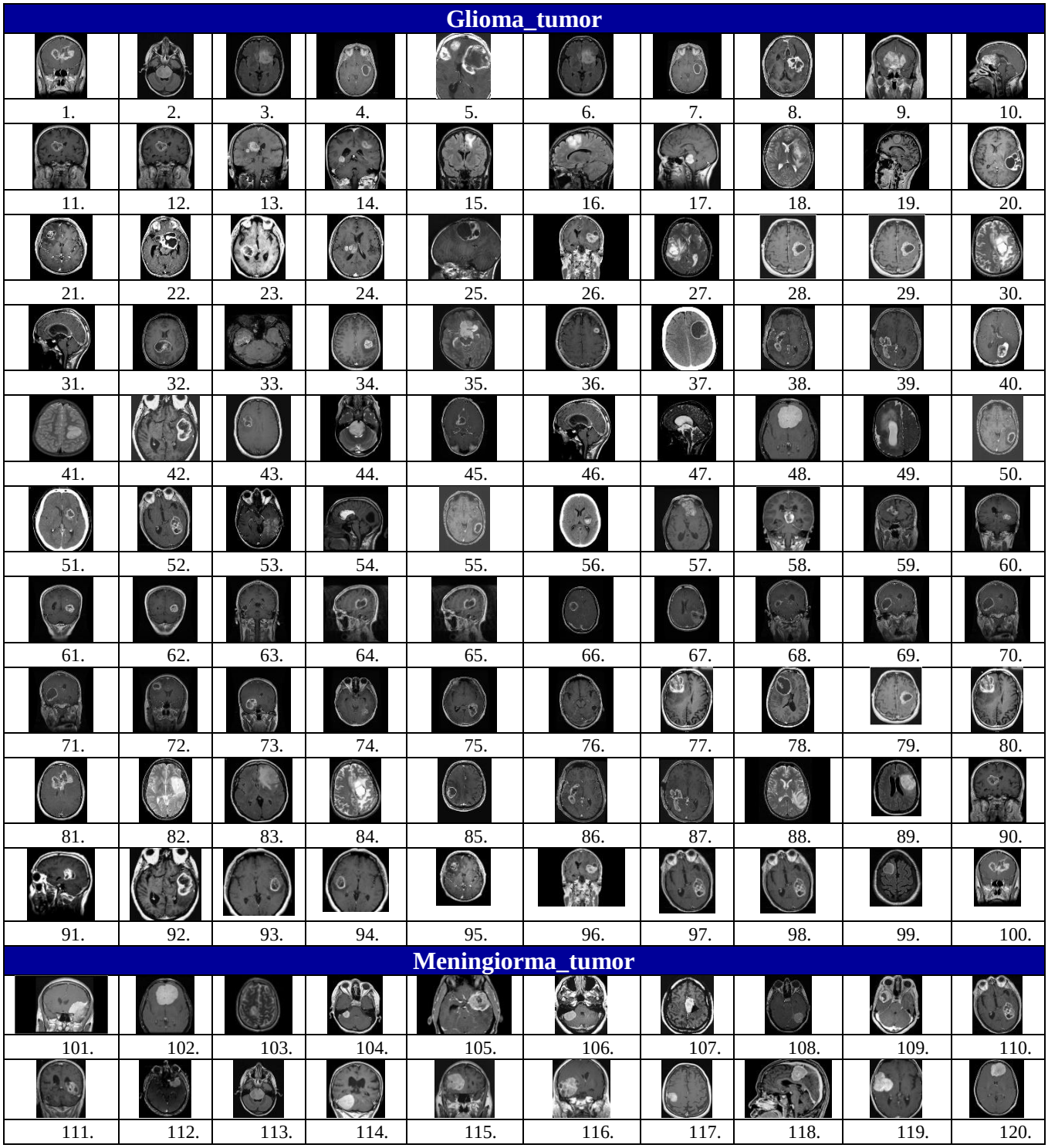
| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| C21 |  | | | | | | | | | |
| | **201.** | **202.** | **203.** | **204.** | **205.** | **206.** | **207.** | **208.** | **209.** | **210.** |
| C22 |  | | | | | | | | | |
| | **211.** | **212.** | **213.** | **214.** | **215.** | **216.** | **217.** | **218.** | **219.** | **220.** |
| C23 |  | | | | | | | | | |
| | **221.** | **222.** | **223.** | **224.** | **225.** | **226.** | **227.** | **228.** | **229.** | **230.** |
| C24 |  | | | | | | | | | |
| | **231.** | **232.** | **233.** | **234.** | **235.** | **236.** | **237.** | **238.** | **239.** | **240.** |
| C25 |  | | | | | | | | | |
| | **241.** | **242.** | **243.** | **244.** | **245.** | **246.** | **247.** | **248.** | **249.** | **250.** |
| C26 |  | | | | | | | | | |
| | **251.** | **252.** | **253.** | **254.** | **255.** | **256.** | **257.** | **258.** | **259.** | **260.** |
| C27 |  | | | | | | | | | |
| | **261.** | **262.** | **263.** | **264.** | **265.** | **266.** | **267.** | **268.** | **269.** | **270.** |
| C28 |  | | | | | | | | | |
| | **271.** | **272.** | **273.** | **274.** | **275.** | **276.** | **277.** | **278.** | **279.** | **280.** |
| C29 |  | | | | | | | | | |
| | **281.** | **282.** | **283.** | **284.** | **285.** | **286.** | **287.** | **288.** | **289.** | **290.** |
| C30 |  | | | | | | | | | |
| | **291.** | **292.** | **293.** | **294.** | **295.** | **296.** | **297.** | **298.** | **299.** | **300.** |
| C31 |  | | | | | | | | | |
| | **301.** | **302.** | **303.** | **304.** | **305.** | **306.** | **307.** | **308.** | **309.** | **310.** |
| C32 |  | | | | | | | | | |
| | **311.** | **312.** | **313.** | **314.** | **315.** | **316.** | **317.** | **318.** | **319.** | **320.** |
| C33 |  | | | | | | | | | |
| | **321.** | **322.** | **323.** | **324.** | **325.** | **326.** | **327.** | **328.** | **329.** | **330.** |
| C34 |  | | | | | | | | | |
| | **331.** | **332.** | **333.** | **334.** | **335.** | **336.** | **337.** | **338.** | **339.** | **340.** |
| C35 |  | | | | | | | | | |
| | **341.** | **342.** | **343.** | **344.** | **345.** | **346.** | **347.** | **348.** | **349.** | **350.** |
| C36 |  | | | | | | | | | |
| | **351.** | **352.** | **353.** | **354.** | **355.** | **356.** | **357.** | **358.** | **359.** | **360.** |
| C37 |  | | | | | | | | | |
| | **361.** | **362.** | **363.** | **364.** | **365.** | **366.** | **367.** | **368.** | **369.** | **370.** |

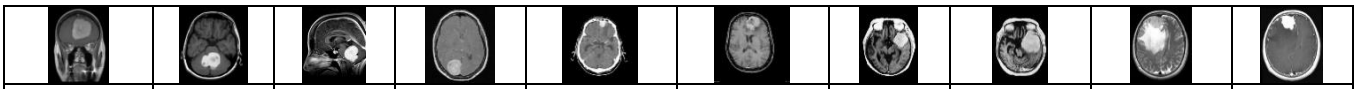
| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| C38 |  |  |  |  |  |  |  |  |  |  |
| | **371.** | **372.** | **373.** | **374.** | **375.** | **376.** | **377.** | **378.** | **379.** | **380.** |
| C39 |  |  |  |  |  |  |  |  |  |  |
| | **381.** | **382.** | **383.** | **384.** | **385.** | **386.** | **387.** | **388.** | **389.** | **390.** |
| C40 |  |  |  |  |  |  |  |  |  |  |
| | **391.** | **392.** | **393.** | **394.** | **395.** | **396.** | **397.** | **398.** | **399.** | **400.** |

# Annexe 3 Bases de données médicales

## 3.1 Base de données IRM du cerveau

394 images répertoriées en 4 classes : Glioma_tumor, Meningiorma_tumor, No_tumor et Pituitary_tumor.

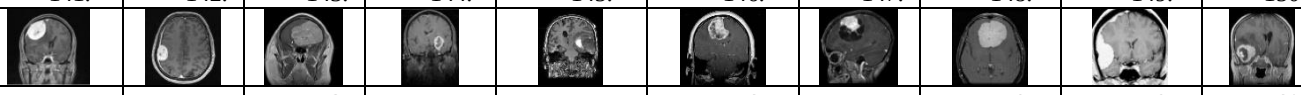




| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 121. | 122. | 123. | 124. | 125. | 126. | 127. | 128. | 129. | 130. |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 131. | 132. | 133. | 134. | 135. | 136. | 137. | 138. | 139. | 140. |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 141. | 142. | 143. | 144. | 145. | 146. | 147. | 148. | 149. | 150. |



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 151. | 152. | 153. | 154. | 155. | 156. | 157. | 158. | 159. | 160. |

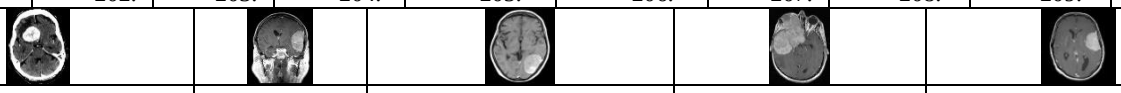
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 161. | 162. | 163. | 164. | 165. | 166. | 167. | 168. | 169. | 170. |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 171. | 172. | 173. | 174. | 175. | 176. | 177. | 178. | 179. | 180. |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 181. | 182. | 183. | 184. | 185. | 186. | 187. | 188. | 189. | 190. |

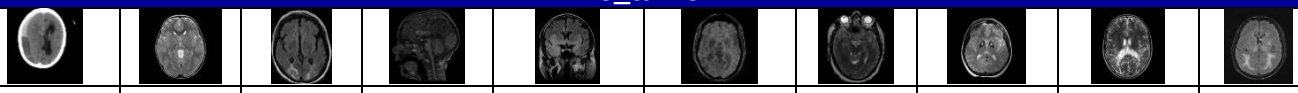
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 191. | 192. | 193. | 194. | 195. | 196. | 197. | 198. | 199. | 200. |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 201. | 202. | 203. | 204. | 205. | 206. | 207. | 208. | 209. | 210. |



| | | | | |
|---|---|---|---|---|
| 211. | 212. | 213. | 214. | 215. |

## No_tumor

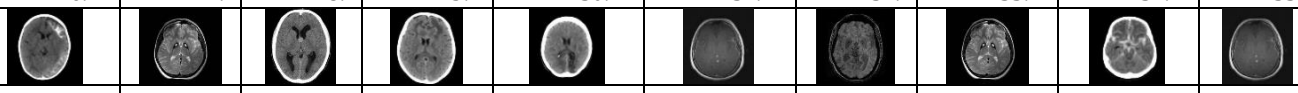


| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 216. | 217. | 218. | 219. | 220. | 221. | 222. | 223. | 224. | 225. |

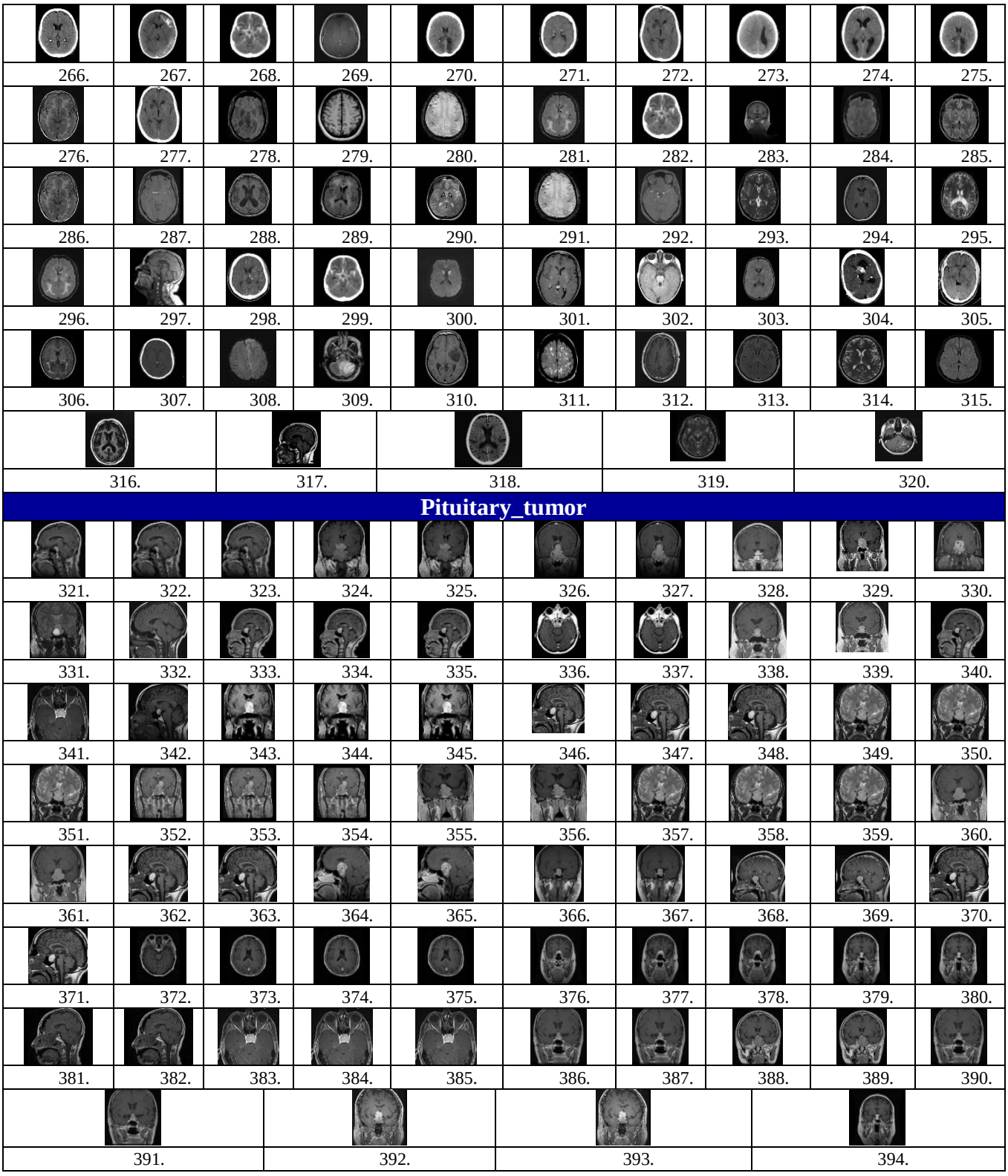
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 226. | 227. | 228. | 229. | 230. | 231. | 232. | 233. | 234. | 235. |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 236. | 237. | 238. | 239. | 240. | 241. | 242. | 243. | 244. | 245. |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 246. | 247. | 248. | 249. | 250. | 251. | 252. | 253. | 254. | 255. |

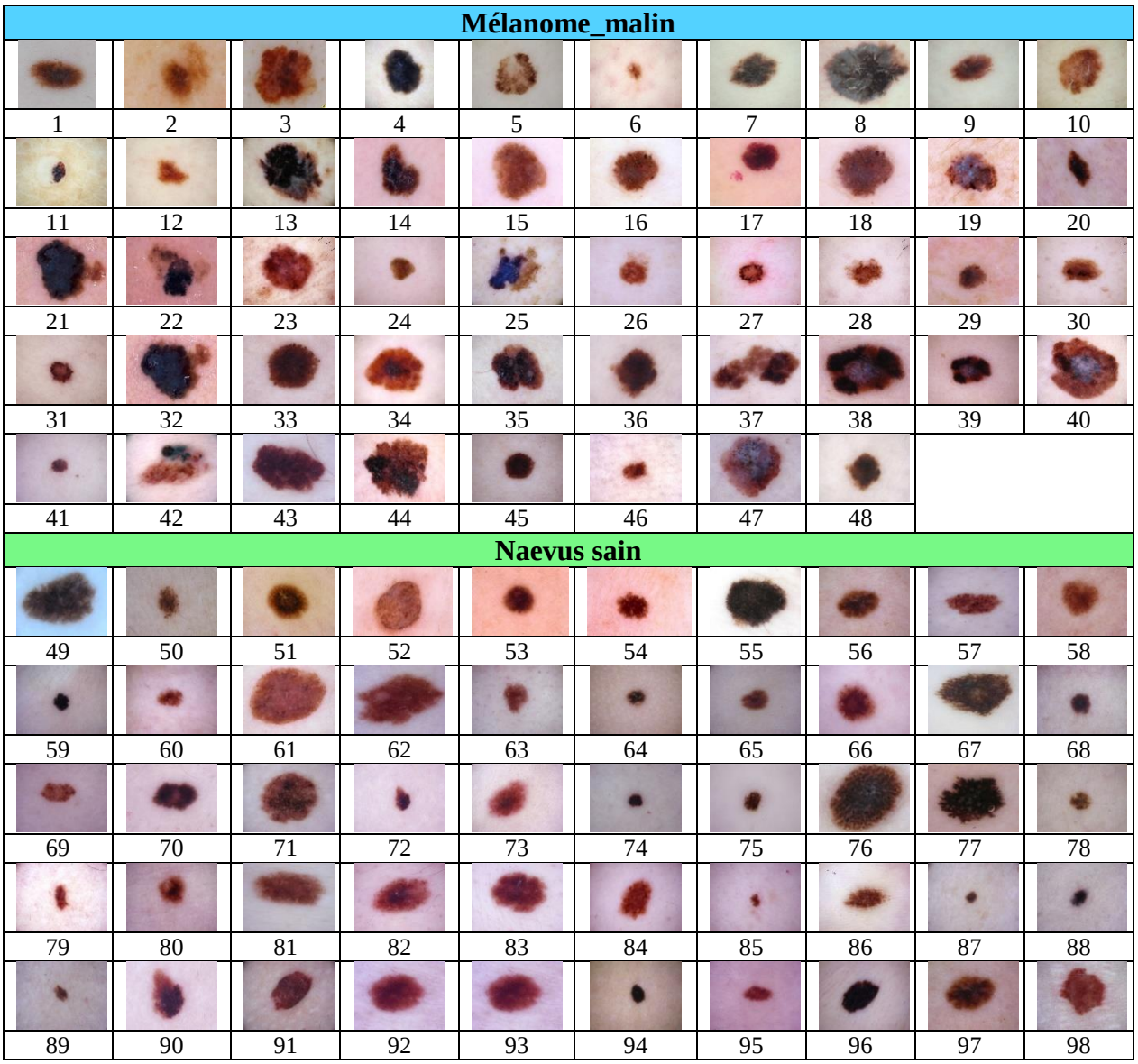


| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 256. | 257. | 258. | 259. | 260. | 261. | 262. | 263. | 264. | 265. |


266. 267. 268. 269. 270. 271. 272. 273. 274. 275.
276. 277. 278. 279. 280. 281. 282. 283. 284. 285.
286. 287. 288. 289. 290. 291. 292. 293. 294. 295.
296. 297. 298. 299. 300. 301. 302. 303. 304. 305.
306. 307. 308. 309. 310. 311. 312. 313. 314. 315.
316. 317. 318. 319. 320.
Pituitary_tumor
321. 322. 323. 324. 325. 326. 327. 328. 329. 330.
331. 332. 333. 334. 335. 336. 337. 338. 339. 340.
341. 342. 343. 344. 345. 346. 347. 348. 349. 350.
351. 352. 353. 354. 355. 356. 357. 358. 359. 360.
361. 362. 363. 364. 365. 366. 367. 368. 369. 370.
371. 372. 373. 374. 375. 376. 377. 378. 379. 380.
381. 382. 383. 384. 385. 386. 387. 388. 389. 390.
391. 392. 393. 394.

- **3.2 Base de données de mélanomes**

98 images RVB répertoriées en 2 classes : mélanome malin et naevus sain.



# Références


[1] M. HALKIDI, Y. Batistakis, and M. Vazirgiannis, 'On Clustering Validation Techniques', *Journal of Intelligent Information Systems*, vol. 17, pp. 107–145, Dec. 2001.

[2] E. Dimitriadou, S. Dolničar, and A. Weingessel, 'An examination of indexes for determining the number of clusters in binary data sets', *Psychometrika*, vol. 67, no. 1, pp. 137–159, Mar. 2002.

[3] M. D. Levine and A. M. Nazif, 'Dynamic Measurement of Computer Generated Image Segmentations', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 2, pp. 155–164, Mar. 1985.

[4] L. Lebart, A. Morineau, and J-P. Fénelon, 'Traitement des données statistiques', 2000.

[5] N. H. Timm, 'Applied Multivariate Analysis', New York: Springer-Verlag, 2002.

[6] T. Caliński and J. Harabasz, 'A dendrite method for cluster analysis', *Communications in Statistics*, vol. 3, no. 1, pp. 1–27, Jan. 1974.

[7] S. Sharma, 'Applied Multivariate Techniques', *Technometrics*, vol. 39, no. 1, pp. 100–101, Feb. 1997.

[8] J. C. Dunn, 'Well-Separated Clusters and Optimal Fuzzy Partitions', *Journal of Cybernetics*, vol. 4, no. 1, pp. 95–104, Jan. 1974.

[9] D. L. Davies and D. W. Bouldin, 'A Cluster Separation Measure', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[10] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, 'Understanding of Internal Clustering Validation Measures', in *2010 IEEE International Conference on Data Mining*, Dec. 2010, pp. 911–916.

[11] M. Halkidi, M. Vazirgiannis, and Y. Batistakis, 'Quality Scheme Assessment in the Clustering Process', in *Principles of Data Mining and Knowledge Discovery*, Sep. 2000, pp. 265–276.

[12] P. J. Rousseeuw, 'Silhouettes: A graphical aid to the interpretation and validation of cluster analysis', *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov. 1987.

[13] X.L. Xie and G. Beni, 'A validity measure for fuzzy clustering', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, Aug. 1991.

[14] F. BLANCHARD, M. HERBIN, and P. VAUTROT, 'Vers une classification non supervisée basée sur un nouvel indice de connectivité', *20ème colloque GRETSI sur le traitement du signal et des images*, pp. 30–31, 2005.

[15] D. J. Hand, 'Assessing the Performance of Classification Methods', *International Statistical Review*, vol. 80, no. 3, pp. 400–414, 2012.

[16] G. Forestier, C. Wemmert, and P. Gançarski, 'Background Knowledge Integration in Clustering Using Purity Indexes', in *Knowledge Science, Engineering and Management*, vol. 6291, Y. Bi and M.-A. Williams, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 28–38, 2010.

[17] P. Clark and T. Niblett, 'The CN2 induction algorithm', *Mach Learn*, vol. 3, no. 4, pp. 261–283, Mar. 1989.

[18] A. Amelio and C. Pizzuti, 'Is Normalized Mutual Information a Fair Measure for Comparing Community Detection Methods?', *in Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Paris, 2015.

[19] M. Soltani, 'Partitionnement des images hyperspectrales de grande dimension spatiale par propagation d'affinité', *Thèse de doctorat de l'université de Rennes 1*, 2014.

[20] C. E. Guerra, N. S. da Silva Caldas, and A. J. N. Andrade, 'Lithology mapping by a hybridization of the firefly and affinity propagation algorithms', *Journal of Petroleum Science and Engineering*, vol. 158, pp. 222–233, Sep. 2017.

[21] J. Thomas, J. Jin, J. Dauwels, S. S. Cash, and M. B. Westover, 'Clustering of interictal spikes by dynamic time warping and affinity propagation', in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 749–753.

[22] S. Latha, D. Samiappan, P. Muthu, and R. Kumar, 'Fully Automated Integrated Segmentation of Carotid Artery Ultrasound Images Using DBSCAN and Affinity Propagation', *J. Med. Biol. Eng.*, vol.41, pp. 260–271, Jan. 2021.

[23] A. Busch, T. Homeier-Bachmann, M. Y. Abdel-Glil, A. Hackbart, H. Hotzel, and H. Tomaso, 'Using affinity propagation clustering for identifying bacterial clades and subclades with whole-genome sequences of Francisella tularensis', *PLOS Neglected Tropical Diseases*, vol. 14, no. 9, p. e0008018, Sep. 2020.

[24] R. Santana, C. Bielza, and P. Larrañaga, 'Affinity propagation enhanced by estimation of distribution algorithms', in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, New York, NY, USA, Jul. 2011, pp. 331–338.

[25] H. Zhu, J. Xu, J. Hu, and J. Chen, 'Medical Image Segmentation Using Improved Affinity Propagation', in *Computational Modeling of Objects Presented in Images. Fundamentals, Methods, and Applications*, 2017, pp. 208–215.

[26] T. Ren, W. Zeng, N. Wang, L. Chen, and C. Wang, 'A novel approach for fMRI data analysis based on the combination of sparse approximation and affinity propagation clustering', *Magn Reson Imaging*, vol. 32, no. 6, pp. 736–746, Jul. 2014.
[27] Z. Geng, R. Zeng, Y. Han, Y. Zhong, and H. Fu, 'Energy efficiency evaluation and energy saving based on DEA integrated affinity propagation clustering: Case study of complex petrochemical industries', *Energy*, vol. 179, pp. 863–875, Jul. 2019.
[28] Y. Han, H. Wu, M. Jia, Z. Geng, and Y. Zhong, 'Production capacity analysis and energy optimization of complex petrochemical industries using novel extreme learning machine integrating affinity propagation', *Energy Conversion and Management*, vol. 180, pp. 240–249, Jan. 2019.
[29] S. Yang, Z. Liu, J. Li, S. Wang, and F. Yang, 'Anomaly Detection for Internet of Vehicles: A Trust Management Scheme with Affinity Propagation', *Mobile Information Systems*, vol. 2016, Mar. 2016.
[30] A. P. Dempster, N. M. Laird, and D. B. Rubin, 'Maximum Likelihood from Incomplete Data via the EM Algorithm', *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.
[31] C. Cortes and V. Vapnik, 'Support-vector networks', *Mach Learn*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[32] K. Chehdi and C. Cariou, 'The true false ground truths: What interest?', in *Image and Signal Processing for Remote Sensing XXII*, vol. 10004, Oct. 2016, pp. 213–228.
[33] K. Chehdi and C. Cariou, 'Learning or assessment of classification algorithms relying on biased ground truth data: what interest?', *Journal of applied remote sensing*, vol. 13, no. 03, pp. 1, Jul. 2019.
[34] J. MacQueen, 'Some methods for classification and analysis of multivariate observations', *Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.
[35] X. Li, X. Lu, J. Tian, P. Gao, H. Kong, and G. Xu, 'Application of fuzzy c-means clustering in data analysis of metabolomics', *Anal Chem*, vol. 81, no. 11, pp. 4468–4475, Jun. 2009.
[36] J. C. Bezdek, R. Ehrlich, and W. Full, 'FCM: The fuzzy c-means clustering algorithm', *Computers & Geosciences*, vol. 10, no. 2, pp. 191–203, Jan. 1984.
[37] Y. Linde, A. Buzo, and R. Gray, 'An Algorithm for Vector Quantizer Design', *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, Jan. 1980.
[38] Rui Xu and D. Wunsch, 'Survey of clustering algorithms', *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005.
[39] B. J. Frey and D. Dueck, 'Clustering by Passing Messages Between Data Points', *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.
[40] J. M. Peña, J. A. Lozano, and P. Larrañaga, 'An empirical comparison of four initialization methods for the K-Means algorithm', *Pattern Recognition Letters*, vol. 20, no. 10, pp. 1027–1040, Oct. 1999.
[41] S. Bubeck, M. Meilă, and U. von Luxburg, 'How the initialization affects the stability of the κ-means algorithm', *ESAIM: Probability and Statistics*, vol. 16, pp. 436–452, 2012.
[42] A. Ben Ayed, M. Ben Halima, and A. M. Alimi, 'Survey on clustering methods: Towards fuzzy clustering for big data', in *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, Aug. 2014, pp. 331–336.
[43] Bang Huang and Linbo Xie, 'An improved LBG algorithm for image vector quantization', in *2010 3rd International Conference on Computer Science and Information Technology*, vol. 6, Jul. 2010, pp. 467–471.
[44] B. Fritzke, 'The LBG-U method for vector quantization – an improvement over LBG inspired from neural networks', *Neural Processing Letters*, vol. 5, pp. 35–45, 1997.
[45] J. C. Dunn, 'A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters', *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, Jan. 1973.
[46] J. C. Bezdek, 'Pattern Recognition with Fuzzy Objective Function Algorithms', Boston, MA: Springer US, 1981.
[47] C. Hung, S. Kulkarni, and B. Kuo, 'A New Weighted Fuzzy C-Means Clustering Algorithm for Remotely Sensed Image Classification', *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 543–553, Jun. 2011.
[48] K. Chehdi, A. Taher, and C. Cariou, 'Stable and unsupervised fuzzy C-means method and its validation in the context of multicomponent images', *J. Electron. Imaging*, vol. 24, no. 6, p. 061117, Dec. 2015.
[49] C. Rosenberger and K. Chehdi, 'Unsupervised clustering method with optimal estimation of the number of clusters: application to image segmentation', in *Proceedings 15th International Conference on Pattern Recognition*. ICPR-2000, Sep. 2000, vol. 1, pp. 656–659.
[50] W. Hang, F. Chung, and S. Wang, 'Transfer Affinity Propagation-based Clustering', *Inf. Sci.*, vol. 348, no. C, pp. 337–356, Jun. 2016.
[51] S. Lerm, A. Saeedi, and E. Rahm, 'Extended Affinity Propagation Clustering for Multi-source Entity Resolution', BTW 2021, pp. 217–236, 2021.

[52] S. Park, H.-S. Jo, C. Mun, and J.-G. Yook, 'RRH Clustering Using Affinity Propagation Algorithm with Adaptive Thresholding and Greedy Merging in Cloud Radio Access Network', *Sensors (Basel)*, vol. 21, no. 2, p. 480, Jan. 2021.
[53] J. A. Cardille, J. C. White, M. A. Wulder, and T. Holland, 'Representative Landscapes in the Forested Area of Canada', *Environmental Management*, vol. 49, no. 1, pp. 163–173, Jan. 2012.
[54] P. Li, H. Ji, B. Wang, Z. Huang, and H. Li, 'Adjustable preference affinity propagation clustering', *Pattern Recognition Letters*, vol. 85, pp. 72–78, 2017.
[55] F. Shang, L. C. Jiao, J. Shi, F. Wang, and M. Gong, 'Fast affinity propagation clustering: A multilevel approach', *Pattern Recognition*, vol. 45, no. 1, pp. 474–486, Jan. 2012.
[56] D. Xia, F. Wu, X. Zhang, and Y. Zhuang, 'Local and global approaches of affinity propagation clustering for large scale data', *Journal of Zhejiang University-SCIENCE A*, vol. 9, no. 10, pp. 1373–1381, Oct. 2008.
[57] Y. Yang, H. Ha, F. C. Fleites, and S. Chen, 'A Multimedia Semantic Retrieval Mobile System Based on HCFGs', *IEEE MultiMedia*, vol. 21, no. 1, pp. 36–46, Jan. 2014.
[58] H. Xiao and P. Guo, 'Iris Image Analysis Based on Affinity Propagation Algorithm', in *Advances in Neural Networks – ISNN 2009*, May 2009, pp. 943–949.
[59] C. Wang, J. Lai, C. Y. Suen, and J. Zhu, 'Multi-Exemplar Affinity Propagation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2223–2237, Sep. 2013.
[60] Z.-J. Zha, L. Yang, T. Mei, M. Wang, and Z. Wang, 'Visual query suggestion', in *Proceedings of the seventeen ACM international conference on Multimedia - MM '09*, Beijing, China, 2009, pp. 15–24.
[61] K. Lindorff-Larsen and J. Ferkinghoff-Borg, 'Similarity Measures for Protein Ensembles', *PLOS ONE*, vol. 4, no. 1, p. e4203, Jan. 2009.
[62] G. Gan and M. K.-P. Ng, 'Subspace clustering using affinity propagation', *Pattern Recognition*, vol. 48, no. 4, pp. 1455–1464, Apr. 2015.
[63] Z. Zhu, S. Jia, and Z. Ji, 'Towards a Memetic Feature Selection Paradigm [Application Notes]', *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 41–53, May 2010.
[64] K. Guo, W. Guo, Y. Chen, Q. Qiu, and Q. Zhang, 'Community discovery by propagating local and global information based on the MapReduce model', *Information Sciences*, vol. 323, pp. 73–93, Dec. 2015.
[65] T. Zhou, M. Qi, J. Jiang, X. Wang, S. Hao, and Y. Jin, 'Person Re-identification based on nonlinear ranking with difference vectors', *Information Sciences*, vol. 279, pp. 604–614, Sep. 2014.
[66] R. Shang, S. Luo, W. Zhang, R. Stolkin, and L. Jiao, 'A multiobjective evolutionary algorithm to find community structures based on affinity propagation', *Physica A: Statistical Mechanics and its Applications*, vol. 453, pp. 203–227, Jul. 2016.
[67] S. Taheri and A. Bouyer, 'Community Detection in Social Networks Using Affinity Propagation with Adaptive Similarity Matrix', *Big Data*, vol. 8, no. 3, pp. 189–202, May 2020.
[68] X. Bi, B. Guo, L. Shi, Y. Lu, L. Feng, and Z. Lyu, 'A New Affinity Propagation Clustering Algorithm for V2V-Supported VANETs', *IEEE Access*, vol. 8, pp. 71405–71421, 2020.
[69] L. Wang *et al.*, 'An Improved Integrated Clustering Learning Strategy Based on Three-Stage Affinity Propagation Algorithm with Density Peak Optimization Theory', *Complexity*, vol. 2021, p. e6666619, Jan. 2021.
[70] A. Bandi, K. Joshi, and V. Mulwad, 'Affinity Propagation Initialisation Based Proximity Clustering For Labeling in Natural Language Based Big Data Systems', in *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, May 2020, pp. 1–7.
[71] K. Chehdi, M. Soltani, and C. Cariou, 'Pixel classification of large-size hyperspectral images by affinity propagation', *JARS*, vol. 8, no. 1, p. 083567, Aug. 2014.
[72] S. Boriah, V. Chandola, and V. Kumar, 'Similarity Measures for Categorical Data: A Comparative Evaluation', in *Proceedings of the 2008 SIAM International Conference on Data Mining*, 0 vols, Society for Industrial and Applied Mathematics, 2008, pp. 243–254.
[73] F. C. Lourenço, V. Lobo, F. Bação, and G. de Informação, 'Binary-based similarity measures for categorical data and their application in Self- Organizing Maps', *JOCLAD 2004-XI Jornadas de Classificacao e Anlise de Dados*, April 2004, pp. 1-8.
[74] R. Deshpande, B. VanderSluis, and C. L. Myers, 'Comparison of Profile Similarity Measures for Genetic Interaction Networks', *PLOS ONE*, vol. 8, no. 7, p. e68664, Jul. 2013.
[75] A. Strehl, E. Strehl, J. Ghosh, and R. Mooney, 'Impact of Similarity Measures on Web-page Clustering', in *In Workshop on Artificial Intelligence for Web Search (AAAI 2000)*, 2000, pp. 58–64.

[76] Zhang Zhang, Kaiqi Huang, and Tieniu Tan, 'Comparison of Similarity Measures for Trajectory Clustering in Outdoor Surveillance Scenes', in *18th International Conference on Pattern Recognition (ICPR'06)*, Aug. 2006, vol. 3, pp. 1135–1138.
[77] Aysha Al Khalifa, Maciej Haranczyk, and John Holliday, 'Comparison of Nonbinary Similarity Coefficients for Similarity Searching, Clustering and Compound Selection, *Journal of Chemical Information and Modeling*, vol. 49, pp. 1193–201, May 2009.
[78] L. Pujo-Menjouet, 'Cours d'Analyse 3 Fonctions de plusieurs variables', Cours Université Claude Bernard, Lyon I.
[79] J. Irani, N. Pise, and M. Phatak, 'Clustering Techniques and the Similarity Measures used in Clustering: A Survey', *International Journal of Computer Applications*, vol. 134, no. 7, pp. 9–14, Jan. 2016.
[80] Y. Sohn and N. S. Rebello, 'Supervised and Unsupervised Spectral Angle Classifiers', *Photogrammetric Engineering & Remote Sensing*, vol. 68, no. 12, pp. 1271–1280, Dec. 2002.
[81] Chein-I Chang, 'Spectral information divergence for hyperspectral image analysis', in *IEEE 1999 International Geoscience and Remote Sensing Symposium. IGARSS'99 (Cat. No.99CH36293)*, vol. 1, Jun. 1999, pp. 509–511.
[82] J. Soler, F. Tence, L. Gaubert, and C. Buche, 'Data Clustering and Similarity', in *26th International Florida Artificial Intelligence Research Society Conference*, May 2013.
[83] A. S. Shirkhorshidi, S. Aghabozorgi, and T. Y. Wah, 'A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data', *PLoS One*, vol. 10, no. 12, p. e0144059, Dec. 2015.
[84] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*. ASA-SIAM Series on Statistics and Applied Mathematics, 2007.
[85] A. K. Jain, M. N. Murty, and P. J. Flynn, 'Data clustering: a review', *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.
[86] T. Strauss and M. J. von Maltitz, 'Generalising Ward's Method for Use with Manhattan Distances', *PLOS ONE*, vol. 12, no. 1, p. e0168288, Jan. 2017.
[87] K. M. Ponnmoli, 'Analysis of Face Recognition using Manhattan Distance Algorithm with Image Segmentation', *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 7, pp. 18–27, July 2014.
[88] G. Khosla, N. Rajpal, and J. Singh, 'Evaluation of Euclidean and Manhanttan metrics in Content Based Image Retrieval system', in *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, Mar. 2015, pp. 12–18.
[89] Y. Qian, F. Yao, and S. Jia, 'Band selection for hyperspectral imagery using affinity propagation', *IET Computer Vision*, vol. 3, no. 4, pp. 213–222, Dec. 2009.
[90] C. Yang, S. Liu, L. Bruzzone, R. Guan, and P. Du, 'A Feature-Metric-Based Affinity Propagation Technique for Feature Selection in Hyperspectral Image Classification', *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 5, pp. 1152–1156, Sep. 2013.
[91] J. Yu and C. Jia, 'Convergence Analysis of Affinity Propagation', in *Knowledge Science, Engineering and Management*, Berlin, Heidelberg, 2009, pp. 54–65.
[92] D. W. Aha, D. Kibler, and M. K. Albert, 'Instance-based learning algorithms', *Mach Learn*, vol. 6, no. 1, pp. 37–66, Jan. 1991.
[93] C. M. Bishop, 'Neural Networks for Pattern Recognition', Birmingham, UK, 1995.
[94] A. Krizhevsky, I. Sutskever, and G. E. Hinton, 'ImageNet classification with deep convolutional neural networks', *Commun. ACM*, vol. 60, no. 6, pp. 84–90, June 2012.
[95] C. L. Giles, G. M. Kuhn, and R. J. Williams, 'Dynamic recurrent neural networks: Theory and applications', *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 153–156, Mar. 1994.
[96] M. Shi, T. Zhang, and Y. Zeng, 'A Curiosity-Based Learning Method for Spiking Neural Networks', *Frontiers in Computational Neuroscience*, vol. 14, 2020.
[97] N. M. Nawi, A. Khan, M. Z. Rehman, H. Chiroma, and T. Herawan, 'Weight Optimization in Recurrent Neural Networks with Hybrid Metaheuristic Cuckoo Search Techniques for Data Classification', *Mathematical Problems in Engineering*, vol. 2015, p. e868375, Oct. 2015.
[98] B. Guan *et al.*, 'Automatic detection and localization of thighbone fractures in X-ray based on improved deep learning method', *Computer Vision and Image Understanding*, vol. 216, p. 103345, Jan. 2022.
[99] A. Mohammed, I. Farup, M. Pedersen, S. Y. YILDIRIM, and Ø. Hovde, 'PS-DeVCEM: Pathology-sensitive deep learning model for video capsule endoscopy based on weakly labeled data', *Comput. Vis. Image Underst.*, vol. 201, p. 103062, 2020.

[100] R. El Jurdi, C. Petitjean, P. Honeine, V. Cheplygina, and F. Abdallah, 'High-level prior-based loss functions for medical image segmentation: A survey', *Computer Vision and Image Understanding*, vol. 210, p. 103248, Sep. 2021.
[101] S. Brossette, A. Sprague, J. Hardin, K. Waites, W. Jones, and S. Moser, 'Association Rules and Data Mining in Hospital Infection Control and Public Health Surveillance', *Journal of the American Medical Informatics Association : JAMIA*, vol. 5, pp. 373–81, Jul. 1998.
[102] M. Q. Hatem, 'Skin lesion classification system using a K-nearest neighbor algorithm', *Visual Computing for Industry, Biomedicine, and Art*, vol. 5, no. 1, pp. 1–10, Mar. 2022.
[103] J. Yao, A. Dwyer, R. M. Summers, and D. J. Mollura, 'Computer-aided diagnosis of pulmonary infections using texture analysis and support vector machine classification', *Acad Radiol*, vol. 18, no. 3, pp. 306–314, Mar. 2011.
[104] S. Banerjee, R. G. VidalMata, Z. Wang, and W. J. Scheirer, 'Report on UG2+ challenge Track 1: Assessing algorithms to improve video object detection and classification from unconstrained mobility platforms', *Computer Vision and Image Understanding*, vol. 213, p. 103297, Dec. 2021.
[105] R. G. VidalMata *et al.*, 'Bridging the Gap Between Computational Photography and Visual Recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4272–4290, Dec. 2021.
[106] L. Oliva-Teles, R. Pinto, R. Vilarinho, A. P. Carvalho, J. A. Moreira, and L. Guimarães, 'Environmental diagnosis with Raman Spectroscopy applied to diatoms', *Biosensors and Bioelectronics*, vol. 198, p. 113800, Feb. 2022.
[107] H. Kalesse-Los, W. Schimmel, E. Luke, and P. Seifert, 'Evaluating cloud liquid detection against Cloudnet using cloud radar Doppler spectra in a pre-trained artificial neural network', *Atmospheric Measurement Techniques*, vol. 15, no. 2, pp. 279–295, Jan. 2022.
[108] J. Forough and S. Momtazi, 'Ensemble of deep sequential models for credit card fraud detection', *Applied Soft Computing*, vol. 99, p. 106883, Feb. 2021.
[109] B. Stojanović *et al.*, 'Follow the Trail: Machine Learning for Fraud Detection in Fintech Applications', *Sensors*, vol. 21, no. 5, p. 1594, Jan. 2021.
[110] K. Wang, L. Cheng, and B. Yong, 'Spectral-Similarity-Based Kernel of SVM for Hyperspectral Image Classification', *Remote Sensing*, vol. 12, p. 2154, Jul. 2020.
[111] Y. Ding, X. Zhao, Z. Zhang, W. Cai, and N. Yang, 'Multiscale Graph Sample and Aggregate Network With Context-Aware Learning for Hyperspectral Image Classification', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4561–4572, 2021.
[112] J. Alameddine, K. Chehdi, and C. Cariou, 'Hierarchical Unsupervised Partitioning of Large Size Data and Its Application to Hyperspectral Images', *Remote Sensing*, vol. 13, no. 23, p. 4874, Jan. 2021.
[113] C. J. C. Burges, 'A Tutorial on Support Vector Machines for Pattern Recognition', *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, Jun. 1998.
[114] J. Han, J. Pei, and M. Kamber, 'Data Mining: Concepts and Techniques', 3rd ed, University of Illinois: Urbana-Champaign, 2011.
[115] B. Sun *et al.*, 'SuperTML: Two-Dimensional Word Embedding for the Precognition on Structured Tabular Data', in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2019, pp. 2973–2981.
[116] R. A. Fisher, 'The Use of Multiple Measurements in Taxonomic Problems', *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
[117] F. Hu, Y. Zhu, J. Liu, and L. Li, 'An efficient Long Short-Term Memory model based on Laplacian Eigenmap in artificial neural networks', *Applied Soft Computing*, vol. 91, p. 106218, Jun. 2020.
[118] M. Vasic, C. T. Chalk, S. Khurshid, and D. Soloveichik, 'Deep Molecular Programming: A Natural Implementation of Binary-Weight ReLU Neural Networks', in *Proceedings of the 37th International Conference on Machine Learning*, 2020, pp. 9701–9711.
[119] T. Chen and C. Guestrin, 'XGBoost: A Scalable Tree Boosting System', in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA, Aug. 2016, pp. 785–794.
[120] E. Hartuv and R. Shamir, 'A Clustering Algorithm Based on Graph Connectivity', *Information Processing Letters*, vol. 76, pp. 175–181, Dec. 2000.
[121] Y. Dong, Y. Zhuang, K. Chen, and X. Tai, 'A hierarchical clustering algorithm based on fuzzy graph connectedness', *Fuzzy Sets and Systems*, vol. 157, no. 13, pp. 1760–1774, Jul. 2006.
[122] Zhang Huijuan and Sun Shixuan, 'A Graph Clustering algorithm based on shared neighbors and connectivity', in *2013 8th International Conference on Computer Science Education*, Apr. 2013, pp. 761–764.

[123] H. Chen, Z. Yu, Q. Yang, and J. Shao, 'Attributed graph clustering with subspace stochastic block model', *Information Sciences*, vol. 535, pp. 130–141, Oct. 2020.
[124] X. Huang, H. Cheng, and J. X. Yu, 'Dense community detection in multi-valued attributed networks', *Information Sciences*, vol. 314, pp. 77–99, Sep. 2015.
[125] G. Bello-Orgaz, S. Salcedo-Sanz, and D. Camacho, 'A Multi-Objective Genetic Algorithm for overlapping community detection based on edge encoding', *Information Sciences*, vol. 462, pp. 290–314, Sep. 2018.
[126] C. Zhong, D. Miao, and P. Fränti, 'Minimum spanning tree based split-and-merge: A hierarchical clustering method', *Information Sciences*, vol. 181, no. 16, pp. 3397–3410, Aug. 2011.
[127] C. Zhao, W.-L. Hwang, C.-L. Lin, and W. Chen, 'Greedy orthogonal matching pursuit for subspace clustering to improve graph connectivity', *Information Sciences*, vol. 459, pp. 135–148, Aug. 2018.
[128] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, 'A density-based algorithm for discovering clusters in large spatial databases with noise', in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, Aug. 1996, pp. 226–231.
[129] A. Hinneburg and D. A. Keim, 'An Efficient Approach to Clustering in Large Multimedia Databases with Noise', 1998, pp. 58–65.
[130] A. Bryant and K. Cios, 'RNN-DBSCAN: A Density-Based Clustering Algorithm Using Reverse Nearest Neighbor Density Estimates', *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1109–1121, Jun. 2018.
[131] Qixiang Ye, Wen Gao, and Wei Zeng, 'Color image segmentation using density-based clustering', in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Apr. 2003, vol. 3, p. III–345.
[132] K. Mahesh Kumar and A. Rama Mohan Reddy, 'A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method', *Pattern Recognition*, vol. 58, pp. 39–48, Oct. 2016.
[133] S. F. Galán, 'Comparative evaluation of region query strategies for DBSCAN clustering', *Information Sciences*, vol. 502, pp. 76–90, Oct. 2019.
[134] Y. Geng, Q. Li, R. Zheng, F. Zhuang, R. He, and N. Xiong, 'RECOME: A new density-based clustering algorithm using relative KNN kernel density', *Information Sciences*, vol. 436–437, pp. 13–30, Apr. 2018.
[135] Y.-F. Li, L.-H. Lu, and Y.-C. Hung, 'A New Clustering Algorithm Based on Graph Connectivity', in *Intelligent Computing*, Cham, pp. 442–454, 2019.
[136] S. Dodel, J. M. Herrmann, and T. Geisel, 'Functional connectivity by cross-correlation clustering', *Neurocomputing*, vol. 44–46, pp. 1065–1070, Jun. 2002.
[137] J.-S. Lee and S. Olafsson, 'A meta-learning approach for determining the number of clusters with consideration of nearest neighbors', *Information Sciences*, vol. 232, pp. 208–224, May 2013.
[138] S. Soor, A. Challa, S. Danda, B. S. D. Sagar, and L. Najman, 'Extending K-Means to Preserve Spatial Connectivity', *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018, pp. 6959–6962.
[139] X. Huang and W. Lai, 'Clustering graphs for visualization via node similarities', *Journal of Visual Languages & Computing*, pp. 225–253, Jun. 2006.
[140] D. Dueck and B. J. Frey, 'Non-metric affinity propagation for unsupervised image categorization', in *2007 IEEE 11th International Conference on Computer Vision*, Oct. 2007, pp. 1–8.
[141] K. Mikolajczyk, B. Leibe, and B. Schiele, 'Multiple Object Class Detection with a Generative Model', in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*, New York, NY, USA, 2006, pp. 26–36.
[142] D. G. Lowe, 'Distinctive Image Features from Scale-Invariant Keypoints', *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[143] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, 'Coding facial expressions with Gabor wavelets', in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, Apr. 1998, pp. 200–205.
[144] 'Yale Face Database | vision.ucsd.edu'. http://vision.ucsd.edu/content/yale-face-database.
[145] 'The Database of Faces'. https://cam-orl.co.uk/facedatabase.html.
[146] 'ISIC Archive'. https://www.isic-archive.com/#!/topWithHeader/wideContentTop/main.

# Liste des Figures

**Figure 1.** Ensemble de données ($J_1$) caractérisées par deux attributs sans la présence des individus identiques et leur représentation par rapport aux attributs $\alpha 1$ et $\alpha 2$..... 45
**Figure 2.** Image originale, image test construite et image VT..... 46
**Figure 3.** Partitionnement de l'image hyperspectrale de la Figure 2 par l'AP (λ = 0.9). (a) : avec tous les pixels de l'image ; (b) : avec les pixels non dupliqués et l'ensemble des exemplaires des pixels dupliqués..... 48
**Figure 4**. Individus caractérisés par deux attributs $a1$ et $a2$ ($\varepsilon ijk > 1$). ..... 59
**Figure 5**. Individus caractérisés par deux attributs $a1$ et $a2$ ($\varepsilon ijk \in ]0,1[$). ..... 61
**Figure 6.** Image hyperspectrale synthétique (bandes visualisés : 5, 15 et 25) et images VT..... 62
**Figure 7.** Signatures spectrales moyennes ± écart-type des 9 classes de la VT de l'image hyperspectrale synthétique de la Figure 6. ..... 63
**Figure 8.** Résultats de partitionnement de l'image hyperspectrale synthétique de la Figure 6 par l'AP avec $pmed$ en fonction du choix de l'indice de similarité. ..... 65
**Figure 9.** Résultats de partitionnement de l'image hyperspectrale synthétique de la Figure 6 par l'AP avec $pmin$ en fonction du choix de l'indice de similarité. ..... 65
**Figure 10**. CCR de l'AP partitionnant l'image hyperspectrale synthétique de la Figure 6, en fonction des indices de similarité. ..... 66
**Figure 11.** Résultats du partitionnement de l'image hyperspectrale synthétique de la Figure 6 par des méthodes non supervisées et semi-supervisées. ..... 83
**Figure 12**. Image hyperspectrale de taille 630 × 1800 pixels (100 bandes) affichée en mode RVB et points de la VT. ..... 86
**Figure 13.** Signatures spectrales des points de la VT par classe de l'image de la Figure 12. Ligne 1 : (a) Algues vertes (Ulva armoricana) ; (b) Algues brunes (Fucus serratus) ; Substrat ((c) Roches+Galets et (d) Sable) et Ligne 2 : signature spectrale moyenne correspondante ± écart-type de chaque classe. ..... 86
**Figure 14.** Résultat du partitionnement optimal obtenu par la méthode HUP-OAP appliquée à l'image hyperspectrale de la Figure 12 (5 classes localisées, *LN* = 0.17 au niveau 4). ..... 89
**Figure 15**. Signature spectrale moyenne ± écart-type de chaque classe de la Figure 14 obtenue par HUP-OAP sur l'image hyperspctrale réelle de la Figure 12 : (a) Algues vertes ; (b) Algues brunes ; (c) Roches+Galets ; (d) Sable ; (e) Eau. ..... 89
**Figure 16.** Transformation de l'image originale de la Figure 6 : étape par étape..... 110
**Figure 17.** Résultats de partitionnement par HUP-OAPM-RSM de l'image hyperspectrale synthétique de la Figure 6 et de ses deux images transformées utilisant 4 et 8 blocs..... 111
**Figure 18.** Image hyperspectrale transformée 630 × 1800 pixels (bandes visualisées : 5, 27, 48) et son résultat de partitionnement optimal obtenu par la méthode HUP-OAPM-RSM (4 classes, *LN* = 0.18 au niveau 7). ..... 114
**Figure 19.** Signature spectrale moyenne ± écart-type de chaque classe obtenue par HAUP-OAP-RSM sur l'image hyperspectrale réelle de la Figure 18 (partition optimale : niveau 7, 4 classes). ..... 115
**Figure 20.** Organigramme de la méthode non supervisée et autonome de validation ou de sélection des échantillons d'apprentissage. ..... 123
**Figure 21.** Les 3 espèces de fleurs IRIS. ..... 127
**Figure 22.** CCR du SVM en faisant varier ses trois paramètres d'entrée pour chaque ensemble d'apprentissage des données IRIS : SH-TS, MSH-TS, CGT-TS et OGT-TS. ..... 130
**Figure 23**. CCR obtenus par KNN en faisant varier le paramètre *K* pour chaque ensemble d'apprentissage des données IRIS..... 131
**Figure 24.** CCR obtenus par ANN en fonction de ses deux paramètres d'entrée suivant les quatre ensembles d'apprentissage des données IRIS. ..... 132
**Figure 25.** CCR obtenus par les méthodes KNN, SVM et ANN en fonction des ensembles d'apprentissage sur les données IRIS avec le réglage optimal de leurs paramètres d'entrée..... 134
**Figure 26.** Résultats du partitionnement de l'image hyperspectrale synthétique de la Figure 2 par HUP-OAP, (a) : image partitionnée et (b) : signatures spectrales moyennes (en abscisse le nombre de bandes spectrales et en ordonnée la réflectance) des deux sous-classes de pêchers (Pêchers et New classe de Pêchers)..... 136

**Figure 27.** Echantillons d'apprentissage sélectionnés de l'image hyperspectrale synthétique de la Figure 2 par la méthode proposée. ..... 136
**Figure 28.** CCR obtenus par SVM en faisant varier ses trois paramètres d'entrée en fonction des quatre ensembles d'apprentissage de l'image hyperspectrale synthétique de la Figure 2. ..... 138
**Figure 29**. CCR obtenus par KNN en faisant varier le paramètre *K* suivant les quatre ensembles d'apprentissage de l'image hyperspectrale synthétique de la Figure 2..... 139
**Figure 30.** CCR obtenus par ANN en faisant varier ses deux paramètres d'entrée suivant les quatre ensembles d'apprentissage de l'image hyperspectrale synthétique de la Figure 2. ..... 140
**Figure 31.** CCR des méthodes supervisées SVM, KNN et ANN en fonction des ensembles d'apprentissage sur l'image hyperspectrale synthétique de la Figure 2 avec l'ensemble optimal de paramètres. ..... 141
**Figure 32.** Echantillons d'apprentissage sélectionnés de l'image hyperspectrale réelle par la méthode proposée..... 142
**Figure 33.** CCR obtenus par SVM en faisant varier ses trois paramètres d'entrée selon les trois ensembles d'apprentissage de l'image hyperspectrale réelle de la Figure 12. ..... 144
**Figure 34**. CCR obtenus par KNN en faisant varier le paramètre *K* suivant les ensembles d'apprentissage SH-TS, MSH-TS et CGT-TS de l'image hyperspectrale réelle de la Figure 12. ..... 144
**Figure 35.** CCR obtenus par ANN en faisant varier ses deux paramètres d'entrée selon les ensembles d'apprentissage SH-TS, MSH-TS et CGT-TS de l'image hyperspectrale réelle de la Figure 12. ..... 145
**Figure 36.** CCR des méthodes supervisées SVM, KNN et ANN suivant les ensembles d'apprentissage SH-TS, MSH-TS et CGT-TS sur l'image hyperspectrale réelle de la Figure 12 avec le réglage optimal de leurs paramètres. ..... 146
**Figure 37.** Résultats de partitionnement par HUP-DIA de l'image hyperspectrale synthétique de la Figure 6 (nombre de classes 9, *LN* = 0.26). ..... 161
**Figure 38.** Résultat de partitionnement optimal par la méthode HUP-DIA de l'image hypersepctrale réelle de la Figure 12 (4 classes, *LN* = 0.16 au niveau 6)..... 163
**Figure 39.** Signature spectrale moyenne ± écart-type de chaque classe obtenue par HAUP-DIA (partition optimale : niveau 6, 4 classes) sur l'image hypersepctrale réelle de la Figure 12. ..... 163
**Figure 40.** Performances des méthodes développées sur l'image hyperspectrale synthétique de la Figure 2..... 168
**Figure 41.** Signatures spectrales moyennes de la classe Pêchers subdivisée en sous-classes par les trois méthodes proposées de l'image hyperspectrale synthétique de la Figure 2. ..... 168
**Figure 42.** Image hyperspectrale Cieza acquise par le capteur AISA Eagle et zones d'intérêt à partitionner. ..... 169
**Figure 43.** Signatures spectrales moyennes $\pm$ écart types des 6 classes de la VT de l'image hyperspectrale réelle de la Figure 42. ..... 170
**Figure 44**. Résultat de partitionnement hiérarchique par niveau obtenu par la méthode HUP-OAP sur l'image complète de la Figure 42 et les zones d'intérêt, $Nc$ : nombre estimé de classes, *LN* : valeur du critère d'optimisation. ..... 172
**Figure 45.** Signatures spectrales ± écart type des classes de la VT subdivisées de l'image hyperspectrale réelle de la Figure 42. ..... 173
**Figure 46**. Résultat de partitionnement hiérarchique par niveau obtenu par la méthode HUP-OAPM-RSM sur l'image complète de la Figure 42 et les zones d'intérêt, $Nc$ : nombre estimé de classes, *LN* : valeur du critère d'optimisation. ..... 175
**Figure 47.** Signatures spectrales ± écart type des classes de la VT divisées de l'image hyperspectrale réelle de la Figure 42. ..... 176
**Figure 48**. Résultat de partitionnement hiérarchique par niveau obtenu par la méthode HUP-DIA sur l'image complète de la Figure 42 et les zones d'intérêt, $Nc$ : nombre estimé de classes, *LN* : valeur du critère d'optimisation. ..... 178
**Figure 49.** Signatures spectrales ± écart type des classes de la VT divisées de l'image hyperspectrale réelle de la Figure 42. ..... 179
**Figure 50.** Classe obtenue par la méthode HUP-OAP avec confusion entre les expressions HA et SA3 de la femme KR. ..... 182
**Figure 51.** Classe obtenue par la méthode HUP-OAPM-RSM avec confusion entres les expressions AN et DI de la femme KA. ..... 184
**Figure 52.** Exemple d'une classe obtenue par HUP-OAP avec $\zeta = 1$ : confusion des individus (1), (2) et (3) des classes $C_3$, $C_4$ et $C_8$ avec ceux des individus (4) et (5) de la classe $C_{15}$ et nombre de mises en correspondance. ..... 187
**Figure 53.** La vraie classe d'appartenance de l'individu (1) d'après les données de la VT et nombre de correspondances ..... 187

**Figure 54.** Exemple de mises en correspondance entre l'individu (1) et les individus (2), (15) et (9). ..... 187
**Figure 55.** Exemple d'un individu (79) mal classé par HUP-OAP et nombre de correspondances. ..... 189
**Figure 56.** La vraie classe d'appartenance de l'individu (79) et nombre de correspondances. ..... 189

# Liste des Tableaux

**Tableau 1.** Base de données 1. ..... 45
**Tableau 2.** Résultats de partitionnement par l'AP sur les sept ensembles de données. ..... 45
**Tableau 3.** Numéro des bandes spectrales et longueur d'onde associée des images hyperspectrales de la Figure 2. .. 46
**Tableau 4.** Détails des 5 classes de la VT de l'image synthétique hyperspectrale de la Figure 2. ..... 47
**Tableau 5.** Résultats d'estimation du nombre de classes ($Nc$) et le nombre d'itérations ($Niter$) par l'AP suivant λ (a : application de l'AP à tous les pixels de l'image de la Figure 2, b : application de l'AP aux pixels non dupliqués et exemplaires des pixels dupliqués). ..... 48
**Tableau 6.** Matrice de similarité entre individus de la Figure 4 (distances $d1$ et $d2$) dans le cas où $\varepsilon ijk > 1$ ..... 60
**Tableau 7.** Matrice de similarité entre individus de la Figure 5 (distances $d1$ et $d2$) dans le cas où $\varepsilon ijk \in ]0,1[$. .... 61
**Tableau 8.** Détails des classes de la VT de l'image hyperspectrale synthétique de la Figure 6. ..... 62
**Tableau 9.** Correspondance bande spectrale/longueur d'onde de l'image hyperspectrale synthétique de la Figure 6. 64
**Tableau 10.** Matrice de confusion du résultat du partitionnement de l'image hyperspectrale synthétique de la Figure 6 par HUP-OAP. ..... 84
**Tableau 11**. Résultats de partitionnement par HUP-OAP de l'image hyperspectrale synthétique de la Figure 6 suivant la taille des blocs. ..... 84
**Tableau 12**. Performances de la méthode développée et des cinq autres méthodes comparées sur l'image hyperspectrale synthétique de la Figure 6. ..... 84
**Tableau 13.** Nombre estimé de classes par la méthode HUP-OAP et valeurs du critère d'optimisation $LN$ pour chaque partition de l'image hyperspectrale réelle de la Figure 12. ..... 88
**Tableau 14.** Taux de couverture de chaque classe de la Figure 14 obtenue au niveau 4 par HUP-OAP. ..... 89
**Tableau 15.** Temps CPU et espace mémoire de l'image hyperspctrale réelle partitionnée de la Figure 12 par HUP−OAP. ..... 89
**Tableau 16.** Performances de la méthode développée HUP-OAP, U-OFCM, S-OFCM, FCM et $K$−means sur l'image hyperspctrale réelle de la Figure 12. ..... 90
**Tableau 17.** Comparaison des performances des méthodes de partitionnement sur l'image hypespectrale synthétique de la Figure 6. ..... 112
**Tableau 18.** Moyenne et variance de l'erreur de mise à jour des matrices $R$ et $A$ calculées par AP originale et HUP-OAPM-RSM sur l'image hypespectrale synthétique de la Figure 6. ..... 112
**Tableau 19.** Nombre estimé de classes par la méthode HUP-OAPM-RSM sur l'image hyperspctrale réelle de la Figure 18 par niveau de partitionnement et valeurs du critère d'optimisation $LN$ pour chaque partition. ..... 114
**Tableau 20.** Taux de couverture de chaque classe obtenue au niveau 7 par HUP-OAPM-RSM sur l'image hyperspectrale réelle de la Figure 18. ..... 115
**Tableau 21.** Performance de la méthode développée HUP-OAPM-RMS, U-OFCM, S-OFCM, FCM et $K$-means sur l'image hypespectrale réelle de la Figure 18. ..... 115
**Tableau 22.** Paramètres d'entrée des méthodes supervisées SVM, KNN et ANN. ..... 125
**Tableau 23.** Évaluation de la partition obtenue par HUP-OAP sur les données originales de la VT d'Iris en utilisant le critère de distance $L1$-norme. ..... 128
**Tableau 24.** Nombre d'échantillons d'apprentissage obtenus par la méthode proposée sur les données IRIS. ..... 129
**Tableau 25.** CCR (%) des méthodes KNN, SVM et ANN en fonction de la sélection automatique des échantillons d'apprentissage sur les données IRIS avec le réglage optimal de leurs paramètres d'entrée. ..... 134
**Tableau 26.** Performances des méthodes supervisées avec sélection d'échantillons d'apprentissage, des méthodes d'apprentissage profond et de la méthode non supervisée sur les données IRIS. ..... 134
**Tableau 27.** Nombre d'échantillons d'apprentissage par classe de l'image hyperspectrale synthétique de la Figure 2 obtenue par la méthode proposée. ..... 136

**Tableau 28.** CCR (%) des méthodes KNN, SVM et ANN en fonction de la sélection automatique des échantillons d'apprentissage sur l'image hyperspectrale synthétique de la Figure 2 avec le réglage optimal de leurs paramètres d'entrée. ..... 140
**Tableau 29.** Taux de couverture au sol de chaque classe obtenue au niveau 4 par HUP-OAP sur l'image hyperspectrale réelle de la Figure 12. ..... 141
**Tableau 30.** Nombre d'échantillons d'apprentissage par classe de l'image hyperspectrale réelle de la Figure 12 obtenue par la méthode proposée. ..... 142
**Tableau 31.** CCR en % des méthodes de partitionnement supervisées SVM, KNN et ANN sur l'image hyperspectrale réelle de la Figure 12 en fonction de la sélection des échantillons d'apprentissage avec le réglage optimal de leurs paramètres. ..... 146
**Tableau 32.** Les paramètres SVM donnant les meilleures performances sur les 3 bases de données (IRIS, Images de la Figure 2 et de la Figure 12). ..... 147
**Tableau 33**. Le paramètre *K* de KNN fournit la meilleure performance sur les 3 bases de données (IRIS, Images de la Figure 2 et de la Figure 12). ..... 147
**Tableau 34**. Les paramètres ANN donnant les meilleures performances sur les 3 bases de données (IRIS, Images de la Figure 2 et de la Figure 12). ..... 147
**Tableau 35.** Temps CPU (s) par HUP-OAP avec et sans parallélisation sur les trois bases de données (IRIS, Images de la Figure 2 et de la Figure 12). ..... 147
**Tableau 36.** Comparaison des performances des méthodes de partitionnement sur l'image hyperspectrale synthétique de la Figure 6. ..... 161
**Tableau 37.** Comparaison de trois méthodes de partitionnement développées sur l'image hyperspectrale synthétique de la Figure 6. ..... 162
**Tableau 38.** Nombre estimé de classes par la méthode HUP-DIA de l'image hyperspectrale réelle de la Figure 12 et la valeur du critère d'optimisation *LN* pour chaque partition. ..... 163
**Tableau 39.** Taux de couverture de chaque classe obtenue au niveau 6 par HUP-DIA sur l'image hypersepctrale réelle de la Figure 12. ..... 164
**Tableau 40.** Performance de la méthode développée HUP-DIA, U-OFCM, S-OFCM, FCM et *K*-means sur l'image hypersepctrale réelle de la Figure 12. ..... 164
**Tableau 41.** Comparaison des performances des méthodes développées sur l'image hypersepctrale réelle de la Figure 12. ..... 164
**Tableau 42**. Détails des classes de la VT de l'image hyperspectrale réelle de la Figure 42. ..... 170
**Tableau 43.** Matrice de confusion correspondant à la partition optimale de l'image hyperspectrale réelle de la Figure 42 obtenue par HUP-OAP au niveau 3, où le nombre estimé de classes est 10. ..... 172
**Tableau 44.** Matrice de confusion correspondant à la partition optimale de l'image hyperspectrale réelle de la Figure 42 obtenue par HUP-OAPM-RSM au niveau 4, où le nombre estimé de classes est 9. ..... 175
**Tableau 45.** Matrice de confusion correspondant à la partition optimale de l'image hyperspectrale réelle de la Figure 42 obtenue par HUP-DIA au niveau 4, où le nombre estimé de classes est 9. ..... 178
**Tableau 46.** Synthèse des performances des trois méthodes développées pour le partitionnement de l'image hyperspectrale réelle de la Figure 42. ..... 179
**Tableau 47.** Les 7 expressions faciales de JAFFE. ..... 180
**Tableau 48.** Partitionnement des données JAFFE par HUP-OAP : nombre estimé de classes ($Nc$), valeur de *LN* et temps CPU par partition. ..... 181
**Tableau 49.** Matrice de confusion et CCR par HUP-OAP sur la base de données JAFFE au niveau 1. ..... 181
**Tableau 50**. Partitionnement des données JAFFE par HUP-OAPM-RSM : nombre estimé de classes ($Nc$), valeur de *LN* et temps CPU par partition. ..... 183
**Tableau 51.** Matrice de confusion et CCR par HUP-OAPM-RSM sur la base de données JAFFE au niveau 1. ..... 183
**Tableau 52**. Partitionnement des données JAFFE par HUP-DIA : nombre estimé de classes ($Nc$), valeur de *LN* et temps CPU par partition. ..... 184
**Tableau 53.** Matrice de confusion et CCR par HUP-DIA sur JAFFE au niveau 1. ..... 185
**Tableau 54**. Les onze configurations de la base de données YALE. ..... 186
**Tableau 55.** Résultats d'estimation du nombre de classes ($Nc$) sur les données YALE, le CCR-SCVT et l'indice *LN* obtenus par HUP-OAP au niveau 1 en fonction de $\zeta$. ..... 187

**Tableau 56.** Résultats d'estimation du nombre de classes ($Nc$) de la base ORL par HUP-OAP en fonction de $\zeta$ : le CCR-SCVT et l'indice *LN*. ..... 188
**Tableau 57**. Performances des trois méthodes développées pour la reconnaissance faciale avec expressions. ..... 189
**Tableau 58.** Résultats de partitionnement par HUP-OAP pour la détection de tumeur cérébrale par IRM. ..... 190
**Tableau 59.** Résultats de partitionnement par HUP-OAPM-RSM pour la détection de tumeur cérébrale par IRM. 191
**Tableau 60.** Résultats de partitionnement par HUP-DIA pour détection de tumeur cérébrale par IRM. ..... 191
**Tableau 61.** Résultats par niveau de partitionnements des trois méthodes développées pour la détection de mélanomes. ..... 191