

**Université Saad Dahlab Blida**

N° D'ordre:.....



Faculté des sciences

**Département d'informatique**

Mémoire Présenté par:

**MOUALHI Mohamed**

**En vue d'obtenir le diplôme de master**

Domaine: Mathématique et informatique

Filière: Informatique

Spécialité: Informatique

Option: Ingénierie de logiciel



**Thème:**

**Classification thématique et détection de changement des images satellitaires en utilisant les algorithmes de machines d'apprentissage**

Promotrice:

Pr ABED Hafida

Encadrant:

Dr ZERROUKI Nabil

MA-004-491-1

**Soutenu le:**

Dr CHIKHI Nacim Fateh

Président

Mr DOUGA Yassine

Examineur

Pr ABED Hafida

Promotrice

Dr ZERROUKI Nabil

Encadrant

Année universitaire: 2017/ 2018

## Résumé

L'utilisation des données de télédétection dans la cartographie de l'état de la surface et de son évolution facilite énormément la planification, l'évaluation, et la surveillance des surfaces terrestres; particulièrement si les résultats sont intégrés dans des systèmes d'information géographique **SIG**. Comparée aux techniques dites classique comme la photo-interprétation, la télédétection spatiale présente des avantages incontestables en termes de rapidité, de facilité de mise à jour, mais aussi de superficie couverte.

Dans ce projet, nous proposons un système de classification thématique et de détection de changements à base d'algorithme de machines d'apprentissage supervisé.

Pour permettre une classification précise, plusieurs attributs sont proposés dans la caractérisation des pixels dans la scène, à savoir les caractéristiques radiométriques et les caractéristiques de texture.

## Abstract

The use of remote sensing data in surface mapping and its evolution greatly facilitates land surface planning, assessment and monitoring; especially if the results are integrated into geographic information systems **GIS**. Compared to so-called classical techniques such as photo-interpretation, spatial remote sensing has undeniable advantages in terms of speed, ease of updating, but also covered area.

In this project, we propose a thematic classification and change detection system based on supervised machine learning algorithms.

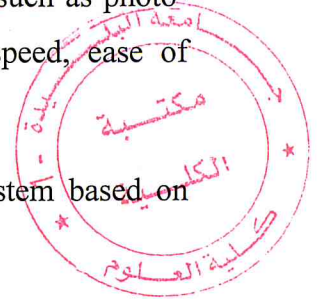
To allow precise classification, several attributes are proposed in the characterization of pixels in the scene, namely radiometric characteristics and texture characteristics.

## ملخص

إن استخدام بيانات الاستشعار عن بعد في رسم خرائط حالة سطح الأرض وتطورها يسهل إلى حد كبير في عمليات التخطيط، التقييم والمراقبة؛ لا سيما إذا تم دمج النتائج في نظم المعلومات الجغرافية. مقارنةً مع ما يطلق عليه التقنيات الكلاسيكية مثل تفسير الصور، إن الاستشعار عن بعد له عدة مزايا لا يمكن إنكارها من حيث السرعة وسهولة التحديث، ولكن أيضا من حيث حجم المنطقة المغطاة.

في هذا المشروع، نقترح نظامًا للتصنيف المواضيعي و للكشف عن التغيير يعتمد على خوارزميات التعلم الآلي الخاضعة للإشراف.

من أجل أكثر دقة في عملية التصنيف، يتم اقتراح العديد من السمات لتحديد خصائص البكسل في المشهد، بالإضافة إلى الخصائص الإشعاعية، تم استخدام خصائص النسيج.



## *REMERCIEMENTS*

*AVANT DE PRÉSENTER MON TRAVAIL, IL EST DE MON DEVOIR D'EXPRIMER MES REMERCIEMENTS TOUT D'ABORD AU TOUT-PUISSANT ALLAH QUI M'A DONNÉ LA FORCE DE RÉALISER CE TRAVAIL.*

*À MA FAMILLE TOUT PARTICULIÈREMENT MA FEMME QUI A ÉTÉ UN GRAND SOUTIEN TOUT AU LONG DE LA PÉRIODE DE RÉALISATION DE CE TRAVAIL.*

*Je suis profondément reconnaissant à ma promotrice, Madame ABED Hafida pour son soutien, implication, expérience, son temps qu'elle a investi pour suivre et analyser les résultats de ce travail et enrichir mes connaissances durant toute cette année.*

*Je tiens à remercier mon Encadrant Monsieur ZERROUKI Nabil pour son aide et engagement à mon côté, de m'avoir accompagné dans la maîtrise de mes connaissances. Ainsi que pour ses nombreux conseils qu'il n'a cessé de me prodiguer.*

*Je remercie les membres du jury pour avoir accepté d'examiner ce travail.*

## *Dédicaces*

*Grace à Dieu voilà notre travail est terminé et il est temps pour moi de partager Ma joie avec tous ceux qui m'ont soutenu et encouragé.*

*À travers ce modeste mémoire, je tiens à présenter mes sincères dédicaces à Mr CHAFAA Nabil qui m'a permis de suivre mes études.*

*Mon père et ma mère qui ont consacré leur vie à notre éducation et à faire notre bonheur et qui nous encouragent toujours à terminer nos études en espérant voir les fruits de leurs sacrifices.*

*A mes adorables filles Lea et Maylis.*

*A mes chers frères et sœurs.*

*A l'ensemble des amis que j'ai connu pendant mes études et à eux qui m'ont prodigué leurs vifs conseils, encouragements et témoigné de leur amitié.*

**MOUALHI mohamed**

# Table des matières

INTRODUCTION GENERALE .....	1
CHAPITRE 1: généralités sur l'utilisation des algorithmes de machine d'apprentissage dans la télédétection.....	4
1.1 Introduction .....	5
1.2 Problématique.....	6
1.3 Extraction des caractéristiques .....	6
1.4 Description des données: multi-spectrales, hyper-spectrales .....	7
1.5 Description des différentes caractéristiques utilisées dans la télédétection.....	8
1.5.1 Valeurs radiométriques dans le spectre du visible.....	8
1.5.2 Indice de végétation par différence normalisé (NDVI) .....	8
1.5.3 Caractéristiques de texture.....	9
1.6 Conclusion .....	12
CHAPITRE 2: Classification thématique et détection de changement en utilisant les algorithmes de machine d'apprentissage.....	13
2.1 Introduction .....	14
2.2 Classification non supervisée .....	14
2.3 Classification supervisée .....	15
2.4 Conclusion .....	24
CHAPITRE 3: Analyse et Conception .....	25
3.1 Introduction .....	26
3.2 Analyse .....	26
3.2.1 présentation des logiciels utilisés.....	26
3.2.2 Présentation du système existant .....	27
3.2.3 Étude préliminaire .....	30
3.3 Conception.....	32
3.3.1 Diagramme de cas d'utilisation.....	32
3.3.2 Diagramme de classes.....	35
3.3.3 Diagramme de séquence .....	38
3.4 Conclusion .....	41
CHAPITRE 4: Réalisation.....	42
4.1 Introduction .....	43
4.2 Architecture du système .....	43
4.3 Environnement et technologies logicielles .....	44
4.3.1 Plateforme de développement.....	44

4.3.2 Bibliothèques utilisées .....	45
4.4 Implémentation .....	46
4.4.1 Phase 1: Extraction des caractéristiques .....	47
4.4.2 Phase 2: Formation DE L'ensemble de données d'apprentissage .....	48
4.4.3 Phase 3: Préparation des ensembles de données pour la classification .....	50
4.4.4 Phase 4: Classification .....	50
4.5 Interface graphique .....	53
4.6 Conclusion .....	57
CHAPITRE 5: Résultats et interprétation .....	58
5.1 Introduction .....	59
5.2 Description des données utilisées .....	59
5.2.1 Sztaki AirChange Benchmark .....	59
5.3 Critères d'évaluation .....	60
5.3.1 Matrice de confusion .....	60
5.3.2 Taux de reconnaissance .....	60
5.3.3 Le coefficient de Kappa .....	61
5.3.4 Le coefficient F-measure .....	61
5.4 Les résultats obtenus .....	62
5.4.1 Les résultats de la classification thématique .....	63
5.4.2 Les résultats de la détection du changement .....	65
5.5 Conclusion .....	67
CONCLUSION GENERALE ET PERSPECTIVES .....	68
ANNEXE .....	71
BIBLIOGRAPHIE .....	72

## **LISTE DES ABREVIATIONS**

- SIG:** Système d'Information Géographique.
- NDVI:** Normalized Difference Vegetation Index.
- PIR:** Proche Infra-Rouge.
- GLCM:** Gray-Level Co-Occurrence Matrix.
- SSIM:** Structural Similarity Index.
- SVM:** Support Vector Machines.
- RBF:** Radial Basis Function.
- MLP:** Multi layer perceptron.
- PC:** Personal Computer.
- JS:** Java Script.
- JSON:** JavaScript Object Notation.
- HTTP:** hypertext transfer protocol.
- MIME:** Multipurpose Internet Mail Extensions.
- RFC:** Requests For Comments.
- R:** valeur radiométrique Red «Rouge».
- G:** valeur radiométrique Green «vert».
- B:** valeur radiométrique Bleu.
- E:** Entropie.
- D:** Dissimilarité.
- P:** Pixel.

## LISTE DES FIGURES:

Figure 1.1 Signatures spectrales des thèmes naturels [Harsanyi, 1994].	7
Figure 1.2 Illustration du principe du calcul de la matrice de cooccurrence de niveau de gris selon une distance $d=1$ et une orientation $\theta=0$ (direction horizontale).	10
Figure 2.1 Exemple d'architecture globale d'un réseau de neurones.	16
Figure 2.2 Schéma de l'algorithme de définition des hyperplans de séparation.	18
Figure 2.3 Exemple de classification AdaBoost.	23
Figure 3.1 Diagramme de cas d'utilisation du système existant.	28
Figure 3.2 Diagramme de classe du système existant.	29
Figure 3.3 Diagramme de cas d'utilisation	34
Figure 3.4 Diagramme de classes	37
Figure 3.5 Diagramme de séquence du cas d'utilisation «Entrainer le classifieur»	39
Figure 3.6 Diagramme de séquence du cas d'utilisation «Evaluer le modèle prédictif»	40
Figure 3.7 Diagramme de séquence du cas d'utilisation «Classifier».	41
Figure 4.1 Architecture client-serveur	43
Figure 4.2 Plateforme de développement	45
Figure 4.3 Extraction des caractéristiques.	47
Figure 4.4 Calcul des caractéristiques contextuelles.	48
Figure 4.5 Formation d'ensemble de données d'apprentissage «classification thématiques»	49
Figure 4.6 Formation d'ensemble de données d'apprentissage «détection du changement».	49
Figure 4.7 Manipulation d'ensembles des données.	50
Figure 4.9 Sélection du mode d'utilisation.	53
Figure 4.10 Chargement des images satellitaires.	53
Figure 4.11 Sélection des échantillons d'apprentissage.	54
Figure 4.12 Sélection et paramétrage des classifieurs.	55
Figure 4.13 Sélection du modèle prédictif.	55
Figure 4.14 Boutons de déclenchement des actions «Apprentissage, Evaluation, Classification».	55
Figure 4.15 Résultats affichés.	56
Figure 5.1 Une paire d'images de la base de données SZTAKI AirChange Benchmark.	59
Figure 5.2 Comparaison des résultats de la classification thématique.	63
Figure 5.3 Comparaison des résultats de la détection du changement.	66



## LISTE DES TABLEAUX

Tableau 1. 1 Les paramètres de texture de Haralick définis à partir de la matrice de cooccurrence [Haralick, 1973].....	11
Tableau 5. 1 Comparaison des performances entre les différents algorithmes de classification thématique. ....	64
Tableau 5.2 Comparaison des performances entre les différentes méthodes de détection du changement. ....	66

# **INTRODUCTION GENERALE**

## Introduction générale

De nos jours, l'analyse et le traitement de données de télédétection particulièrement la classification thématique, représente un domaine de recherches très intéressant. D'autre part, la détection de changements en utilisant des données multi-dates est devenue une préoccupation majeure dans le domaine de cartographie de la couverture terrestre, en particulier les régions touchées par des phénomènes tels que la déforestation ou l'avancement du tissu urbain au détriment des surfaces agricoles.

L'utilisation des images satellitaires ou aérienne pour la cartographie thématique s'effectue principalement à travers un premier processus de classification dont le but est de regrouper dans une même entité ou classe, tous les pixels similaires sur la base d'une ressemblance de caractéristiques. D'autre part, la détection de changement se fait à travers une deuxième étape de comparaison entre plusieurs images couvrant la même région mais prises à des dates différentes.

Dans le présent travail de recherche qui fait l'objet de ce mémoire de Master, nous nous intéressons à la classification thématique et à la détection de changements en utilisant les techniques de machines d'apprentissage. De manière plus précise, plusieurs attributs sont utilisés pour caractériser le pixel dans une scène, à savoir les valeurs radiométriques et les caractéristiques contextuelles (entropie et dissimilarité). Ces attributs sont regroupés dans un vecteur caractéristique qui sera utilisé par la suite dans la classification.

Le processus de classification et de détection que nous avons mis en œuvre a été appliqué sur une base de données réelle (SZTAKI AirChange Benchmark) couvrant plusieurs milieux terrestres tel que urbain, végétation, surfaces humide, etc. Afin d'évaluer de manière significative le système proposé, plusieurs mesures statistiques ont été évoquées. Les résultats obtenus ont permis de mettre en évidence les avantages des techniques de machines d'apprentissage dans la classification des données de télédétection.

Afin d'organiser ce mémoire en parties cohérentes, nous avons structuré ce manuscrit en cinq chapitres. Dans le Chapitre 1, nous décrivons les données de télédétection ainsi que les caractéristiques les plus utilisées dans la description de ces données. Nous évoquons par la suite dans le chapitre 2 les aspects généraux de la classification thématique et la détection de changement dans les images satellitaires. Nous détaillons également les différentes méthodes de machine d'apprentissage utilisées telles que: Adaboost, les réseaux de neurones et les Séparateurs à Vaste Marge, etc. Dans le Chapitre 3, nous présentons les parties analyse et conception du système proposé dans ce travail. Pour cela, une étude conceptuelle de système

proposé est élaborée en détail. Le chapitre 4 est consacré à l'implémentation du système en présentant son architecture, les différents langages de programmation et bibliothèques utilisés, ainsi que l'interface graphique du système.

Dans le chapitre 5, nous décrivons dans un premier lieu les données utilisées et nous présentons par la suite une analyse globale des résultats obtenus et leurs interprétations. Pour finir, nous concluons ce mémoire en résumant les principales contributions de ce travail où des orientations futures et des perspectives sont aussi mentionnées.

**CHAPITRE 1: GENERALITES SUR L'UTILISATION DES  
ALGORITHMES DE MACHINE D'APPRENTISSAGE DANS LA  
TELEDETECTION**

## 1.1 INTRODUCTION

Avec l'accroissement effrayant des phénomènes tels que la déforestation ou l'avancement anarchique du tissu urbain au détriment des espaces verts, l'étude et le suivi de l'occupation des sols deviennent, désormais la préoccupation essentielle des autorités locales. La cartographie de l'état de la surface et de son évolution aide énormément à la surveillance, l'organisation et l'évaluation des occupations thématique du sol, particulièrement si ces cartes sont intégrées dans un système d'information géographique **SIG** comme une information de base.

Depuis son apparition, la télédétection spatiale ou même l'imagerie aérienne sont exploitées pour la cartographie des sols. Comparée aux techniques de relevé de type classique (photo-interprétation), la télédétection spatiale (passive ou active) présente des avantages incontestables en termes de coût, de rapidité, de facilité de mise à jour, mais aussi de superficie couverte et de périodicité des acquisitions.

L'utilisation des images satellitaires ou aérienne pour la cartographie thématique s'effectue principalement à travers un processus de classification dont le but est de regrouper dans une même entité ou classe, tous les pixels similaires sur la base d'une ressemblance de propriétés (ou caractéristiques). D'autre part, l'avancement imminent des méthodes de classification (ou de séparation), basées généralement sur les machines d'apprentissage a rendu leurs utilisation presque indispensable dans ce genre de problème [Dai, 1999].

Il est à noter que le concept de la classification apporte des améliorations conséquentes à la compréhension des images satellitaires. Pour cela, nous proposons dans ce travail l'utilisation de deux phases de classification, à savoir une classification statique et une classification dynamique.

La **classification statique** consiste en la reconnaissance des différentes classes thématiques (végétation, milieu urbain, etc.) au sein de la même image satellitaire ou aérienne (décrivant une seule scène). Cette classification est appelée statique car l'identification est basée sur chaque image séparément et non pas sur une séquence d'images.

La deuxième étape consiste en la détection de changement. Cette détection est appelée aussi **classification dynamique**, puisqu'elle est basée sur une série d'images multi-temporelle couvrant la même scène mais prises dans des dates différentes. Les intervalles de temps entre les prises peut aller jusqu'à quelques années, afin de permettre l'évolution dans l'occupation du sol.

## 1.2 PROBLEMATIQUE

Plusieurs méthodes et approches de classification d'images satellitaires et aériennes ont été proposées dans la littérature. Cependant, de nombreuses méthodes dites de détection de changement sont basées sur les approches classiques qui utilisent généralement des méthodes de reconnaissances simples et qui ne nécessitent pas forcément de modèle mathématique, parmi ces méthodes on peut citer la classification par seuillage ou par calcul de distances [Timothy, 2006]. Ces méthodes de détection de changement se heurtent à certaines difficultés inhérentes au contenu des images satellitaires en général, en particulier les images à très haute résolution spatiale (hétérogénéité spatiale et spectrale des classes). En effet, ces méthodes s'avèrent limitées pour la séparation des changements réels (généralement irréversible tel que l'avancement urbain) des changements réversibles dus généralement aux conditions climatiques (telle que la variation des couleurs de la végétation d'une scène à une autre, les passages des nuages ou encore les ombres des objets dans l'image).

Il y a aussi des méthodes de détection de changement qui utilisent uniquement les valeurs radiométriques comme caractéristiques des pixels [Jin, 2013]. Ces méthodes ne peuvent pas prendre en compte l'hétérogénéité spectrale des pixels appartenant aux différentes classes thématiques d'occupation du sol (pour une classification statique) ou des classes: changée et inchangée (dans le cas d'une classification dynamique). De plus, la mise en place de classes relativement générales, afin de minimiser la confusion, peut ne pas répondre aux besoins réels des utilisateurs.

## 1.3 EXTRACTION DES CARACTERISTIQUES

La multiplication des imageurs spatiaux et spectraux est une véritable opportunité pour l'identification de la nature de l'occupation du sol. Toutefois, l'hétérogénéité de ces données satellitaires ou aérienne, perturbe les méthodes de classification classique [Shat, 2008], dite purement spectrale, qui sont plus adaptées à l'analyse d'image de résolution moyenne. La tendance actuelle est au développement de méthodes permettant d'un côté une bonne séparabilité entre les différentes classes et d'un autre côté une détection des changements robuste. Pour cette raison, l'extraction de caractéristiques est considérée comme une étape cruciale dans le processus de classification et de détection. Ces caractéristiques ont un impact direct sur la performance du système de classification. Pour cela, les caractéristiques extraites des pixels dans l'image doivent être significatives et représentatives. Dans ce travail, nous avons utilisé des caractéristiques contextuelles telles que l'entropie et la dissimilarité en plus

## 1.5 DESCRIPTION DES DIFFERENTES CARACTERISTIQUES UTILISEES DANS LA TELEDETECTION

Plusieurs caractéristiques pour décrire la couverture terrestre ont été proposées dans la littérature, dans cette partie nous décrivons deux types de caractéristiques les plus utilisés à savoir (i) **pixélique** telles que les valeurs radiométriques, l'indice de végétation, NDVI, et (ii) caractéristiques de **texture**, comme l'entropie, l'énergie, la dissimilarité, etc.

### 1.5.1 VALEURS RADIOMETRIQUES DANS LE SPECTRE DU VISIBLE

La variation thématique de la couverture terrestre (végétation, sol nu, etc.) correspond aussi à une variation des réflectances par rapport aux différentes longueurs d'ondes existante. Dans cette section, nous décrivons la variation des valeurs radiométriques (liées aux degrés de réflectances) dans le domaine visible des longueurs d'ondes (entre 0.3 et 0.7  $\mu\text{m}$ ). Il est à noter que la réflectance d'une surface dépend de plusieurs paramètres tels que la couleur, la teneur en eau dans cette surface, etc. Les surface humides sont généralement caractérisées par la couleur sombre (bleu) ce qui augmente leurs degré d'absorption de la lumière (peu de réflectance). Alors que la réflectance de la végétation dépend surtout de la teneur d'eau, ou lorsque cette dernière est importante cela se traduit par une faible réflectance [Kruse, 1993].

### 1.5.2 INDICE DE VEGETATION PAR DIFFERENCE NORMALISE (NDVI)

L'indice de végétation par différence normalisé, appelé aussi Normalized Difference Vegetation Index (**NDVI**) est construit à partir des données de réflexion dans les canaux rouge (**R**) et proche infra-rouge (**PIR**). L'indice de végétation normalisé met en valeur la différence entre la bande visible du rouge et celle du proche infrarouge.

$$\text{NDVI} = \frac{\text{PIR} - \text{R}}{\text{PIR} + \text{R}} \quad (1.1)$$

Cet indice est fortement lié à la réflectance des zones végétales. Les valeurs du NDVI sont comprises entre -1 et +1, les valeurs négatives correspondant aux surfaces autres que les couverts végétaux, comme la neige, l'eau ou les nuages pour lesquelles la réflectance dans le rouge est supérieure à celle du proche infrarouge. Pour les sols nus, les réflectances étant à peu près du même ordre de grandeur dans le rouge et le proche infrarouge, le NDVI présente des valeurs proches de 0.

Les formations végétales quant à elles, ont des valeurs de NDVI positives, généralement comprises entre 0,1 et 0,7. Les valeurs les plus élevées correspondant aux couverts les plus denses [Shao, 2016].



### 1.5.3 CARACTERISTIQUES DE TEXTURE

La texture est un phénomène qui implique la description et l'interaction entre les primitives (éléments de base et motifs) qui composent la texture [Haralick, 1979]. Par ailleurs, la signature texturale est une mesure indépendante de la taille et de la forme de l'image d'origine mais dépend de la taille et de la forme de la fenêtre d'analyse. Certaines mesures de texture représentent une caractéristique texturale spécifique de l'image telles que: l'homogénéité, le contraste et l'entropie. Les autres mesures caractérisent la complexité et la nature des transitions de niveaux de gris présentes dans l'image. On peut différencier deux approches: l'approche structurale et l'approche statistique. Dans la première approche, on considère la texture comme une agrégation de primitives [Ratté, 1999]. Les différents types de primitives, leur orientation et leur forme. Dans l'approche statistique, on utilise des paramètres statistiques où la texture est modélisée comme une fonction de niveau de gris. L'objectif de l'approche statistique est de caractériser les propriétés stochastiques (probabilistes) de la distribution spatiale des niveaux de gris [Haralick, 1979].

#### 1.5.3.1 Matrice de cooccurrence

La matrice de cooccurrence, développée par **Haralick** est une méthode statistique d'analyse de texture qui représente une caractérisation de la distribution spatiale des niveaux de gris dans une image monospectrale.

La matrice **GLCM** (Gray-Level Co-Occurrence Matrix) mesure la probabilité d'apparition des paires de valeurs de pixels situés à une certaine distance dans l'image.

L'élément  $(i, j)$  de la matrice **GLCM** est égal à la probabilité  $P(i, j, d, \theta)$  associée au nombre de fois dans l'image où un pixel de niveau de gris  $i$  apparaît à une distance relative  $d$  (en pixels) d'un pixel de niveau de gris  $j$  et selon une orientation  $\theta$  donnée. Les directions angulaires  $\theta$  classiquement utilisées sont 0, 45, 90 et 135 degrés. Le pixel de niveau de gris  $i$  est nommé pixel de référence et le pixel  $j$  est appelé pixel voisin comme illustré dans la figure 1.2 [Benelcadi, 2014].

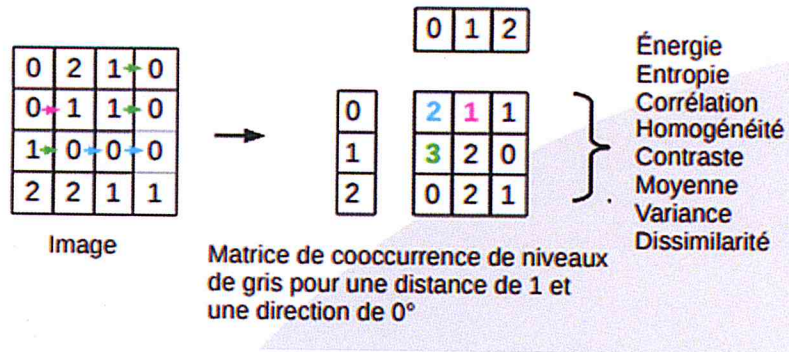


Figure 1.2 Illustration du principe du calcul de la matrice de cooccurrence de niveau de gris selon une distance  $d=1$  et une orientation  $\theta=0$  (direction horizontale).

Plusieurs mesures de texture sont calculées à partir de la matrice de cooccurrence. En effet, ces paramètres de texture ont été regroupés en trois catégories:

- **Par rapport au contraste:** Les paramètres de ce groupe sont reliés au contraste, utilisant un poids lié à la diagonale de la matrice **GLCM**. Ce groupe comprend (contraste, dissimilarité et l'homogénéité).
- **Par rapport à l'ordre:** Ce groupe renseigne sur la régularité des pixels de l'image. Autrement dit le nombre d'occurrence des couples de niveau de gris doit être le même pour les autres couples de niveau de gris différents. Ces mesures sont basées sur les fréquences d'apparition des couples de niveaux de gris dans l'image. Si ces couples se répètent de la même fréquence, l'image est ordonnée, sinon elle est désordonnée. Ce groupe comprend (l'énergie et l'entropie).
- **Statistique:** Il s'agit des mesures statistiques calculées à partir de la matrice **GLCM** dont il regroupe la moyenne, la variance et la corrélation.

Le Tableau 1.1 présente les principales mesures ainsi que leurs formules de calcul:

<b>Le contraste</b>	$\sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} P(i,j) *  i - j ^2$
<b>La dissimilarité</b>	$\sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} P(i,j) *  i - j $
<b>L'homogénéité (inverse different moment)</b>	$\sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} P(i,j) * \frac{1}{1 + (i - j)^2}$
<b>L'énergie (Second moment angulaire)</b>	$\sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} P(i,j)^2$
<b>L'entropie</b>	$\sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} P(i,j) * (-\ln (P(i,j) )$
<b>La moyenne</b>	$\mu_i = \sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} i * P(i,j) ,$ $\mu_j = \sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} j * P(i,j)$
<b>La variance</b>	$\sigma_i^2 = \sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} P(i,j) * (i - \mu_i)^2 ,$ $\sigma_j^2 = \sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} P(i,j) * (j - \mu_j)^2$
<b>La corrélation</b>	$\sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} P(i,j) * \left[ \frac{(i - \mu_i) * (j - \mu_j)}{\sqrt{\sigma_i^2 * \sigma_j^2}} \right]$

Tableau 1. 1 Les paramètres de texture de Haralick définis à partir de la matrice de cooccurrence [Haralick, 1973].

Plusieurs remarques peuvent être faites à partir de la définition des paramètres de Haralick.

- Le **contraste** et la **dissimilarité** ont des formules très proches faisant intervenir respectivement  $|i - j|^2$  et  $|i - j|$ . Il n'est pas étonnant qu'une **forte corrélation** entre les deux paramètres soit observée.

- Le **contraste** et l'**homogénéité** sont inversement proportionnel  $|i - j|^2$  et  $\frac{1}{1 + |i - j|^2}$ . Ce qui engendre une redondance dans l'information extraite.

- L'**entropie** et l'**énergie** sont deux mesures **fortement corrélées** ( $p(i,j)^2$  et  $p(i,j) \ln p(i,j)$ ).

En conséquence, nous nous sommes intéressés dans ce travail uniquement à **la dissimilarité** et **l'entropie** pour la caractérisation de texture.

En plus des caractéristiques radiométriques, les caractéristiques de texture sont également utilisées afin de constituer le vecteur caractéristique offrant une meilleure description des données de l'image, pour permettre une classification plus précise.

### 1.5.3.2 entropie

L'entropie est l'une des caractéristiques de texture la plus utilisée. Cette mesure se construit sur la probabilité qu'une valeur soit présente dans l'histogramme. Si tous les niveaux de radiation apparaissent avec la même fréquence, la situation est optimale, car l'image offre ainsi un maximum d'information, alors son entropie est également maximale. Dans notre cas, l'entropie est une variable qui *exprime le degré de désordre ou de structuration* dans l'image. Elle se définit à partir de la probabilité d'apparition d'une valeur par rapport à l'ensemble des valeurs dans un masque dans l'image (une petite partie). L'entropie est définie par l'équation suivante:

$$E(i, j) = - \sum \sum P(i, j) \ln P(i, j) \quad (1.2)$$

Notons que l'entropie maximale croît avec le nombre de niveau de gris disponibles dans la scène connu aussi pour l'indice de codage de l'image.

### 1.5.3.4 Dissimilarité

La dissimilarité est une mesure obtenue à partir du voisinage de chaque pixel. Cette caractéristique mesure l'indice global de similarité structurelle (structural similarity index - SSIM). La dissimilarité est définie par l'équation suivante:

$$D(i, j) = - \sum \sum P(i, j) |i - j| \quad (1.3)$$

En effet, l'indice de dissimilarité est calculé par le balayage d'une fenêtre glissante (élément structurant). Le choix de la taille de l'élément structurant est primordial dans la construction de la matrice de la dissimilarité. L'idée générale est que la taille de la fenêtre soit adaptée au type d'image utilisée comme la résolution de l'image, etc. afin d'extraire convenablement les caractéristiques locales de chaque pixel.

## 1.6 CONCLUSION

Après l'étape d'extraction de caractéristiques, chaque pixel est alors représenté par un vecteur caractéristique correspondant (cinq valeurs). L'étape suivante consiste à affecter chaque pixel à une classe thématique en se basant sur ses caractéristiques. Nous évoquerons en détails (principe, avantages, etc.) dans le chapitre suivant les différents algorithmes de classification utilisés dans ce travail à savoir les réseaux de neurones, les SVMs, Adaboost, le Plus Proche Voisin (KNN) et le classifieur Bayésien.

**CHAPITRE 2: CLASSIFICATION THEMATIQUE ET DETECTION  
DE CHANGEMENT EN UTILISANT LES ALGORITHMES DE  
MACHINE D'APPRENTISSAGE**

## 2.1 INTRODUCTION

Nous rappelons que le but essentiel de la classification des images satellitaires est la cartographie thématique. Le processus de classification s'effectue principalement à travers des regroupements des pixels similaires (sur la base d'une ressemblance de caractéristiques) dans une même entité ou classe.

Les approches de classification en utilisant les machines d'apprentissage sont des méthodes basées généralement sur un fondement mathématique, permettent la résolution des problèmes plus complexes tels que les problèmes non-linéaires. Il existe plusieurs catégories d'algorithmes de machine d'apprentissage, à savoir les méthodes non-supervisées, les méthodes supervisées, semi-supervisées, etc [Duda, 2012]. Dans le présent travail, nous avons utilisé plusieurs algorithmes supervisés pour la classification thématiques et la détection de changement, à savoir Adaboost, les réseaux de neurones, KNN, classifieur Bayésien et les SVMs. Ces méthodes supervisées comprennent deux principales étapes: (i) en premier lieu, affecter à l'algorithme certaines pixels avec leurs classes correspondantes comme échantillons d'apprentissage et (ii) par la suite attribuer le reste des pixels de l'image en fonction de leurs caractéristiques via le modèle construit dans la première phase. Dans les parties suivantes, nous évoquerons plus de détails sur chaque méthode de classification, la phase d'apprentissage, la construction du modèle, l'estimation de ces paramètres et l'affectation des nouveaux échantillons, etc. [Duda, 2012].

## 2.2 CLASSIFICATION NON SUPERVISEE

Dans les méthodes non supervisées, on ne dispose pas de l'ensemble des échantillons d'apprentissage. Le nombre de classes et les règles d'affectation à ces classes doivent être établis seulement à partir d'observations, sans faire référence à une base d'entraînement. Le groupement des individus est réalisé sur la base de similarités et il est généralement conditionné par le choix du nombre de classes. L'utilisateur n'intervient qu'une fois que la classification est effectuée pour interpréter le contenu des classes, sans faire appel à d'autres hypothèses sur les images ou sur les classes. Notons que les techniques de ce type sont couramment utilisées dans un but exploratoire. On retrouve plusieurs méthodes de classification non supervisée qui découlent de l'algorithme de base Isodata [Duda, 1973]. Parmi celles-ci, on cite une méthode très connue et couramment utilisée qui est la méthode des K-means.

## 2.3 CLASSIFICATION SUPERVISEE

Lors de l'utilisation d'une méthode de classification supervisée, l'analyste identifie des échantillons assez homogènes de l'image qui sont représentatifs de différents types de classes d'information. Ces échantillons forment un ensemble de données d'apprentissage. La sélection de ces données est basée sur les connaissances a priori de l'analyste sur la région. Les échantillons de chaque classe sont utilisés pour entraîner le système à définir les classes et ensuite reconnaître des régions aux propriétés similaires à chaque classe.

### 2.3.1 Réseaux de neurone

Dans un premier lieu, nous avons appliqué les réseaux de neurones comme algorithme de classification afin de distinguer entre les différentes classes thématiques. La classification basée sur les réseaux de neurones, comme son nom l'indique, est une méthode qui s'inspire du fonctionnement du cerveau humain. Pour cela, un réseau est constitué généralement d'un nombre de cellules ou de neurones (nodes) interconnectées. De nombreuses variantes sont définies dans la littérature, selon le choix de la cellule élémentaire, de l'architecture du réseau et de la dynamique du réseau, etc. Une cellule élémentaire peut manipuler des valeurs soit binaires ou réelles et différentes fonctions peuvent être utilisées dans le calcul des sorties du réseau de neurones. Plusieurs modèles de réseaux de neurones ont été proposés, tels que le modèle de Hopfield, le modèle du Perceptron multi-couches (PMC) qui est surtout connu par l'appellation Multi layer perceptron (MLP), etc.

La structure MLP dans les réseaux de neurones est la structure la plus utilisée durant ces dernières années, pour laquelle de nombreux algorithmes d'apprentissage ont été développés. Les réseaux MLP sont des modèles non linéaires composés d'un nombre de nœuds organisés en plusieurs couches. Ces réseaux comptent suffisamment de nœuds cachés (présentés dans des couches cachées) afin de gérer les quantités de données à classifier. Il a été rapporté par E. Alpaydin dans son célèbre livre [Alpaydin, 2014], que les réseaux MLP peuvent faire une approximation de n'importe quelle fonction à une certaine précision souhaitée. En d'autres termes, les réseaux MLP sont considérés comme des fonctions d'approximations universelles [Alpaydin, 2014], [Duda, 2012].

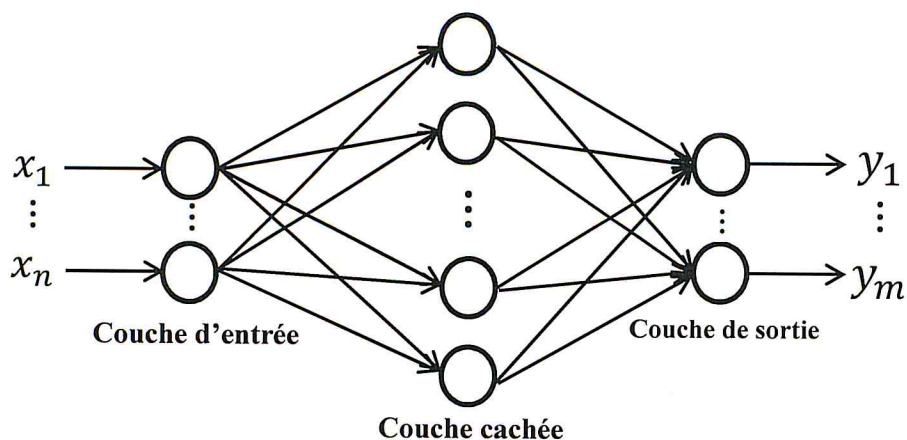


Figure 2.1 Exemple d'architecture globale d'un réseau de neurones.

En règle générale, un réseau MLP nécessite trois couches, ou plus, de nœuds de traitement, à savoir: une couche d'entrée qui reçoit les variables d'entrée (dans le cas présent, ce sont les caractéristiques extraites dans la phase précédente), une ou plusieurs couches cachées et une couche de sortie (en attribuant souvent chaque nœud de la couche de sortie à une classe).

Le principe du fonctionnement d'un réseau de neurones est que lorsque les données (ou les caractéristiques) sont présentées à la couche d'entrée ( $x_i$ ), les nœuds de réseau effectuent des calculs dans les couches successives jusqu'à ce qu'une valeur de sortie soit calculée à chacun des nœuds de sortie ( $y_i$ ). Ce signal de sortie devrait en principe fournir la classe appropriée pour les données d'entrées, c'est-à-dire que nous prévoyons avoir une valeur de sortie élevée sur le nœud de la classe correspondante et des faibles valeurs de sortie sur les autres nœuds (qui correspondent au reste des classes). Pour parvenir à cette affectation correcte, le réseau de neurones (MLP) doit d'abord passer par une phase d'apprentissage supervisée en utilisant l'algorithme itératif de rétro-propagation.

Dans cette phase d'apprentissage, un ensemble d'échantillons de vecteurs caractéristiques, appelés ensembles d'entraînement est présenté au réseau de neurone, où chaque échantillon d'entraînement est lui affecté la sortie désirée correspondante (situé au nœud correspondant).

Par la suite, le processus commence par une affectation des poids. Les valeurs initiales des poids sont de faibles valeurs et surtout aléatoires. La phase cruciale durant l'entraînement d'un réseau de neurones consiste à fixer les poids et cela en fonction des échantillons d'entraînements et leurs résultats désirés. L'ajustement de ces poids se fait itérativement par un processus appelé rétro-propagation. Le principe consiste à ramener au maximum la valeur de sortie du réseau de neurones (obtenue) vers la valeur désirée, en modifiant les différents paramètres du réseau (dans le cas présent, c'est les poids). Pour cela, une fonction d'erreur E



entre les sorties obtenue et désirée est définie; où les poids sont ajustés afin de minimiser cette mesure d'erreur:

$$E(w_{ij}) = 1/2 \sum_{i=1}^m (Y_{désirée} - Y_{obtenue})^2 \quad (2.1)$$

La règle d'apprentissage de la rétrogradation est généralement basée sur la descente en gradient:

$$\Delta w_{ij} = -\rho \frac{\partial E}{\partial w_{ij}} \quad (2.2)$$

Où  $\rho$  est un facteur d'apprentissage qui indique la taille relative de la variation des poids. Les poids sont par la suite modifiés dans la direction qui réduira l'erreur:

$$w_{ij}(t + 1) = w_{ij}(t) + \Delta w_{ij}(t) \quad (2.3)$$

Où  $t$  représente le nombre d'itération. La pertinence de l'algorithme de rétro-propagation réside dans la simplicité de fonctionnement du système itératif. Il est à noter que le cas idéal correspondant à une erreur nulle est quasi impossible dans les cas réels, d'où la définition d'un critère d'arrêt, appliqué soit sur l'erreur, ou sur le nombre d'itération.

### 2.3.2 SVM

L'introduction des SVMs comme méthode de classification avait pour but de résoudre les problèmes de discrimination à deux classes. En réalité, il est difficile de se concentrer uniquement sur les problèmes de classifications bi-classes. Beaucoup de problématiques réelles présentent une multitude de classes telles que la classification de thématique de la couverture terrestre. Pour cela, plusieurs méthodes ont été suggérées pour étendre l'application des SVM aux problèmes de discrimination à plus de deux classes. Nous nous concentrerons dans un premier lieu sur le principe de classification des SVMs à deux classes. Supposant un vecteur caractéristique (représentant un pixel donné)  $\mathbf{X}$  de dimension  $\mathbf{n}$  qui est en même temps l'entrée du classifieur SVM. La classe à laquelle appartient l'individu est représentée par  $\mathbf{y} \in \{-1, 1\}$ ; où une des classes possible est représentée soit par -1 soit par 1.

Comme toute méthode de classification supervisée, les SVMs utilisent un ensemble de données d'apprentissage pour lesquelles les classes sont déjà connues et s'en servent pour construire les règles de décision. Supposons que nous disposons d'un ensemble d'apprentissage de  $l$  données de la forme  $(x_i; y_i) \in \mathbb{R}^n \times \{-1, 1\}$  ( $i = 1, \dots, l$ ). L'idée la plus basic des SVMs dans la classification est d'utiliser un hyperplan de séparation afin de distinguer les échantillons

d'entraînement des différentes classes. Pour le cas des données linéairement séparables, cela signifie qu'il existe un hyperplan dans  $R^n$  tel que toutes les données appartenant à la classe 1 et -1 se retrouvent de part et d'autre de l'hyperplan vérifiant l'équation:  $w \cdot x + b = 0$ ; les données appartenant à la classe +1 doivent vérifier l'inégalité  $w \cdot x + b > 0$ , alors que les échantillons appartenant à la classe -1 doivent vérifier l'inégalité  $w \cdot x + b < 0$ .

Avec  $w = (w_1, \dots, w_n) \in R^n$  le vecteur des coefficients de l'hyperplan et  $b \in R$  un scalaire appelé le biais. La définition de l'hyperplan optimale, revient à résoudre la fonction indicatrice suivante:

$$y = \text{classe}(x) \quad (2.4)$$

$$y = \text{signe}(w \cdot x + b) \quad (2.5)$$

$$\text{signe}(w \cdot x + b) = \begin{cases} -1 & \text{Si } w \cdot x + b < 0 \\ 0 & \text{Si } w \cdot x + b = 0 \\ +1 & \text{Si } w \cdot x + b > 0 \end{cases} \quad (2.6)$$

Cependant, dans le cas des données linéairement séparables, la résolution de la fonction indicatrice dans la séparation des deux classes peut avoir une infinité d'hyperplans. L'idée des SVMs est de choisir le meilleur hyperplan de séparation, en vérifiant la condition de la marge de séparation. Cette marge  $M$  est définie par rapport à la distance euclidienne  $\text{dist}$  entre l'échantillon  $x$  et l'hyperplan:

$$M = \min\{\text{dist}(x_i, w, b) : i = 1, \dots, l\} \quad (2.7)$$

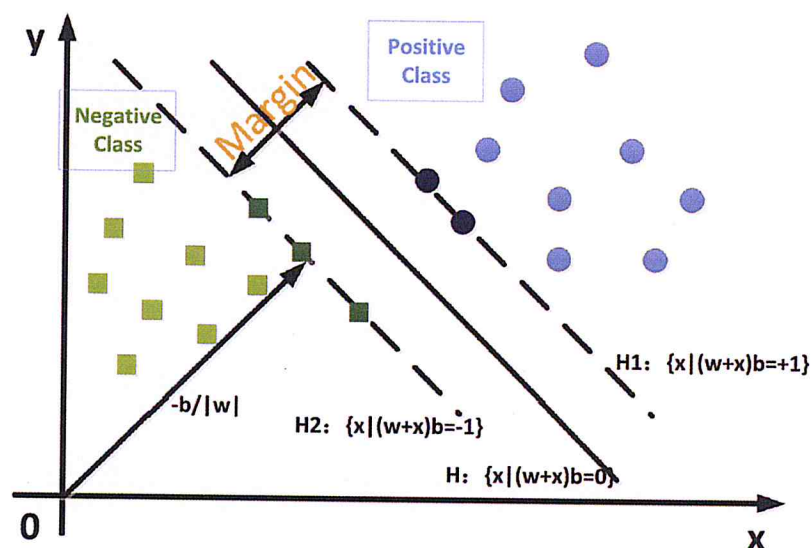


Figure 2.2 Schéma de l'algorithme de définition des hyperplans de séparation.

Un hyperplan optimal séparant les données doit s'écrire sous la forme canonique, ce qui va diminuer le nombre d'hyperplans (infinité d'hyperplan qui vérifient la condition de séparation). En considérant l'hyperplan canonique, l'hyperplan séparant les données devient unique correspondant à une marge maximale. Autrement dit, le vecteur des coefficients de l'hyperplan sera exprimé comme étant le point qui minimise une fonction strictement convexe (rappelons que les fonctions strictement convexes n'ont qu'un unique minimum global). Pour cela, l'hyperplan optimal doit vérifier la condition suivante:

$$\min\{dist(w \cdot x_i + b) : i = 1, \dots, l\} = 1 \quad (2.8)$$

Les seuls points où  $y_i(w \cdot x_i + b) = 1$  sont ceux qui sont sur les deux droites de la marge et correspondent à la distance maximal (c'est les éléments les plus proches, mais de classes différentes). Ces points sont appelés les vecteurs supports (voir figure 2.2).

Cependant, il existe de nombreux cas de données (tel que les caractéristiques extraites pour la classification de thématique) pour lesquels il est impossible de les séparer avec un hyperplan. Pour remédier à ce problème, les SVMs appliquent une transformation sur les données de sorte qu'une fois transformées, elles soient linéairement ou quasi linéairement séparables. L'espace où se trouvent les données avant d'être transformées est appelé l'espace d'entrée (input space), alors qu'après transformation, les données se trouvent dans ce qu'on appelle l'espace de redescription (ou feature space). Il suffit alors de trouver l'hyperplan dans l'espace de redescription qui sépare le mieux ces données transformées. La fonction de transformation est définie comme suit:

$$\begin{aligned} \phi: R^n &\rightarrow R^r \\ x &\rightarrow \phi(x) \end{aligned} \quad (2.9)$$

Où  $r$  et  $n$  représentent les dimensions des espaces de redescription et d'entrée respectivement, avec  $r > n$ , ce qui signifie que la transformation est une projection dans un espace de dimension supérieure afin de pouvoir les séparer linéairement.

Afin de trouver l'hyperplan séparateur optimal, on procède de la même manière que précédemment (le cas linéairement séparable section 3.2) et cela revient à résoudre l'équation suivante:

$$y_i(w \cdot \phi(x) + b) = 1 \quad (2.10)$$

La fonction indicatrice associée à ce problème est alors définie comme suit:

$$\text{Classe}(x) = \text{signe}\left(\sum_{i=1}^l (y_i \phi(x_i) \phi(x)) + b\right) \quad (2.11)$$

Si la transformation utilisée est appropriée, la résolution du problème permet de trouver un séparateur non linéaire avec la marge la plus grande possible, en utilisant ainsi les vecteurs du support dans le cas des données non-linéairement séparables. Toutefois, l'utilisation des transformations pose certains problèmes car les calculs se font avec les données transformées, c'est-à-dire dans l'espace de redescription. Or, comme la dimension de cet espace est bien souvent beaucoup plus grande que celle de l'espace d'entrée, ces calculs peuvent devenir extrêmement longs et difficile à effectuer. Ce qui amène à l'utilisation des fonctions noyaux  $K$  (kernels):

$$\begin{aligned} K: R^n \times R^n &\rightarrow R \\ (x_i, x_j) &\rightarrow \phi(x_i) \phi(x_j) \end{aligned} \quad (2.12)$$

Le principe de base de l'utilisation des noyaux est de prendre en entrée deux points (représentant deux échantillons d'entraînement) dans l'espace d'entrée et calculer le produit scalaire dans l'espace de redescription. L'avantage des noyaux est que le calcul peut se faire directement à partir des données de l'espace d'entrée. Grâce au concept de noyau, la fonction indicatrice peut alors s'écrire de la manière suivante:

$$\text{Classe}(x) = \text{signe}\left(\sum_{i=1}^l (y_i K(x_i, x)) + b\right) \quad (2.13)$$

Lorsque la fonction noyau est connue, la transformation  $\phi(x)$  n'apparaît nulle part, il n'est donc pas nécessaire d'effectuer la transformation sur les données dans le calcul ni se préoccuper de la dimension de l'espace de redescription.

Plusieurs types de noyaux sont utilisés, cela dépend essentiellement de l'application et des données. Les noyaux les plus fréquemment utilisés sont:

- Linéaire:  $K(x_i, x_j) = (x_i \cdot x_j)$
- Polynomial:  $K(x_i, x_j) = (x_i \cdot x_j + 1)^d$
- Gaussien (Radial Basis Function RBF):  $K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$
- Multi-quadratique inverse:  $K(x_i, x_j) = \frac{1}{\sqrt{(x_i \cdot x_j)(x_i \cdot x_j) + \beta}}$

Nous avons présenté jusqu'à présent le principe de séparation des SVMs entre deux classes. Il est à noter qu'initialement les SVMs ont été conçues pour résoudre des problèmes binaires ou bi-classes. Cependant, les problèmes du monde réel exigent souvent la discrimination entre

plusieurs classes. Pour remédier à cette limitation, les SVMs décomposent les problèmes de classification multi-classes en une série de problèmes bi-classe, de sorte que les SVM standard puissent être directement appliquées. Deux approches de classification SVM multi-classes sont connues et utilisées, à savoir l'approche "un contre tous" ou "un contre un".

### 2.3.3 Adaboost

En plus des réseaux de neurones et les SVMs, nous avons appliqué aussi une méthode de classification en cascade à savoir l'algorithme adaboost (en utilisant les arbres de décision comme méthodes basiques de classification ou faibles classifieurs). La classification adaboost est une classification supervisée, où les caractéristiques sont nécessaires durant la phase d'entraînement.

Il faut bien noter que la classification adaboost est très différente de celle basée sur des règles; une phase d'apprentissage est appliquée pour chaque classifieur faible. La performance d'un classifieur faible est légèrement meilleure que celle obtenue par une classification aléatoire (correspondante à une performance de 0.5). En d'autres termes, l'algorithme adaboost essaie de générer des classifieurs complémentaires en entraînant chaque nouveau faible classifieur **uniquement sur les erreurs commises par le faible classifieur précédent**. D'où le but principal de cette classification est de combiner plusieurs classifieurs faibles (weak classifiers) afin de construire au final un classifieur puissant. Les classifieurs faibles sont généralement sélectionnés en fonction de leur temps ou espace de complexité [Bourges, 2012]. Une valeur de **poids  $w_i$  est alors incrémenté** pour chaque échantillon **mal classé**, afin d'augmenter son influence durant la prochaine étape d'apprentissage. Le choix des faibles classifieurs  $c_i$  est basé sur la minimisation du taux d'erreurs de classification enregistré  $\epsilon_i$ . Le résultat d'un faible classifieur doit être meilleur de celui obtenu par une classification aléatoire [Wang, 2004].

Les principales étapes de classification par l'algorithme AdaBoost sont résumées comme suit:

**En entrée:** soit un ensemble d'apprentissage  $S = ((x_1, y_1), \dots, (x_m, y_m))$ , où  $x_i$  et  $y_i$  représente les échantillons de pixels dans l'image et leurs classes correspondantes;  $m$  représente le nombre d'échantillons d'entraînement.

**Initialisation:** dans cette étape l'algorithme commence par:

- Définir le nombre d'itérations  $T$ ;

- Initialiser tous les poids des échantillons de la première itération  $w_1(i) = \frac{1}{m}$ .

**Pour chaque itération ( $t = 1, \dots, T$ ):**

- L'algorithme sélectionne un faible classifieur  $C_t: R^d \rightarrow \{-1, +1\}$  et effectue une classification suivant la distribution des poids  $w_i$ .
- Calculer par la suite l'erreur de classification à chaque itération,

$$\varepsilon_t = \sum_{t=1}^N w_t(i) [Y_t \neq h_j(x)], \text{ et } C_t(x) = \text{Argmin } \varepsilon_t \quad (2.14)$$

- Où tout classifieur  $C_t$  correspondant à une erreur  $\varepsilon_t > 0,5$  sera rejeté (le seuil de confiance est fixé à 0,5), ce qui rend le classifieur meilleur par rapport à une classification au hasard (correspondante à une erreur de 0.5).
- Augmenter les poids correspondant aux échantillons mal classés:

$$w_{t+1}(i) = \frac{w_t(i) \exp(-\alpha_t Y_t h_t(x_t))}{Z_t} \quad (2.15)$$

Où  $Z_t$  représente le facteur de normalisation:

$$Z_t = \sum_{i=1}^m w_t(i) \exp(-\alpha_t Y_t h_t(x_t)) \quad (2.16)$$

$$\text{et } \alpha_t = \frac{1}{2} \left( \ln \frac{1-\varepsilon_j}{\varepsilon_j} \right) \quad (2.17)$$

**En sortie:** l'algorithme combine les faibles classifieurs obtenus à chaque itération afin de construire le classifieur final  $H_t$ . Avec:

$$H_T(x) = \text{sign}(\sum_{t=1}^T \alpha_i \times h_t(x)) \quad (2.18)$$

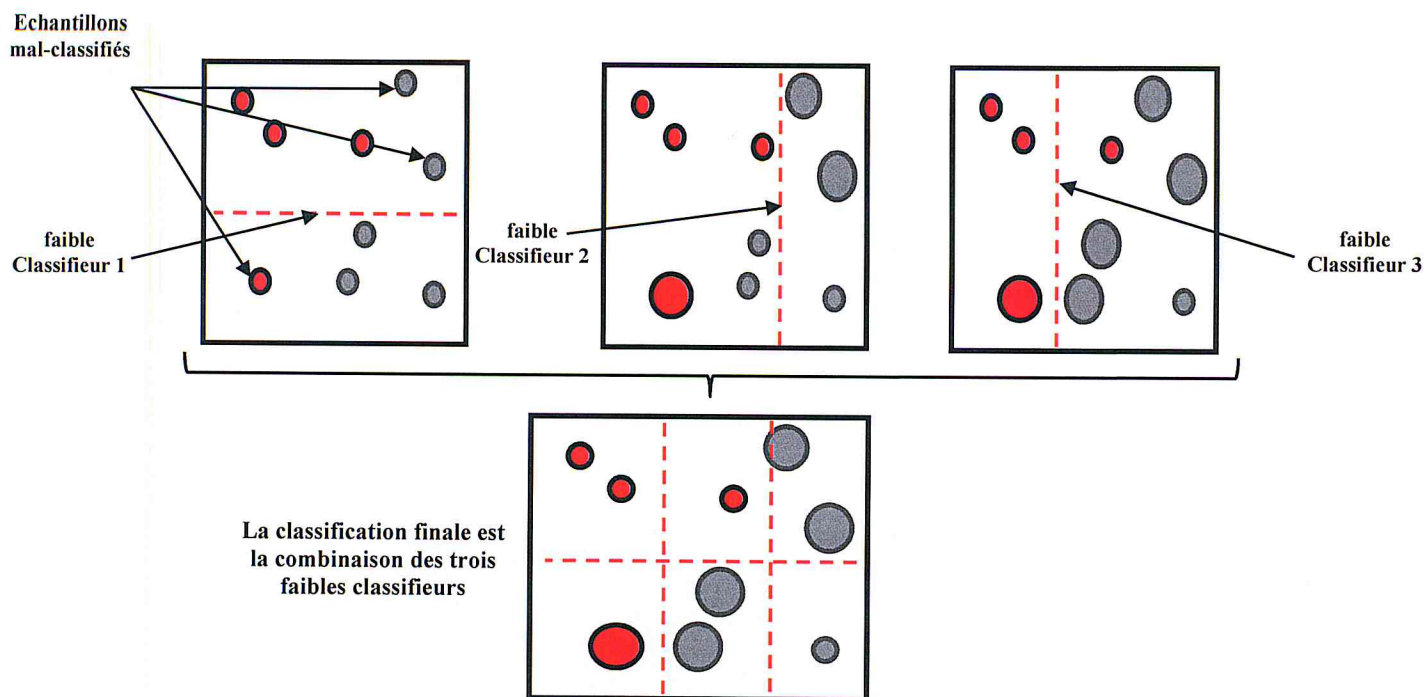


Figure 2.3 Exemple de classification AdaBoost

La figure 2.3 illustre un exemple de classification adaboost, où le processus est répété trois fois correspondant à  $T = 3$ . On remarque que, les poids sont optimisés itérativement par rapport aux échantillons mal classés et sont affectés de poids plus importants dans l'itération suivante. En plus, le faible classifieur est appliqué uniquement sur les échantillons mal-classifiés de l'itération précédente. Au final, on peut constater que les échantillons des deux classes sont correctement séparés.

Dans la procédure AdaBoost, de nombreux classifieurs faibles peuvent être combinés afin d'obtenir le classifieur final. Malgré l'utilisation des classifieurs faibles, la classification AdaBoost peut atteindre des performances élevées avec moins de paramètres à régler par rapport à d'autres méthodes de classification puissantes tels que les réseaux de neurones et les SVMs. L'implémentation d'un classifieur AdaBoost nécessite uniquement: (i) un classifieur faible; et (ii) le nombre de boosts utilisés durant la phase d'entraînement.

### 2.3.4 Le plus proche voisin

La classification par le plus proche voisin (kNN) est l'une des méthodes de classification les plus simples. Cet algorithme est recommandé lorsqu'il y a peu de connaissances préalables sur la distribution des données à classifier. La classification par les K plus proches voisins est principalement caractérisée par deux paramètres à savoir le nombre des voisins proches K et la distance utilisée comme mesure de similarité entre les échantillons et le pixel à classifier [Altman, 1992].

### 2.3.5 Le classifieur Bayésien

La classification par le théorème de Bayes est une méthode simple basée sur la maximisation de la probabilité d'appartenance du pixel à la classe. Le principe général de cette classification est basé sur la formule de Bayes suivante:

$$P(C_k/P(i,j)) = \frac{P(C_k) P(P(i,j)/C_k)}{P(P(i,j))} \quad (2.19)$$

Etant  $p(i,j)$  le pixel à classifier et  $C_k$  est une classe thématique.

La principale limitation de cet algorithme est le calcul de la probabilité conditionnelle  $P(P(i,j)/C_k)$ , ce qui a engendré l'hypothèse d'indépendance conditionnelle des probabilités. Cela ramène le calcul du terme à l'équation suivante:

$$P(C_k/P(i,j)) = \text{Argmax } P(C_k) (\prod P(P(i,j) / C_k)) \quad (2.20)$$

Dans notre travail, les classifieurs KNN et bayésien ont été surtout utilisés dans la phase de comparaison afin d'évaluer les résultats obtenus des différentes méthodes de classification.

## 2.4 CONCLUSION

Dans ce chapitre, nous avons présenté la phase de classification thématique des images de télédétection. Nous avons aussi expliqué le principe de base de la classification en général, ainsi que le principe de fonctionnement des différents algorithmes de classification utilisés dans ce travail, à savoir les réseaux de neurones, les Séparateurs à Vaste Marge, Adaboost, les plus proches voisins et le classifieur de Bayes. Comme nous l'avons déjà évoqué ces algorithmes sont appliqués pour les deux modes d'application, à savoir la classification thématique et la détection de changements. Le chapitre suivant est consacré à l'analyse et la conception du système proposé dans ce travail.



## **CHAPITRE 3: ANALYSE ET CONCEPTION**

## 3.1 INTRODUCTION

L'objectif de ce chapitre est de présenter la phase analyse et conception du système développé pour la classification thématique et la détection de changement des images satellitaires multi-dates. Pour cela, plusieurs algorithmes de machines d'apprentissage ont été utilisés.

Ce chapitre est divisé en deux principales sections. Dans un premier temps, nous procéderons à une brève analyse des logiciels et systèmes existants, Cette partie consiste à comprendre le contexte du système afin de distinguer les spécifications fonctionnelles et non fonctionnelles. Dans un second temps, nous proposerons une étude conceptuelle de système proposé.

## 3.2 ANALYSE

L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences de l'utilisateur. Il s'agit de livrer des spécifications pour permettre de concevoir le système.

### 3.2.1 PRESENTATION DES LOGICIELS UTILISES

Plusieurs logiciels ont été développés et commercialisés spécifiquement pour le traitement, le prétraitement, l'analyse et enfin la classification des images de télédétection.

La plupart des systèmes existants s'appuient sur les méthodes classiques de classification des images de télédétection, telle que la classification par minimisation de distance, la classification par Maximum de Vraisemblance, la classification par seuil, la méthode de classification dite «objet» (où elle nécessite une étape supplémentaire de segmentation afin de définir les objets). Cependant, la plupart de ces logiciels n'englobent pas les techniques de machine d'apprentissage à l'exception des dernières versions (**eCognition developer**) où quelques algorithmes tels que Machines à vecteur de support et le Réseau de neurones artificiel ont été rajoutés.

On peut citer quelques exemples de ces logiciels:

- **ENVI** (Environment for Visualizing Images): est un logiciel professionnel de la société «EXELIS» permettant la visualisation, le traitement, l'analyse, et la présentation de nombreux types d'images numériques, dont les images satellites.

En particulier, **ENVI** permet de travailler sur différents types de données (multi spectrale, hyper spectrale, radar), d'intégrer des données de type matriciel (raster) et vectoriel. Il permet

entre autres de contraster les images, de les corriger géométriquement, de les classifier et de réaliser des analyses à l'aide de données d'élévations, etc.

**ENVI** est associé aussi avec un langage de programmation **IDL** (Interactive Data Language).

- **eCognition**: est un logiciel commercial pour l'approche orientée objet a été lancé en 2003, dédié à l'interprétation d'images avec une approche cognitive par l'utilisation d'information non seulement sur les pixels mais aussi sur les objets. Cette approche "orientée objets" permet d'intégrer dans la procédure d'analyse d'image des informations spectrales, spatiales (taille, position des objets, ...) et relationnelles (position d'un objet par rapport à une classe).

Bien que ces logiciels soit accessible et facile à utiliser par les utilisateurs qui ne maîtrisent pas la programmation, il reste qu'ils n'incluent pas tous les types d'algorithmes de machines d'apprentissage et qu'ils imposent des restrictions et des limitations (légales, fonctionnelles, matérielles). De plus, cela ne permet pas l'adaptation aux propres besoins de l'utilisateur, ou même le rajout d'autres fonctionnalités car le code source n'est pas publié. Certains utilisateurs se plaignent aussi d'un manque d'ergonomie.

### 3.2.2 PRESENTATION DU SYSTEME EXISTANT

Un système a été développé avec le langage de programmation **Matlab**. Ce système est basé sur l'utilisation des techniques de machine d'apprentissage pour réaliser la classification thématique et la détection du changement.

- **Matlab**: est un langage de programmation de haut niveau destiné au calcul scientifique. On le trouve dans divers types d'applications, la modélisation, la simulation, l'analyse et la visualisation de données, création d'application avec interfaces utilisateurs. Il existe un grand nombre de toolboxes, «boîtes à outils» étendant les fonctionnalités de base de Matlab à un certain type de problème. On trouve ainsi des toolboxes dans les domaines du traitement du signal, de la commande des systèmes, de la logique floue, de la simulation, etc...

Pour réaliser des systèmes de classification d'images de télédétection avec **Matlab**, il faut faire appel aux fonctionnalités de la Statistics and Machine Learning Toolbox qui fournit des fonctions et des applications pour décrire, analyser et modéliser les données. Les algorithmes de régression et de classification permettent de tirer des inférences à partir des données et de construire des modèles prédictifs.

Après analyse du système existant, implémenté selon le paradigme de programmation procédurale avec le langage de programmation **Matlab**. Ce programme est décomposé en modules, avec des suites d'instruction simple.

Nous avons essayé d'élaborer un diagramme de cas d'utilisation et un «semblant» diagramme de classe pour cette partie d'Analyse:

### 3.2.2.1 Diagramme de cas d'utilisation du système existant

Le système existant permet à l'utilisateur d'effectuer de la classification thématique et la détection du changement à l'aide des algorithmes de machine d'apprentissage, et cela sans une interface graphique qui facilite l'interaction de l'utilisateur avec le système et sans gestion de la persistance des données qui assure la sauvegarde et la récupération des données (données extraites des images satellitaires, modèles prédictifs) sans avoir à refaire les mêmes traitements à chaque fois.

Nous proposons ci-dessous le diagramme de cas d'utilisation de ce système:

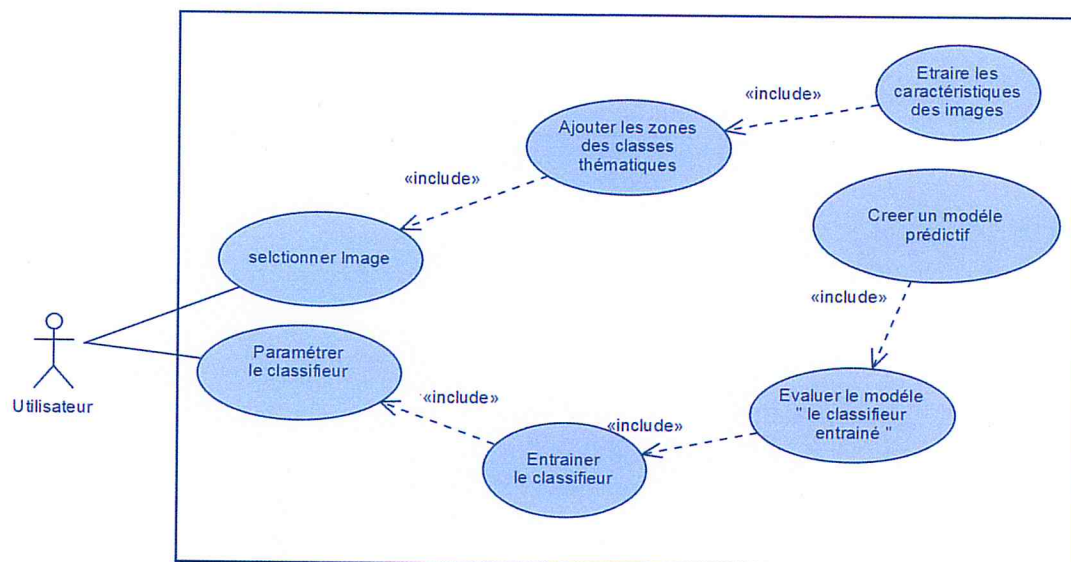


Figure 3.1 Diagramme de cas d'utilisation du système existant.

### 3.2.2.2 Diagramme de classes du système existant

En analysant la structure du code du système existant, un semblant diagramme de classe «Figure 3.2» est élaboré afin de mieux assimiler l'organisation de ses données, ainsi que les liens entre elles.

On peut constater, d'après ce diagramme, que la maintenance et la réutilisation des fonctions de ce programme est en fait très compliquée.

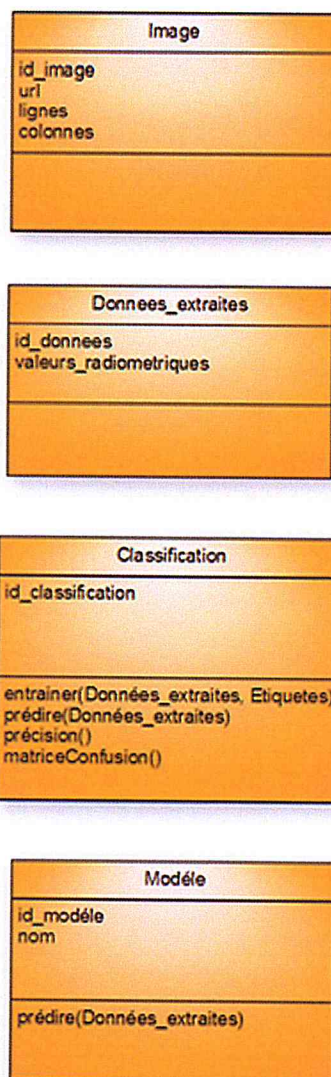


Figure 3.2 Diagramme de classe du système existant.

Cependant, nous pensons que **Matlab** a quelques lacunes fondamentales. La plupart d'entre elles découlent de sa nature commerciale:

Les algorithmes sont propriétaires, ce qui signifie que nous ne pouvons pas voir le code de certain de ces algorithmes que nous utilisons et que nous devons nous assurer qu'ils ont été implémentés correctement.

**Matlab** est assez cher, ce qui signifie que le code écrit en **Matlab** ne peut être utilisé que par des personnes disposant de fonds suffisants pour acheter une licence.

Naturellement, le **Mathworks** met des restrictions sur la portabilité du code, la possibilité d'exécuter le code sur un ordinateur de quelqu'un d'autre. Nous pouvons exécuter notre application "compilée" en utilisant **Matlab Component Runtime (MCR)**, mais notre application portable doit correspondre exactement à la version du MCR installé, ce qui peut être gênant étant donné que **Mathworks** publie une nouvelle version tous les **6 mois**.

Comparé à d'autre langage de programmation open source tel que **python** ou **java**, l'ajout des produits tiers reste difficile étant donné le caractère propriétaire de **Matlab**.

Vu le besoin spécifique des utilisateurs par rapport à l'application, l'utilisation d'un logiciel vaste (**Matlab**) ou trop orienté (**ENVIE**) reste inappropriée. Pour cela, l'objectif de notre travail est de concevoir et développer une solution mieux adaptée aux exigences des utilisateurs par l'utilisation des algorithmes de machines d'apprentissage dans la classification thématique et la détection des changements.

### 3.2.3 ÉTUDE PRELIMINAIRE

Nous rappelons que le projet réalisé dans ce travail est destiné à la communauté scientifique (étudiants, chercheurs, etc.) ainsi qu'aux spécialistes du domaine de télédétection tels que les géodésiens, les analystes etc. Dans cette section, nous nous intéressons aux besoins des utilisateurs à travers les spécifications fonctionnelles et non fonctionnelles pour aboutir à un système de qualité qui répond aux problématiques existantes.

#### 3.2.3.1 Besoins fonctionnels

L'objectif principal du système proposé est de permettre à l'utilisateur ou l'analyste de manipuler et d'analyser les données de télédétection. Le système offre les fonctionnalités suivantes:

- a. Chargement, affichage et manipulation des images satellitaires;
- b. Choix de mode d'utilisation «classification thématique où détection de changement»;
- c. Interaction avec l'image affichée pour pouvoir sélectionner les classes thématiques;
- d. Choix entre plusieurs méthodes de classification «différents algorithmes de machine d'apprentissage utilisés pour la classification et la détection des changements» ainsi que l'ajustement des paramètres des classifieurs;
- e. Sélection des échantillons d'entraînement des classifieurs pour l'ensemble des classes;
- f. Sauvegarde du modèle »classifieur entraîné» pour éviter l'étape d'apprentissage à chaque utilisation;
- g. Sélection d'un modèle parmi ceux sauvegardés auparavant. Cela est utilisé soit pour une nouvelle classification «prédiction» ou même pour une évaluation de ses performances sur d'autres données;

- h. Visualisation des résultats de classification thématiques, de détection de changement et des différentes évaluations.

### 3.2.3.2 Besoins non fonctionnels

Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur les performances et sur le rendement du système, ce qui signifie qu'elles ne doivent pas être négligées, pour cela il faut répondre aux exigences suivantes:

- a. **Fiabilité:** L'application doit fonctionner de façon cohérente sans erreurs et doit être satisfaisante;
- b. **Gestion des erreurs:** Les ambiguïtés doivent être signalées par des messages d'erreurs bien organisés et personnalisés pour bien guider l'utilisateur et le familiariser avec le système. Tous les messages d'erreurs ne doivent pas être affichés à l'utilisateur. Certains messages sont propres à l'administrateur, afin d'éliminer le risque de divulguer des informations sur le système;
- c. **Ergonomie de l'interface:** L'interface utilisateur doit permettre à l'analyste d'effectuer les tests appropriés sans se soucier de la programmation, elle doit être adaptée à l'utilisateur sans qu'il ne fournisse aucun effort de point de vue navigation entre les différentes fonctionnalités, couleurs et mise en textes utilisés;
- d. **Sécurité:** le système doit respecter surtout la confidentialité des données des utilisateurs qui reste l'une des contraintes les plus importantes dans les applications web;
- e. **Aptitude à la maintenance et la réutilisation:** Le système doit être conforme à une architecture standard et claire permettant sa maintenance et sa réutilisation;
- f. **Compatibilité et portabilité:** Une application web quel que soit son domaine, son éditeur et son langage de programmation ne peut être fiable qu'avec une compatibilité avec tous les navigateurs web et tous les moyens que ce soit **PC, IPAD** ou mobiles;
- g. **Persistance des données:** La gestion de la persistance des données est un mécanisme responsable de la sauvegarde et de la récupération des données. Ces mécanismes garantissent que le système peut conserver les données extraites des images satellitaires afin d'éviter de les extraire à nouveau pour améliorer les

performances. Ces données peuvent être enregistrées sur disque, ou éventuellement sur un serveur distant;

- h. **Performance:** afin d'être acceptée par l'utilisateur, notre système doit respecter ce critère pour assurer le bon fonctionnement des différents modules avec un temps de réponse optimal.

Dans cette partie nous avons présenté une étude des systèmes existants, les lacunes qu'ils comprennent ainsi que les solutions que nous proposons pour pallier à ces problèmes. Nous avons aussi évoqué les besoins fonctionnels et non fonctionnels qui sont indispensables pour mieux faciliter le travail à réaliser.

### 3.3 CONCEPTION

Dans cette partie nous allons aborder l'étude conceptuelle de notre système, pour apporter plus de détails sur les fonctionnalités du système et clarifier les aspects techniques.

#### 3.3.1 DIAGRAMME DE CAS D'UTILISATION

Pour modéliser la vue fonctionnelle du système, nous nous servons du diagramme de cas d'utilisation. Ce dernier est destiné à représenter les besoins des utilisateurs par rapport au système. [Joseph Gabay, 2004]

Nous proposons ci-dessous les cas d'utilisation de notre système:

- Sélection du mode d'utilisation «classification thématique ou détection de changement»;
- Chargement des images sert à charger et afficher les images satellitaires vers le serveur;
- Sélection des classes thématiques d'entraînement dans le cas où le mode choisi est la classification thématique;
- Choix de type classifieur;
- Paramétrage de classifieur;
- Entraîner le classifieur;
- Sauvegarde des classifieurs entraînés «modèles prédictifs»;
- Extraction et sauvegarde des caractéristiques extraites;



- Sélection du modèle prédictif;
- Evaluation du modèle prédictif «Evaluer»;
- Utilisation du modèle prédictif pour la classification thématique où la détection de changement »Classifier»;
- Affichage des résultats.

D'où la présentation de notre diagramme de cas d'utilisation (Figure. 3.3)

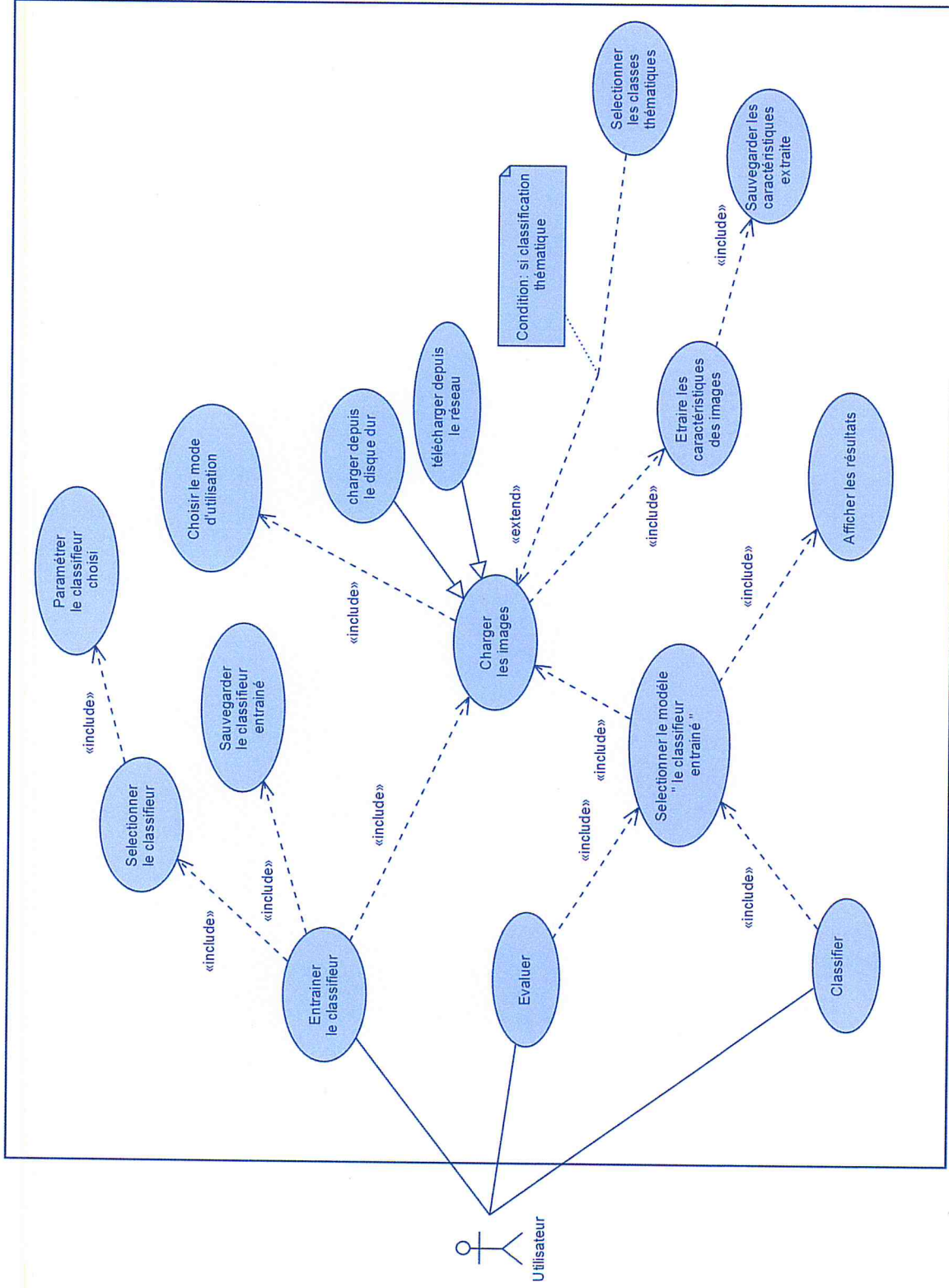


Figure 3.3 Diagramme de cas d'utilisation

### 3.3.2 DIAGRAMME DE CLASSES

Un diagramme de classes est une collection d'éléments de modèle statique, tels que des classes, des interfaces et leurs relations, connectés entre eux comme un graphe [Kettani, 2001]

Il représente la description statique du système en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements. C'est le diagramme pivot de l'ensemble de la modélisation d'un système. [Joseph Gabay, 2004]

Les classes de notre système sont les suivantes:

- **Image:** c'est la classe modélisant l'image numérique;
- **Data\_image:** c'est la classe modélisant les caractéristiques d'une image numérique restituée;
- **Data\_set:** Une instance de cette classe contient les caractéristiques utilisés pour la classification (données radiométriques, mesure d'entropie et mesure de dissimilarité);
- **Training\_set:** Une instance de cette classe contient les caractéristique utilisés pour l'entraînement ou pour l'évaluation;
- **Classifieur:** c'est la classe modélisant un algorithme de machine d'apprentissage donné;
- **Modèle:** Une instance de cette classe représente un modèle prédictif;
- **Evaluation:** c'est la classe responsable sur la phase de l'évaluation d'un modèle prédictif donné;
- **Classification:** c'est la classe responsable du processus de classification thématique ou de détection des changements;
- **Classe\_détection\_changement:** c'est la classe modélisant les étiquettes de détection de changement «Changé, Inchangé»;
- **Classe\_thématique:** c'est la classe modélisant les étiquettes thématiques «Végétation, Urbain, Sols nus, Milieu humides»;
- **Classifieur\_type:** c'est la classe modélisant les types un algorithme de machine d'apprentissage utilisés «Plus proches voisins, Machines à vecteur de support, Réseaux de neurones, Réseau bayésien naïf, Adaboost»;

- **Classification\_type**: c'est la classe modélisant les modes d'utilisation du système «Classification thématique, Détection du changement»;

Afin de représenter une vue globale sur les différentes fonctions du système et les informations et les méthodes réalisées par les acteurs, nous avons modélisé les classes de notre système dans la Figure 3.4.

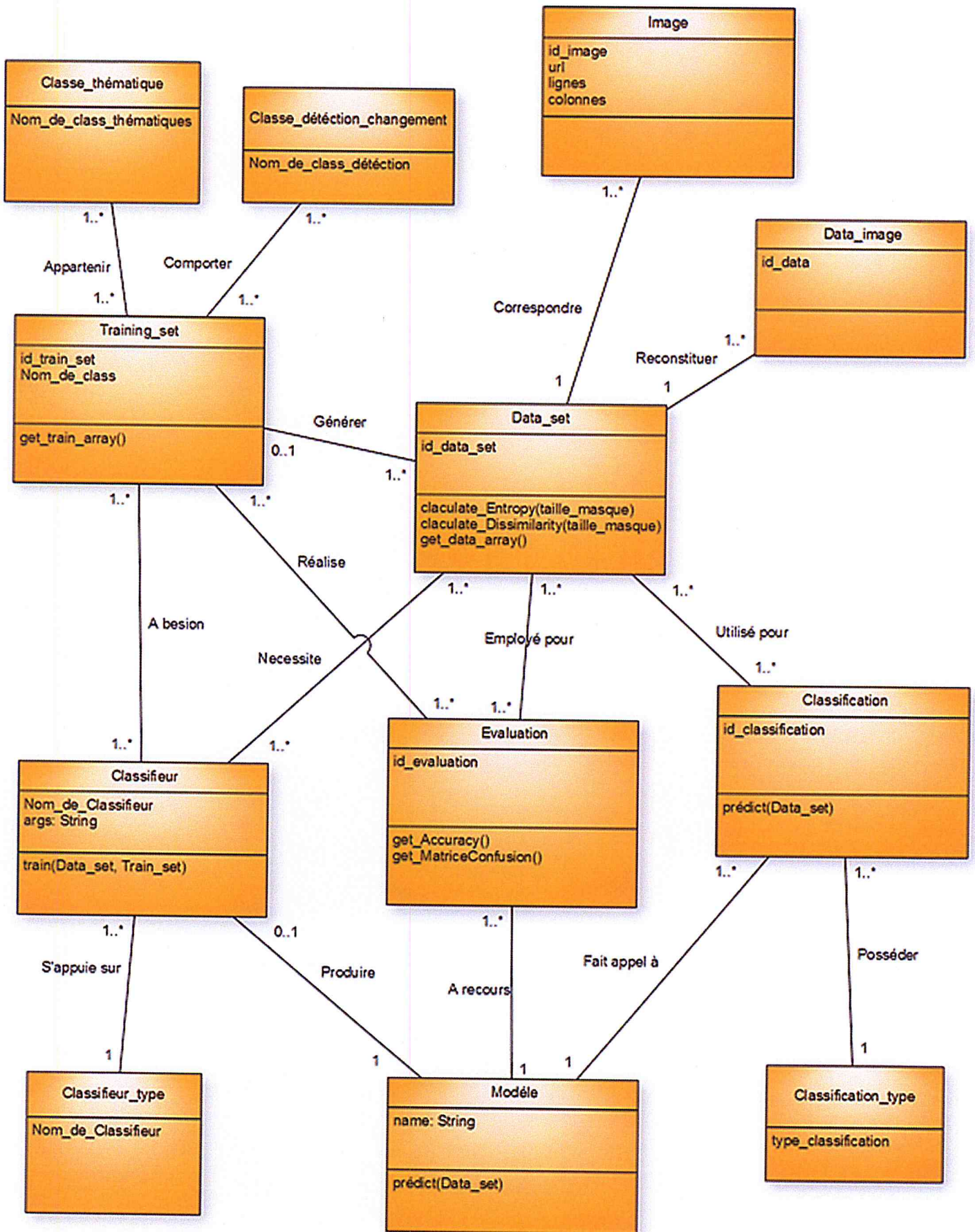


Figure 3.4 Diagramme de classes

### 3.3.3 DIAGRAMME DE SEQUENCE

Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages. Les diagrammes de séquences permettent de décrire comment les éléments du système interagissent entre eux et avec les acteurs: [P. Gérard, 2013]

- Les objets au cœur d'un système interagissent en s'échangeant des messages;
- Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine);

L'objectif des diagrammes de séquences:

- Représenter les interactions entre les objets en précisant la chronologie des échanges des messages;
- Représenter les scénarios possibles d'un cas d'utilisation donné;
- Faire ressortir les acteurs, les objets et les messages.

### 3.3.3.1 Diagramme de séquence du cas d'utilisation «Entraîner le classifieur»

Le diagramme de séquence de la figure 3.5 représente l'enchaînement des actions et des interactions entre l'utilisateur et le système de manière générale pour réaliser la phase d'apprentissage des algorithmes.

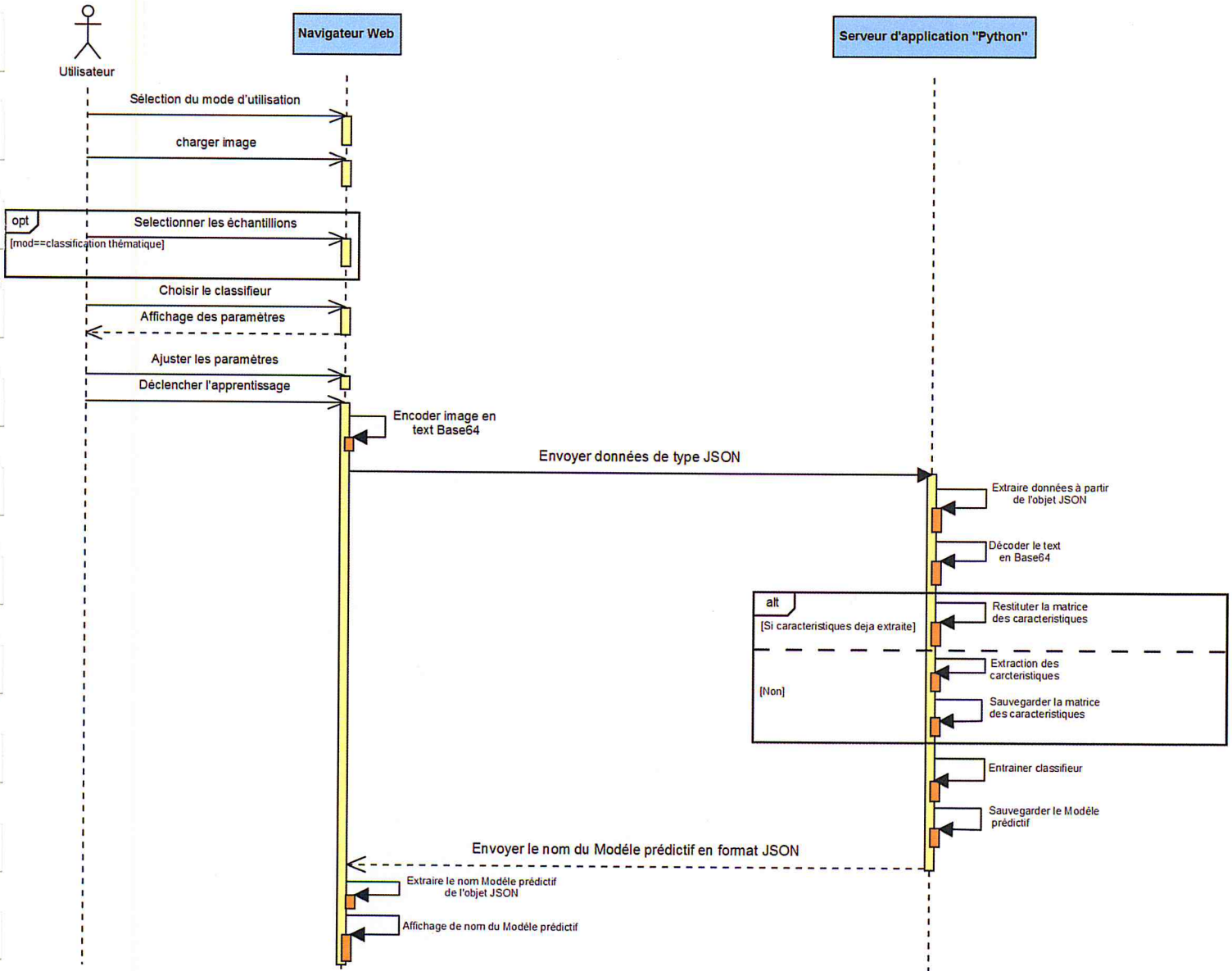


Figure 3.5 Diagramme de séquence du cas d'utilisation «Entraîner le classifieur»

Notre système est donc représenté par le **Navigateur Web** «interface utilisateur» et le **Serveur d'application** «traitements métier».

### 3.3.3.2 Diagramme de séquence du cas d'utilisation «Evaluer le modèle prédictif»

Le diagramme de séquence de la figure 3.6 représente l'enchaînement des actions et des interactions entre l'utilisateur et le système pour réaliser la phase d'évaluation des performances d'un modèle prédictif.

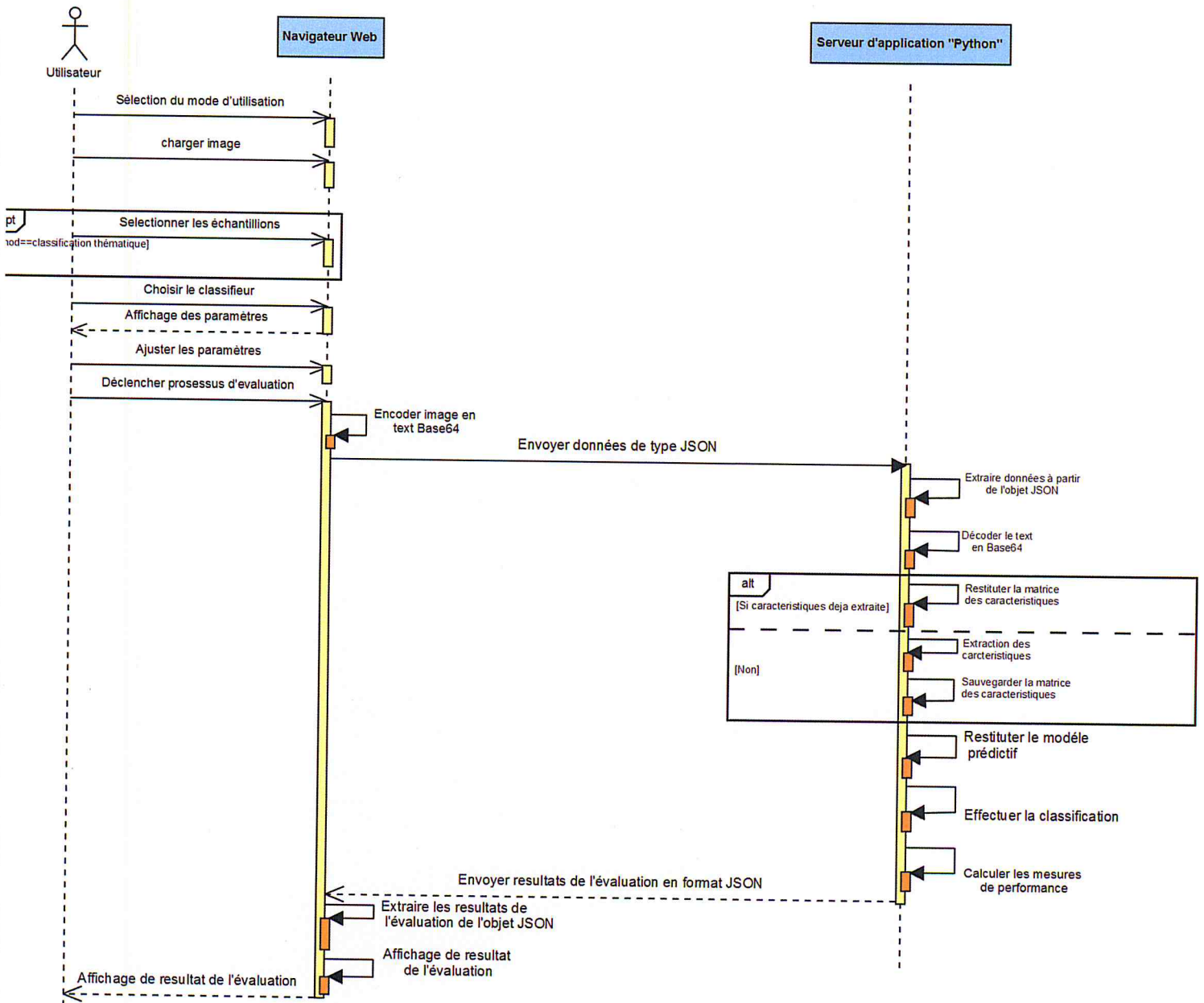


Figure 3.6 Diagramme de séquence du cas d'utilisation «Evaluer le modèle prédictif»



### 3.3.3.3 Diagramme de séquence du cas d'utilisation «Classifier»

Le diagramme de séquence de la figure 3.7 représente l'enchaînement des actions et des interactions entre l'utilisateur et le système pour réaliser la classification thématique ou la détection du changement avec un modèle prédictif sauvegardé.

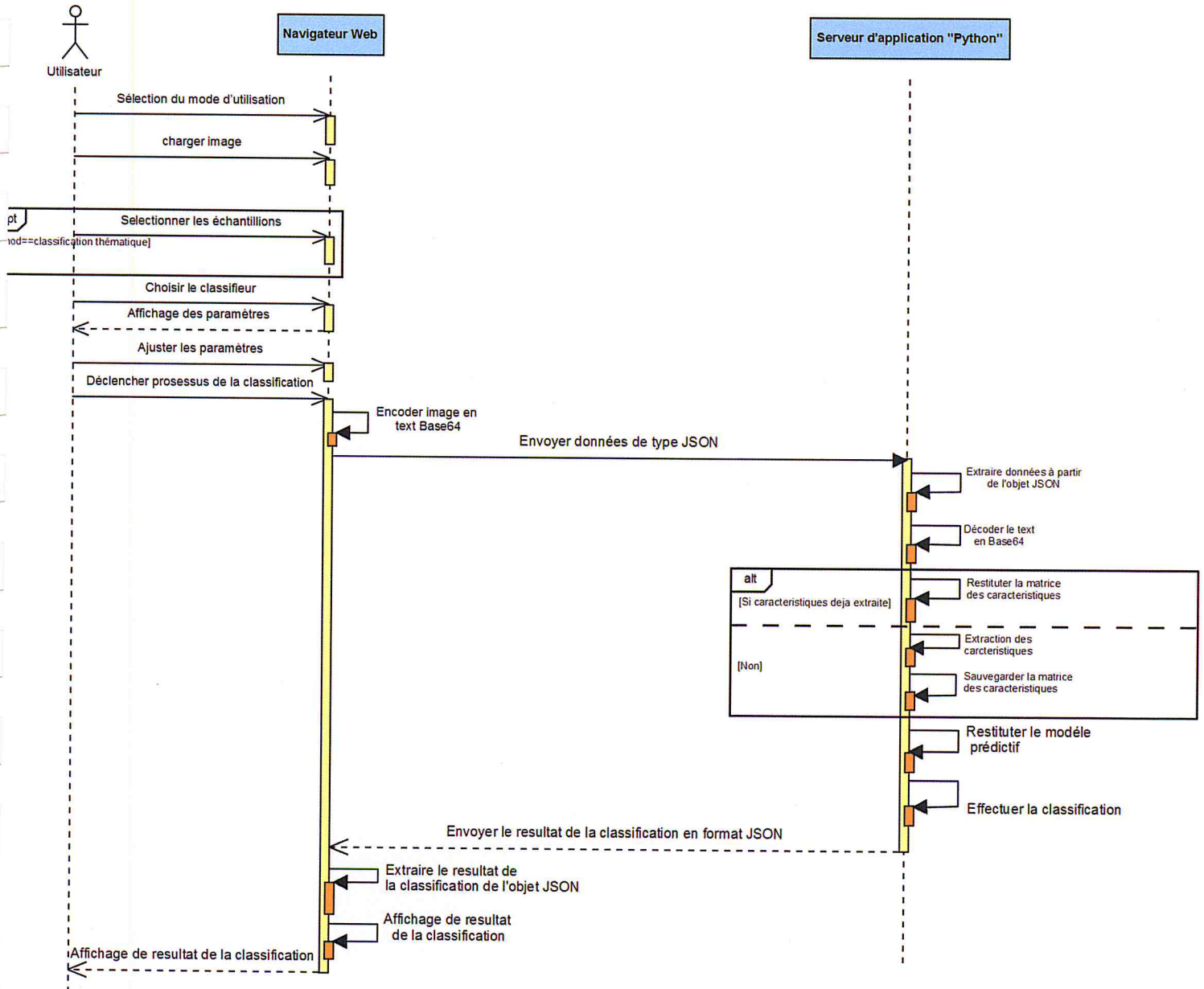


Figure 3.7 Diagramme de séquence du cas d'utilisation «Classifier».

## 3.4 CONCLUSION

Dans ce chapitre, nous avons analysé le système existant et présenté les différents diagrammes développés lors de la conception du projet. Cette étude adoptée contribue à la mise en œuvre de notre système, qui sera abordée dans le prochain chapitre.

## **CHAPITRE 4: REALISATION**

## 4.1 INTRODUCTION

En se basant sur l'étude conceptuelle faite dans le chapitre précédent, nous allons montrer dans ce chapitre les étapes d'implémentation de notre système. Premièrement, nous commencerons par une description de l'architecture du système proposé. Dans un second lieu, nous présenterons les différents langages de programmation et bibliothèques utilisés dans la réalisation de notre application. En troisième partie, nous donnerons une vue d'ensemble sur l'implémentation et la réalisation du système. Nous terminerons cette partie par les interfaces graphiques du système.

## 4.2 ARCHITECTURE DU SYSTEME

L'organisation de l'application suit les principes du développement web, qui implique la création d'un serveur web qui permet le chargement des données de télédétection "images satellitaires" à travers l'interface graphique.

Dans cette architecture, l'élément le plus important est l'utilisation d'un mécanisme de communication entre les deux applications (client-serveur). Ce dialogue est illustré par la Figure 4.1.

- Le client (généralement un navigateur web) qui est chargé de l'affichage des ressources et des données et le dialogue avec l'utilisateur. Ce client demande un service ou une ressource au serveur.
- Le serveur effectue le traitement, fournit la ressource ou renvoie le résultat au client, il effectue le traitement métier des données, correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative, ainsi que l'accès aux données persistantes "correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive".

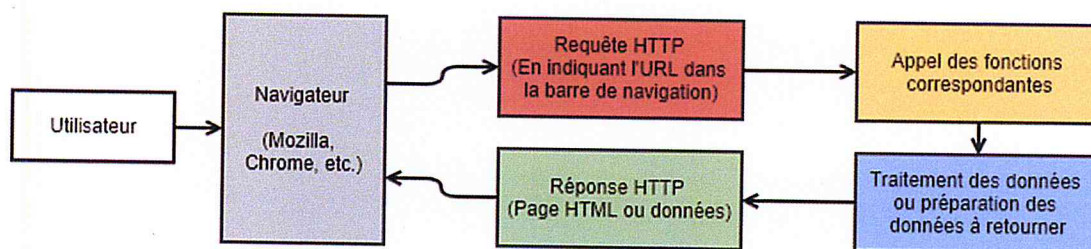


Figure 4.1 Architecture client-serveur

Le fait que l'application est déployée sur le réseau «Internet ou réseau local», l'un des plus grands défis est l'échange des données volumineuse (images satellitaires multi-spectrales), entre le

poste client et le serveur distant surtout que tous les traitements se font au niveau du serveur. L'interface utilisateur n'est exploitée que pour le chargement des images, l'affichage des résultats, la sélection des échantillons et des paramètres des algorithmes de classification.

Pour remédier à cela, le chargement et le transfert des images vers le serveur se fait en utilisant le codage **base64** (voir annexe) qui est une chaîne de caractères au lieu de les transférer sous le format initial. Cette approche nous évite la manipulation des images avec leurs tailles réelles (la résolution et le niveau de codage) d'où le gain considérable dans le nombre de requêtes envoyées au serveur web par le navigateur.

L'utilisateur peut aussi soumettre un ou plusieurs fichiers à classifier. Ils sont tous enregistrés en utilisant le nom indiqué lors de la soumission pour les identifier individuellement. Afin de rentabiliser ce transfert, nous avons utilisé le format **JSON** qui permet l'envoi de différents types de données telles que les chaînes de caractères (images déjà codée en **base64**), les tableaux (les coordonnées des échantillons ou les caractéristiques), booléens (les paramètres de classifieurs).

## 4.3 ENVIRONNEMENT ET TECHNOLOGIES LOGICIELLES

### 4.3.1 PLATEFORME DE DEVELOPPEMENT

Nous avons choisi **Python** comme langage de programmation, qui est un langage et une plateforme de développement de logiciel libre et complet, orienté objet, très facile à apprendre. Python est utilisé par un public très large: développeurs web et professionnels, chercheurs en intelligence artificielle ou en bioinformatique, administrateurs système ou même programmeurs occasionnels.

Enfin, dans le domaine de l'apprentissage automatique, **Python** se démarque tout particulièrement en offrant une pléthore de bibliothèques de très haute qualité, couvrant la plupart des types d'apprentissage automatique, le tout accompagné d'une communauté large et dynamique [Zelle, 2004].

Pour réaliser l'interface utilisateur, nous avons utilisé:

- **JavaScript**: incorporé dans un document **HTML**. **JavaScript** est un langage de programmation qui permet d'apporter des améliorations au langage **HTML** en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non pas du serveur web.

- **JQuery**: est une bibliothèque JavaScript libre et multiplateforme qui porte sur l'interaction entre JavaScript (comprenant **Ajax**) et **HTML**, et a pour but de simplifier des commandes communes de JavaScript. [Osmani, 2012]

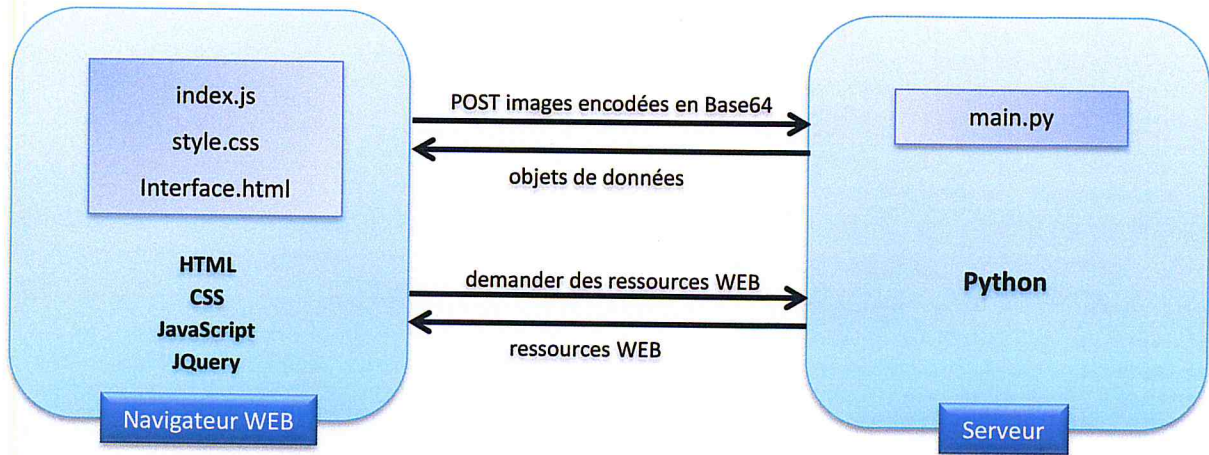


Figure 4.2 Plateforme de développement.

Nous avons également utilisé des langages de développement web: **HTML** (HyperText Markup Language), **CSS** (Cascade Style Sheet) pour réaliser l'interface du système. En effet, nous avons fait appel au langage Python pour les traitements coté serveur et les outils de développement web (JavaScript, HTML, CSS) pour la réalisation de l'interface utilisateur (coté client) comme l'illustre la figure 4.2.

#### 4.3.2 BIBLIOTHEQUES UTILISEES

Une **bibliothèque** ou **librairie logicielle** (ou encore, bibliothèque de programmes) est un ensemble de classes regroupées afin de pouvoir être utilisées sans avoir à les implémenter à nouveau. Les classes sont structurées par leur appartenance à un même domaine conceptuel (mathématique, graphique, etc.). De plus, ces librairies offrent:

- Une structure de projet déjà testée et fiable;
- Gain de temps considérable;
- une meilleure adaptation par rapport aux besoins du système;
- Mieux se concentrer sur le problème métier du système au lieu de reprogrammer des fonctionnalités déjà existantes.

Une des grandes forces du langage **Python** réside dans le nombre important de bibliothèques logicielles externes disponibles, qui le rend un des langages privilégié pour le domaine

d'apprentissage automatique. Dans le cadre de notre travail, nous avons fait appel aux bibliothèques suivantes:

- **Numpy**: est une bibliothèque logicielle destinée à manipuler des matrices ou des tableaux multidimensionnels, elle fournit de multiples fonctions permettant notamment de créer un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier.
- **scikit-image**: est une collection d'algorithmes pour le traitement d'image sous python.
- **Matplotlib**: est une bibliothèque qui, combinée avec les bibliothèques python de calcul scientifique *numpy* et *scipy*, constitue un puissant outil pour tracer et visualiser des données.
- **Pandas**: est une bibliothèque permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles.
- **Flask**: est un Framework open-source de développement web sous Python qui s'occupe de traitement des requêtes.
- **Scikit-learn**: Il existe plusieurs bibliothèques Python qui fournissent des implémentations solides d'une gamme d'algorithmes d'apprentissage automatique. L'un des plus connus est **Scikit-learn**, un package qui fournit des versions efficaces d'un grand nombre d'algorithmes d'apprentissage automatique, couvrant en grande partie l'activité typique du data analyste. Il offre plusieurs techniques de prétraitement des données, apprentissage supervisé et non supervisé, sélection de modèle, validation et métrique d'erreur. **Scikit-Learn** se caractérise par une API propre et uniforme. L'avantage de cette uniformité est qu'une fois qu'on comprend l'utilisation de base et la syntaxe pour un type de modèle, le passage à un nouveau modèle ou algorithme est très simple.

#### 4.4 IMPLEMENTATION

Elle consiste à réaliser le système conformément aux critères définis dans les phases d'analyse et de conception. L'implémentation de notre système peut être divisée en quatre phases.

### 4.4.1 PHASE 1: EXTRACTION DES CARACTERISTIQUES

- Notre approche de classification supervisée d'images numériques consiste à utiliser l'information spectrale issue des différentes bandes spectrales pour classer chaque pixel séparément. Cette information réside dans les valeurs radiométriques de chaque pixel (dans le spectre du visible).
- Dans notre cas, nous avons utilisé les réponses radiométriques R G B (plus les caractéristiques de texture et/ou de voisinage E et D) dans le processus de classification comme le montre la figure 4.3.
- Pour chaque image numérique, il fallait extraire les valeurs radiométriques des pixels ensuite calculer les valeurs contextuelles Entropie et Dissimilarité, afin de construire la matrice des caractéristiques de l'image «data set».

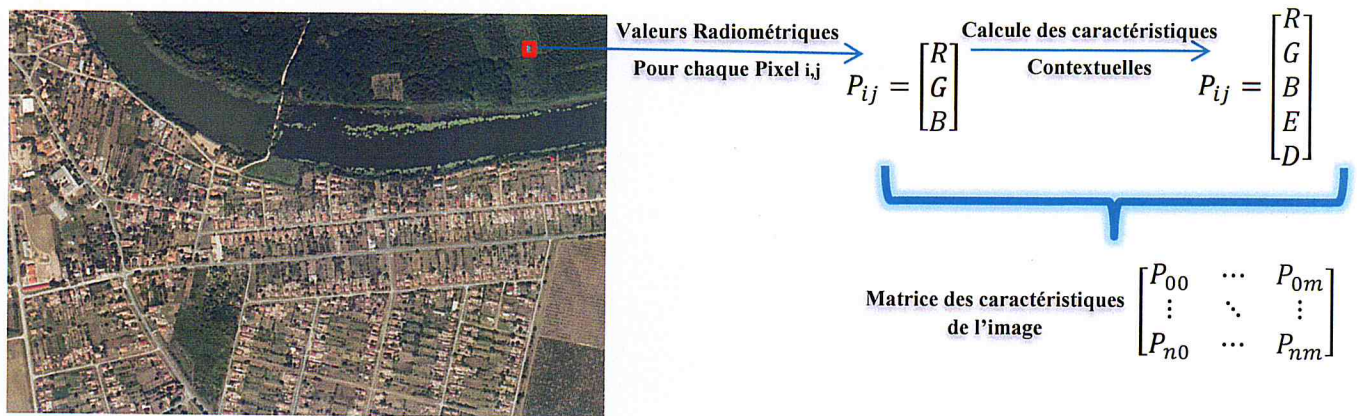


Figure 4.3 Extraction des caractéristiques.

#### 4.4.1.1 Choix de la taille optimale de la fenêtre pour l'extraction des paramètres de texture

L'application d'un algorithme pour le calcul de valeurs **contextuelles** nécessite l'ajustement d'un ou plusieurs paramètres tels que la taille de l'élément structurant utilisé dans le balayage. Les valeurs de ces paramètres dépendent essentiellement des types d'images, la résolution, etc. Le choix de la taille optimale de la fenêtre est primordial dans le calcul des paramètres de texture où il a un impact direct sur la précision de la classification. En effet, si la taille de la fenêtre est trop petite, il n'y aura pas assez d'information spatiale extraite afin de caractériser la transition dans les niveaux de gris entre voisins. Par contre, si la taille de la fenêtre est trop grande, la transition est noyée dans le masque et cela peut créer des limites de transition larges entre deux types de voisins et provoquer un chevauchement des deux types de régions.

Concernant les données de télédétection utilisées dans ce travail et en se basant sur la connaissance préalable de ces données, nous avons sélectionné les paramètres optimaux dans l'extraction des caractéristiques. Pour cela, **la taille du masque** utilisée pour le calcul de **l'entropie** a été fixée à **11** pixels, et celle utilisée pour le calcul de la valeur de **dissimilarité** à **3** pixels de voisinage comme le montre la figure 4.4. Ces valeurs ont été obtenues suite à des tests faits empiriquement.

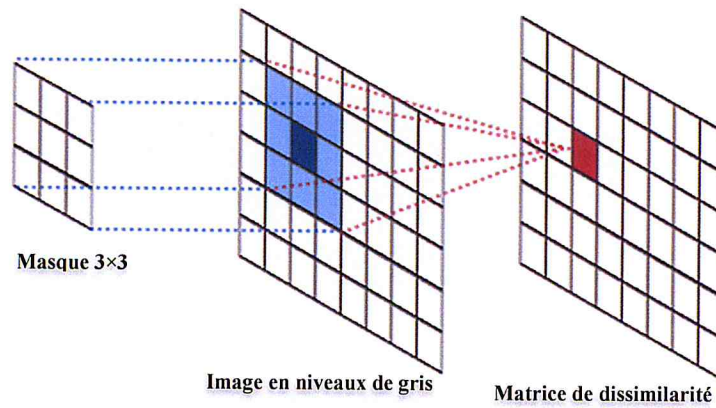


Figure 4.4 Calcul des caractéristiques contextuelles.

#### 4.4.2 PHASE 2: FORMATION DE L'ENSEMBLE DE DONNEES D'APPRENTISSAGE

Nous rappelons que l'approche utilisée dans ce système est basée les méthodes de classification supervisée. Où l'analyste (en se basant sur ses connaissances par rapport aux régions géographiques et les types de surfaces présents dans l'image) identifie puis sélectionne des échantillons assez homogènes de l'image qui sont représentatifs de différents types de surfaces (classes thématiques). Ces échantillons forment l'**ensemble de données d'apprentissage**.

Le choix des pixels échantillons se fait par la sélection des zones rectangulaires de l'image, puis leur assignation à une des classes ou thèmes particuliers (Végétation, Urbain, Sols nus, Milieu humides).



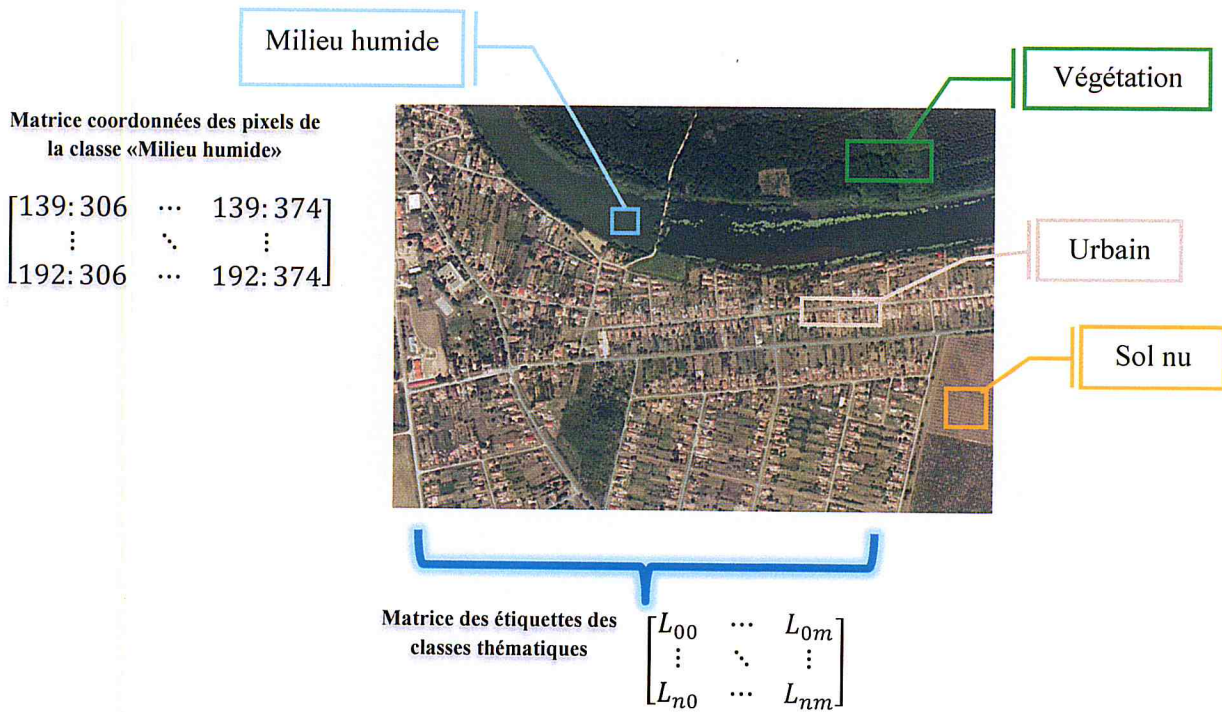


Figure 4.5 Formation d'ensemble de données d'apprentissage «classification thématique»

Les échantillons sélectionnés seront structurés dans des matrices ainsi que les coordonnées correspondantes. De plus, chaque pixel sélectionné est associé à l'étiquette de la classe correspondante comme l'illustre la figure 4.5.

Dans le cas de la détection du changement entre deux images décrivant la même zone d'observation, les données d'entraînement sont extraites directement de l'image de vérité terrain binaire correspondante, après normalisation «Mise à l'échelle» comme l'illustre la figure 4.6.

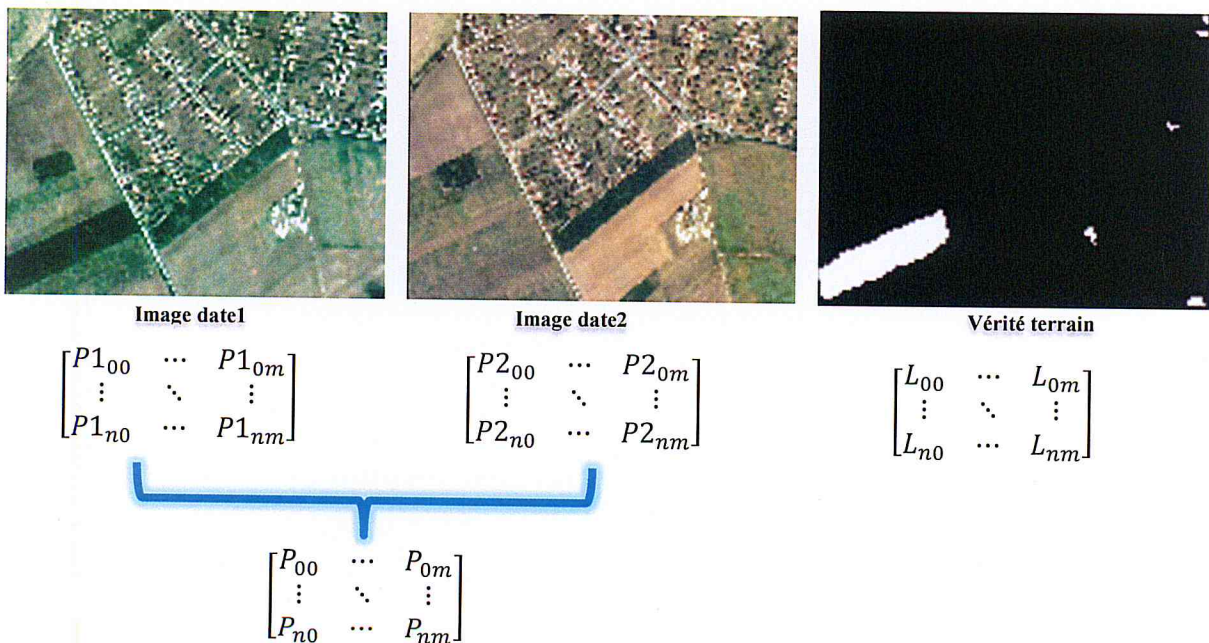


Figure 4.6 Formation d'ensemble de données d'apprentissage «détection du changement».

### 4.4.3 PHASE 3: PREPARATION DES ENSEMBLES DE DONNEES POUR LA CLASSIFICATION

Les matrices des données caractéristiques et des données d'apprentissage doivent être transformées en vecteurs pour pouvoir les utiliser avec les algorithmes de classification «en utilisant la bibliothèque **Scikit-learn**». Pour ce faire on manipule les tableaux avec la bibliothèque **Numpy**, comme l'illustre la figure 4.7.

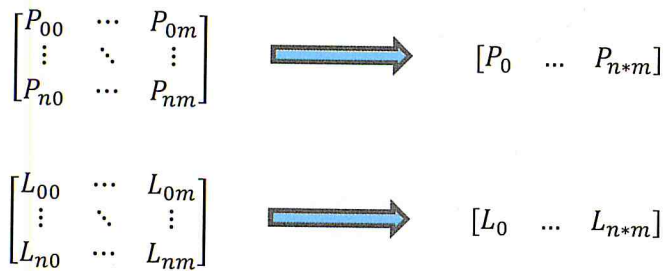
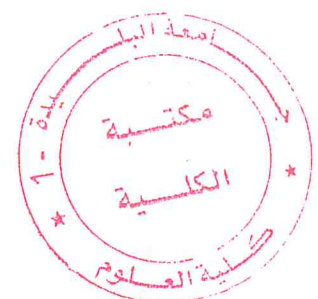


Figure 4.7 Manipulation d'ensembles des données.

### 4.4.4 PHASE 4: CLASSIFICATION

Notre approche est basée sur la classification supervisée. Cette dernière peut être divisée en trois étapes:

- **Apprentissage:** Les données d'apprentissage sont constituées d'un ensemble d'échantillons. Chaque échantillon est une paire composée d'un vecteur de caractéristiques et d'une étiquette. L'algorithme de classification supervisé analyse les données d'apprentissage étiquetées et établit des règles de classification appropriées afin de produire un classifieur entraîné.
- **Essai:** des nouvelles données «essai» sont classifiés par le classifieur entraîné en classes (étiquettes) sur la base des règles de classification générées.
- **Evaluation:** Les étiquettes classées sont ensuite comparées aux données d'essai étiquetées pour évaluer la performance de la classification. La Figure 4.8 donne un aperçu du processus de classification supervisée.



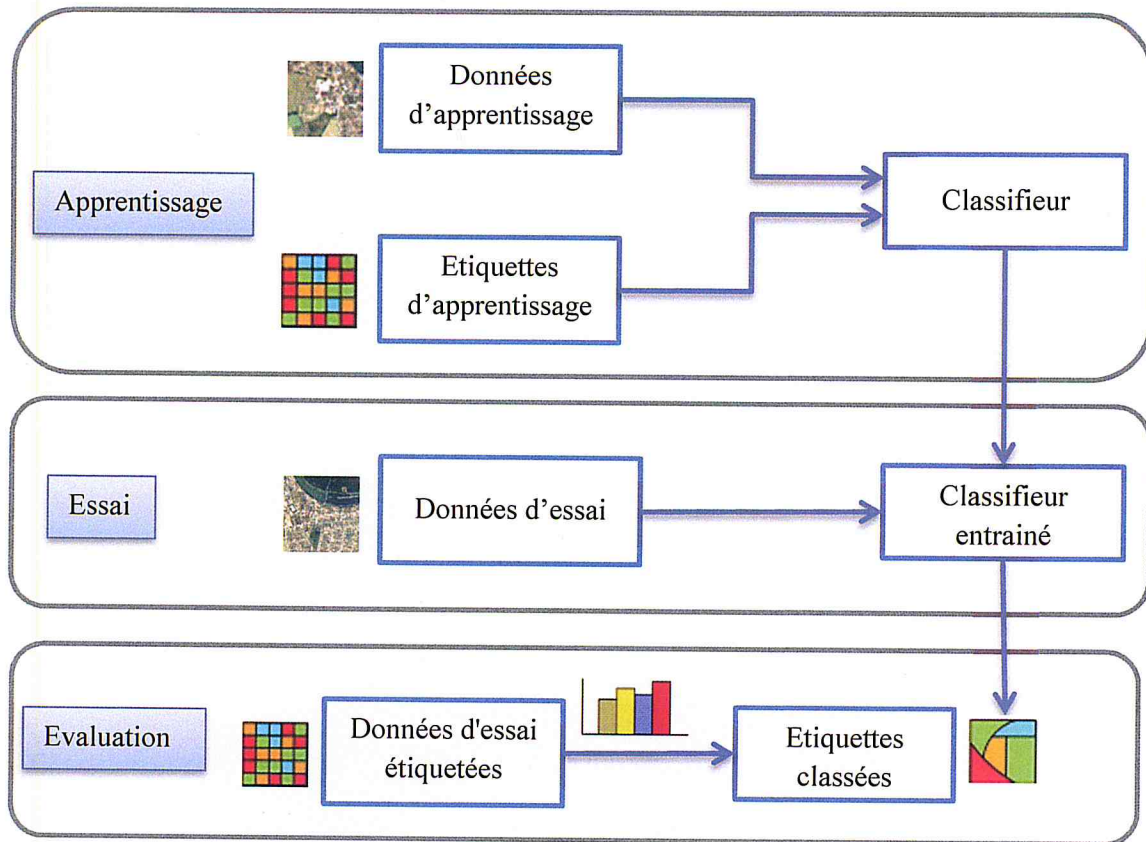


Figure 4.8 Processus de classification supervisée.

#### 4.4.4.1 Apprentissage

Dans cette étape, nous avons utilisé cinq types «les plus utilisés» d'algorithmes de machine d'apprentissage à savoir (Plus proches voisins, machines à vecteur de support, Réseaux de neurones, réseau bayésien naïf, Adaboost). Chacun d'entre eux est configurable par l'utilisateur à travers l'interface graphique.

Tous ces algorithmes sont implémentés en utilisant la bibliothèque **Scikit-learn** de **Python**, il suffit d'importer la classe correspondant au classifieur choisi par l'utilisateur.

Après avoir importé les classes nécessaires, l'objet **classifieur** est initialisé «instancié» avec les paramètres définis par l'utilisateur, puis l'algorithme est alimenté par les données d'apprentissage avec la fonction **fit()**.

Enfin, on aboutit à un **classifieur entraîné** ou ce qu'on appelle modèle prédictif, qui sera par la suite sauvegardé pour une utilisation ultérieure.

#### 4.4.4.2 Test et évaluation des performances du modèle prédictif

Une évaluation rigoureuse des performances d'un algorithme est une étape indispensable à son déploiement. La meilleure méthode pour évaluer un classifieur consiste à calculer sa métrique de précision sur un ensemble **d'échantillons de tests, indépendant** de l'ensemble d'échantillons d'apprentissage mais généré dans les **mêmes conditions**.

Dans cette étape, l'utilisateur a choisi un modèle prédictif via l'interface, l'objet contenant ce modèle est restauré. Nous pouvons alors utiliser cet objet "modèle" pour faire des prédictions sur l'ensemble de tests (en utilisant la fonction **predict()**). La fonction **predict()** renvoie une prédiction pour chaque instance de données de l'ensemble de test. Nous pourrions ensuite afficher nos prédictions pour avoir une idée de ce que le modèle a déterminé.

En utilisant le tableau des étiquettes de classe réelle «tests», nous pouvons évaluer la **précision** des valeurs prédites du modèle en comparant les deux tableaux (étiquettes de classes réelles «tests» vs. étiquettes de classes prédits). La fonction **accuracy\_score()** est utilisée pour déterminer la précision de notre algorithme de machine d'apprentissage.

La mesure de précision est le rapport entre le nombre de prédictions correctes et le nombre total de prédictions effectuées «équation 4.1».

$$\text{précision} = (\text{prédictions correctes} / \text{prédictions totales}) \times 100 \quad (4.1)$$

Cette mesure de précision donne une **appréciation globale** de la performance d'un classifieur, elle est efficace et significative dans le cas où les classes sont bien équilibrées en nombre d'échantillons. C'est le cas de la classification thématique où nous avons quatre classes (Végétation, Urbain, Sols nus, Milieu humide).

Cependant, lorsqu'il s'agit de la détection des changements et du fait que le changement réel et irréversible à la surface de la terre est généralement trop petit en comparant avec les données inchangées, il en résulte des mesures de précision insignifiantes. Dans ce cas, la matrice de confusion peut être utilisée pour évaluer le modèle. C'est une technique qui permet de situer les performances d'un algorithme de classification en présentant par classe et quantitativement les prédictions correctes et incorrectes. Ce qui permet de donner un aperçu sur les types d'erreurs qui ont été commises.

Avec le langage Python, il existe plusieurs façons pour calculer la matrice de confusion. Soit en implémentant la classe de calcul en utilisant les bibliothèques de calcul matricielle et de

probabilité **Numpy** ou **Pandas**, ou bien en utilisant la fonction **confusion\_matrix()** disponible dans la bibliothèque **Scikit-learn**.

## 4.5 INTERFACE GRAPHIQUE

Notre système permet à l'utilisateur d'effectuer la classification thématique et la détection de changement des images satellitaires multi-date à l'aide d'algorithmes de machine d'apprentissage, par le biais d'une interface graphique accessible depuis le réseau via le navigateur web.

Cette interface permet le chargement et la visualisation d'images satellitaires (localement où sur le réseau), la sélection des échantillons d'apprentissage, la sélection et le paramétrage d'algorithme de machine d'apprentissage, l'entraînement de l'algorithme, la sélection d'un modèle de prédiction parmi ceux sauvegardés (algorithme de machine d'apprentissage entraîné), l'évaluation du modèle où la réalisation d'une classification. Et enfin, la présentation des résultats.

Nous présentons dans la suite un ensemble d'impressions écrans des différents éléments de notre interface graphique.

Tout d'abord l'utilisateur doit choisir le mode d'utilisation du système, réaliser une classification thématique ou bien détection du changement comme le montre la figure ci-dessous.

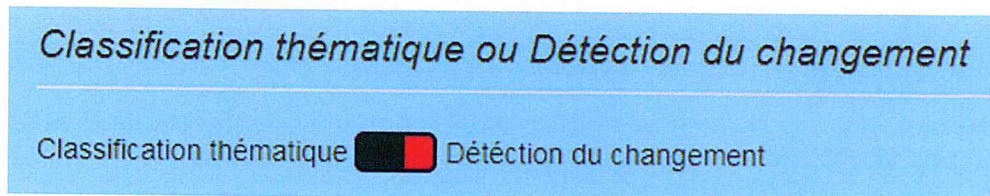


Figure 4.9 Sélection du mode d'utilisation.

L'utilisateur charge les images satellitaires depuis son poste client, il peut aussi les visualiser.

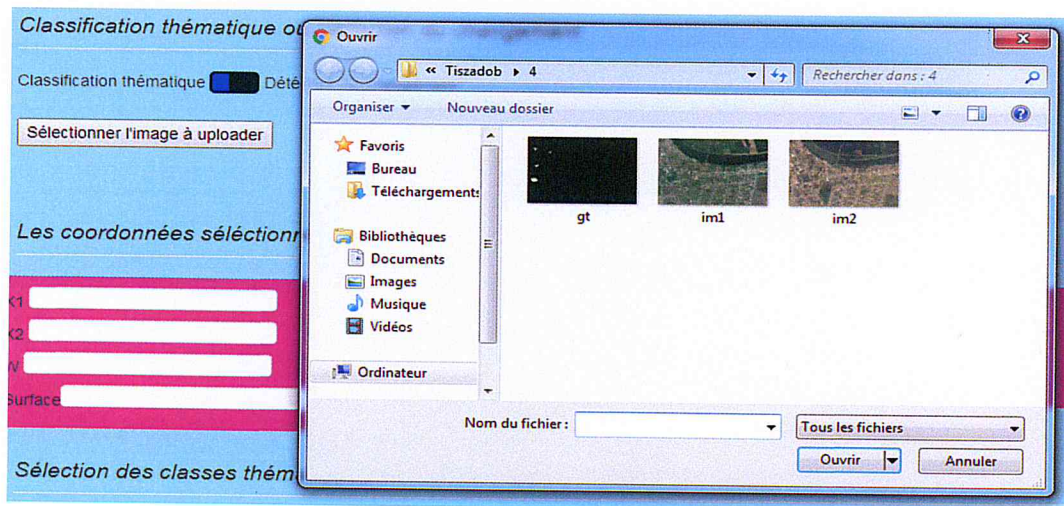


Figure 4.10 Chargement des images satellitaires.

L'interface graphique donne la possibilité à l'utilisateur de sélectionner des échantillons de l'image et de les associer à des classes thématiques «Végétation, Urbain, Sols nus, Milieu humides» afin de constituer les données d'apprentissage, comme illustré dans la figure 4.11. Cette fonctionnalité est réalisée grâce au plugin **JQuery** qui facilite la manipulation des **Canevas** dans les pages **Web**. Ce plugin permet la sélection et l'extraction des coordonnées des images affichées.

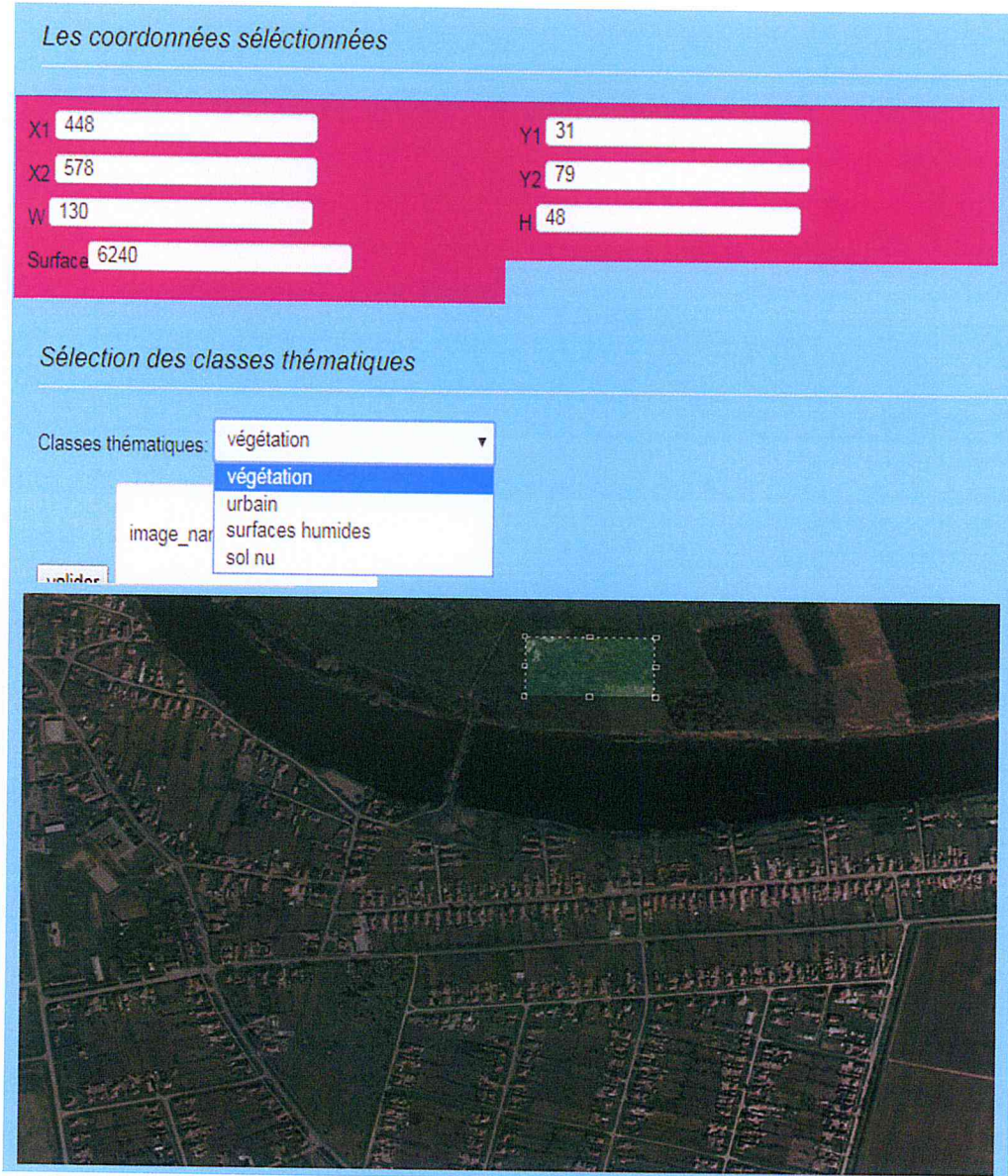


Figure 4.11 Sélection des échantillons d'apprentissage.

L'utilisateur peut choisir entre différents types de classifieurs «algorithme de machine d'apprentissage». Il peut également ajuster les paramètres de chaque type de classifieur comme est illustré dans la figure ci-dessous.

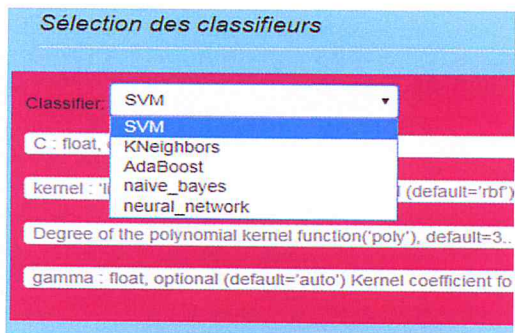


Figure 4.12 Sélection et paramétrage des classifieurs.

L'interface graphique permet aussi le choix du modèle prédictif parmi plusieurs modèles qui sont sauvegardés au niveau du serveur comme le montre la figure 4.13.

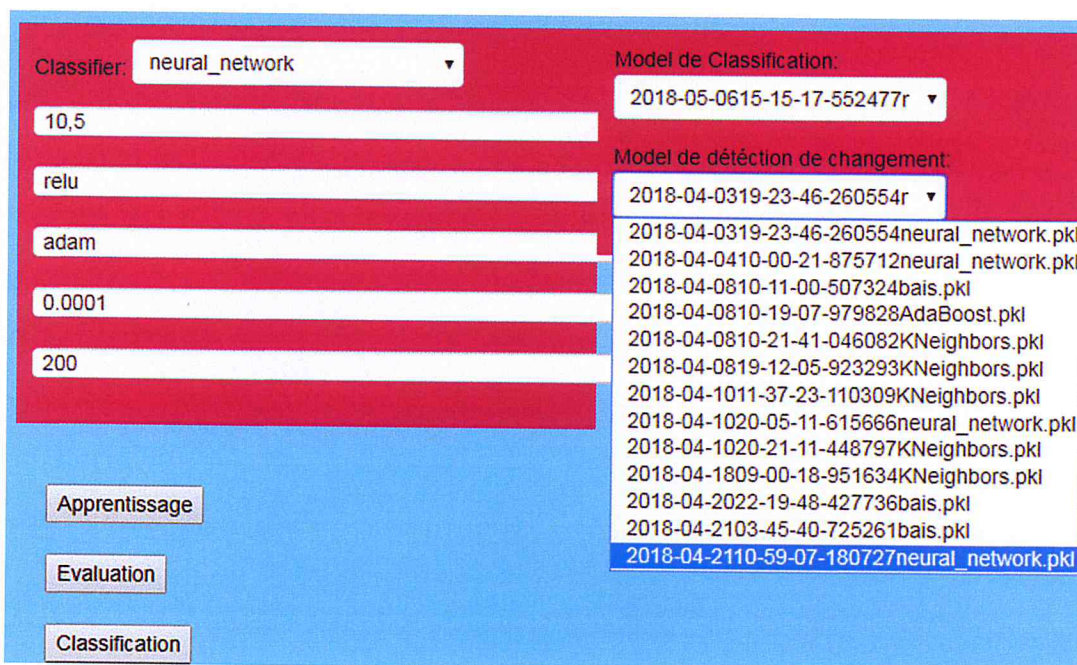


Figure 4.13 Sélection du modèle prédictif.

Suivant le mode d'utilisation choisi, l'utilisateur peut effectuer l'apprentissage d'un classifieur donné, l'évaluation des performances d'un modèle sauvegardé où l'utilisation du modèle sélectionné pour réaliser la classification thématique où la détection du changement, comme le montre la figure 4.14.

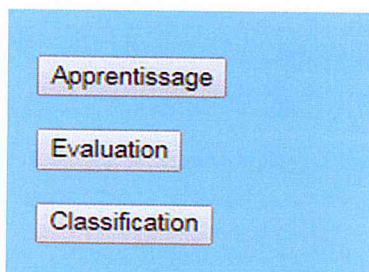
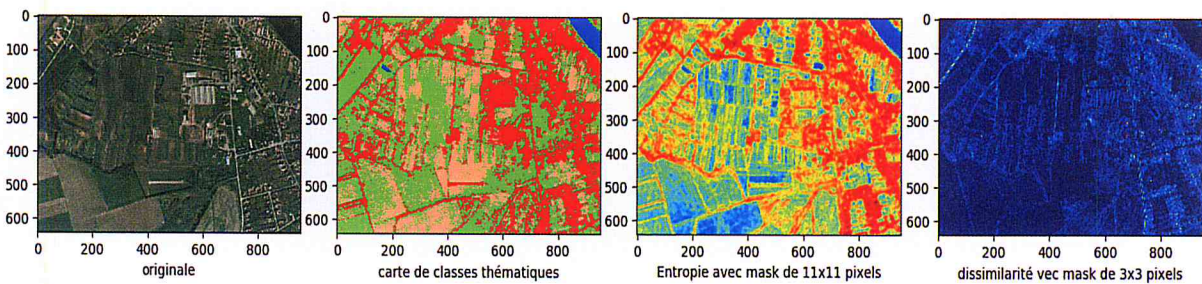
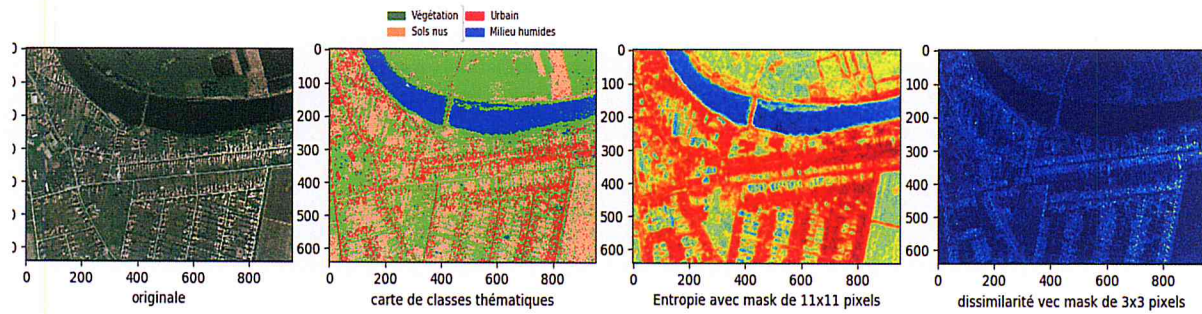
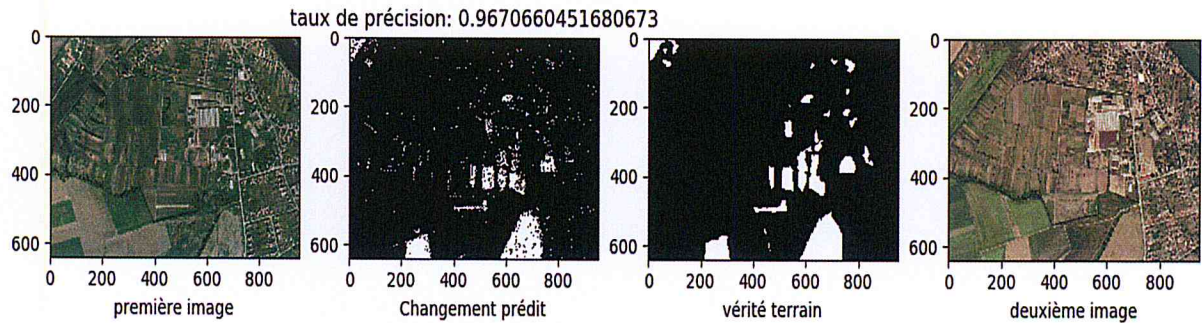


Figure 4.14 Boutons de déclenchement des actions «Apprentissage, Evaluation, Classification».

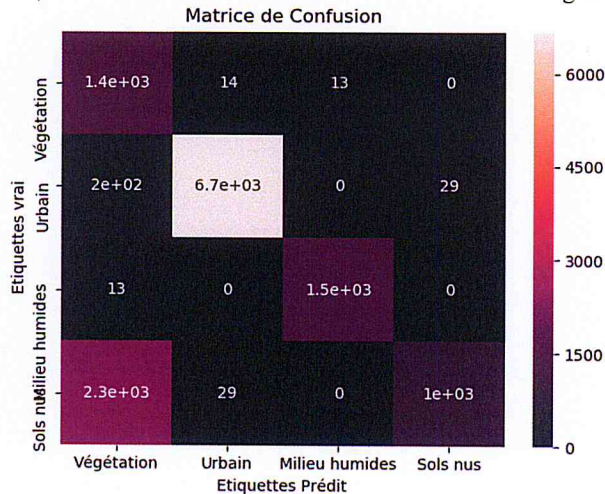
Après la fin du traitement des requêtes utilisateur, les résultats obtenus seront affichés sur l'interface graphique, comme le montre la figure 4.15.



a) Résultat affiché dans le cas de la classification thématique.



b) Résultat affiché dans le cas de la détection du changement.



c) Résultat affiché dans le cas de l'évaluation des performances d'un modèle «matrice de confusion».

Figure 4.15 Résultats affichés.



## 4.6 CONCLUSION

Ce chapitre a été consacré à la mise en œuvre de notre système. A travers la présentation de l'architecture, des différents langages de programmation et bibliothèques utilisés dans la réalisation, l'implémentation et enfin l'interface utilisateur du système. L'évaluation de notre système et la discussion des résultats obtenus seront discutées dans le prochain chapitre.

## **CHAPITRE 5: RESULTATS ET INTERPRETATION**

## 5.1 INTRODUCTION

Le but de ce chapitre est d'évaluer le système développé dans ce travail dans les différentes tâches à savoir l'extraction des caractéristiques, la classification thématique, et la détection de changement. Pour cela, nous avons effectué plusieurs tests sur des données réelles collectées au sein de laboratoires internationaux. En premier lieu nous décrivons les bases de données de télédétection utilisées. Dans un deuxième volet, nous décrivons les mesures d'évaluation utilisées dans ce travail. Il est à noter que plusieurs critères de tests ont été évoqués afin d'assurer une évaluation significative du système. Dans un troisième et dernier volet, nous présentons les résultats obtenus de la classification thématique, la détection de changement ainsi que les discussions et les interprétations correspondantes.

## 5.2 DESCRIPTION DES DONNEES UTILISEES

Afin d'analyser l'efficacité du modèle proposé, une base de donnée de détection de changement a été sélectionnée, à savoir: "Sztaki AirChange Benchmark" [Benedek ,2009].

### 5.2.1 SZTAKI AIRCHANGE BENCHMARK

La base de données SZTAKI AirChange Benchmark a été utilisée dans ce travail [Benedek ,2009]. Cette base comprend 13 paires d'images aériennes de taille 952x640 décrivant la même zone d'observation. La résolution de ces images est de 1.5 mètres, où chaque couple d'images est associé à une image de vérité terrain correspondante.



Figure 5.1 Une paire d'images de la base de données SZTAKI AirChange Benchmark.

Les images constituant chaque paire ont été acquises à différents moments (avec plusieurs années d'intervalle) permettant la génération de changements pertinents tels que: nouvelles régions bâties, opérations de construction, plantation d'un grand groupe d'arbres, etc. Cette base de données contient également des problèmes difficiles dans lesquels des changements réversibles sont considérés comme inchangés, où les variations de la verdure de la végétation, ou même les ombres sont étiquetés comme inchangés dans la vérité terrain. La figure 5.1 représente un échantillon de paires d'images correspondantes aux différentes dates T1 et T2 ainsi que leurs

images de vérité terrain (Ground of truth), en mettant en évidence la zone modifiée avec des points blancs.

### 5.3 CRITERES D'EVALUATION

Afin d'évaluer de manière significative le système proposé, plusieurs mesures statistiques ont été évoquées, telles que la performance de bonne reconnaissance (**Accuracy**), la matrice de confusion, le coefficient de **Kappa** et le **F-measure**. Ces critères d'évaluation sont appliqués dans un premier lieu sur les résultats de classification thématique. Par la suite, une deuxième évaluation a été appliquée sur les résultats de la détection de changement (en considérant l'utilisation des données de références ou vérité terrain), l'utilisation de ces critères permet une évaluation plus rigoureuse.

#### 5.3.1 MATRICE DE CONFUSION

Si l'on dispose d'une réalité de terrain sur l'ensemble de la scène ou sur un ensemble de parcelles uniquement, la matrice de confusion constituera un bon critère d'évaluation de la qualité d'une classification. C'est une matrice carrée établie entre l'image classifiée et la vérité de terrain (base de contrôle). Elle est de taille égale au nombre de classes et ses éléments représentent le nombre (ou le pourcentage) de pixels affectés à chaque classe relativement à leur appartenance aux classes de la vérité de terrain. On choisit arbitrairement de disposer en colonnes de la matrice les données de référence et en lignes le résultat de la classification. Les pourcentages de pixels classés correctement se situent alors sur la diagonale de cette matrice. A partir de cette matrice, différents critères de précision (locaux et globaux) peuvent être déterminés.

#### 5.3.2 TAUX DE RECONNAISSANCE

Ce critère représente le pourcentage de pixels correctement classifiés ou détectés. Ce taux est calculé à partir des moyennes des éléments constituant la diagonale de la matrice de confusion. Il est à noter que le taux de reconnaissance est la mesure d'évaluation la plus largement utilisée. La précision ou le taux de reconnaissance est exprimé comme suit:

$$\text{Taux de reconnaissance} = \frac{tp+tn}{tp+tn+fp+fn} \quad (5.1)$$

Où  $tp$  (true positive ou vrai positif) et  $tn$  (true negative ou vrai négatif) représentent les données correctement classifiées. Pour  $fp$  (false positive, ou faux positif) représente les données mal classées, ce terme est également connu sous le nom de taux de fausses alarmes. Le terme  $fn$  (false negative, ou faux négatif) est le taux de classification erronée ou perdue.

Nous rappelons que certaines problématiques telles que la détection de changement dans la couverture terrestre présentent un déséquilibre dans les données de classes, où le nombre de pixels présentant les régions inchangées dépassent le nombre de séquence représentant les zones changées (unbalanced problems).

### 5.3.3 LE COEFFICIENT DE KAPPA

La deuxième mesure qui peut être extraite à partir de la matrice de confusion est le coefficient de Kappa. C'est une mesure statistique relative à l'accord entre classes. Cette mesure est plus robuste par rapport au taux de reconnaissance puisqu'elle soustrait les cas de classifications correctes obtenus par hasard. Ce coefficient est exprimé comme suit:

$$\text{Kappa} = \frac{P(a) - P(e)}{1 - P(e)} \quad (5.2)$$

Où  $P(a)$  représente la probabilité d'accord entre classes, cette probabilité peut être aussi considérée comme un taux de reconnaissance. La probabilité  $P(e)$  représente l'accord entre classes dû au hasard. La plage du coefficient kappa est  $[-1, 1]$ . La valeur 1 représente une classification parfaite. La valeur 0 indique que l'accord n'est pas meilleur que le hasard, ce qui signifie que le classifieur n'est pas très performant. Un Kappa négatif indiquerait un accord pire que le hasard, ce qui reflète un mauvais résultat de classification (soit un faible algorithme ou la non-correspondance du classifieur avec les données).

### 5.3.4 LE COEFFICIENT F-MEASURE

Le coefficient F-measure est une autre règle d'évaluation efficace dans le domaine de la classification. Cette mesure combine les deux mesures connues sous le nom de précision et rappel (precision and recall). La précision d'une classe est définie par le rapport entre les séquences correctement classées sur les séquences appartenant à cette classe. De même, le rappel pour une classe est défini comme le nombre de vrais positifs divisé par le nombre total d'éléments qui se rapportent réellement à la classe positive. Le coefficient f-measure représente réellement une moyenne harmonique de la précision ( $p$ ) et le rappel ( $r$ ); il est exprimé comme suit:

$$F = (1 + \beta^2) \times \frac{p \times r}{(\beta^2 \times p) + r} \quad (5.3)$$

$$\text{Où } p = \frac{t_p}{t_p + f_p} \quad (5.4)$$

$$\text{Et } r = \frac{t_p}{t_p + f_n} \quad (5.5)$$

Les valeurs de la précision, le rappel et la mesure F varient entre  $[0, 1]$ . La mesure du coefficient F atteint sa meilleure valeur à 1 lorsque  $p$  et  $r$  sont simultanément égaux à 1 et cela correspond à une classification parfaite. La constante  $\beta$  est un paramètre positif qui représente le poids assigné aux types d'erreurs. Dans ce travail, le paramètre  $\beta$  est fixé à 1.

## 5.4 LES RESULTATS OBTENUS

Il est à noter que les performances d'un classifieur dépendent essentiellement du choix de valeurs de ses paramètres. Pour cela, durant la classification thématique et la détection de changement, les paramètres de chaque algorithme (utilisé dans ce système) ont été optimisés pendant la phase d'apprentissage (phase de variation et d'ajustement des paramètres). Les paramètres optimaux doivent correspondre à une précision maximale de classification.

Il n'existe pas de méthode spécifique pour la sélection des valeurs des paramètres. Il est plutôt spécifique à un type de problème.

- Pour le classifieur réseau de neurones, nous avons varié plusieurs paramètres (nombre de neurones dans les couches cachées, nombre maximum d'itération et l'erreur tolérée) afin de sélectionner l'architecture optimale du perceptron.
- Pour le classifieur KNN, le choix du nombre de voisins  $k$  dépend fortement du type de données. Dans ce travail, nous avons varié le paramètre  $k$  de 1 à 20.
- Pour la classification basée sur Naïve Bayes, un modèle gaussien a été adapté comme méthode de distribution où plusieurs valeurs de la moyenne et de l'écart-type ont été utilisées.
- Pour l'algorithme SVM, nous avons testé itérativement les paramètres du noyau, à savoir  $\sigma$  la largeur du noyau Gaussien et  $C$  le paramètre du coût (cost).
- Pour l'algorithme Adaboost, les arbres de décision ont été sélectionnés comme faible classifieur. Nous avons varié le nombre d'itérations ou le nombre de boost.

Il est à noter que ces tests ont été réalisés sur un Intel Core i5 processeur M430, 2.27 GHz avec 4Go de RAM.

Il convient également de noter que les résultats obtenus, présentés ci-dessous, sont en quelque sorte issue de tests des fonctionnalités offertes par le système développé, **l'objectif étant d'évaluer le système développé**. Des tests plus avancés peuvent être effectués dans le cadre de "futurs travaux" afin de trouver la combinaison optimale des paramètres pour chaque classifieur.

### 5.4.1 LES RESULTATS DE LA CLASSIFICATION THEMATIQUE

Nous rappelons que pour le cas de la classification thématique, nous avons pour chaque pixel un vecteur caractéristique de **cinq** valeurs (R, G, B, E et D) qui sera affecté à une des quatre classes thématiques (Végétation, Urbain, Sols nus, Milieu humides), nous avons donc en fait affaire à un problème multi-classe.

#### 5.4.1.1 Les paramètres choisis de la classification thématique

- Pour le classifieur réseau de neurones, cinq couches cachées avec leurs dimensions de neurones respectives (100, 50, 25, 12, 6) ont été sélectionnées avec une erreur tolérée de 0,0001, correspondant à de bonnes performances (taux de reconnaissance et matrice de confusion).
- Pour le classifieur KNN, le choix du nombre de voisins  $k$  a été fixé à 7 correspondant à un taux de reconnaissance plus grand.
- Pour la classification basée sur Naïve Bayes, le modèle gaussien a été choisi.
- Pour le classifieur SVM, la paire  $(\sigma, C)$  correspondante au taux de reconnaissance le plus grand a été sélectionnée ( $\sigma=0.125, C=4$ ).
- Pour l'algorithme adaboost, le nombre d'itérations (ou le nombre de booste) optimal est fixé à  $T = 60$ , qui correspond aux meilleures mesures de performances.

La figure 5.3 montre les résultats de la classification thématique obtenue avec différents algorithmes de classification.

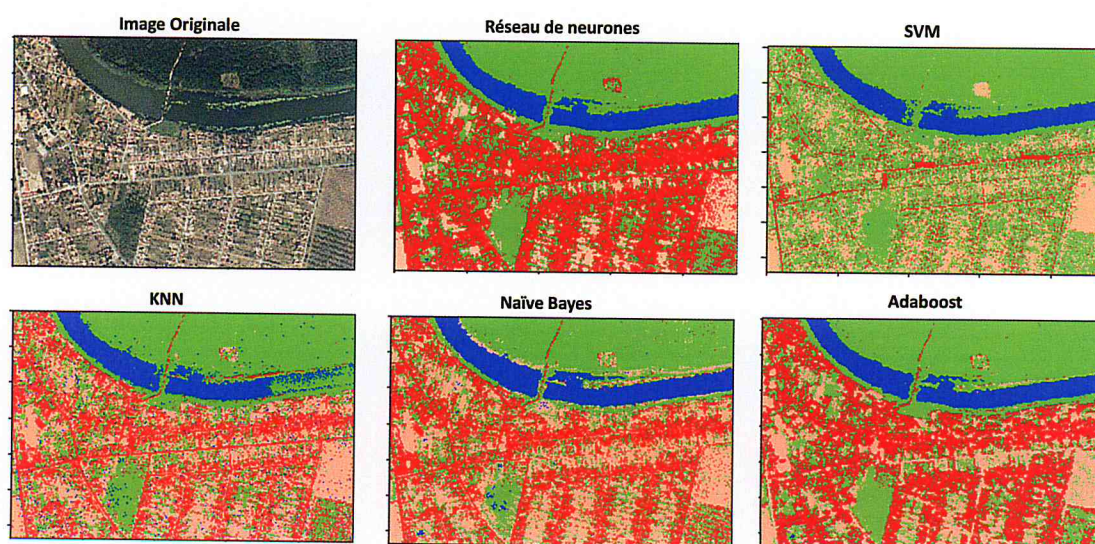


Figure 5.2 Comparaison des résultats de la classification thématique.

### 5.4.1.2 Discussion et Interprétation des résultats de la classification thématique

Les résultats de la classification thématique (présentés dans le tableau 5.1) montrent la distinction entre plusieurs méthodes de classification (telle que les réseaux de neurone, les supports vecteurs machines et l'algorithme adaboost) par rapport aux deux autres méthodes à savoir kNN et Naïve Bayes. La bonne classification revient principalement aux aspects d'apprentissage utilisés par chaque algorithme, tel que la considération de la non-linéarité et la non-structuration des données, etc.

METHODES	TAUX DE RECONNAISSANCE (%)	COEFFICIENT KAPPA	F-MEASURE
NAÏVE BAYES	95.42	0.908	0.954
KNN	96.24	0.9148	0.952
ADABOOST	95.76	0.930	0.953
RESEAUX DE NEURONES	96.85	0.949	0.969
SVM	97.85	0.957	0.978

Tableau 5. 1 Comparaison des performances entre les différents algorithmes de classification thématique.

A partir de tableau 5.1, on remarque que la classification basée sur le théorème de Bayes a présenté le taux de reconnaissance le plus bas, où elle présente également quelques confusions entre les classes. Cela revient principalement à l'hypothèse d'indépendance de cette dernière, ce qui n'est pas toujours valable et même souvent inappropriée pour des données réelles. La classification par le plus proche voisin a présenté un taux de reconnaissance acceptable, mais un coefficient de Kappa faible. Ceci en raison de la complexité de l'algorithme kNN, qui calcule les plus proches voisins à chaque itération (c.-à-d. ré-estimation des distances pour chaque nouveau pixel). On peut constater aussi que la classification en cascade ou adaboost présente un taux de reconnaissance acceptable. Malgré la simplicité du principe (utilisation des faibles classifieurs: arbre de décision), cette classification offre une bonne séparation des classes (coefficient de Kappa raisonnable). La focalisation de chaque classifieur (arbre de décision) sur une partie des caractéristiques uniquement a permis une bonne distinction entre les classes thématiques. On remarque aussi que la classification adaboost est moins sensible au problème de sur-apprentissage, ce qui rend sa mise en œuvre souhaitable pour les applications en temps réel. Cependant, il est à mentionner que le classifieur adaBoost est beaucoup plus sensible aux caractéristiques bruitées et aux données manquantes comparées aux réseaux de neurones et SVMs.



La classification par les réseaux de neurones a présenté un taux de reconnaissance très acceptable. Cela est dû à l'estimation automatique des poids permettant l'obtention des résultats fiables même en présence de données bruitées en entrées du réseau.

En dernier, on peut constater que le classifieur SVM a surpassé les réseaux de neurones (en termes de taux de bonne classification), car les SVMs utilisent une simple interprétation géométrique dans la séparation des données. Les SVMs définissent la solution en minimisant les risques structurels, contrairement aux réseaux de neurones qui utilisent une minimisation empirique des risques. En plus, les résultats de classification par SVM sont moins sensibles aux changements d'ajustement que les réseaux de neurones (nécessitant une ré-estimation des poids).

#### 5.4.2 LES RESULTATS DE LA DETECTION DU CHANGEMENT

Nous rappelons que pour le cas de la détection du changement, nous avons pour chaque pixel un vecteur caractéristique de **dix** valeurs (R1, G1, B1, E1, D1, R2, G2, B2, E2 et D2) qui correspond à une seule classe parmi les deux états (Changé, Inchangé), nous avons donc en fait affaire à un problème de classification binaire.

##### 5.4.2.1 Les paramètres choisis de la détection du changement

- Pour le classifieur réseau de neurones, deux couches cachées avec leurs dimensions de neurones respectives (50, 25) ont été sélectionnées avec une erreur tolérée de 0,00005, correspondant à de bonnes performances pour le cas de la détection du changement.
- Pour le classifieur KNN, le choix du nombre de voisins **k** a été fixé à 3 correspondant aux meilleures performances.
- Pour la classification basée sur Naïve Bayes, le modèle gaussien a été choisi.
- Pour le classifieur SVM, la paire ( $\sigma$ , C) correspondante au taux de reconnaissance le plus grand a été sélectionnée ( $\sigma=0.125$ , C=1).
- Pour l'algorithme adaboost, le nombre d'itérations (ou le nombre de booste) optimal est fixé à  $T = 80$ .

La figure 5.4 montre les résultats de la détection du changement obtenue avec différents algorithmes de classification.

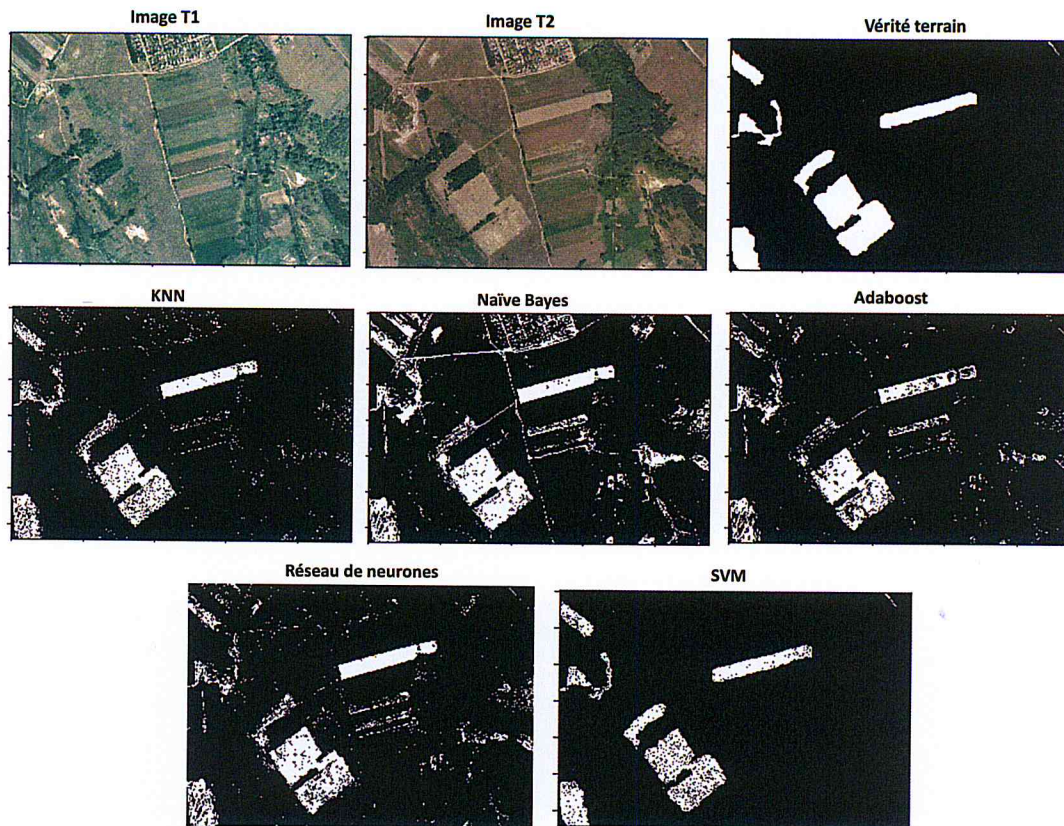


Figure 5.3 Comparaison des résultats de la détection du changement.

#### 5.4.2.2 Discussion et Interprétation des résultats de la détection du changement

Les résultats (présentés dans le tableau 5.2) obtenues de la détection du changement avec différents algorithmes de machine d'apprentissage (telle que les réseaux de neurone, les supports vecteurs machines, adaboost, kNN et Naïve Bayes).

METHODES	TAUX DE RECONNAISSANCE (%)	COEFFICIENT KAPPA	F-MEASURE
NAÏVE BAYES	91.73	0.598	0.925
KNN	96.54	0.783	0.964
ADABOOST	94.38	0.646	0.942
RESEAUX DE NEURONES	95.05	0.700	0.950
SVM	98.52	0.902	0.984

Tableau 5.2 Comparaison des performances entre les différentes méthodes de détection du changement.

A partir de tableau 5.2, on remarque que la détection du changement basée sur le théorème de Bayes a présenté les plus bas taux de reconnaissance et coefficient Kappa comparé aux autres méthodes de classification. Il revient principalement à l'hypothèse d'indépendance de cet algorithme. L'algorithme de plus proche voisin a également présenté un taux de reconnaissance acceptable, mais un coefficient de Kappa faible. Ceci en raison de la complexité de l'algorithme kNN, qui calcule les plus proches voisins à chaque itération (c.-à-d. ré-estimation des distances

pour chaque nouveau pixel). On peut constater aussi que la classification en cascade ou adaboost présente le taux de reconnaissance faible et un coefficient Kappa faible, ce qui induit à une mauvaise séparation des classes de changement.

La classification par les réseaux de neurones a présenté un taux de reconnaissance acceptable, mais un faible coefficient Kappa, Cela est dû (i) au type de données caractéristiques «vecteur de 10 dimensions» et (ii) à deux classes non équilibrées.

En dernier, on peut constater que le classifieur SVM a surpassé l'ensemble des méthodes de classification. Cela est dû principalement à sa capacité dans la séparation géométrique des données, en utilisant la fonction du noyau.

Il faut également noter que le temps d'exécution dépend de plusieurs facteurs et varie d'un algorithme à un autre, en fonction de la taille des échantillons d'apprentissage, du nombre/dimension des couches cachées du réseau de neurones, de la valeur du paramètre C du SVM et de la valeur du paramètre K du KNN, etc.

## 5.5 CONCLUSION

Ce chapitre a été consacré à la présentation des différents tests effectués et résultats obtenues avec le système développé, où nous avons présenté les données de télédétection utilisées, les principales mesures de performances des algorithmes de machine d'apprentissage, les résultats et interprétations pour chaque application (classification thématique et détection du changement).

## **CONCLUSION GENERALE ET PERSPECTIVES**

## Conclusion générale et perspectives

Dans ce travail, nous avons en premier lieu mené plusieurs recherches sur les systèmes de détection de changement proposés dans la littérature. Inspiré par certaines limitations rencontrées auparavant, les motivations principales de ce travail étaient de trouver (i) une représentation adéquate et robuste aux pixels dans la scène et (ii) d'exploiter les techniques de machine d'apprentissage (machine learning techniques) dans la classification thématique et la détection de changement.

Plusieurs attributs sont ainsi proposés pour caractériser le pixel dans la scène, à savoir les valeurs radiométriques ainsi que l'utilisation des caractéristiques contextuelles telle que l'entropie et la dissimilarité. L'avantage de ces caractéristiques est leurs représentations significatives des pixels. En plus, le concept des caractéristiques contextuelles peut apporter des améliorations conséquentes, dans la mesure où l'on ne tient pas seulement compte des propriétés spectrales des pixels, mais aussi de leurs organisations géographiques dans la scène afin de réduire la probabilité de confusion entre les classes.

Durant la phase de classification thématique, plusieurs techniques de machines d'apprentissage sont utilisées, telles que Adaboost, le plus proche voisin, la classification par le théorème de Bayes, les réseaux de neurones et les Séparateur à Vaste Marge afin de séparer entre quatre classes définies à savoir végétation, urbain, sol nu et eau (surfaces humides). Ces techniques sont aussi utilisés pour la phase de détection de changements ce qui aide énormément à la séparation des changements réels des changements réversibles.

En effet, on tient à noter que plusieurs combinaisons d'apprentissage (appliquée sur les différentes données) peuvent mener à une meilleure classification ou détection de changement. En réalité, il n'y a pas d'algorithme unique qui mène toujours à la classification la plus précise. Malgré les résultats prometteurs obtenus dans ce projet, il y a encore des cas qui soulèvent un certain nombre de questions et fournissent plusieurs perspectives à ce travail:

- Nous proposons l'utilisation des images hyper-spectrales afin d'avoir une description plus fine des régions observées et, en conséquence, améliorer encore la précision des classifieurs.
- Nous proposons aussi d'élargir les caractéristiques extraites, surtout quand il s'agit de données hyper-spectrales telles que les indices thématiques (végétation et surfaces humides), etc.

- Il serait aussi intéressant d'explorer d'autres approches basées sur les Réseaux de Neurones Artificiels Profonds (Deep Neural Networks- DNN). Ces approches font partie des grandes tendances actuelles. Où ils ont pris de l'ampleur et de l'importance dans le domaine des machines d'apprentissage, surtout dans l'analyse des données volumineuses (Big Data Analysis).
- Nous proposons également de migrer l'architecture du système vers des systèmes distribués afin d'améliorer les performances: la mise en commun de plusieurs unités de calcul distantes et autonomes (serveurs) permet d'effectuer des traitements parallèles «Répartition de charge» dans un temps plus court.

# ANNEXE

**Encodage base64:** est un codage de l'information utilisant 64 caractères, choisis pour être disponibles sur la majorité des systèmes. Défini en tant qu'encodage MIME dans le RFC 20451, il est principalement utilisé pour la transmission de messages (courrier électronique) sur Internet. Il est par ailleurs défini en propre dans le RFC 4648 [Ousman, 2012].

**JSON:** (Notation Objet issue de JavaScript) est un format léger d'échange de données. Il est facile à lire ou à écrire pour des **humains**. Il est aisément analysable en générale par des **machines**. Il est basé sur un sous-ensemble du langage de programmation JavaScript. Ces propriétés font de **JSON** un langage d'échange de données idéal.

Il se base sur deux structures:

- Une collection de couples nom/valeur. Divers langages la réifient par un objet, un enregistrement, une structure, un dictionnaire, une table de hachage, une liste typée ou un tableau associatif.
- Une liste de valeurs ordonnées. La plupart des langages la réifient par un tableau, un vecteur, une liste ou une suite.

Ces structures de données sont universelles. Pratiquement tous les langages de programmation modernes les proposent sous une forme ou une autre. Il est raisonnable qu'un format de données interchangeable avec des langages de programmation se base aussi sur ces structures. [Bray, 2017]

**MIME:** Multipurpose Internet Mail Extensions ou Extensions multifonctions du courrier Internet, est un standard internet qui étend le format de données des courriels pour supporter des textes en différents codage de caractères autres que l'ASCII, des contenus non textuels, des contenus multiples, et des informations d'en-tête en d'autres codages que l'ASCII [Freed, 1996].

**RFC:** Les requests for comments littéralement «demande de commentaires», sont une série numérotée de documents officiels décrivant les aspects techniques d'Internet, Ce sont des documents de travail qui décrivent pratiquement l'ensemble des informations relatives à l'Internet. Certains décrivent les protocoles de l'Internet, ainsi que leurs évolutions, d'autres sont simplement informatifs.

## BIBLIOGRAPHIE

- [**Altman, 1992**]: Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), 175-185.
- [**Benedek, 2009**]: C. Benedek and T. Szir'anyi, "Change detection in optical aerial images by a multilayer conditional mixed markov model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 10, pp. 3416–3430, 2009.
- [**Benelcadi, 2014**]: Benelcadi, H. (2014). Apport de l'analyse texturale des images radar à haute résolution spatiale pour la cartographie des forêts tropicales (Doctoral dissertation, Paris Est).
- [**Bray, 2017**]: Bray, T. (2017). The JavaScript object notation (json) data interchange format.
- [**Dai, 1999**]: X. Dai and S. Khorram, "Remotely sensed change detection based on artificial neural networks," *Photogrammetric engineering and remote sensing*, vol. 65, pp. 1187–1194, 1999.
- [**Freed, 1996**]: Freed, N., & Borenstein, N. (1996). Multipurpose internet mail extensions (MIME) part one: Format of internet message bodies (No. RFC 2045).
- [**Harsanyi, 1994**]: Harsanyi, J. C., & Chang, C. I. (1994). Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach. *IEEE Transactions on geoscience and remote sensing*, 32(4), 779-785.
- [**Jin, 2013**]: S. Jin, L. Yang, P. Danielson, C. Homer, J. Fry, and G. Xian, "A comprehensive change detection method for updating the National Land Cover Database to circa 2011," *Remote Sensing of Environment*, vol. 132, pp. 159–175, 2013.
- [**Joseph Gabay, 2004**]: Joseph Gabay. (2004). *Merise et UML - Pour la modélisation des systèmes d'information -... -* Librairie Eyrolles (éd. 5ème édition). (O. Dunod, Éd.).
- [**Kettani, 2001**]: Nasser Kettani , Dominique Mignet , Pascal Paré , Camille Rosenthal-Sabroux. (2001). *De Merise à UML (éd. 2e édition)*. (Eyrolles, Éd.) Eryolles france edition.
- [**Kruse, 1993**]: Kruse, F. A., Lefkoff, A. B., Boardman, J. W., Heidebrecht, K. B., Shapiro, A. T., Barloon, P. J., & Goetz, A. F. H. (1993). The spectral image processing system (SIPS)—interactive visualization and analysis of imaging spectrometer data. *Remote sensing of environment*, 44(2-3), 145-163.
- [**Ling, 2011**]: F. Ling, W. Li, Y. Du, and X. Li, "Land cover change mapping at the subpixel scale with different spatial-resolution remotely sensed imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 1, pp. 182–186, 2011.
- [**Osmani, 2012**]: Osmani, A. (2012). *Learning JavaScript Design Patterns: A JavaScript and jQuery Developer's Guide*. «O'Reilly Media, Inc.».
- [**Ousman, 2012**]: Ousman, Y. Z. P. (2012). Une nouvelle approche pour la détection des spams se basant sur un traitement des données catégorielles (Doctoral dissertation, Université de Sherbrooke).



- [**P. Gérard, 2013**]: P. Gérard. (2013). UML Cours 5: Diagramme de séquences. Récupéré sur <http://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours05.html>.
- [**Ratté, 1999**]: Ratté, K. (1999). Identification de la composante «formations meubles» avec des images R \$ O de RADARSAT pour la cartographie des districts écologiques de l'île d'Anticosti (Québec).
- [**Roques, 2018**]: Roques, P. (2018). UML 2.5 par la pratique: Etudes de cas et exercices corrigés. Editions Eyrolles.
- [**Shao, 2016**]: Shao, Y., Lunetta, R. S., Wheeler, B., Iames, J. S., & Campbell, J. B. (2016). An evaluation of time-series smoothing algorithms for land-cover classifications using MODIS-NDVI multi-temporal data. Remote Sensing of Environment, 174, 258-265.
- [**Timothy, 2006**]: Timothy, A., James, B., & Landenberger, R. (2006). Segmentation and classification of high resolution imagery for mapping individual species in a closed canopy, deciduous forest. Department of Geology and Geography, West Virginia University, Morgantown WV 26506-6300, USA.
- [**Zelle, 2004**]: Zelle, J. M. (2004). Python programming: an introduction to computer science. Franklin, Beedle & Associates, Inc.

