

N° d'ordre :

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie
HOUARI BOUMEDIENE

Faculté d'Électronique et d'Informatique

Thèse présentée par

Hassiba NEMMOUR

Pour l'obtention de Magister en Électronique
Option : Électronique des Systèmes

Méthodes de Détection de Changement dans les Images Optiques de Télédétection



Soutenue le 12 novembre 2003 devant le jury composé de :

Mr	A.	ADANE	Professeur	USTHB	Président
Mr	Y.	CHIBANI	Maître de conférences	USTHB	Directeur de thèse
Mr	Y.	SMARA	Maître de conférences	USTHB	Examineur
Mme	A.	SERIR	Docteur d'état	USTHB	Examineur
Mr	S.	CHITROUB	Docteur d'état	USTHB	Examineur

Remerciements

Je tiens tout d'abord à remercier la personne qui est à l'origine de ce travail Mr Y. CHIBANI maître de conférences à l'USTHB qui a consacré à l'encadrement de ma thèse un temps et une disponibilité considérables, auxquels j'ai été d'autant plus sensible que son emploi du temps est très chargé. J'ai ainsi, pu profiter de ses qualités d'encadrement. Je le remercie infiniment pour sa patience et pour sa tolérance dont je n'arriverai jamais à décrire. Je veux lui dire que le travail avec lui était un grand plaisir pour moi. Je tiens absolument à lui exprimer ma profonde gratitude et j'espère qu'il acceptera encore longtemps de bien vouloir travailler avec moi.

Cette thèse résume mon parcours d'initiation à la recherche en traitement d'images au sein du laboratoire de traitement du signal dans la faculté d'électronique et d'informatique de l'USTHB. Je tiens à exprimer ma gratitude au professeur A. HOUACINE pour m'avoir acceptée dans son laboratoire. Je le remercie aussi pour ses conseils judicieux et pour les encouragements qu'il m'a toujours apporté. Je profite aussi pour exprimer ma gratitude aux personnes du laboratoire.

Je remercie vivement Mr ADANE pour avoir accepté de présider le jury de soutenance de ma thèse. Mes remerciements vont également à Mme SERIR pour avoir accepté d'examiner le travail présenté.

Je remercie tout particulièrement Mr Y. SMARA pour avoir accepté d'évaluer mon travail et aussi, pour les efforts qu'il a effectué en nous enseignant durant l'année théorique. Pour cette raison, je veux adresser des remerciements très très très particuliers à Mr BERKANI et à Mme BELHADJ AÏSSA.

Je n'ai pas oublié ma promotion de post graduation, je ne veux pas citer les noms mais je tiens à saluer Karima pour son soutien et pour les discussions fructueuses que j'ai eu avec elle.

Le dernier remerciement est de droit à mes parents (les trois) et à toute ma famille qui ont donné une importance particulière pour mes études. Sans eux j'aurais certainement eu beaucoup de difficultés pour achever ce travail.

Table des Matières

Introduction Générale	1
------------------------------------	---

Chapitre 1 :

Techniques de Détection de Changement

1.1. Approches de détection de changement.....	5
1.2. Méthodes de détection de changement	6
1.2.1. Différence des images	6
1.2.2. Différence des indices de végétation	6
1.2.3. Rapport des images	6
1.2.4. Analyse en composantes principales.....	7
1.2.5. Analyse du vecteur de changement.....	7
1.2.6. Comparaison par post classification	8
1.2.7. Analyse composite multitemporelle.....	8
1.2.8. Analyse de l'intercorrélation.....	8
1.2.9. Autres techniques.....	8
1.2.10. Etudes comparatives des méthodes de détection de changement	9
1.3. Prétraitements des images.....	11
1.3.1. Corrections géométriques	12
1.3.2. Corrections radiométriques	12
1.3.3. Normalisation radiométrique	12
1.4. Zone d'étude	13
1.5. Conclusion	17

Chapitre 2 :

Utilisation des Réseaux de Neurones pour la Détection de Changement

2.1. Introduction.....	18
2.2. Principe de la détection de changement par le réseau multicouche.....	19
2.3. Algorithmes d'apprentissage du réseau multicouche	20
2.4. Résultats expérimentaux	22
2.4.1. Evaluation des règles d'apprentissage	22
2.4.1.1. Critères d'évaluation.....	22
2.4.1.2. Vitesse de convergence.....	23
2.4.1.3. Influence du nombre de nœuds	24
2.4.1.4. Influence du nombre d'itérations.....	25

2.4.2.	Evaluation de la détection de changement.....	26
2.4.2.1.	Analyse simultanée.....	26
2.4.2.1.1.	Critères d'évaluation.....	26
2.4.2.1.2.	Matrices de confusion.....	26
2.4.2.1.3.	Courbes <i>ROC</i>	29
2.4.2.1.4.	Appréciation visuelle.....	30
2.4.2.2.	Analyse comparative.....	36
2.4.2.2.1.	Critères d'évaluation.....	36
2.4.2.2.2.	Appréciation visuelle.....	36
2.4.2.3.	Comparaisons statistiques des analyses simultanée et comparative.....	41
2.4.2.3.1.	Critères d'évaluation.....	41
2.4.2.3.2.	Taux moyen de bonne reconnaissance.....	41
2.4.2.3.3.	Courbes <i>ROC</i>	43
2.5.	Conclusion.....	43

Chapitre 3 :

Détection de Changement par les Opérateurs de la Logique Floue

3.1.	Utilisation de la logique floue en télédétection.....	45
3.2.	Application à la détection de changement.....	45
3.3.	Théorie de la logique floue.....	46
3.3.1.	Introduction.....	46
3.3.2.	Sous ensembles flous et degré d'appartenance.....	46
3.3.3.	Principale extension.....	47
3.3.4.	Opérations sur les ensembles flous.....	47
3.3.5.	Normes triangulaires.....	48
3.3.6.	Conormes triangulaires.....	49
3.3.7.	Négation.....	49
3.4.	Méthode adoptée pour la classification et la détection de changement.....	50
3.5.	Résultats expérimentaux.....	51
3.5.1.	Etude empirique sur le choix du paramètre de fuzzification.....	51
3.5.2.	Etude comparative des approches <i>AS</i> et <i>AC</i> floues.....	55
3.5.2.1.	Critères d'évaluation.....	55
3.5.2.2.	Evaluation quantitative.....	55
3.5.2.3.	Evaluation qualitative.....	55
3.5.3.	Etude comparative des méthodes floue et neuronale.....	65
3.6.	Utilisation des combinaisons neurofloues pour la détection de changement.....	66
3.6.1.	Comparaison entre le modèle neuroflou et le réseau de neurones.....	67
3.6.2.	Appréciation visuelle.....	68
3.7.	Conclusion.....	73

Chapitre 4 :

Amélioration de la Détection de Changement par Combinaison de Classifieurs

4.1. Introduction.....	74
4.2. Combinaison de classifieurs : état de l'art.....	74
4.3. Méthodes de combinaison	75
4.4. Combinaison par les intégrales floues	76
4.4.1. Définitions.....	76
4.4.1.1. Définition de la mesure floue.....	76
4.4.1.2. Mesure floue λ	77
4.4.1.3. Intégrale floue de Sugeno	77
4.4.1.4. Intégrale discrète de Choquet	78
4.4.2. Méthode de combinaison	78
4.5. Application à la combinaison des classifieurs neuronaux	79
4.5.1. Combinaison par l'intégrale de Sugeno	80
4.5.1.1. Evaluation statistique.....	80
4.5.1.2. Appréciation visuelle	81
4.5.2. Comparaison entre l'intégrale de Sugeno et l'intégrale de Choquet	84
4.6. Application à la combinaison des approches <i>AS</i> et <i>AC</i> floues.....	85
4.6.1. Schéma de Combinaison.....	85
4.6.2. Présentation des résultats	86
4.6.2.1. Evaluation statistique.....	86
4.6.2.2. Appréciation visuelle	86
4.7. Conclusion	88

Conclusion Générale	89
----------------------------------	-----------

Annexes

A : Algorithmes d'apprentissage du Perceptron Multicouche.....	91
A.1. Algorithme de rétropropagation du gradient.....	91
A.1.1. Calcul détaillé.....	91
A.1.2. Adaptation des poids dans les couches cachées.....	92
A.1.3. Lissage de la règle d'adaptation.....	94
A.2. Algorithme de filtrage de Kalman.....	95
A.2.1. Principe	95
A.2.2. Filtre de Kalman.....	97
A.2.3. Algorithme	98
B : Calcul de la Matrice de Confusion et du Coefficient <i>Kappa</i>.....	100
B.1. Matrice de confusion	100
B.2. Coefficient <i>Kappa</i>	101

C : La Courbe Receiver Operating Characteristic : <i>ROC</i>	102
D : Présentation des Résultats des Tests de Simulation	104
D.1. Utilisation des Réseaux de Neurones pour la Détection de Changement	105
D.2. Détection de Changement par les Opérateurs de la Logique Floue	108
D.3. Amélioration de la Détection de Changement par Combinaison de Classifieurs	109
D.4. Evaluation sur la scène entière	110
E : Statistiques du Chapitre 3	113
E.1. Statistiques de l'approche <i>AS</i>	113
E.2. Statistiques de l'approche <i>AC</i>	115
E.3. Variations du degré d'appartenance en fonction du paramètre de fuzzification	117
Références Bibliographiques	121

Liste des Tableaux

<u>Tableau</u>	<u>Page</u>
1.1 Récapitulation des comparaisons des méthodes de détection de changement.....	11
2.1 Données de la base d'entraînement.....	23
2.2 Matrice de confusion du réseau 1cc entraîné par l'algorithme RP	27
2.3 Matrice de confusion du réseau 1cc entraîné par l'algorithme FK	27
2.4 Matrice de confusion du réseau 2cc entraîné par l'algorithme RP	28
2.5 Matrice de confusion du réseau 2cc entraîné par l'algorithme FK	28
2.6 Erreurs de surestimation dans les différentes classes.....	29
2.7 Données d'apprentissage utilisées dans les processus de classification	36
2.8 Comparaison des analyses <i>AS</i> et <i>AC</i> selon les valeurs du <i>TMBR</i> pour différents réseaux	41
3.1 Normes triangulaires de base	48
3.2 Conormes triangulaires de base	49
3.3 Distances moyennes du pixel typique de chaque classe par rapport aux centroïdes de toutes les classes	52
3.4 Degrés d'appartenance global et par classe calculés pour chaque approche .	55
3.5 Taux de bonne reconnaissance de l'approche <i>AS</i> floue crispée.....	65
3.6 Taux de bonne reconnaissance des réseaux de neurones et des combinaisons neurofloues.....	67
3.7 Taux de surestimation des réseaux de neurones et des combinaisons neurofloues.....	67
4.1. Taux de bonne reconnaissance des réseaux à une couche cachée et du processus de combinaison.....	80
4.2. Taux de bonne reconnaissance des réseaux à deux couches cachées et du processus de combinaison.....	81
4.3. <i>TBR</i> du processus de combinaison par les intégrales de Sugeno et de Choquet.....	84
4.4. <i>TBR</i> du processus de combinaison et des approches <i>AS</i> et <i>AC</i> floues crispées.....	86

Annexes

D.1. Données de la base d'entraînement.....	104
D.2. Valeurs du khat pour les différents réseaux	107
D.3. <i>TMBR</i> des approches <i>AS</i> et <i>AC</i>	108
D.4. Degrés d'appartenance par classe et global des approches <i>AS</i> et <i>AC</i> floues.....	108
D.5. <i>TBR</i> obtenus par les méthodes neuronale et neurofloue	109
D.6. Performance du processus de combinaison comparativement aux réseaux individuels (1cc).....	109

D.7. Performance du processus de combinaison comparativement aux réseaux individuels (2 cc).....	110
E.1. Matrice inverse de covariance de la classe <i>eau</i> $\rightarrow$ <i>eau</i>	113
E.2. Matrice inverse de covariance de la classe <i>construction</i> $\rightarrow$ <i>construction</i>	113
E.3. Matrice inverse de covariance de la classe <i>végétation</i> $\rightarrow$ <i>végétation</i>	113
E.4. Matrice inverse de covariance de la classe <i>sol nu</i> $\rightarrow$ <i>sol nu</i>	113
E.5. Matrice inverse de covariance de la classe <i>végétation</i> $\rightarrow$ <i>sol nu</i>	114
E.6. Matrice inverse de covariance de la classe <i>construction</i> $\rightarrow$ <i>sol nu</i>	114
E.7. Matrice inverse de covariance de la classe <i>eau</i> $\rightarrow$ <i>sol nu</i>	114
E.8. Matrice inverse de covariance de la classe <i>X</i> $\rightarrow$ <i>nuages</i>	114
E.9. Vecteurs moyennes	114
E.10. Matrice inverse de covariance de la classe <i>eau</i>	115
E.11. Matrice inverse de covariance de la classe <i>construction</i>	115
E.12. Matrice inverse de covariance de la classe <i>végétation</i>	115
E.13. Matrice inverse de covariance de la classe <i>sol nu</i>	115
E.14. Vecteurs moyennes	115
E.15. Matrice inverse de covariance de la classe <i>eau</i>	116
E.16. Matrice inverse de covariance de la classe <i>construction</i>	116
E.17. Matrice inverse de covariance de la classe <i>végétation</i>	116
E.18. Matrice inverse de covariance de la classe <i>sol nu</i>	116
E.19. Matrice inverse de covariance de la classe <i>nuages</i>	116
E.20. Vecteurs moyennes	116

Liste des Figures

<u>Figure</u>	<u>Page</u>
1.1. Image acquise en 1989 représentée en composition colorée.....	14
1.2. Image acquise en 1991 représentée en composition colorée.....	14
1.3. Composition colorée de la partie extraite de l'image de 1989	15
1.4. Composition colorée de la partie extraite de l'image de 1991	15
1.5. Image classifiée de la surface d'étude de 1989	16
1.6. Image classifiée de la surface d'étude de 1991	16
2.1. Principe de la classification par un réseau à une couche cachée.....	19
2.2. Principe de la détection de changement par le réseau de neurones.....	20
2.3. Minimisation de l'erreur au niveau d'un nœud	21
2.4. Evolution de l'erreur quadratique moyenne au cours du temps pour un réseau à une couche cachée comportant 20 nœuds.....	23
2.5. Evolution de l'erreur quadratique moyenne au cours du temps pour un réseau à deux couches cachées comportant 20 nœuds chacune	24
2.6. Influence du nombre de nœuds pour un réseau à une couche cachée	24
2.7. Influence du nombre de nœuds pour un réseau à deux couches cachées	25
2.8. Evolution de la valeur du <i>Khat</i> en fonction du nombre d'itérations pour le réseau à une couche cachée	25
2.9. Evolution de la valeur du <i>Khat</i> en fonction du nombre d'itérations pour le réseau comportant deux couches cachées.....	26
2.10. Courbes <i>ROC</i> pour un réseau à une couche cachée	30
2.11. Courbes <i>ROC</i> pour un réseau à deux couches cachées	30
2.12. Image obtenue par <i>AS</i> en utilisant un réseau à une couche cachée entraîné par l'algorithme RP	32
2.13. Image obtenue par <i>AS</i> en utilisant un réseau à une couche cachée entraîné par l'algorithme FK	33
2.14. Image obtenue par <i>AS</i> en utilisant un réseau à deux couches cachées entraîné par l'algorithme RP	34
2.15. Image obtenue par <i>AS</i> en utilisant un réseau à deux couches cachées entraîné par l'algorithme FK	35
2.16. Image obtenue par <i>AC</i> en utilisant un réseau à une couche cachée entraîné par l'algorithme RP	37
2.17. Image obtenue par <i>AC</i> en utilisant un réseau à une couche cachée entraîné par l'algorithme FK	38
2.18. Image obtenue par <i>AC</i> en utilisant un réseau à deux couches cachées entraîné par l'algorithme RP	39
2.19. Image obtenue par <i>AC</i> en utilisant un réseau à deux couches cachées entraîné par l'algorithme FK	40
2.20. <i>TMBR</i> obtenu par les analyses simultanée et comparative.....	42
2.21. Courbes <i>ROC</i> obtenues en utilisant un réseau à une couche cachée entraîné par l'algorithme FK	43
2.22. Courbes <i>ROC</i> obtenues en utilisant un réseau à deux couches cachées entraîné par l'algorithme FK	43

3.1.	Evolution du degré d'appartenance dans la classe <i>végétation</i> $\rightarrow$ <i>sol nu</i> en fonction du paramètre de fuzzification.....	52
3.2.	Evolution du degré d'appartenance dans les autres classes.....	53
3.3.	Evolution du degré d'appartenance dans la classe <i>construction</i> $\rightarrow$ <i>sol nu</i> en fonction du paramètre de fuzzification.....	53
3.4.	Evolution du degré d'appartenance dans les autres classes.....	53
3.5.	Evolution du degré d'appartenance dans la classe <i>eau</i> $\rightarrow$ <i>sol nu</i> en fonction du paramètre de fuzzification.....	54
3.6.	Evolution du degré d'appartenance dans les autres classes.....	54
3.7.	Degré d'appartenance dans la classe <i>eau</i> $\rightarrow$ <i>eau</i> obtenu par <i>AS</i>	56
3.8.	Degré d'appartenance dans la classe <i>eau</i> $\rightarrow$ <i>eau</i> obtenu par <i>AC</i>	56
3.9.	Degré d'appartenance dans la classe <i>construction</i> $\rightarrow$ <i>construction</i> obtenu par <i>AS</i>	57
3.10.	Degré d'appartenance dans la classe <i>construction</i> $\rightarrow$ <i>construction</i> obtenu par <i>AC</i>	57
3.11.	Degré d'appartenance dans la classe <i>végétation</i> $\rightarrow$ <i>végétation</i> obtenu par <i>AS</i>	58
3.12.	Degré d'appartenance dans la classe <i>végétation</i> $\rightarrow$ <i>végétation</i> obtenu par <i>AC</i>	58
3.13.	Degré d'appartenance dans la classe <i>sol nu</i> $\rightarrow$ <i>sol nu</i> obtenu par <i>AS</i>	59
3.14.	Degré d'appartenance dans la classe <i>sol nu</i> $\rightarrow$ <i>sol nu</i> obtenu par <i>AC</i>	59
3.15.	Degré d'appartenance dans la classe <i>végétation</i> $\rightarrow$ <i>sol nu</i> obtenu par <i>AS</i>	60
3.16.	Degré d'appartenance dans la classe <i>végétation</i> $\rightarrow$ <i>sol nu</i> obtenu par <i>AC</i>	60
3.17.	Degré d'appartenance dans la classe <i>construction</i> $\rightarrow$ <i>sol nu</i> obtenu par <i>AS</i>	61
3.18.	Degré d'appartenance dans la classe <i>construction</i> $\rightarrow$ <i>sol nu</i> obtenu par <i>AC</i>	61
3.19.	Degré d'appartenance dans la classe <i>eau</i> $\rightarrow$ <i>sol nu</i> obtenu par <i>AS</i>	62
3.20.	Degré d'appartenance dans la classe <i>eau</i> $\rightarrow$ <i>sol nu</i> obtenu par <i>AC</i>	62
3.21.	Degré d'appartenance dans la classe <i>X</i> $\rightarrow$ <i>nuages</i> obtenu par <i>AS</i>	63
3.22.	Degré d'appartenance dans la classe <i>X</i> $\rightarrow$ <i>nuages</i> obtenu par <i>AC</i>	63
3.23.	Composition colorée des degrés d'appartenance dans les trois classes de changement (<i>AS</i>).....	64
3.24.	Composition colorée des degrés d'appartenance dans les trois classes de changement (<i>AC</i>).....	64
3.25.	Courbe <i>ROC</i> du processus flou crispé.....	66
3.26.	Combinaisons neurofloues.....	66
3.27.	Comparaison entre les réseaux de neurones et les combinaisons neurofloues selon la valeur du <i>Khat</i>	68
3.28.	Image obtenue par la combinaison neurofloue du réseau avec une couche cachée entraîné par l'algorithme <i>RP</i>	69
3.29.	Image obtenue par la combinaison neurofloue du réseau avec une couche cachée entraîné par l'algorithme <i>FK</i>	70
3.30.	Image obtenue par la combinaison neurofloue du réseau avec deux couches cachées entraîné par l'algorithme <i>RP</i>	71
3.31.	Image obtenue par la combinaison neurofloue du réseau avec deux couches cachées entraîné par l'algorithme <i>FK</i>	72
4.1.	Approche de combinaison parallèle des classifieurs.....	75
4.2.	<i>Khat</i> des réseaux à une couche cachée et du processus de combinaison.....	80
4.3.	<i>Khat</i> des réseaux à deux couches cachées et du processus de combinaison.....	81
4.4.	Image produite par la combinaison des réseaux à une couche cachée.....	82
4.5.	Image produite par la combinaison des réseaux à deux couches cachées.....	83
4.6.	<i>Khat</i> obtenu par les intégrales de Sugeno et de Choquet.....	84

4.7. Schéma de combinaison des approches <i>AS</i> et <i>AC</i>	85
4.8. Image produite par la combinaison des approches <i>AS</i> et <i>AC</i> floues.....	87

Annexes

A.1. Perceptron avec une seule couche cachée	91
A.2. Portion linéaire d'un neurone	95
D.1. Composition colorée des images multispectrales.....	104
D.2. Evolution de l'erreur pour un réseau à une couche cachée	105
D.3. Evolution de l'erreur pour un réseau à deux couches cachées	105
D.4. Influence du nombre de nœuds pour un réseau à une couche cachée	106
D.5. Influence du nombre de nœuds pour un réseau à deux couches cachées	106
D.6. Influence du nombre d'itérations pour un réseau à une couche cachée	106
D.7. Influence du nombre d'itérations pour un réseau à deux couches cachées	107
D.8. Courbes <i>ROC</i>	107
D.9. Image obtenue par <i>AS</i>	110
D.10. Image obtenue par <i>AC</i>	111
D.11. Image obtenue par le processus <i>AS</i> flou crispé	111
D.12. Image obtenue par le modèle neuroflou	112
D.13. Image obtenue par la combinaison des réseaux à une couche cachée.....	112
E.1. Evolution du degré d'appartenance dans la classe <i>eau</i> → <i>eau</i>	117
E.2. Evolution du degré d'appartenance dans les autres classes.....	117
E.3. Evolution du degré d'appartenance dans la classe <i>construction</i> → <i>construction</i> ...	117
E.4. Evolution du degré d'appartenance dans les autres classes.....	118
E.5. Evolution du degré d'appartenance dans la classe <i>végétation</i> → <i>végétation</i>	118
E.6. Evolution du degré d'appartenance dans les autres classes.....	118
E.7. Evolution du degré d'appartenance dans la classe <i>sol nu</i> → <i>sol nu</i>	119
E.8. Evolution du degré d'appartenance dans les autres classes.....	119
E.9. Evolution du degré d'appartenance dans la classe <i>X</i> → <i>nuages</i>	119
E.10. Evolution du degré d'appartenance dans les autres classes.....	120

Introduction Générale

Le couvert de la terre subit des changements continus dans le temps qui conduisent peu à peu à la modification globale de l'environnement. Généralement, ces changements sont causés par deux types de facteurs (Lambin et Stahler, 1994):

- *Facteurs naturels* : où le couvert de la terre subit des modifications à cause des phénomènes naturels tels que les changements à long terme du climat et les successions de végétation.
- *Facteurs humains* : en relation avec les activités de l'homme qui modifient de plus en plus le couvert des surfaces à travers des actions directes comme la déforestation et l'urbanisation.

La quantification et la cartographie de ces modifications qui se manifestent par des changements dans les propriétés du sol à savoir l'émissivité, l'albédo et la rugosité, sont d'une importance exceptionnelle pour plusieurs applications. Cela nécessite un suivi continu de l'état du sol. Ainsi, les images fournies par la télédétection sont extrêmement utiles. Elles présentent l'avantage par rapport aux autres types de données d'être directement exploitables à cause de leur nature numérique. De plus, grâce à leur aspect multispectral, l'acquisition à des intervalles quasi-réguliers dans le temps et la possibilité d'accéder à des lieux inaccessibles par les moyens traditionnels, la télédétection est devenue la source principale d'informations. L'usage de ce type de données a ouvert la porte vers l'exploitation d'un nombre important de techniques de traitement qu'on ne pouvait pas utiliser auparavant. Par conséquent, il y a eu une amélioration notable de l'exploration des propriétés du sol et donc une meilleure interprétation des changements qui peuvent l'affecter.

La télédétection spatiale peut être définie comme étant la science qui capture l'information à partir d'un objet sans un contact physique avec lui. Elle fournit deux types de données :

1. Les images *SAR* (Synthetic Aperture Radar) acquises dans la partie des hyperfréquences du spectre issues de la télédétection active où le capteur envoie des impulsions et reçoit la réponse rétrodiffusée par les objets au sol. Ces données obtenues à des intervalles courts dans le temps, permettent d'étudier les phénomènes rapides liés essentiellement aux facteurs naturels (Almutairi, 2000).
2. Les images optiques multispectrales issues de la télédétection passive sont acquises dans la partie visible et infrarouge du spectre. Elles sont obtenues en mesurant l'énergie réfléchie à partir des objets illuminés par le rayonnement solaire. Ces images contiennent alors, des informations sur les propriétés physiques et chimiques du sol. Généralement, elles sont utilisées pour la cartographie et pour l'étude des phénomènes subtils qui se déroulent dans le couvert de la terre (Almutairi, 2000).

Le principe de la détection de changement est fondé sur l'identification des différences des états d'un objet, d'une surface ou d'un processus en l'observant à différents instants (Lambin et Stahler, 1994). C'est donc une comparaison des images multitemporelles de la scène étudiée pour extraire les zones qui ont subi des changements. Cette fonction devient de plus en plus complexe du fait que l'on veut détecter en plus du changement et de sa position, son

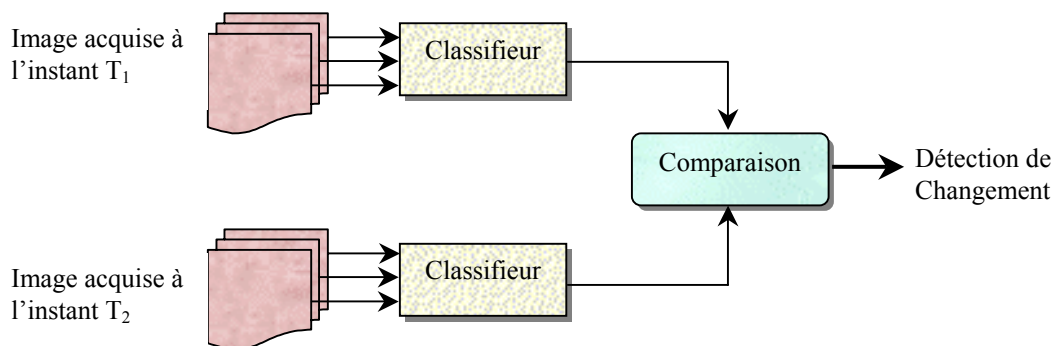
étendue spatiale et sa nature pour en déduire les causes. Les techniques utilisées dans ce cadre sont nombreuses et se basent sur les prémisses suivantes :

1. Le changement dans les propriétés du sol, se manifeste par un changement dans les valeurs radiométriques des pixels.
2. Les changements dans les valeurs radiométriques causés par les changements du sol sont plus importants que ceux qui sont dus aux autres facteurs (notons la différence dans les conditions atmosphériques, la différence dans l'angle d'incidence, le bruit dû aux capteurs, etc.) (Almutairi, 2000).
3. Il faut distinguer les véritables changements de ceux qui sont liés aux autres facteurs (cités en 2) et des changements prévus ou naturels comme les changements saisonniers.

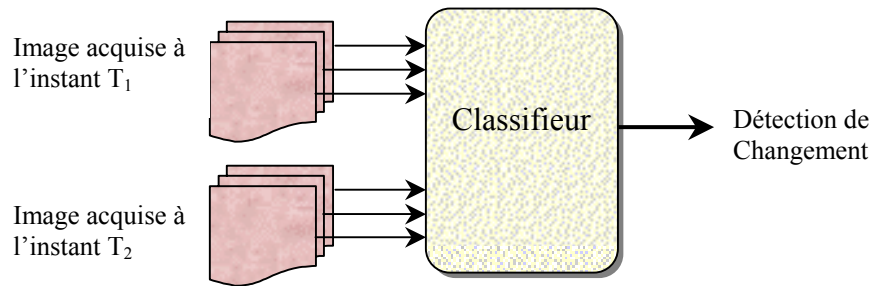
Généralement, toutes les méthodes de détection de changement sont utilisées selon deux approches. L'approche procédant par *analyse simultanée* des données multitemporelles et l'approche procédant par *analyse comparative* de ces données (Lambin et Stahler, 1994). Par ailleurs, il existe une autre classification des méthodes de détection de changement qui se base sur le niveau de l'information utilisée. On distingue alors trois niveaux : *le niveau pixel*, *le niveau primitif* et *le niveau objet* (ces notions seront détaillées plus loin).

Nous nous intéressons dans cette étude aux méthodes capables de détecter plusieurs thèmes de changement. L'analyse comparative est intuitivement utilisée car elle fournit une information complète sur les modifications de la scène. Cette approche concerne en effet, la comparaison par post classification (*CPC*) qui est fondée sur la comparaison des images préalablement classifiées pour identifier les classes de changement. Bien que simple à mettre en œuvre, elle possède malheureusement certaines limitations. D'une part, la classification indépendante des images prises à plusieurs dates ne permet pas de montrer les changements subtils à l'intérieur d'une classe (Lambin et Stahler, 1994). D'autre part, les erreurs produites lors de la classification altèrent la précision de la détection de changement (Deer, 1998). En revanche, l'approche d'analyse simultanée utilisant les méthodes de classification fournit le même niveau d'information et peut être plus robuste.

Ainsi, notre étude s'orientera vers la comparaison des performances des analyses simultanée et comparative de données.



Détection de changement par analyse comparative



Détection de changement par analyse simultanée

Néanmoins, les méthodes conventionnelles de la classification sont des techniques statistiques impliquant des hypothèses sur les données. Par exemple la classification par le maximum de vraisemblance qui est l'une des méthodes les plus utilisées suppose que les données ont une distribution gaussienne. Dans ce cas, la performance de la classification dépend de la qualité des données d'apprentissage utilisées (Foody et al., 1995). Pour éviter ce problème plusieurs techniques sont proposées incluant les classifieurs non paramétriques et les classifieurs flous.

Cette thèse est organisée comme suit :

Dans le premier chapitre nous présentons les différentes méthodes utilisées pour la détection de changement dans les images multispectrales de télédétection. Nous spécifions les approches d'analyse simultanée et comparative et les classes opérant au niveau pixel, primitif et objet. Nous rappelons aussi, les prétraitements appliqués aux images avant l'application d'une méthode de détection de changement.

Dans le deuxième chapitre nous exploitons la méthode de classification par les réseaux de neurones pour détecter les changements. L'apprentissage est effectué en premier lieu par l'algorithme standard de rétropropagation du gradient. Pour améliorer cette étape, nous utilisons une variante fondée sur l'utilisation du filtrage de Kalman (dont les équations d'établissement sont données dans l'annexe A). Ainsi, nous effectuons une évaluation des résultats d'apprentissage et de détection de changement des deux algorithmes et nous comparons entre autres les performances des analyses simultanée et comparative. Nous utilisons la matrice de confusion et les courbes *ROC* (dont les méthodes de calcul sont données respectivement dans les annexes B et C) pour l'évaluation des résultats.

Le troisième chapitre est consacré à l'application d'une méthode de classification *Soft* découlant de la théorie des ensembles flous. Cette approche est utilisée pour surmonter les limitations de la comparaison par post classification traditionnelle. Nous évaluons d'abord les performances des deux approches de détection de changement en exploitant la distance de Mahalanobis pour établir le modèle du degré d'appartenance. Puis, afin d'améliorer la performance des résultats obtenus, nous proposons un autre modèle pour le calcul du degré d'appartenance fondé sur une combinaison neurofloue. Pour évaluer ses performances, nous la comparons par rapport à l'approche neuronale.

Après avoir effectué les différents tests sur des images multitemporelles ainsi que sur des images de simulation il s'est révélé qu'aucun détecteur n'est capable de fournir les taux de bonne reconnaissance les plus élevés dans toutes les classes d'intérêt. Nous abordons alors, le problème d'amélioration de la détection de changement dans le quatrième chapitre. Nous nous inspirons des méthodes de combinaison (ou fusion) de classifieurs pour améliorer les résultats obtenus auparavant. Ainsi, deux tests de combinaison sont effectués. Dans le premier

nous combinons trois réseaux de neurones opérant par l'approche d'analyse simultanée. Ensuite, dans le second test nous établissons la combinaison des analyses comparative et simultanée en exploitant le classifieur flou. Nous utilisons pour cela, la méthode de combinaison par les intégrales floues qui présente l'avantage de recalculer les importances (ou les poids) de tous les classifieurs à chaque décision.

En conclusion, nous rappelons les objectifs du travail et les perspectives de recherche.

Chapitre 1

Techniques de Détection de Changement

Résumé: Plusieurs facteurs comme les variations radiométriques et les distorsions géométriques, influencent sur la qualité des images et rendent l'analyse du changement une tâche fastidieuse. Ceci nous impose de jeter un coup d'œil sur les techniques utilisées pour éliminer leurs effets. Ce chapitre résume en premier lieu les différentes techniques de détection de changement dans les images optiques de télédétection. Ensuite, une brève revue est introduite sur les prétraitements appliqués aux images avant l'application d'une méthode de détection de changement.

1.1 Approches de détection de changement

Plusieurs classifications des méthodes de détection de changement existent dans la littérature. Lambin et Stahler (1994) partagent les différentes méthodes en deux approches liées à la façon selon laquelle on traite les données en entrée. La première regroupe toutes les méthodes procédant par analyse simultanée des données multitemporelles (il s'agit des méthodes où les images multitemporelles subissent le même traitement) et, la seconde approche est celle des méthodes qui effectuent une analyse comparative des données (La comparaison par post classification). Dai et Khorram (1997) indiquent que les méthodes de détection de changement se divisent principalement en deux autres approches en se référant cette fois-ci à la nature du résultat. On distingue alors, les méthodes qui fournissent des masques de changement et les méthodes qui font une extraction catégorique du changement. La première approche regroupe donc les méthodes qui discriminent les changements des non changements en utilisant un seuil alors que la seconde regroupe les méthodes qui fournissent la nature du changement. Cette dernière concerne particulièrement trois méthodes : l'analyse du vecteur de changement, la comparaison par post classification et la classification multitemporelle. Par ailleurs, Deer (1998) propose une autre classification qui se base sur le type ou le niveau de l'information utilisée dans le processus de détection. On distingue ainsi trois classes :

- *La classe opérant au niveau pixel (bas niveau):* Elle regroupe toutes les méthodes qui utilisent directement les valeurs des pixels dans le processus de détection de changement. Cette classe concerne donc un nombre important de techniques comme la différence, le rapport, l'analyse composite, etc.
- *La classe opérant au niveau primitif (niveau transformée ou caractéristique):* Cette classe regroupe les méthodes qui se basent sur les résultats issus d'une transformation des valeurs des images pour détecter les changements. La transformation peut avoir une interprétation significative radiométrique (indice de végétation) ou spatiale (les lignes et les contours) ou bien, une interprétation non significative (comme les composantes de l'analyse principale et la texture) (Deer, 1998). Elle peut être en outre, linéaire ou non linéaire. Plusieurs études ont été effectuées dans ce cadre (Hayes et Sader, 1999 ; Caroll, 2000 ; Fisher et Levien, 2001 a, b).

- *La classe opérant au niveau objet (haut niveau)* : Le principe est de faire la classification indépendante de chaque image et comparer ensuite les étiquettes des pixels pour illustrer les changements. Cette approche concerne donc la comparaison par post classification et la classification multitemporelle.

Nous introduisons dans la section suivante les méthodes qui sont largement utilisées pour la détection de changement dans les images satellitaires.

1.2 Méthodes de détection de changement

1.2.1 Différence des images

Une méthode directe et très simple pour détecter les changements consiste à calculer la différence entre les valeurs d'un pixel à deux instants. Le changement se manifeste par un résultat positif ou négatif alors qu'une valeur proche de zéro indique l'absence du changement. Cette technique est très sensible aux variations radiométriques entre les images. De plus, le choix de la bande spectrale est lié au type du changement que l'on veut étudier. Si l'intérêt porte sur les changements des surfaces végétales, généralement, on utilise la bande du proche infrarouge. En revanche, la bande rouge est efficace pour l'étude des changements dans les zones urbaines (Almutairi, 2000). Plusieurs variantes ont été proposées pour améliorer cette technique comme la différence des logarithmes des variances locales (Wiemker, 1997 ; Canty et Niemeys, 1998), la différence sélective (Bruzzone et Serpico, 1997), la différence normalisée (Fisher et Levien, 2001b ; Sung et Byung, 2001 ; Sung et al., 2002). Cependant, la contrainte principale de cette technique est le choix difficile du seuil qui dépend des paramètres statistiques comme la moyenne et la déviation standard des pixels (Almutairi, 2000 ; Sung et Byung, 2001 ; Sung et al., 2002).

1.2.2 Différence des indices de végétation

C'est une technique très utilisée pour étudier l'évolution du couvert végétal au cours du temps. Plusieurs indices de végétation ou *NDVI* sont proposés dans la littérature mais le plus utilisé est la différence normalisée entre les bandes rouge et proche infrarouge. Ainsi, les modifications du couvert végétal sont mises en évidence en prenant la différence entre les indices de végétation de deux dates. Fisher et Levien (2001, a, b) ont noté que cette méthode est très efficace pour ce type d'étude. Par ailleurs, Lambin et Stahler (1994) ont utilisé les images *NDVI* au lieu des images originales pour décrire l'évolution de cet indice à travers son vecteur de changement. Ils révèlent que l'indice de végétation fournit une interprétation universelle des processus de changement et peut aboutir à une définition des directions inhérentes et stables du changement.

1.2.3 Rapport des images

Cette méthode consiste à calculer le rapport entre les valeurs d'un pixel ou d'un masque de pixels en utilisant une seule bande. Si les images sont normalisées le changement est détecté quand le rapport est différent de 1. Cacdac (1998) a rapporté que dans ce cas le contraste entre les pixels de changement et les pixels sans changement est élevé et le résultat est donc plus clair que celui de la différence. Cette méthode est très utilisée pour détecter les changements dans les images *SAR* car elle permet de réduire les effets du *Speckle*.

1.2.4 Analyse en composantes principales

L'analyse en composantes principales (*ACP*) est utilisée dans le traitement des images de télédétection pour réduire l'espace de données en ne gardant que les informations significatives représentées dans un nouvel espace de façon qu'elles soient non corrélées. Le calcul des composantes principales exploite les vecteurs propres et les valeurs propres de la matrice de covariance (ou de corrélation pour l'*ACP* standardisée) extraite des images initiales. Après cette transformation, il y a projection orthogonale sur le plan principal qui doit être calculé avec soin pour être le plus discriminant possible. La méthode de calcul des composantes principales est donnée par (Richards, 1986) .

L'analyse en composantes principales fait partie des méthodes initialement utilisées pour la détection de changement dans les images multispectrales. Elle consiste à calculer les composantes principales en combinant les différentes bandes des images multitemporelles. Généralement, le résultat obtenu sous forme d'un vecteur d'axes principaux de la même dimension que le vecteur initial est évalué visuellement. Comme le premier axe regroupe les pixels corrélés (les pixels inchangés) et les derniers axes sont toujours noyés dans le bruit, les changements apparaissent dans les axes 2, 3 et 4. Almutairi (2000) a noté que dans l'image composite issue de la combinaison de ces axes chaque thème de changement prend une couleur particulière. Il a conclu que cette technique est efficace pour détecter les changements dans les zones urbaines. Par ailleurs, d'autres études ont présenté des évaluations quantitatives de cette méthode. D'une part, Huang et al. (2000) proposent l'utilisation d'un seuil pour chaque composante principale et, d'autre part Canty et Niemeyer (1999) proposent une méthode Bayésienne pour le choix automatique du seuil. En outre, d'autres formes d'utilisation de cette technique existent. Notons la variante qui utilise une seule bande spectrale de chaque date et fournit donc deux composantes principales où la première composante représente les pixels inchangés et la seconde représente les pixels de changement. Ainsi, plusieurs façons d'interprétation des différents axes de changement sont proposées dans différentes études (Wiemker, 1997 ; Wiemker et al., 1997 ; Canty et Niemeyer, 1999).

L'analyse en composantes principales permet donc d'effectuer la détection automatique des changements. Cependant, son inconvénient majeur réside dans le fait que les dernières composantes sont fortement bruitées et peuvent ainsi, masquer certaines informations pouvant se trouver à l'intérieur.

1.2.5 Analyse du vecteur de changement

Le vecteur de changement est la différence entre les trajectoires (valeurs) successives dans le temps d'un pixel dans l'espace multispectral. Le vecteur calculé possède une amplitude et un angle qui indiquent respectivement le taux et la nature du changement. Théoriquement cette technique est applicable dans un espace de données de dimension quelconque mais dans la pratique on utilise l'espace *rouge-proche infrarouge*. Almutari (2000) a noté que c'est une technique efficace pour la détection de changement. Lambin et Stahler (1994) ont utilisé la distance de Mahalanobis au lieu de la distance euclidienne combinée avec l'analyse en composantes principales. Ils ont obtenu des résultats suffisants pour détecter et catégoriser les changements interannuels entre les trajectoires de l'indice de végétation. Huang et al. (1998, 2000) ont rapporté que l'analyse du vecteur de changement (*AVC*) appliquée dans l'espace *rouge-proche infrarouge* est suffisante pour détecter les changements dans les zones urbaines et végétales.

1.2.6 Comparaison par post classification

Durant plusieurs années la comparaison par post classification était la méthode la plus utilisée pour détecter les changements. Elle consiste à faire la classification indépendante de chaque image et identifier le changement lorsque les étiquettes d'un pixel dans les deux images classifiées sont différentes. Cette méthode fournit en plus des positions la nature et le taux de changement dans la scène. En outre, les deux types de classification supervisée et non supervisée peuvent être utilisés (Almutairi, 2000). Deer (1998) a rapporté que la comparaison par post classification produit des résultats qui ne sont pas précis et qu'elle souffre d'un certain nombre de problèmes tels que:

1. Les erreurs dans l'une des images classifiées ou dans les deux, altèrent la précision de la détection de changement (par des fausses alarmes ou des non détections).
2. La comparaison des classifications de plusieurs dates ne permet pas de détecter les changements subtils à l'intérieur d'une classe.

En revanche cette méthode permet de réduire quelques fausses détections liées généralement aux facteurs naturels tels que les changements saisonniers. L'évaluation de cette technique se fait souvent visuellement dans l'image issue de la comparaison.

1.2.7 Analyse composite multitemporelle

Cette technique est utile pour la détection de changement lorsqu'on ne possède pas des connaissances sur le terrain. Elle est simple à mettre en œuvre et consiste à combiner des images bitemporelles ou multitemporelles pour faire une classification non supervisée de la zone étudiée. L'image obtenue en sortie contient automatiquement des classes de changement et des classes de non changement. Almutairi (2000) a appliqué cette méthode en utilisant l'algorithme *ISODATA* de clustering et a effectué une partition des images bitemporelles en 50 clusters dont 8 représentaient des classes de changement. Il a noté que cette technique ignore un taux assez important de changement dans la scène.

1.2.8 Analyse de l'intercorrélation

L'analyse de l'intercorrélation (cross correlation analysis : *CCA*) permet de surmonter plusieurs limitations des méthodes conventionnelles de détection de changement. Son avantage provient de sa capacité à éliminer les problèmes liés à la variation radiométrique entre les deux images comparées. En utilisant la classification de la première date, on calcule les paramètres statistiques (moyenne et variance) de chaque classe dans les bandes spectrales de l'image de la seconde date. Ensuite, pour chaque pixel on calcule le coefficient d'intercorrélation entre sa valeur dans la seconde image et les paramètres statistiques de sa classe. Un changement se manifeste par une valeur importante de ce coefficient. Civco et al. (2002) ont obtenu des résultats satisfaisants par l'application de cette technique sur des images TM. Cependant, la précision de ces résultats est fortement liée à la qualité de classification de l'image de la première date.

1.2.9 Autres techniques

Le choix d'une méthode de détection de changement est lié principalement au type de changement et parfois à la base de données disponible. Dans le but d'améliorer la précision de la détection de changement l'intégration de diverses techniques provenant de plusieurs

domaines de recherche est souvent nécessaire. Aussi, l'utilisation des informations fondées sur la texture et le contour est une autre approche pour détecter les changements dans les images. Les textures sont utilisées comme étant des transformations spatiales dont les variations temporelles fournissent des informations sur les changements. Ainsi, la comparaison des variances locales ou d'autres paramètres de texture permet de vérifier le changement indépendamment des variations radiométriques qui affectent les valeurs des pixels. En 1996 Nielson a introduit la technique *MAD* (Multivariate Alteration Detection) une transformée dont les coefficients sont calculés à partir d'une analyse canonique des images bitemporelles (Simpson, 1998). Les images résultantes sont souvent représentées en composition colorée pour mettre en évidence les changements. Par ailleurs, l'utilisation des réseaux de neurones constitue une autre voie pour détecter les changements. Un premier travail a été effectué par Dai et Khorram en 1997 (Dai et Khorram, 1997) avec le modèle du perceptron multicouche en utilisant l'approche d'analyse simultanée. Il y a eu par la suite plusieurs travaux dérivant de cette approche (Civco et al., 2000 ; Clifton, 2001 ; Civco et al., 2002). Ensuite, en 1998 Deer a utilisé la théorie des ensembles flous avec l'analyse comparative de données. Néanmoins, les méthodes classiques utilisant les compositions colorées et les indices de végétation restent valables et subissent des traitements supplémentaires pour être plus robustes.

La majorité de ces techniques se basent sur l'utilisation du seuil pour vérifier l'existence du changement. Ce paramètre est fixé soit d'une façon interactive où l'analyste fait varier les valeurs jusqu'à l'obtention d'un résultat satisfaisant, soit en utilisant des mesures statistiques comme la variance de la distance de Mahalanobis. Pour toutes ces méthodes, l'estimation de la précision est souvent compliquée. Cela provient du fait que le critère utilisé fondé sur la matrice de confusion nécessite une base de données de vérité terrain pour deux ou plusieurs dates de la même scène. Cependant, cette exigence est rarement satisfaite. Par ailleurs, la qualité de la détection de changement peut être liée à d'autres facteurs comme la précision de l'alignement géométrique des images multitemporelles. A cet effet, l'appréciation visuelle du résultat reste le moyen le plus efficace pour évaluer la qualité de la détection.

1.2.10 Etudes comparatives des méthodes de détection de changement

L'étude des diverses techniques de détection de changement montre qu'un choix judicieux dépend fortement de l'application envisagée.

- Pour l'objectif de contrôler le changement dans les zones forestières tropicales humides Hayes et Sader (1999) ont comparé les méthodes suivantes :
 - La différence des indices de végétation normalisés,
 - L'analyse en composantes principales,
 - L'analyse composite utilisant les images *NDVI*.

Ils ont évalué la précision de ces méthodes en utilisant l'analyse *Kappa* et ont obtenu le résultat suivant : L'analyse composite *RGB-NDVI* a donné la précision la plus élevée suivie de la différence des *NDVI*. L'*ACP* fournit par ailleurs, la précision la plus faible. Il est utile de signaler que les images de classification issues de la *RGB-NDVI* et de la différence des *NDVI* sont très semblables.

- Pour le but de détecter les changements dans les zones des activités nucléaires Canty et Niemeyer (1998, 1999) ont comparé les méthodes suivantes :

- La technique *MAD*,
- l'analyse en composantes principales standard et,
- l'analyse en composantes principales itérative.

Ils ont rapporté que la technique *MAD* et l'*ACP itérative* fournissent des résultats similaires et plus précis que ceux de l'*ACP standard*.

- Almutairi (2000) a étudié l'évolution des extensions urbaines en comparant les méthodes suivantes :
 - La différence des images,
 - la comparaison par post classification (avec la classification par le maximum de vraisemblance),
 - l'analyse du vecteur de changement,
 - l'analyse en composantes principales,
 - l'analyse composite multitemporelle avec l'algorithme *ISODATA*.

Il a constaté que :

1. La technique de différence utilisant les bandes rouges normalisées illustre la globalité des changements urbains.
 2. L'analyse en composantes principales fournit des résultats difficiles à interpréter.
 3. L'analyse du vecteur de changement fournit de bons résultats mais la valeur optimale du seuil est difficile à trouver.
 4. L'analyse composite multitemporelle fournit de résultats non satisfaisants.
 5. La comparaison par post classification est la meilleure méthode bien que les erreurs dans les images classifiées apportent des compromis pour plusieurs types de surfaces.
- Huang et al. (2000) ont comparé :
 - La différence des images,
 - l'analyse en composantes principales et,
 - l'analyse du vecteur de changement.

Ils ont noté que la différence des images et l'*ACP* donnaient des résultats plus précis que l'analyse du vecteur de changement. Ce résultat est lié à la résolution spatiale des images (des images *MODIS* de résolution modérée).

- Pour l'objectif de chercher la méthode la plus efficace pour détecter plusieurs classes de changement, Civco et al. (2002) ont comparé les résultats de :
 - La comparaison par post classification traditionnelle,
 - l'analyse de l'intercorrélation,
 - le réseau de neurones entraîné par l'algorithme de rétropropagation et,
 - la classification orientée objet.

Ils ont noté les résultats suivants :

1. La comparaison par post classification et l'analyse de l'intercorrélation ont donné des résultats comparables.

2. Le réseau de neurones fournit une précision suffisamment élevée de 92% mais l'évaluation sur l'image entière était moins satisfaisante en produisant plusieurs erreurs d'omission et de commission.
3. La méthode de la classification orientée objet est assez complexe. Son principe est d'effectuer une segmentation indépendante multirésolution pour chaque image et de développer ensuite, une base de règles pour déterminer les régions de changement. Cette technique est efficace pour détecter certains types de changement non détectés par les réseaux de neurones.

Par ailleurs, les auteurs ont conclu qu'il n'existe pas une méthode meilleure pour toutes les applications de détection de changement.

Ainsi, le tableau 1.1 résume les résultats des différentes comparaisons :

Références	Comparaison des méthodes	Meilleure méthode	Mauvaise méthode
Cacdac (1998)	<i>ACP</i> , différence, rapport	<i>ACP</i>	Différence
Hayes et Sader (1998)	La DIFF- <i>NDVI</i> , l'analyse composite avec <i>RGB-NDVI</i> et l' <i>ACP</i>	La <i>RGB-NDVI</i>	L' <i>ACP</i>
Canty et Niemeyer (1999, 2000)	<i>ACP</i> , <i>ACP</i> itérative, <i>MAD</i>	<i>ACP</i> itérative et <i>MAD</i>	<i>ACP</i>
Almutairi (2000)	La différence, la <i>CPC</i> , le <i>AVC</i> , l' <i>ACP</i> et l'analyse composite	La <i>CPC</i>	L'analyse composite
Huang et al (2000)	<i>ACP</i> , différence et <i>AVC</i>	<i>ACP</i>	<i>AVC</i>
Willhauck (2000)	Classification orientée objet et classification ponctuelle	Classification orientée objet	Classification ponctuelle
Civco et al (2002)	La <i>CPC</i> traditionnelle, la <i>CCA</i> , Réseau de neurones et la classification orientée objet	Classification orientée objet et le réseau de neurones	La <i>CPC</i> et la <i>CCA</i>

Tableau 1.1. Récapitulation des comparaisons des méthodes de détection de changement

1.3 Prétraitements des images

Avant l'application d'une méthode de détection de changement, il est souvent nécessaire de pré-traiter les images en y apportant des corrections géométriques et radiométriques. Cette exigence est essentielle car les images ne sont pas acquises dans les mêmes conditions d'éclairement et de prise de vue. Ainsi, nous rappelons brièvement les étapes principales pour les corriger.

1.3.1 Corrections géométriques

Les déformations géométriques sont causées par différents facteurs comme la rotation et la courbure de la terre, la non idéalité des capteurs, les variations d'attitude et d'altitude du satellite, l'effet panoramique, etc. L'élimination de ces déformations peut être déterministe en corrigeant l'effet de chaque facteur indépendamment des autres. Cependant, l'existence de plusieurs sources aléatoires de distorsion conduit généralement à opter à une correction globale qui comporte les étapes suivantes (Deer, 1998) :

1. *Le choix de la référence* : l'image est corrigée soit par rapport à une image, soit par rapport au terrain ou bien par rapport à une carte.
2. *L'établissement du modèle de correction* : cette étape consiste à extraire des points d'appui à partir de la référence et de l'image à corriger. Ces points doivent être choisis avec soin, doivent être nombreux et répartis uniformément dans l'image. On les utilise pour calculer les coefficients d'un polynôme qui sera le modèle de distorsion. Généralement, un polynôme du deuxième ordre est le plus souvent utilisé.
3. *Le rééchantillonnage radiométrique* : cette étape applique le modèle inverse de distorsion pour affecter les valeurs radiométriques aux pixels. Cette étape dite aussi interpolation se fait soit par un interpolateur K-plus proche voisin, bilinéaire ou bicubique.

Ces corrections jouent un rôle important dans les applications de détection de changement. Deer (1998) a cité quelques travaux qui ont étudié l'influence de la précision de la correction géométrique sur la précision de la détection de changement. Il a rapporté qu'une erreur inférieure à 0.2 pixels est nécessaire pour avoir un taux d'erreur inférieur à 10% dans la détection de changement.

1.3.2 Corrections radiométriques

Le rôle de la correction radiométrique est d'éliminer les variations liées essentiellement à l'atmosphère. Elle comporte les opérations suivantes (Deer, 1998):

1. *Elimination du bruit dû aux conditions externes de prise de données* : cela concerne les effets atmosphériques et topographiques. Les effets atmosphériques sont corrigés par la théorie des points noirs et les effets topographiques sont corrigés en utilisant la modélisation de la transmittance de l'atmosphère avec le *MNT* de la région.
2. *Le filtrage du bruit dû à l'instrument de mesure* : chaque constructeur fournit les paramètres de calibration des capteurs de ses satellites. (Pour plus de détails sur les corrections géométriques et radiométriques des images voir Richards, 1986).

1.3.3 Normalisation radiométrique

Bien que les corrections radiométriques et géométriques fournissent des valeurs plus précises de l'image corrigée, cependant lors du traitement multitemporel, il existe toujours des différences dans les images prises à différentes dates. Ces variations radiométriques apparaissent et posent ainsi, des problèmes pour plusieurs méthodes de détection de changement car les conditions de prise de vue ne sont jamais les mêmes. En particulier, cela concerne les méthodes qui utilisent de simples comparaisons entre les valeurs des images

(comme la différence et le rapport). Dans ce cas, la normalisation des valeurs radiométriques des images multitemporelles est indispensable. Généralement, la méthode de régression linéaire est utilisée où la référence est l'image la plus récente (Hayes et Sader, 1999 ; Kim et Lee, 2000 ; Fisher et Levien, 2001, a). La normalisation s'effectue bande par bande et consiste à extraire à partir des images deux ensembles de pixels dont les thèmes sont inchangés dans le temps comme le sable, les routes, les toits et les eaux. De plus, les valeurs des pixels de chaque ensemble doivent se trouver dans l'une des extrémités de l'histogramme. C'est à dire il faut collecter le plus grand nombre de pixels des thèmes inchangés les plus réfléchissants et les plus sombres dans la scène. On calcule ensuite la moyenne de chaque ensemble dans chaque bande et pour toutes les dates. Puis, on calcule les paramètres de l'équation de régression linéaire pour chaque bande spectrale. Si cette étape est faite avec précision, elle facilite l'application de plusieurs méthodes. Pour les méthodes différence et rapport, elle permet d'éliminer le rôle du seuil.

1.4 Zone d'étude

La zone d'étude se trouve à l'est de la ville d'Alger. Les images utilisées sont des XS fournies par le satellite *SPOT* dans les bandes vert (XS_1), rouge (XS_2) et proche infrarouge (XS_3). La première scène a été acquise en 1989 et la seconde en 1991. Cependant, ces images (représentées par les figures 1.1 et 1.2, respectivement) ne couvrent pas exactement la même zone. La première image couvre une surface de 190.32 Km² qui se trouve à l'est d'Alger alors que la seconde couvre une surface de 189.42 km² incluant une portion de la ville d'Alger. Ainsi le chevauchement entre les deux scènes s'étend sur une largeur d'environ 800 pixels. Un problème posé par cette zone de chevauchement est la présence d'une couverture nuageuse. Pour extraire une zone de recouvrement qui va nous servir à notre étude, nous avons effectué un recalage par prise de points d'appui en utilisant le logiciel *ENVI*. La zone de chevauchement contient en plus des surfaces cachées par les nuages une partie importante de la mer. Ceci dit, la sélection des points d'appui était un peu fastidieuse mais nous avons quand même pu collecter 41 points d'appui. Nous avons utilisé un polynôme du deuxième ordre et un interpolateur bicubique. En outre, nous étions obligés d'accepter une erreur globale de 0.12 pixels. Ainsi, nous avons choisi une zone de 400×400 pixels dans laquelle se trouvent les différents thèmes de la scène mais il nous a été impossible d'éviter la présence des nuages. Les deux parties extraites sont représentées par les figures 1.3 et 1.4.

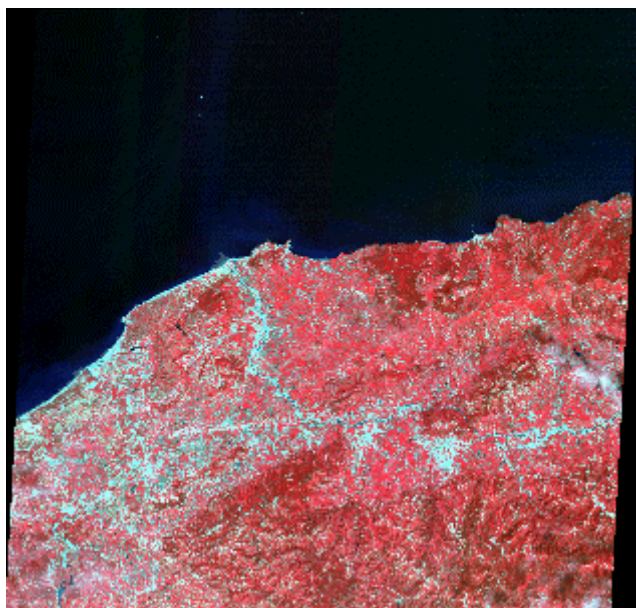


Figure 1.1. Image acquise en 1989 représentée en composition colorée



Figure 1.2. Image acquise en 1991 représentée en composition colorée

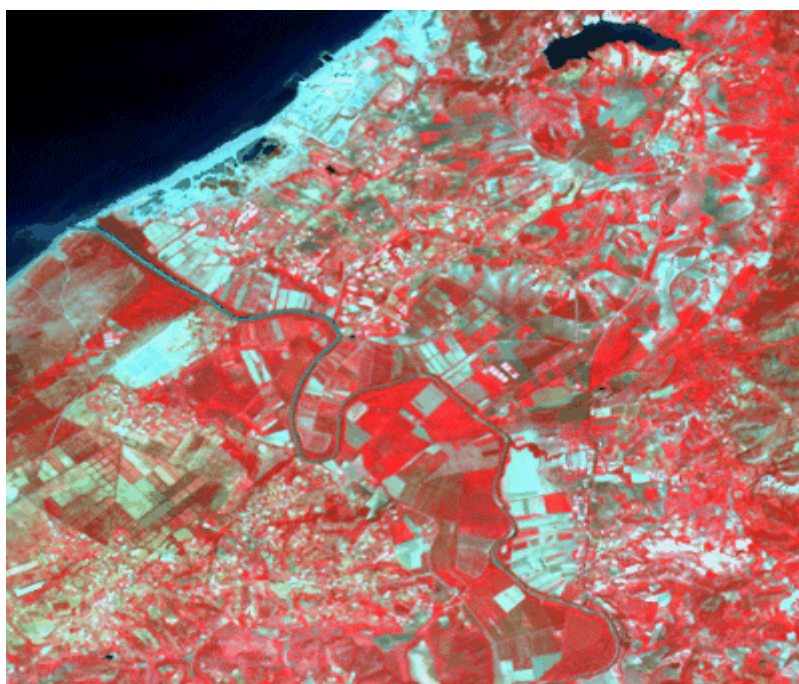


Figure 1.3. Composition colorée de la zone extraite de l'image prise en 1989

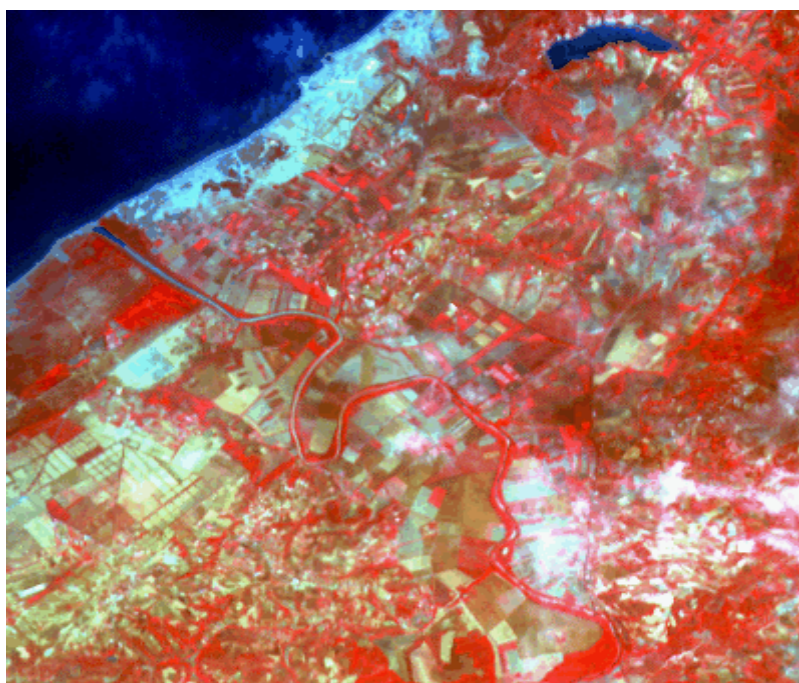


Figure 1.4. Composition colorée de la zone extraite de l'image prise en 1991

Dans toutes les méthodes testées et évaluées, nous nous sommes intéressés aux thèmes suivants : eau, construction, végétation et sol nu. Les images obtenues par la classification indépendante de chaque date sont fournies dans les figures 1.5 et 1.6.

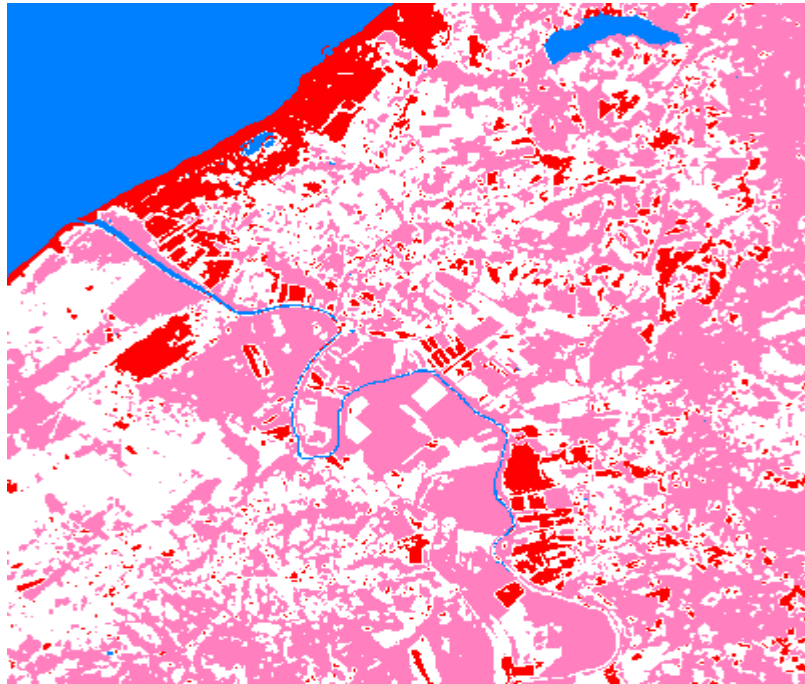


Figure 1.5. Image classifiée de la surface d'étude de 1989

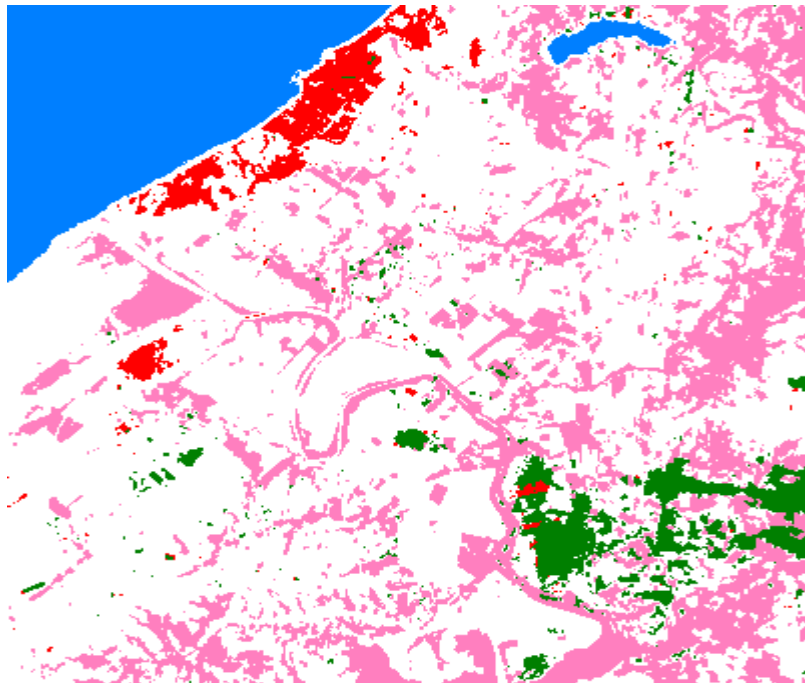


Figure 1.6. Image classifiée de la surface d'étude de 1991

	Eau,		construction,		végétation,		sol nu,		nuages
---	------	---	---------------	---	-------------	---	---------	---	--------

1.5 Conclusion

Le choix de la technique utilisée lors d'une application de détection de changement dépend de l'information recherchée et de la base de données disponible. Il s'est avéré que les procédures de classification appliquées avec les approches simultanée et comparative offrent une information entière sur le changement. Cependant, les classifieurs conventionnels présentent l'inconvénient d'être fondés sur l'utilisation des hypothèses statistiques sur les données. Ainsi, dans les chapitres suivants, nous effectuerons des études comparatives entre les deux approches en exploitant des techniques non paramétriques de la classification supervisée.

La dispersion des nuages dans l'image de la seconde date étant importante, nous avons décidé de prendre en considération dans notre étude la classe ($X \rightarrow \text{nuages}$). Par ailleurs, nous avons validé les méthodes appliquées sur des images de simulation. Quelques résultats de ces tests sont donnés dans l'annexe *D* et sont ordonnés par chapitre.

Chapitre 2

Utilisation des Réseaux de Neurones pour la Détection de Changement

***Résumé :** Ce chapitre a pour objectif la comparaison des performances des analyses simultanée et comparative pour la détection de changement en exploitant les réseaux de neurones. Le modèle utilisé est le perceptron multicouche entraîné par l'algorithme standard de rétropropagation du gradient de l'erreur. Pour accélérer l'apprentissage du réseau nous utilisons le filtrage de Kalman comme règle d'apprentissage. Les deux algorithmes sont comparés en termes de rapidité dans la phase d'apprentissage et de précision dans la phase d'exécution du processus de détection de changement.*

2.1 Introduction

La majorité des méthodes utilisées dans la littérature n'indiquent pas la nature de la modification qui est une information cruciale pour toute application de détection de changement. Par ailleurs, pour pouvoir établir une comparaison entre l'analyse simultanée et l'analyse comparative il faut que le processus employé dans les deux soit le même. Ainsi, l'utilisation des techniques de classification s'avère indispensable. Intuitivement, les techniques paramétriques conventionnelles de la classification supervisée sont choisies. Ces méthodes statistiques nécessitent des hypothèses puissantes sur les données et font appel à des modélisations particulières. Par exemple, la classification par le maximum de vraisemblance utilise une distribution Gaussienne sur les classes. Puisque ces méthodes sont gourmandes en calculs et en données nous avons préféré l'approche neuronale. Cette approche utilisée avec succès dans le domaine de la classification possède, en outre, une grande capacité de généralisation (Touzet, 1992).

La nécessité d'avoir des outils souples pour le traitement automatique de l'information était à la base de l'essor des réseaux de neurones. L'idée était de comprendre le mécanisme de traitement de l'information dans le cerveau humain et s'en inspirer pour établir une modélisation mathématique utile pour diverses applications.

L'objectif de ce chapitre consiste à comparer l'efficacité de l'approche d'analyse simultanée à celle de l'approche d'analyse comparative pour la détection de changement. Nous utilisons pour cela, la technique de classification par le perceptron multicouche dont le principe est donné dans la section suivante.

2.2 Principe de la détection de changement par le réseau multicouche

Lors d'une application de classification l'architecture du réseau multicouche est organisée comme suit : chaque nœud dans le vecteur d'entrée correspond à une bande spectrale et chaque nœud dans la couche de sortie correspond à l'une des classes d'intérêt. Le nombre de couches cachées et le nombre de nœuds cachés sont fixés par l'utilisateur (Figure 2.1). L'apprentissage est la phase de développement d'un réseau de neurones durant laquelle son comportement est modifié jusqu'à l'obtention du comportement désiré. L'apprentissage supervisé fait appel à des exemples de comportement sous forme de paires d'entrée / sortie. Il procède par la modification des poids synaptiques du réseau en accordant la réponse de ce dernier aux exemples selon un algorithme d'entraînement. A l'issue de l'apprentissage, les poids sont fixés : c'est alors la phase d'utilisation du réseau.

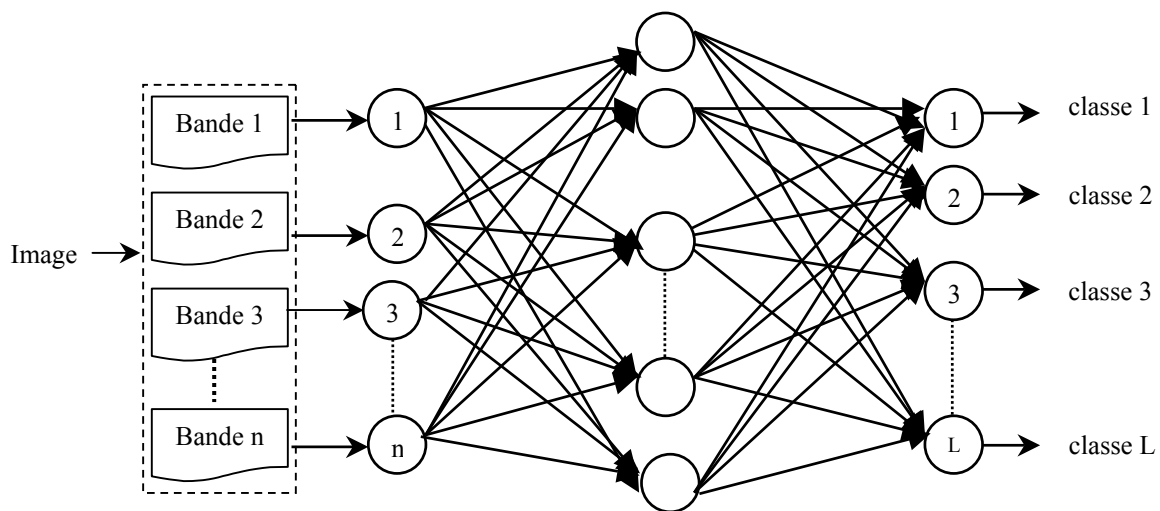


Figure 2.1. Principe de la classification par un réseau à une couche cachée

Par ailleurs, si on voudrait utiliser un perceptron multicouche pour effectuer la détection de changement par analyse simultanée de données, le principe reste le même que celui de la classification. Les images multitemporelles sont alignées pour former le vecteur d'entrée du réseau. En revanche, les classes d'intérêt sont des classes de changement et des classes de non changement (Figure 2.2). Pour des données bitemporelles on distingue par exemple les classes de changement comme étant des classes du type :

végétation (dans la première image) → *sol* (dans la seconde image).

Et les classes de non changement telles que :

végétation (dans la première image) → *végétation* (dans la seconde image).

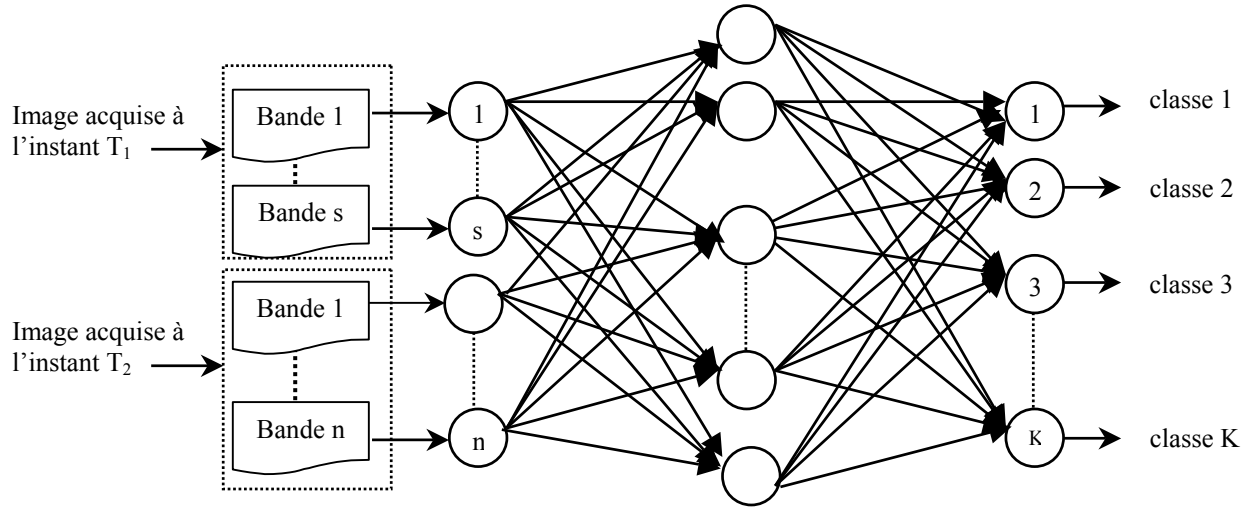


Figure 2.2. Principe de la détection de changement par le réseau de neurones

2.3 Algorithmes d'apprentissage du réseau multicouche

L'algorithme de rétropropagation du gradient est la règle standard d'apprentissage du perceptron multicouche. Son principe consiste à modifier les poids synaptiques du réseau en minimisant l'erreur quadratique moyenne E entre la sortie actuelle et la sortie désirée (respectivement : Y , Y_d) du réseau.

$$E = \sum_{j=1}^M (Y_j - Y_{d_j})^2 \quad (2.1)$$

M : nombre d'échantillons.

Les poids sont modifiés selon l'équation (2.2).

$$w(t+1) = w(t) + \Delta w(t+1) \quad (2.2)$$

$\Delta w(t+1)$ est le changement d'un poids à l'itération $t+1$. Il est calculé par :

$$\Delta w(t+1) = -\eta \frac{\partial E}{\partial w} + \alpha \Delta w(t) \quad (2.3)$$

η : pas d'apprentissage.

α : momentum.

L'utilisation du perceptron multicouche pour la détection de changement date de quelques années et reste encore un thème de recherche. Dai et Khorram (1997) ont utilisé un perceptron avec deux couches cachées pour étudier 16 classes dont 12 sont des classes de changement. Ils ont rapporté que le réseau de neurones montre une grande efficacité comme technique de détection de changement mais l'algorithme de rétropropagation nécessite une longue durée d'apprentissage ce qui n'est pas adéquat lors de l'étude des changements rapides. Pour étudier l'évolution du couvert forestier Civco et al. (2000) ont utilisé un perceptron comportant une

seule couche cachée entraîné par l'algorithme de rétropropagation du gradient. Ils ont pris en considération trois classes : forêt → forêt, forêt → construction et forêt → parking et ont noté des résultats satisfaisants avec 81 nœuds cachés. Cependant, la généralisation sur la scène entière n'avait pas la même précision.

Il est connu que l'algorithme de rétropropagation souffre de plusieurs limitations. On note particulièrement, la lenteur de la phase d'apprentissage du réseau. Scalero et Tepedelenlioglu (1992) disent que la présence de la fonction non linéaire entre les poids et l'erreur est la raison principale pour laquelle l'algorithme de rétropropagation est lent. Alors ils ont proposé une alternative qui ne prend en considération que la partie linéaire de chaque nœud (Figure 2.3). La détermination des poids devient un problème linéaire car l'erreur est minimisée aux entrées des non-linéarités en utilisant le filtre de Kalman.

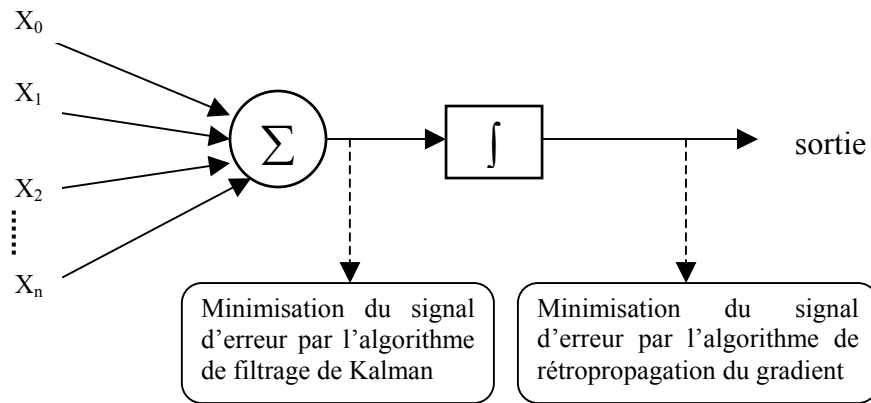


Figure 2.3. Minimisation de l'erreur au niveau d'un nœud

Dans ce cas le changement d'un poids est calculé par :

$$\Delta w(t+1) = k_j e_j \mu_j \quad (2.4)$$

e_j : signal d'erreur dans la couche j .

μ_j : pas d'apprentissage.

k_j : gain de Kalman calculé par l'équation (2.5).

$$k_j(t+1) = \frac{R_j^{-1}(t)x_{j-1}(t+1)}{b_j + x_{j-1}^T(t+1)R_j^{-1}(t)x_{j-1}(t+1)} \quad (2.5)$$

R : matrice d'intercorrélation des données d'apprentissage.

x_{j-1} : vecteur d'entrée de la couche $j-1$.

b_j : facteur d'oubli de la couche j . (Le calcul intégral des deux algorithmes est donné dans l'annexe A).

2.4 Résultats expérimentaux

Plusieurs paramètres sont pris en considération lors de l'apprentissage du réseau. Généralement, les algorithmes d'apprentissage du perceptron multicouche exigent que les données soient arrangées dans l'intervalle $[0, 1]$. Dans notre application nous avons utilisé la normalisation linéaire des radiométries en les divisant par 255. Par ailleurs, les paramètres d'établissement doivent être choisis avec soin pour offrir le meilleur résultat possible pour chaque algorithme. L'initialisation déterministe des poids peut étendre la durée d'apprentissage d'un réseau de petite taille jusqu'à plusieurs jours. Alors nous avons initialisé les poids par des valeurs aléatoires, petites et non nulles pour accélérer la convergence.

Hormis les couches d'entrée et de sortie, on doit fixer le nombre de couches intermédiaires. Sans couches cachées, le réseau n'offre que de faibles possibilités d'adaptation. Par contre avec une couche cachée il est capable avec un nombre suffisant de neurones d'approximer toute fonction continue (discriminer les frontières linéaires entre les classes). En revanche, une seconde couche cachée prend en compte les discontinuités éventuelles (les frontières non linéaires entre les classes) et peut ainsi, approximer n'importe quelle fonction (Herauld et Jutton, 1994). Cette seconde couche est nécessaire pour les applications de détection de changement (Dai et Khorram, 1997). Nos tests seront donc limités aux réseaux à une et à deux couches cachées.

2.4.1 Evaluation des règles d'apprentissage

Dans (Scalero et Tepedelenlioglu, 1992) les performances de l'algorithme de rétropropagation ont été évaluées comparativement à celles de l'algorithme de filtrage de Kalman à travers une application de classification de données. Les résultats obtenus montrent que l'algorithme de filtrage de Kalman est plus efficace. Pour notre part, nous avons choisi d'évaluer les performances de ces deux algorithmes lors du traitement multitemporel en utilisant l'approche d'analyse simultanée.

2.4.1.1 Critères d'évaluation

Les performances des deux algorithmes sont évaluées dans la phase d'apprentissage ainsi que dans la phase de détection de changement. Les tests porteront sur la vitesse de convergence, sa sensibilité envers la variation de l'architecture du réseau ainsi que l'influence de la durée d'apprentissage sur la capacité discriminante du réseau. Les paramètres d'apprentissage des deux algorithmes sont fixés comme suit :

- Algorithme de filtrage de Kalman (FK) :
 - Pas d'apprentissage = 300.
 - Facteur d'oubli = 0.995.
- Algorithme de rétropropagation (RP):
 - Momentum = 0.995.
 - Pas d'apprentissage = 0.08.

Nous nous sommes intéressés aux classes citées dans le tableau 2.1. Nous avons sélectionné pour chacune un ensemble de données contenant les pixels représentatifs. Ces données sont partagées en deux ensembles, un pour l'apprentissage et l'autre pour les tests. A cause du manque des données terrain, nous utilisons la signature spectrale des classes comme référence pour établir la matrice de confusion.

Classes	Description	Données	
		Apprentissage	Test
1	<i>Eau → eau</i>	200	250
2	<i>Végétation → végétation</i>	200	260
3	<i>Construction → construction</i>	200	270
4	<i>Sol nu → sol nu</i>	200	250
5	<i>Construction → sol nu</i>	200	280
6	<i>Végétation → sol nu</i>	200	260
7	<i>Eau → sol nu</i>	200	230
8	<i>X → Nuages</i>	200	230

Table 2.1. Données de la base d'entraînement

2.4.1.2 Vitesse de convergence

Dans ce test nous comparons l'évolution de l'erreur quadratique moyenne à la sortie du réseau au cours du temps. Le résultat est représenté par les figures 2.4 et 2.5.

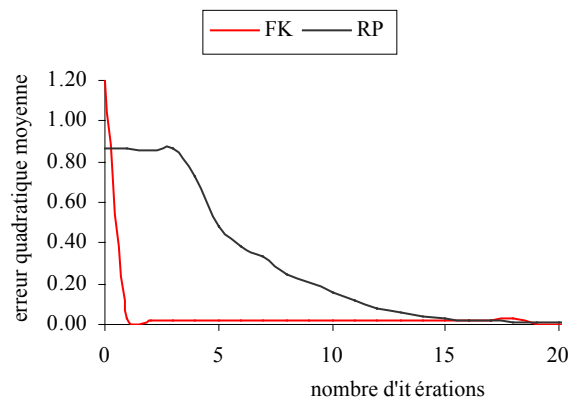


Figure 2.4. Evolution de l'erreur quadratique moyenne au cours du temps pour un réseau à une couche cachée comportant 20 nœuds

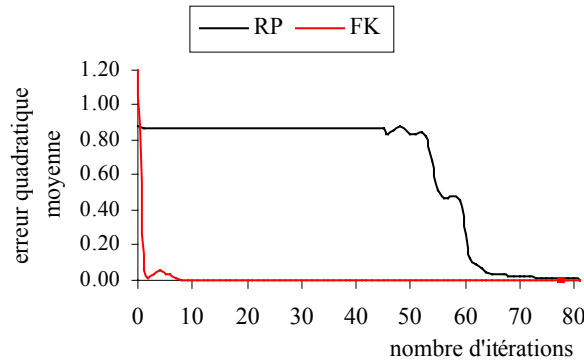


Figure 2.5. Evolution de l'erreur quadratique moyenne au cours du temps pour un réseau à deux couches cachées comportant 20 nœuds chacune

Nous constatons que l'algorithme FK est plus rapide que l'algorithme RP. Par ailleurs, le fait de rajouter une deuxième couche cachée ralentit énormément l'apprentissage par l'algorithme RP. En revanche, l'algorithme FK garde la même vitesse de convergence. Ce constat confirme celui de la simulation (voir l'annexe D : figures D.2, D.3).

2.4.1.3 Influence du nombre de nœuds

Pour illustrer l'influence du nombre de nœuds sur la convergence du réseau et donc sur la vitesse des deux algorithmes, ce test consiste à chercher le nombre d'itérations nécessaires pour atteindre une erreur quadratique moyenne prédéterminée en faisant varier à chaque fois le nombre de nœuds cachés. Les résultats obtenus sont représentés par les figures 2.6 et 2.7 pour un seuil d'erreur fixé à 0.1.

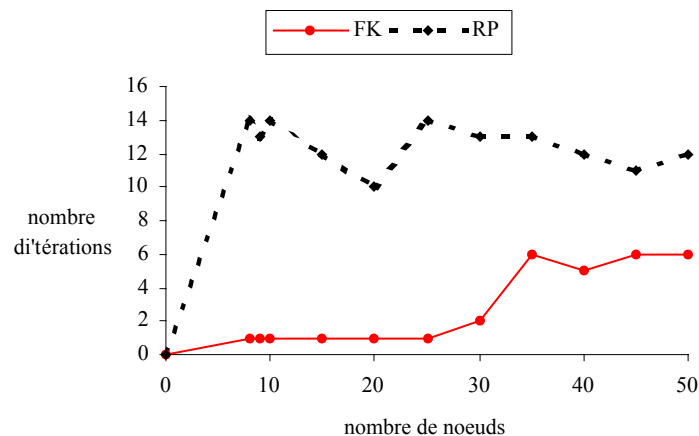


Figure 2.6. Influence du nombre de nœuds pour un réseau à une couche cachée

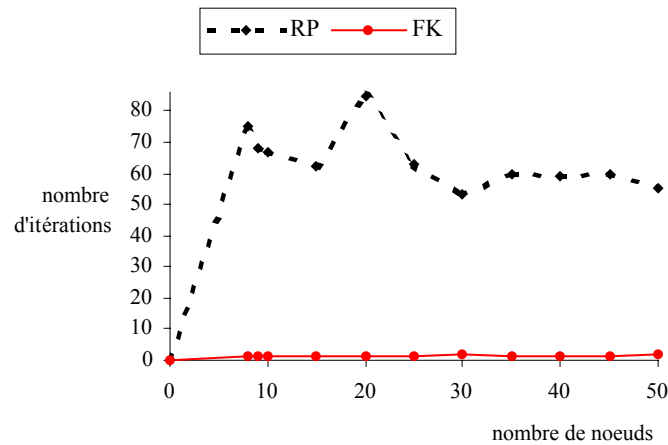


Figure 2.7. Influence du nombre de nœuds pour un réseau à deux couches cachées

Nous constatons que le réseau entraîné par l'algorithme FK atteint le seuil après un nombre d'itérations variant entre 1 et 6 avec une couche cachée et variant entre 1 et 2 avec deux couches cachées. Par ailleurs, l'algorithme de rétropropagation nécessite un nombre d'itérations plus important. Ces résultats sont un peu différents des résultats obtenus dans le test de simulation où la base d'entraînement est beaucoup plus large. Dans ce test, nous avons remarqué que l'algorithme RP exige un nombre d'itérations important avec toutes les architectures alors que l'algorithme FK nécessite une ou deux itérations seulement pour atteindre une erreur quadratique fixée à 0.01 (Voir l'annexe D : figures D.4 et D.5).

2.4.1.4 Influence du nombre d'itérations

Pour montrer l'influence du nombre d'itérations sur la capacité de reconnaissance du réseau, nous avons comparé l'évolution de la statistique *Khat* (qui est une estimation du coefficient *Kappa*, voir l'annexe B) en fonction du nombre d'itérations d'un réseau entraîné par les deux algorithmes. Voici le résultat obtenu pour des réseaux qui contiennent 15 nœuds dans chaque couche cachée. Le seuil de détection est fixé à 0.9 (La réponse du réseau doit être supérieure à 0.9 pour attribuer le pixel à la classe correspondante, dans le cas contraire le pixel est rejeté).

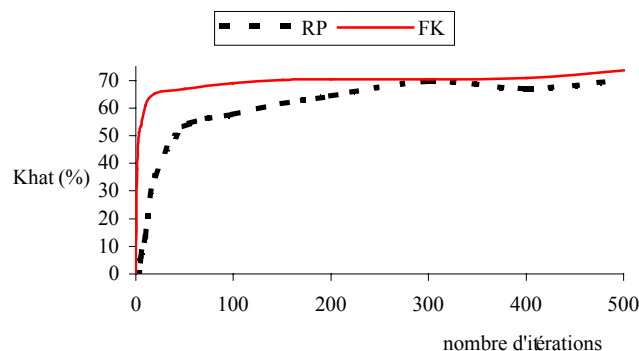


Figure 2.8. Evolution de la valeur du *Khat* en fonction du nombre d'itérations pour un réseau comportant une couche cachée

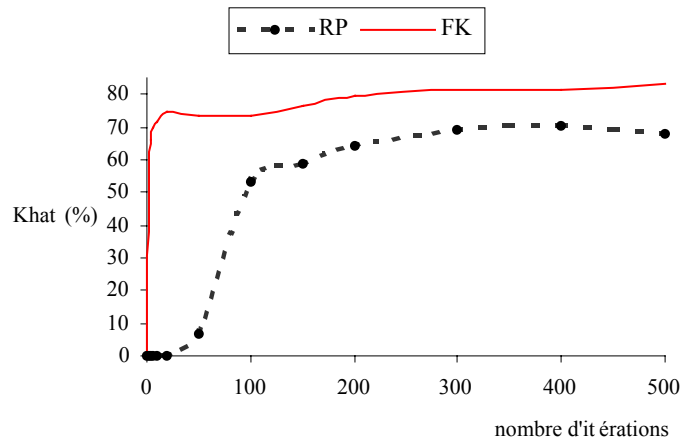


Figure 2.9. Evolution de la valeur du *Khat* en fonction du nombre d'itérations pour un réseau comportant deux couches cachées

L'examen des figures montre que pour le cas de l'algorithme FK le réseau atteint dès les premières itérations sa valeur maximale de *Khat*. Ceci exprime que cet algorithme nécessite peu d'itérations pour atteindre son point de convergence. Par contre, avec l'algorithme RP le *Khat* augmente graduellement après avoir été nul durant quelques itérations. Pour le réseau à une couche cachée on peut obtenir une valeur non nulle du *Khat* après 4 itérations. Cependant, comme le réseau à deux couches cachées nécessite une durée d'apprentissage assez longue, la valeur du *Khat* reste nulle jusqu'à l'itération 20.

2.4.2 Evaluation de la détection de changement

2.4.2.1 Analyse simultanée

2.4.2.1.1 Critères d'évaluation

Les tests de détection de changement consistent à estimer la capacité de chaque algorithme à discriminer les changements réels des non changements. Pour cela, nous utilisons les techniques suivantes : la matrice de confusion et les taux de surestimation, la courbe *ROC* et l'appréciation visuelle du résultat.

2.4.2.1.2 Matrices de confusion

Dans ce test, nous évaluons la capacité de reconnaissance des réseaux de neurones entraînés par les deux algorithmes en utilisant la matrice de confusion (ou la matrice d'erreur dont le calcul est expliqué dans l'annexe B).

- Chaque couche cachée contient 15 nœuds.
- Le seuil de détection est fixé à 0.9.
- Le nombre d'itérations est fixé à 400.

Voici les matrices de confusion obtenues par les différents réseaux (les valeurs sont données en %).

➤ Réseau à une couche cachée :

Classes	1	2	3	4	5	6	7	8
1	99.20	0	0	0	0	0	0	0
2	0	78.84	0	0	0	0	0	0
3	0	0	81.11	0	0.35	1.15	2.17	0
4	0	0	0	45.60	5.71	0	0	0
5	0	0	0.37	0.40	64.28	0	0	0
6	0	0	0	0.40	0	49.61	1.30	0
7	0	0.38	0	0.40	0	0	60.86	2.60
8	0	0	0	0.40	0	0.38	0.43	80.43
Rejet	0.80	20.76	18.51	52.80	29.64	48.84	35.21	16.95

*TMBR = 69.95%

Khat = 66.96%

Tableau 2.2. Matrice de confusion du réseau à une couche cachée entraîné par l'algorithme RP

Classes	1	2	3	4	5	6	7	8
1	99.20	0	0	0	0	0	0	0
2	0	82.30	0	0	0.71	0	0	0
3	0	0	85.92	0	0.35	1.92	3.91	0
4	0	0	0.37	64.00	5.35	0.38	0	0
5	0	0	0	0	64.28	0	0	0
6	0	0	0.37	0.80	0	76.15	3.04	0
7	0	0	0	0	0	0	55.65	2.60
8	0	0	0	0.40	0	0.38	0	57.82
Rejet	0.80	17.69	13.33	34.80	29.28	21.15	37.39	39.56

TMBR = 73.54%

Khat = 70.73 %

Tableau 2.3. Matrice de confusion du réseau à une couche cachée entraîné par l'algorithme FK

* TMBR : taux moyen de bonne reconnaissance.

➤ Réseau à deux couches cachées :

Classes	1	2	3	4	5	6	7	8
1	99.60	0	0	0	0	0	0	0
2	0	93.46	0	0.80	0.35	0	0	0.43
3	0	0	72.22	0	0.35	1.92	0.43	0
4	0	0.74	0.37	73.20	4.92	0.38	0.43	0.43
5	0	0.38	0	0.40	77.21	0	0	0
6	0	0	0	0	0	56.92	4.34	0
7	0	0	0	0	0	0	64.78	3.04
8	0	0	0	0	0	1.15	0.87	50.43
Rejet	0.40	5.40	27.41	25.60	17.14	39.61	29.13	45.65

 $TMBR = 73.79 \%$ $Khat = 70.59 \%$ **Tableau 2.4.** Matrice de confusion du réseau à deux couches cachées entraîné par l'algorithme RP

Classes	1	2	3	4	5	6	7	8
1	99.20	0	0	0	0	0	0	0
2	0	94.61	0.74	0.40	0.71	0	0	0
3	0	0.38	90.00	0	0.35	3.46	5.21	0.43
4	0	2.30	0.37	76.40	12.42	0.38	0	0.43
5	0	0	0.37	0	73.64	0	0	2.6
6	0	0	0.37	0	0	81.92	6.08	0
7	0.4	0	0	0.80	0	0.76	70.43	2.17
8	0	0	0	0.80	0	0.38	0	78.26
Rejet	0.40	2.69	8.14	21.60	12.85	13.07	18.26	16.08

 $TMBR = 83.34 \%$ $Khat = 81.26 \%$ **Tableau 2.5.** Matrice de confusion du réseau à deux couches cachées entraîné par l'algorithme FK

A partir des différentes matrices de confusion nous constatons que :

- Pour chaque algorithme la deuxième couche cachée augmente la précision du réseau.
- La précision globale ($TMBR$) la plus élevée est obtenue par le réseau à deux couches cachées entraîné par l'algorithme FK. Cependant, ce réseau ne fournit pas les meilleures précisions dans toutes les classes. Notons que pour la classe *construction* → *sol nu* (classe 5) le taux le plus élevé est obtenu à partir du réseau à deux couches cachées entraîné par l'algorithme RP.

- Ainsi, le taux de surestimation (taux d'erreur) au niveau de chaque classe peut être calculé en prenant la différence absolue (DF) entre le taux producteur (TP) et le taux utilisateur (TU) de cette classe (Le calcul du TP et du TU est donné dans l'annexe B). Le taux producteur est lié aux erreurs d'omission et le taux utilisateur exprime les erreurs de commission. Pour une classe donnée, une différence proche de zéro indique que la classe en question est bien discriminée. Dans le cas contraire, cela signifie que cette classe présente de fortes ambiguïtés avec les autres classes. Le tableau 2.6 indique le résultat obtenu à partir des matrices de confusion précédentes.

Classe	RP: 1CC*			FK: 1CC			RP: 2CC			FK: 2CC		
	TP	TU	DF	TP	TU	DF	TP	TU	DF	TP	TU	DF
1	0.99	1.00	0.01	0.99	1.00	0.01	0.99	1.00	0.01	0.99	1.00	0.01
2	0.78	1.00	0.22	0.82	0.99	0.17	0.93	0.98	0.05	0.94	0.98	0.04
3	0.81	0.96	0.15	0.85	0.93	0.08	0.72	0.96	0.24	0.90	0.91	0.01
4	0.45	0.87	0.42	0.64	0.90	0.26	0.73	0.85	0.12	0.76	0.82	0.06
5	0.64	0.98	0.34	0.64	1.00	0.36	0.77	0.99	0.22	0.73	0.96	0.23
6	0.49	0.96	0.47	0.76	0.95	0.19	0.56	0.93	0.37	0.82	0.93	0.11
7	0.60	0.94	0.34	0.55	0.95	0.40	0.64	0.95	0.31	0.70	0.94	0.24
8	0.80	0.98	0.18	0.57	0.98	0.41	0.50	0.95	0.45	0.78	0.98	0.20

Tableau 2.6. Erreurs de surestimation dans les différentes classes

Nous notons ainsi, que :

- Pour les différents réseaux les erreurs de surestimation dans les classes de changement sont assez importantes.
- Dans la plupart des classes la deuxième couche cachée permet de réduire le taux d'erreur.
- Le réseau à deux couches cachées entraîné par l'algorithme FK fournit globalement les taux de surestimation les moins élevés montrant ainsi sa grande précision.

Cependant, l'erreur de surestimation au niveau de chaque classe varie d'un réseau à un autre. Par exemple avec le réseau à une couche cachée, l'algorithme RP produit des taux d'erreur de 42% dans la classe *sol nu* → *sol nu* et de 47% dans la classe *végétation* → *sol nu*. En revanche, l'algorithme FK produit des taux d'erreur de 40% dans la classe *eau* → *sol nu* et de 41 % dans la classe *X* → *nuages*.

2.4.2.1.3 Courbes ROC

Pour évaluer l'efficacité de l'algorithme RP comparativement à l'algorithme FK en terme de bonne détection, nous utilisons la courbe ROC (qui mesure la capacité de détection d'un test, voir l'annexe C) pour des réseaux comportant 15 nœuds dans chaque couche cachée et entraînés durant 400 itérations. Les courbes sont tracées en faisant varier le seuil de détection de 1 à 0 selon un pas de - 0.01.

* 1 CC : une couche cachée.

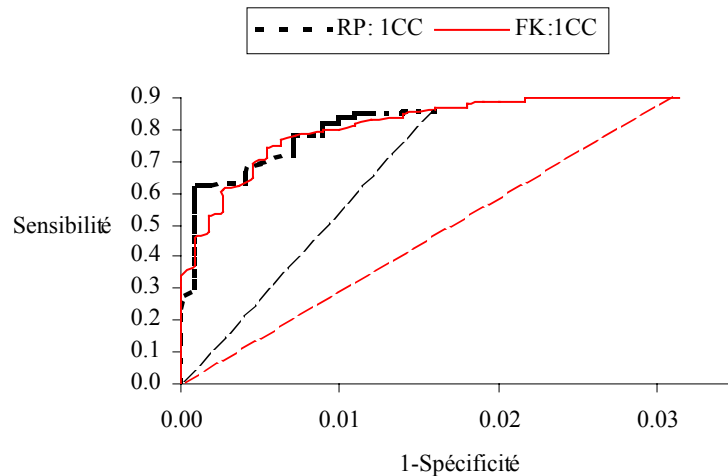


Figure 2.10. Courbes *ROC* pour un réseau à une couche cachée

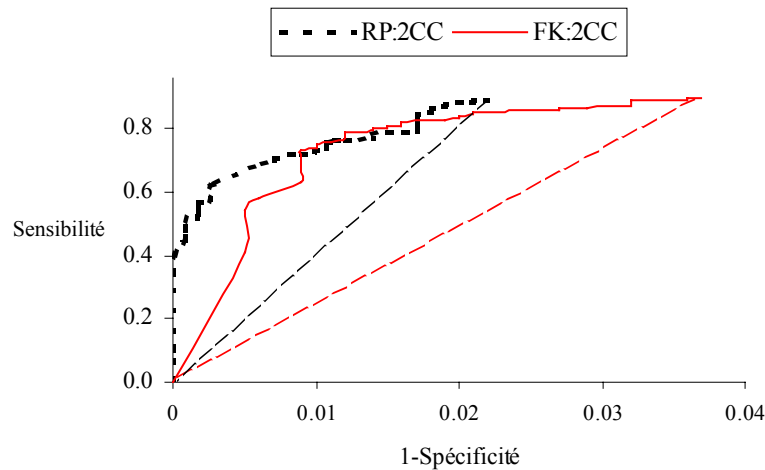


Figure 2.11. Courbes *ROC* pour un réseau à deux couches cachées

Pour les deux réseaux les graphiques montrent bien que l'aire sous la courbe de l'algorithme FK est plus grande que l'aire sous la courbe de l'algorithme RP. Cela confirme que l'algorithme FK est plus efficace au sens de détection de changement.

2.4.2.1.4 Appréciation visuelle

L'évaluation visuelle du résultat sur la scène entière peut être effectuée par différentes façons. Le cas le plus simple consiste à établir une image contenant toutes les classes d'intérêt à la fois. Par ailleurs, comme nous nous intéressons à la détection de changement, il est possible d'établir des images contenant uniquement les classes de changement. Pour notre application,

nous avons préféré ressortir deux images où l'une regroupe les classes de changement et l'autre regroupe les classes de non changement.

Les figures 2.12, 2.13, 2.14, et 2.15 représentent les images produites par le processus de détection de changement en utilisant différents réseaux. Les classes sont représentées par le code couleur suivant :

	<i>Eau → Eau</i>
	<i>Végétation → végétation</i>
	<i>Construction → construction</i>
	<i>Sol nu → Sol nu</i>
	<i>Végétation → sol nu</i>
	<i>Construction → sol nu</i>
	<i>Eau → Sol nu</i>
	<i>X → Nuages</i>

Les images classifiées montrent que :

- *Pour le réseau à une couche cachée* : l'algorithme RP présente une faible détection dans les classes: *végétation → sol nu* et *construction → sol nu*. Par ailleurs, les classes de non changement sont moyennement reconnues. Avec l'algorithme FK, le réseau fournit une meilleure reconnaissance dans les classes *eau → sol nu* et *végétation → sol nu*.
- *Pour le réseau à deux couches cachées* : l'algorithme FK présente une bonne reconnaissance dans toutes les classes contrairement à l'algorithme RP qui présente une faible reconnaissance dans les classes : *sol nu → sol nu*, *végétation → sol nu* et *eau → sol nu*. Ainsi, le réseau à deux couches cachées entraîné par l'algorithme FK produit la meilleure précision sur la scène.
- Les erreurs de confusion dans la classe *eau → sol nu* représentées par des cercles dans la figure 2.12.(a) sont des changements de végétation dans les zones en question. Cette confusion est causée par le fait que le changement dans la rivière s'est accompagné par quelques apparitions de végétation qui ont conduit à ces conflits.

Nous notons globalement que la précision sur l'image entière est différente de celle de la matrice de confusion et que l'algorithme de filtrage de Kalman améliore la précision de la détection de changement.

➤ Réseau à une couche cachée :

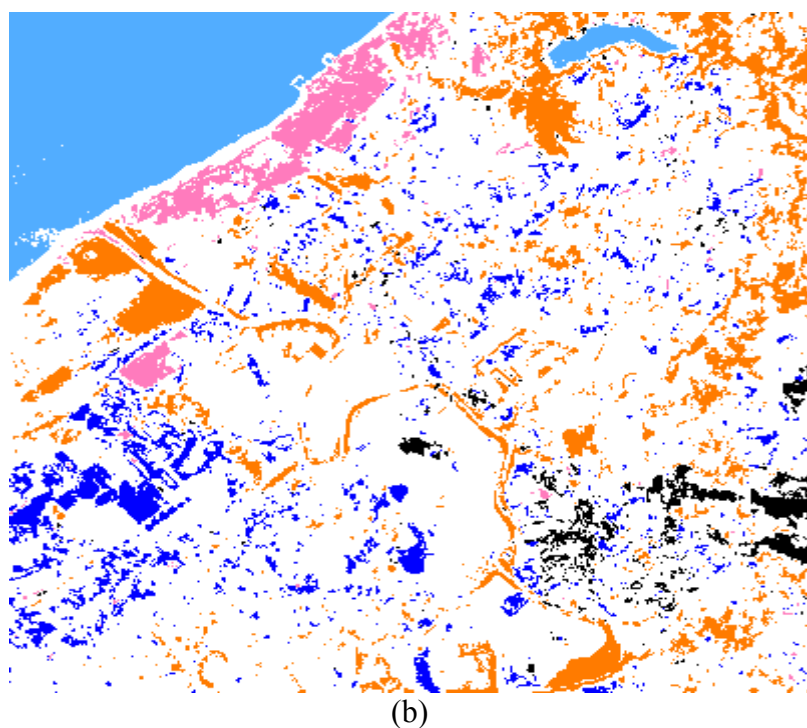
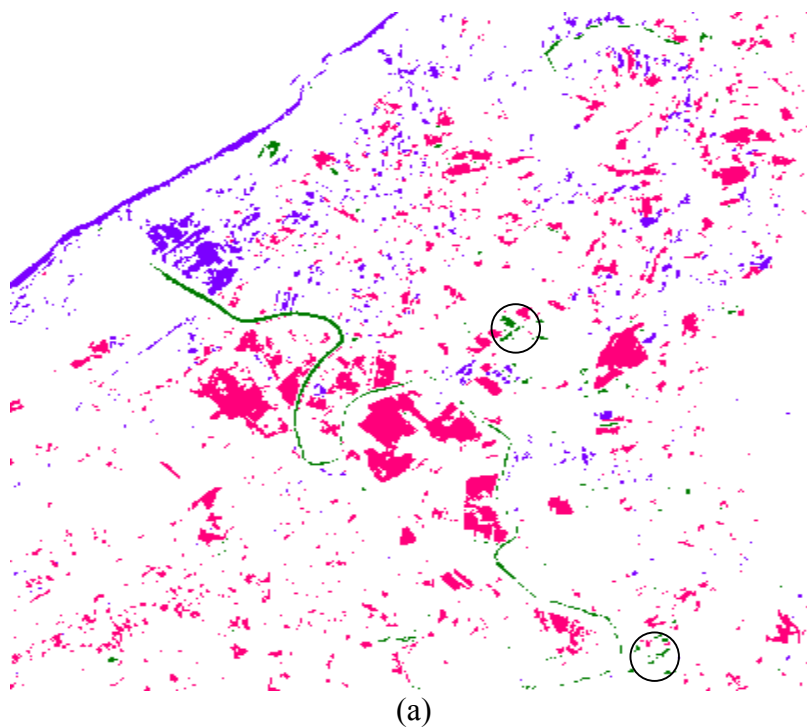
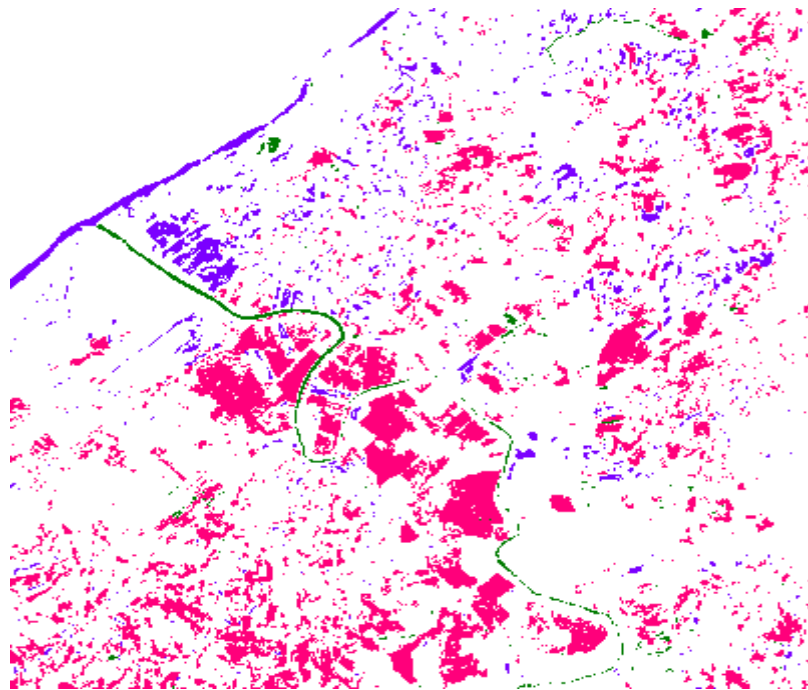
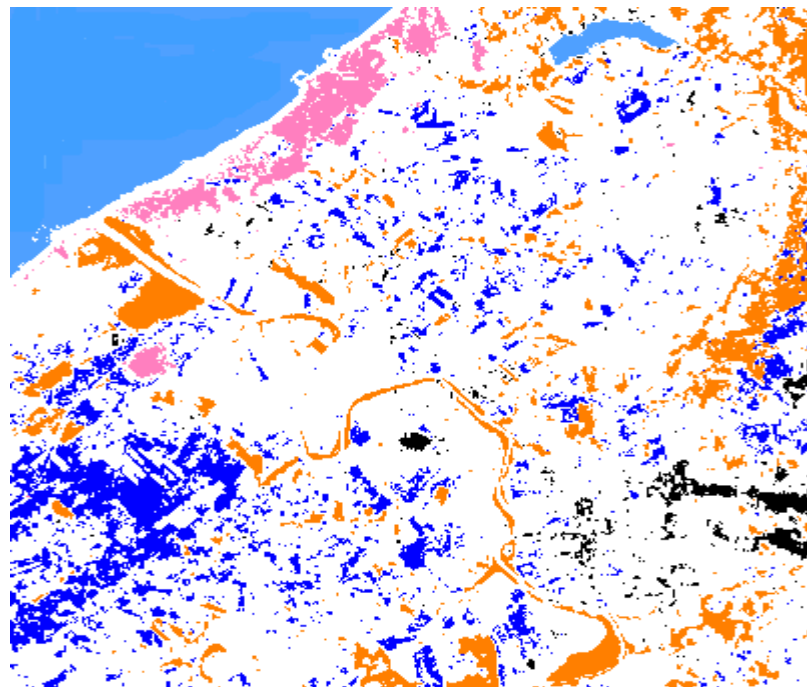


Figure 2.12. Image obtenue par analyse simultanée en utilisant un réseau à une couche cachée avec 15 nœuds entraîné par l'algorithme RP (a : classes de changement, b : classes de non changement)



(a)



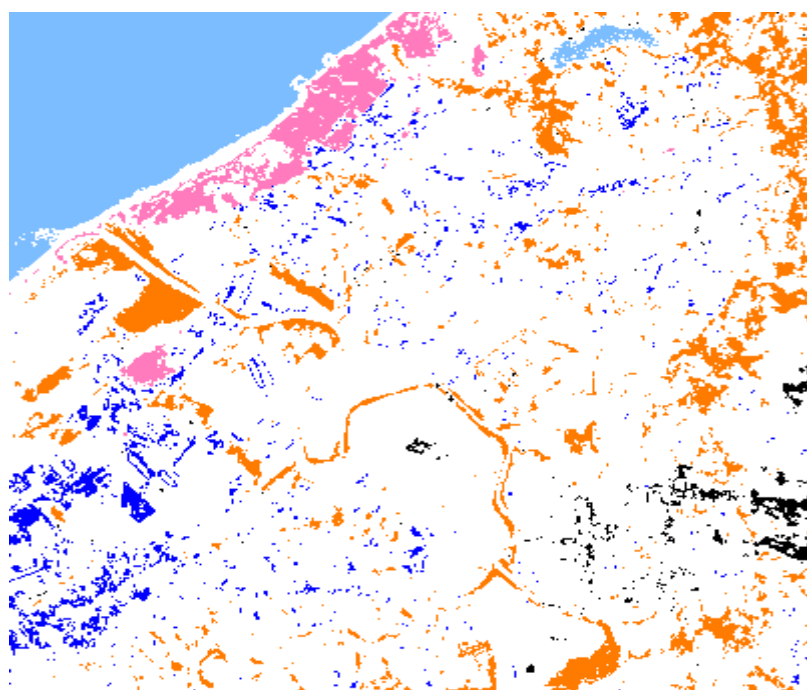
(b)

Figure 2.13. Image obtenue par analyse simultanée en utilisant un réseau à une couche cachée avec 15 nœuds entraîné par l'algorithme FK
(a : classes de changement, b : classes de non changement)

➤ Réseau à deux couches cachées :

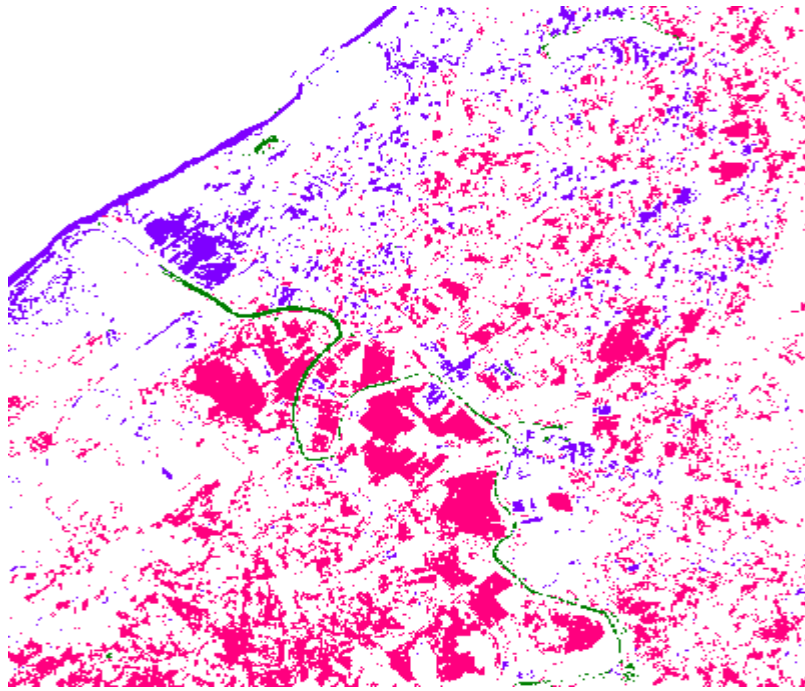


(a)

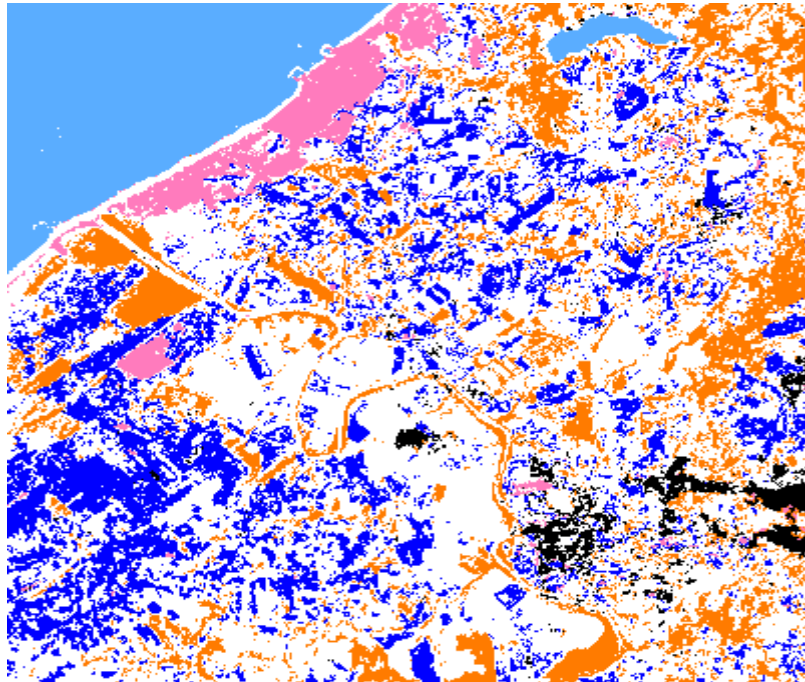


(b)

Figure 2.14. Image obtenue par analyse simultanée en utilisant un réseau à deux couches cachées comportant 15 nœuds entraîné par l'algorithme RP (a : classes de changement, b : classes de non changement)



(a)



(b)

Figure 2.15. Image obtenue par analyse simultanée en utilisant un réseau à deux couches cachées comportant 15 nœuds entraîné par l'algorithme FK (a : classes de changement, b : classes de non changement)

2.4.2.2 Analyse comparative

Nous rappelons que l'analyse comparative est fondée sur la classification indépendante de l'image de chaque date. La comparaison des étiquettes de chaque pixel permet de vérifier l'existence ou non du changement. Ainsi, nous nous sommes intéressés aux classes prédéfinies dans le chapitre précédent (Figures 1.5, 1.6). La base de données représentée par le tableau 2.7 a été sélectionnée pour effectuer l'apprentissage des classifieurs. Dans la phase de validation nous utilisons les données de test de l'analyse simultanée.

Classes	Données d'apprentissage	
	Image de 1989	Image de 1991
Eau	200	200
Construction	200	200
Végétation	200	200
Sol nu	200	200
Nuages		200

Tableau 2.7. Données d'apprentissage utilisées dans les processus de classification

2.4.2.2.1 Critères d'évaluation

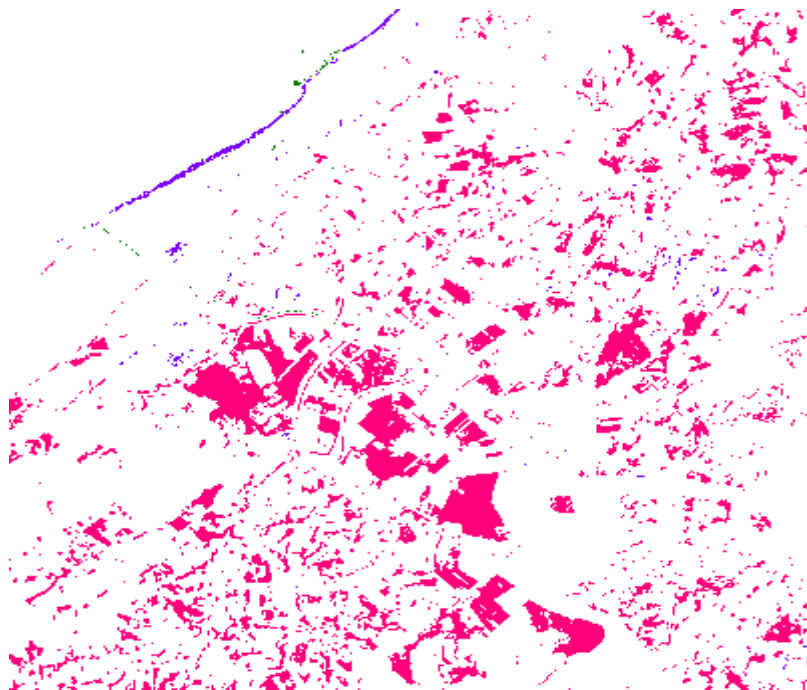
La comparaison par post classification est principalement évaluée d'une façon visuelle. Sur le plan statistique on peut estimer la précision de cette technique en utilisant le taux moyen de bonne reconnaissance dans chaque classe.

2.4.2.2.2 Appréciation visuelle

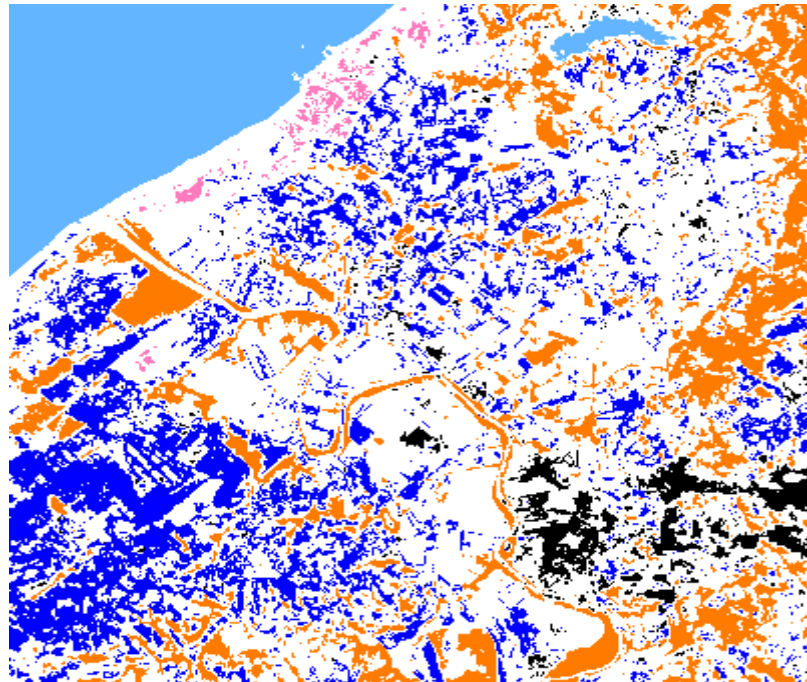
Les figures 2.16, 2.17, 2.18, 2.19 représentent les images obtenues à partir de différentes comparaisons par post classification. Les réseaux utilisés comportent 15 nœuds dans chaque couche cachée et sont entraînés par les algorithmes RP et FK (les paramètres d'apprentissage de chaque algorithme sont les mêmes qu'on a utilisé dans l'analyse simultanée).

En comparant les images représentatives des classes de changement par rapport aux images issues de l'analyse simultanée, on note une forte altération de la détection de changement avec tous les réseaux. Seule la classe *végétation* → *Sol nu* est assez bien détectée lors de la comparaison. Cela montre le manque de précision de l'analyse comparative.

➤ Réseau à une couche cachée :

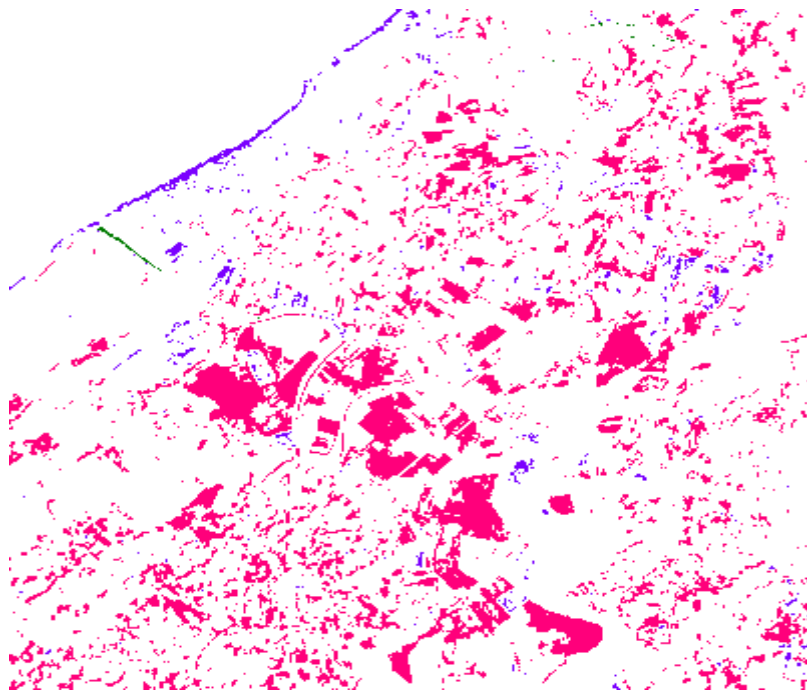


(a)

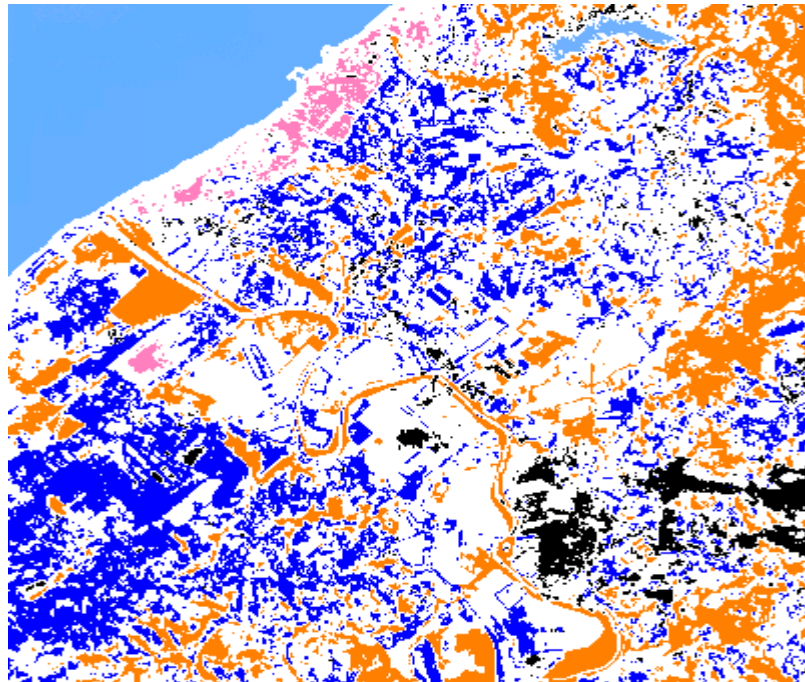


(b)

Figure 2.16. Image obtenue par analyse comparative en utilisant un réseau à une couche cachée entraîné par l'algorithme RP (a : classes de changement, b : classes de non changement)



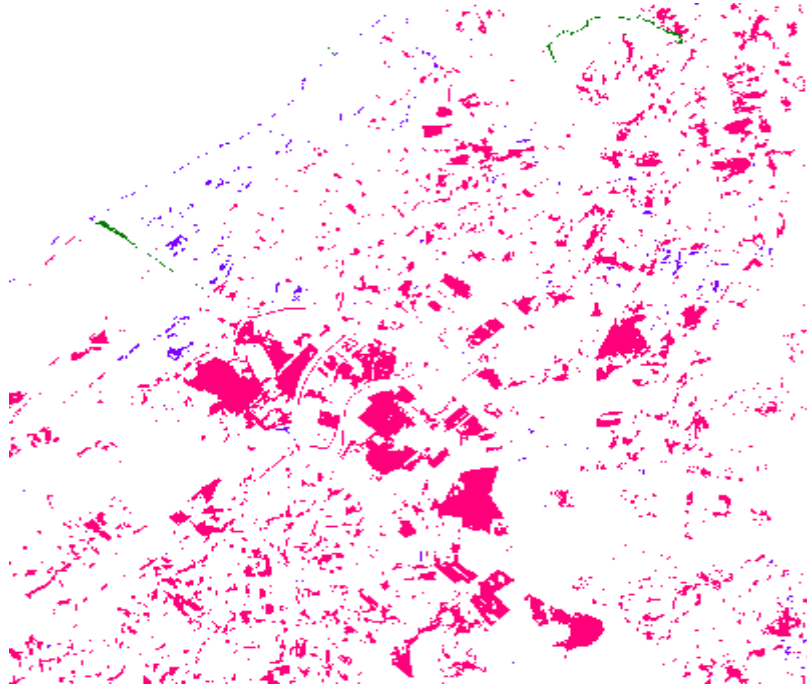
(a)



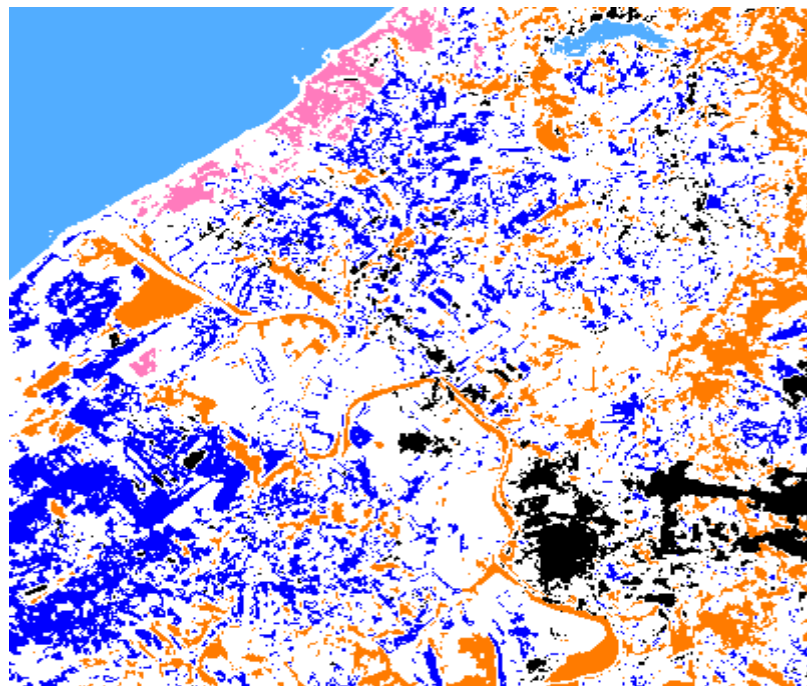
(b)

Figure 2.17. Image obtenue par analyse comparative en utilisant un réseau à une couche cachée entraîné par l'algorithme FK (a : classes de changement, b : classes de non changement)

➤ Réseau à deux couches cachées :

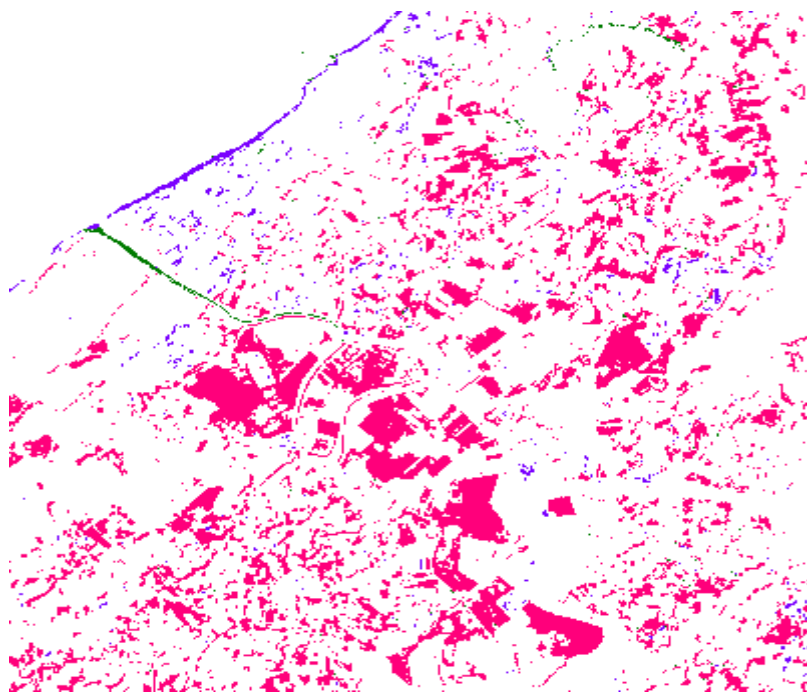


(a)

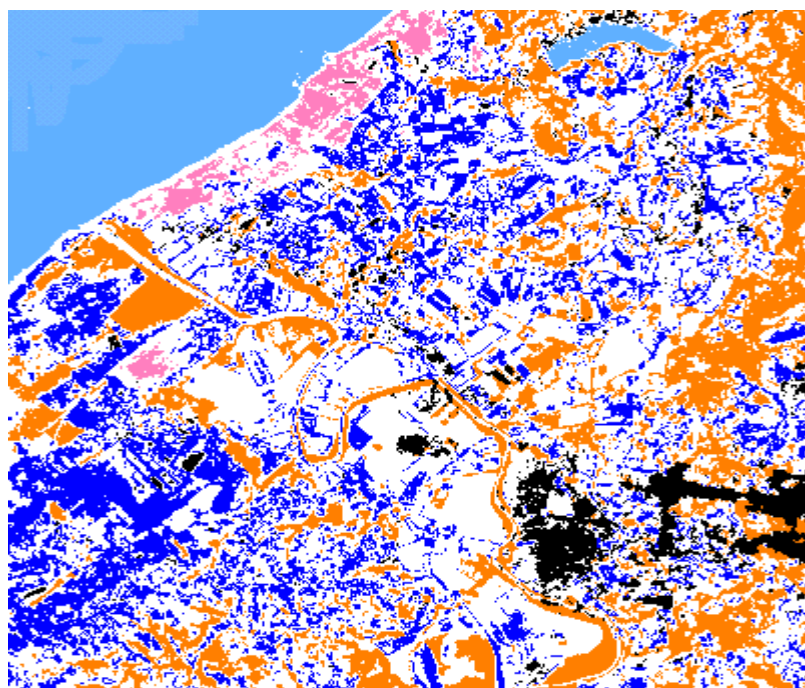


(b)

Figure 2.18. Image obtenue par analyse comparative en utilisant un réseau à deux couches cachées entraîné par l'algorithme RP (a : classes de changement, b : classes de non changement)



(a)



(b)

Figure 2.19. Image obtenue par analyse comparative en utilisant un réseau à deux couches cachées entraîné par l'algorithme FK
(a : classes de changement, b : classes de non changement)

2.4.2.3 Comparaisons statistiques des analyses simultanée et comparative

2.4.2.3.1 Critères d'évaluation

La comparaison statistique entre les deux approches est effectuée par deux tests. Dans le premier nous comparons les analyses simultanée et comparative en terme de précision en utilisant le taux moyen de bonne reconnaissance. Ensuite, dans le second test, nous les comparons en terme de capacité de détection de changement en exploitant les courbes *ROC*.

2.4.2.3.2 Taux moyen de bonne reconnaissance

Nous utilisons le taux moyen de bonne reconnaissance pour quantifier les différences entre les analyses comparative (*AC*) et simultanée (*AS*). Le tableau 2.8 représente les *TMBR* obtenus pour différents réseaux entraînés durant 400 itérations et utilisant un seuil de détection fixé à 0.9.

Nœuds cachés	<i>AC</i> (%)				<i>AS</i> (%)			
	RP: 1CC	FK:1CC	RP: 2CC	FK: 2CC	RP: 1CC	FK:1CC	RP: 2CC	FK :2CC
8	51.52	55.22	50.49	62.06	62.80	66.40	57.43	71.82
10	52.01	59.11	51.77	62.80	66.99	65.12	71.08	78.32
15	53.54	57.63	58.07	61.37	69.95	73.54	73.79	83.34
20	52.80	57.33	53.05	59.55	76.84	70.05	70.54	82.02
25	53.34	52.51	49.50	59.85	71.87	67.88	72.01	79.70
30	54.28	54.70	51.87	57.83	75.07	66.77	75.86	80.19
35	55.46	55.91	58.81	60.21	75.51	68.52	77.09	80.00
40	57.68	54.53	53.69	58.69	69.87	70.50	70.98	79.35
45	56.35	58.07	47.78	57.93	70.24	64.33	73.49	76.15
50	56.55	57.53	56.00	58.52	75.02	62.51	75.76	77.21

Tableau 2.8. Comparaison des analyses comparative et simultanée selon les valeurs du *TMBR* pour différents réseaux

Les résultats du tableau 2.8 (illustrés également sous forme de graphique par la figure 2.20) montrent que l'approche *AS* fournit le meilleur résultat quelle que soit l'architecture neuronale ou l'algorithme utilisé. En effet, l'approche *AC* offre une précision beaucoup plus faible car elle est trop affectée par les taux de rejet et de confusion. Cet inconvénient conduit à des fausses alarmes (faux changements) ou à des non détections.

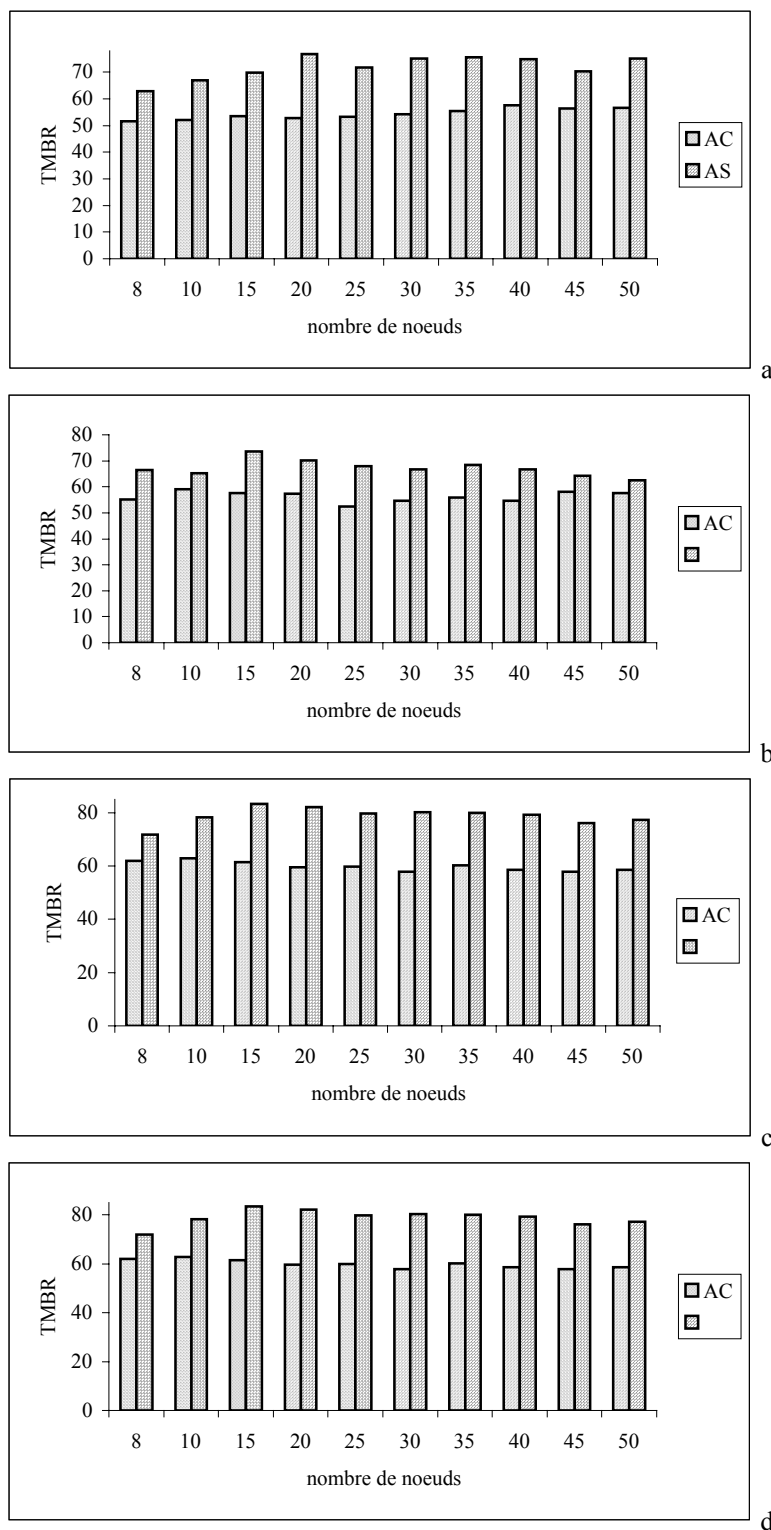


Figure 2.20. TMBR obtenu par les analyses simultanée et comparative (a: RP. 1CC, b: FK. 1CC, c: RP. 2CC, d: FK. 2CC)

2.4.2.3.3 Courbes ROC

Voici une comparaison entre les approches *AS* et *AC* en utilisant les courbes *ROC* (les réseaux sont entraînés par l'algorithme de filtrage de Kalman).

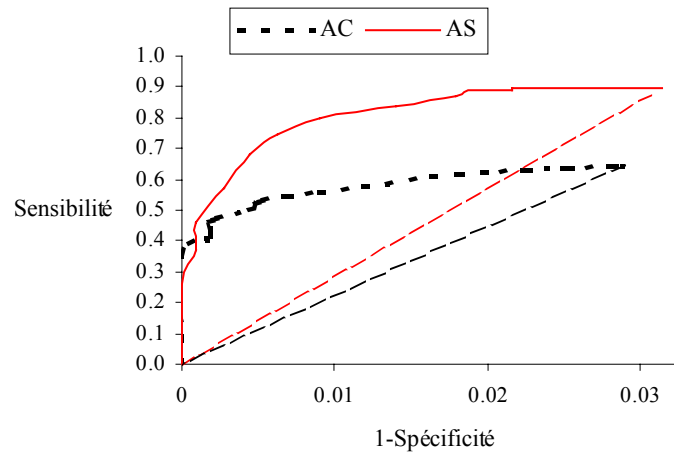


Figure 2.21. Courbes *ROC* obtenues en utilisant un réseau à une couche cachée entraîné par l'algorithme FK

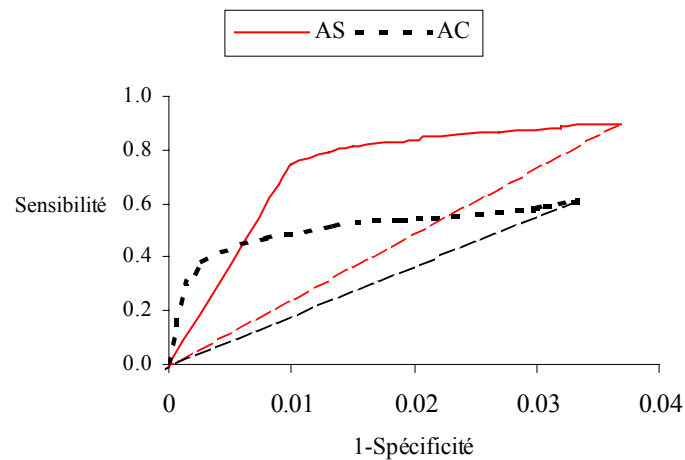


Figure 2.22. Courbes *ROC* obtenues en utilisant un réseau à deux couches cachées entraîné par l'algorithme FK

Les courbes *ROC* confirment à leur tour que l'approche d'analyse simultanée présente de meilleures performances lors de la détection de changement dans les images multispectrales.

2.5 Conclusion

L'objectif de l'étude de détection de changement présentée dans ce chapitre, est la comparaison des performances des analyses simultanée et comparative des données

multitemporelles. Pour cela, nous avons utilisé la classification par les réseaux de neurones qui présente l'avantage d'être plus souple que les techniques paramétriques de la classification supervisée. Les tests effectués ont montré que l'algorithme de filtrage de Kalman est plus approprié que l'algorithme de rétropropagation du gradient. Il permet d'une part, d'accélérer la convergence du réseau et, d'autre part, il augmente la précision de la détection de changement en éliminant quelques confusions liées à la rétropropagation. Par ailleurs, les tests d'évaluation des analyses simultanée et comparative ont permis d'illustrer les performances de l'analyse simultanée et les limitations de l'analyse comparative. Nous concluons ainsi, que le réseau de neurones opérant par analyse simultanée est un détecteur de changement dont la précision et l'efficacité sont considérables.

Chapitre 3

Détection de Changement par les Opérateurs de la Logique Floue

Résumé : Nous utilisons dans ce chapitre la classification par les opérateurs de la logique floue pour la détection de changement. Ce choix est motivé par le fait qu'elle surmonte quelques limitations de la classification Hard. Pour calculer les degrés d'appartenance nous utilisons, d'une part, la distance de Mahalanobis par rapport aux prototypes des classes et, d'autre part, les réseaux de neurones.

3.1 Utilisation de la logique floue en télédétection

Les techniques conventionnelles de la classification attribuent chaque observation (chaque pixel dans l'image) à l'une des classes recherchées. Ces méthodes dites *Hard* ou crispes échouent quand la signature spectrale de la classe est très générale pour décrire proprement un pixel donné (exemple d'une classe de végétation incluant des espèces de différents âges). Par ailleurs, le problème de confusion se pose fréquemment car les frontières entre les classes ne sont pas distinctes. En effet, il est difficile de discriminer les frontières dans une zone où par exemple un changement graduel d'un thème végétal vers un autre thème végétal se déroule. Les pixels qui se trouvent dans cette région contiennent ainsi plus d'une classe à l'intérieur, on les appelle pixels mixtes ou *mixels*. Par conséquent une décision *Hard* concernant ces pixels n'est plus valable. La logique floue a été introduite essentiellement pour résoudre ces ambiguïtés en permettant de raisonner avec le degré d'appartenance d'un pixel dans plusieurs classes.

3.2 Application à la détection de changement

La classification floue ne fournit pas une étiquette indiquant l'appartenance d'un pixel à une classe particulière mais plutôt des degrés d'appartenance de ce pixel dans les différentes classes d'intérêt. La question qui se pose est comment doit-on traiter ces degrés d'appartenance pour détecter les changements ? Dans l'analyse simultanée, ce problème peut être résolu en affectant le pixel à la classe qui correspond à la valeur maximale du degré d'appartenance. Toute la difficulté se pose donc dans l'analyse comparative. Deer (1998) a rapporté qu'il est possible de comparer les degrés d'appartenance d'un pixel issus de deux processus de classification floue en utilisant des opérations arithmétiques. Les deux opérations candidates sont la différence et le rapport. En faisant la soustraction du degré d'appartenance de la première date de la valeur du degré d'appartenance de la deuxième date, un résultat proche de zéro indique l'absence du changement (c'est à dire que la similarité du pixel au prototype de la classe est pratiquement la même dans les deux dates). Un résultat négatif indique que le pixel est actuellement moins similaire à la classe et un résultat positif indique qu'il est plus similaire. Le raisonnement est le même pour le rapport mais on se réfère cette fois à des valeurs proches de l'unité pour exprimer que le pixel est inchangé. Une valeur supérieure ou inférieure à l'unité exprime respectivement que le pixel est plus similaire ou

moins similaire à la classe. Deer (1998) a noté que la division effectue une sorte de compression des valeurs et rend l'interprétation du résultat une tâche difficile. Par conséquent, la différence se montre plus adéquate. Cependant, elle fournit des résultats dans l'intervalle $[-1,+1]$ alors que l'appartenance d'un ensemble flou est une valeur dans l'intervalle $[0,1]$. Même en arrangeant ces résultats dans l'intervalle $[0,1]$ (en rajoutant un 1 à chaque valeur et en divisant le résultat par 2), les valeurs obtenues ne peuvent être interprétées comme étant des degrés d'appartenance. Ainsi, pour y remédier Deer (1998) a proposé l'utilisation des normes triangulaires de la théorie floue pour effectuer la comparaison des images classifiées en opérant sur les degrés d'appartenance.

La théorie floue regroupe trois champs de recherche : les ensembles flous, la logique floue et les mesures floues (Takagi, 1997). Nous nous intéressons dans notre étude aux concepts des ensembles flous. Les opérations possibles sur ces ensembles sont les mêmes que dans le cas de la théorie ensembliste classique à l'exception de quelques lois. Cette théorie étant très vaste, nous introduisons un bref rappel dans le paragraphe suivant en se limitant aux notions générales.

3.3 Théorie de la logique floue

3.3.1 Introduction

Pour modéliser l'imperfection des informations on a souvent recours à la théorie des probabilités ou aux intervalles d'erreurs. L'emploi d'une distribution de probabilité est parfois dû au fait qu'on ne connaît pas d'autres théories de l'incertain. Dubois et Prade (1994, a) disent que l'intervalle d'erreur est fort peu expressif : trop étroit il est peu fiable, trop large il n'est plus informatif. Par ailleurs, la distribution de probabilité contient une information trop riche pour traduire fidèlement la pauvreté relative des informations. En revanche, le cerveau humain possède la possibilité de bien exprimer les imprécisions (il peut par exemple raisonner avec les valeurs d'une variable en employant les termes: très faible, faible, moyenne, élevée ou très élevée, etc.). Pour transmettre cette richesse de raisonnement à un ordinateur pour concevoir des programmes informatiques ou des circuits électroniques ayant un comportement proche des réflexes humains, Zadeh a introduit la théorie des ensembles flous.

Cette théorie s'appuie sur des opérations réalisées sur des sous ensembles flous et se réfère parfois à l'imprécision lexicale des expressions se rapprochant ainsi, de notre forme de raisonnement. Elle permet à la fois la représentation de l'imprécision (sous forme d'ensembles flous) et la quantification de l'incertitude par les degrés d'appartenance (Fuller, 1995). Ainsi, le raisonnement exact devient un cas limite du raisonnement approximatif.

3.3.2 Sous ensembles flous et degré d'appartenance

On définit l'ensemble flou A dans un ensemble X crisp universel non vide comme étant une application caractérisée par une fonction d'appartenance μ_A décrite comme suit (Chauvin, 1995) :

$$\mu_A : X \longrightarrow [0, 1]$$

Le sous ensemble flou permet de graduer l'appartenance d'un élément à une classe donnée où la notion d'appartenance n'est pas tout ou rien.

On définit le degré d'appartenance pour chaque élément x de X dans A comme étant :

$$A = \{ (x, \mu_A(x)) \mid x \in X \}$$

Dubois et Prade (1994, d) ont rapporté qu'il existe trois sémantiques ou interprétations du degré d'appartenance chacune liée à une classe d'application : la classification et l'analyse de données, le raisonnement approximatif et les problèmes de prise de décision. Chaque classe exploite une sémantique particulière du degré d'appartenance respectivement sous terme de similarité, incertitude et préférence.

- *Le degré de similarité* : $\mu_A(x)$ est le degré de proximité de x par rapport aux éléments prototypes de A (centroïdes des classes dans le cas de la classification). C'est la sémantique la plus ancienne du degré d'appartenance définie par Zadeh en 1966.
- *Le degré de préférence* : A représente dans ce cas un ensemble d'objets plus ou moins préférables (ou des valeurs d'une variable de décision x). $\mu_A(x)$ représente une intensité de préférence en faveur de cet objet ou bien la faisabilité de choisir x comme étant la valeur de A .
- *Le degré d'incertitude* : cette interprétation a été proposée aussi par Zadeh quand il a introduit la théorie des possibilités en 1978. $\mu_A(x)$ est alors le degré de possibilité que le paramètre x possède une valeur A .

Si le degré d'appartenance de chaque x_i est μ_i , A peut être défini comme suit :

$$A = \mu_1/x_1 + \mu_2/x_2 + \dots + \mu_n/x_n$$

3.3.3 Principale extension

Afin d'utiliser les nombres et les relations flous dans n'importe quel système intelligent, il faut être capable de faire des opérations arithmétiques avec ces quantités floues c'est à dire : l'arithmétique floue. La principale extension (ou extension principale) nous permet d'étendre n'importe quelle opération aux opérations impliquant les ensembles flous (Fuller, 1995).

Soient X et Y deux ensembles crispes et soit f une application telle que :

$$f: X \longrightarrow Y$$

Dans ce cas, $\forall x \in X \Rightarrow f(x) = y \in Y$.

Si A est un sous ensemble flou de X en utilisant la principale extension, on peut définir $f(A)$ comme étant un sous ensemble flou de Y . En se basant sur cette hypothèse, on a formulé les métriques, les implications et les relations floues. Ainsi, la théorie des ensembles flous fournit plusieurs outils pour manipuler les concepts imprécis.

3.3.4 Opérations sur les ensembles flous

Les opérations de la théorie des ensembles flous ne sont qu'une extension des opérations de la théorie des ensembles crispes. Zadeh a défini en 1965 quatre notions ensemblistes : l'inclusion, la réunion, l'intersection et la complémentation (Chauvin, 1995).

- *L'inclusion* : $A \subset B, \forall w \in \Omega, \mu_A(w) \leq \mu_B(w)$.
- *La réunion* : $\forall i, j, \forall w \in \Omega, \mu_{C_i \cup C_j}(w) = \max\{\mu_{C_i}(w), \mu_{C_j}(w)\}$.
- *L'intersection* : $\forall i, j, \forall w \in \Omega, \mu_{C_i \cap C_j}(w) = \min\{\mu_{C_i}(w), \mu_{C_j}(w)\}$.
- *La complémentation* : l'ensemble flou $\overline{C_i}$ complémentaire de C_i est défini par une application dans $[0, 1]$ telle que :

$$\forall w \in \Omega, \mu_{\overline{C_i}}(w) = 1 - \mu_{C_i}(w)$$

Les opérations d'intersection et de réunion (souvent exprimées respectivement par le *min* et le *max*) peuvent être représentées par plusieurs autres opérations. En particulier, on note les normes triangulaires qui sont très utilisées dans la théorie floue pour modéliser la connectivité logique (Fuller, 1995).

3.3.5 Normes triangulaires

Une application I de $[0, 1]^2$ dans $[0, 1]$ est une norme triangulaire si elle vérifie les propriétés suivantes :

1. La commutativité : $I(x, y) = I(y, x)$.
2. L'associativité : $I(x, I(y, z)) = I(I(x, y), z)$.
3. 1 est l'élément neutre : $I(x, 1) = I(1, x) = x$.
4. La monotonie : $I(x, y) \geq I(z, t)$ si $x \geq z$ et $y \geq t$.

Plusieurs normes triangulaires sont proposées pour modéliser la conjonction (le *and*) floue. Les plus utilisées d'entre elles sont les suivantes (Fuller, 1995):

Le Minimum	$\text{MIN}(x, y) = \min\{x, y\}$
Lukasiewicz	$\text{LAND}(x, y) = \max\{x+y-1, 0\}$
Probabilistique	$\text{PAND}(x, y) = xy$
Weak	$\text{WEAK}(x, y) = \begin{cases} \min\{x, y\} & \text{si } \max\{x, y\} = 1 \\ 0 & \text{sinon} \end{cases}$
Hamacher	$\text{HAND}_\gamma(x, y) = xy / (\gamma + (1 - \gamma)(x + y - xy)), \gamma \geq 0$
Dubois et Prade	$\text{DAND}_\alpha(x, y) = xy / \max\{x, y, \alpha\}, \alpha \in [0, 1]$
Yager	$\text{YAD}_p(x, y) = 1 - \min\{1, [(1-x)^p + (1-y)^p]^{1/p}\}, p > 0$

Tableau 3.1. Normes triangulaires de base

Les T-normes ont toutes la propriété d'être inférieures ou égales à l'opérateur *min* (Chauvin, 1995) :

$$\forall (x, y) \in [0, 1]^2, I(x, y) \leq \min(x, y)$$

Elles effectuent des opérations de conjonction entre les valeurs de x et de y et font au moins confiance à la valeur minimale qui donne l'information la plus pessimiste.

3.3.6 Conormes triangulaires

Une application U de $[0, 1]^2$ dans $[0, 1]$ est une conorme triangulaire si elle vérifie les propriétés suivantes :

1. La commutativité : $U(x, y) = U(y, x)$.
2. L'associativité : $U(x, U(y, z)) = U(U(x, y), z)$.
3. 0 est l'élément neutre : $U(x, 0) = U(0, x) = x$.
4. La monotonie : $U(x, y) \geq U(z, t)$ si $x \geq z$ et $y \geq t$.

Les conormes sont utilisées pour modéliser la disjonction (le *ou*) floue, les plus répandues sont les suivantes :

Le Maximum	$\text{MAX}(x, y) = \max\{x, y\}$
Lukasiewicz	$\text{LOR}(x, y) = \min\{x + y, 1\}$
Probabilistique	$\text{POR}(x, y) = x + y - xy$
Strong	$\text{STRONG}(x, y) = \{ \max\{x, y\} \text{ si } \min\{x, y\} = 0, 1 \text{ sinon } \}$
Hamacher	$\text{HOR}_\gamma(x, y) = (x + y - (2 - \gamma)xy) / (1 - (1 - \gamma)xy), \gamma \geq 0$
Yager	$\text{YOR}_p(x, y) = \min\{1, \sqrt[p]{x^p + y^p}\}, p > 0$

Tableau 3.2. Conormes triangulaires de base

Les T-conormes ont toutes la propriété d'être supérieures ou égales à l'opérateur $\max$ (Chauvin, 1995):

$$\forall (x, y) \in [0, 1]^2, U(x, y) \geq \max(x, y)$$

Elles expriment la disjonction entre les valeurs de x et de y et font au moins confiance à la valeur maximale qui donne ainsi, l'information la plus optimiste.

3.3.7 Négation

La négation est une application N de $[0, 1]$ dans $[0, 1]$ telle que :

1. $N(0) = 1$.
2. $N(N(x)) = x$ (involution).
3. N est strictement décroissante et continue.

L'opération de négation peut se généraliser à toute une famille d'opérations paramétrées, on note celle qui a été introduite au début par Sugeno en 1977 (Deer, 2000) :

$$N_\lambda(x) = (1 - x) / (1 + \lambda x), \lambda > -1$$

Cependant, dans la plupart des cas on utilise l'opération $N(x) = 1 - x$ comme étant l'opérateur de négation. Ainsi, à chaque T-norme I , on peut associer une T-conorme U respectivement à une négation N par la loi de Morgan :

$$U(x, y) = N(I(N(x), N(y)))$$

3.4 Méthode adoptée pour la classification et la détection de changement

Tous les classifieurs *Soft* partagent l'utilisation des concepts de base fournis par la théorie des ensembles flous. La différence entre eux se trouve dans la façon selon laquelle on manipule les données dans les phases d'apprentissage et de validation. Généralement, on utilise soit des modèles du degré d'appartenance basés sur les ensembles flous, soit des fonctions du degré d'appartenance (exemple: classification par le maximum de vraisemblance avec des moyennes et des covariances floues) ou bien, on utilise les règles d'inférence floues (Bárdossy et Samaniego, 2002). Nous adoptons dans notre étude une méthode qui exploite la distance de Mahalanobis pour calculer le degré d'appartenance de chaque pixel.

Pour un pixel x_k , la similarité est mesurée par rapport au centroïde m_i de la classe i par :

$$d_{ik}^2 = (x_k - m_i)^t A (x_k - m_i) \quad (3.1)$$

m_i : vecteur moyenne des pixels d'apprentissage de la classe i calculé dans toutes les bandes spectrales.

A : matrice inverse de covariance des pixels d'apprentissage de la classe i calculée dans toutes les bandes spectrales.

Ainsi, le modèle du degré d'appartenance du pixel x_k dans la classe i est calculé comme suit :

$$\mu_{ik} = \frac{\left(\frac{1}{d_{ik}^2} \right)^n}{\sum_{j=1}^c \left(\frac{1}{d_{jk}^2} \right)^n} \quad (3.2)$$

où : $1 \leq i \leq c$ et $j = 1, \dots, c$ (c étant le nombre de classes).

Le paramètre n est utilisé pour contrôler l'aspect flou des classes. La détermination des degrés d'appartenance de cette manière implique que pour chaque pixel dans l'image, la somme des degrés d'appartenance dans les différentes classes est toujours égale à 1.

$$\sum_{i=1}^c \mu_{ik} = 1 \quad \forall k$$

La *CPC* floue proposée par Deer (1998, 2000) utilise les normes et les conormes triangulaires pour définir les classes de changement.

La situation où le pixel x est dans la classe A à l'instant t_1 est représentée par le degré d'appartenance de x dans A à l'instant t_1 : $\mu_A(x, t_1)$.

De même son appartenance dans la classe B à l'instant t_2 est décrite par : $\mu_B(x, t_2)$.

Alors on peut calculer à quel point cette situation est vraie (c'est à dire : mesurer l'étendu du changement du pixel x) par :

$$MIN(\mu_A(x, t_1), \mu_B(x, t_2)) \quad (3.3)$$

Le MIN représente le degré d'appartenance du pixel x dans la classe de changement (A, B). Il est possible de définir d'autres règles de détection de changement. Par exemple : la situation où le pixel x est dans la classe A ou dans la classe B à l'instant t_1 et est dans la classe C et non pas dans la classe D à t_2 est décrite par :

$$MIN(MAX(\mu_A(x, t_1), \mu_B(x, t_1)), MIN(\mu_C(x, t_2), N(\mu_D(x, t_2)))) \quad (3.4)$$

Par ailleurs, notre but est de comparer les approches d'analyse comparative et simultanée. Habituellement, l'approche comparative est utilisée comme une démarche pour détecter les changements telle que proposée par Deer. Pour notre part, nous souhaitons adopter la classification floue opérant par analyse simultanée afin d'évaluer ses performances. Ainsi, nous avons pensé que l'équation (3.2) peut être exploitée pour calculer le degré d'appartenance dans les classes multitemporelles (classes de non changement et classes de changement) et établir ainsi, la détection de changement par analyse simultanée. Dans ce cas, nous calculons les matrices de covariance et les moyennes sur l'espace bitemporel en utilisant les bandes spectrales de la première date combinées avec les bandes spectrales de la seconde date (les matrices inverses de covariance et les moyennes des différentes classes sont données dans l'annexe E).

3.5 Résultats Expérimentaux

Le modèle du degré d'appartenance décrit par l'équation (3.2) indique que le résultat de la classification est fortement lié au choix du paramètre de fuzzification. En effet, on doit choisir une valeur qui permet à un pixel d'avoir un degré d'appartenance maximal dans sa classe et des degrés d'appartenance minimaux dans toutes les autres classes. Deer (1998) a effectué des études empirique et analytique sur l'évolution du degré d'appartenance en fonction de la valeur de ce paramètre en distinguant les cas des pixels mixtes et des pixels purs. Il a conclu qu'une valeur supérieure à 0.5 arrive à donner le résultat désiré.

3.5.1 Etude empirique sur le choix du paramètre de fuzzification

Dans le cadre de notre application, nous avons étudié l'évolution de la valeur du degré d'appartenance du pixel typique d'une classe en fonction de la valeur du paramètre de fuzzification. Le tableau 3.3 représente les distances moyennes du pixel typique de chaque classe calculées par rapport aux centroïdes de toutes les classes.

D \ C	1	2	3	4	5	6	7	8
D ₁	9.83	50.00	271.23	88.41	37.58	273.97	282.39	526.92
D ₂	11284.60	5.44	213.01	16.80	12.16	100.18	213.10	111.34
D ₃	19914.60	163.20	5.83	47.69	33.62	29.81	31.52	333.27
D ₄	11060.80	45.50	133.00	5.74	14.60	58.06	120.24	105.57
D ₅	10518.40	64.46	168.47	28.42	5.44	87.48	211.45	197.63
D ₆	23676.50	121.32	42.47	33.88	40.11	6.44	47.19	157.44
D ₇	3358.40	76.23	155.17	31.68	45.01	60.47	6.19	176.73
D ₈	20424.70	113.32	140.73	32.59	52.37	41.36	165.14	5.78

Tableau 3.3. Distances (D) moyennes du pixel typique de chaque classe (C) par rapport aux centroïdes de toutes les classes

D_i : la distance moyenne par rapport à la classe i .

En se basant sur le calcul des distances moyennes, la forme générale du degré d'appartenance s'écrit comme suit :

$$y_i = \frac{\left(\frac{1}{D_i}\right)^x}{\sum_{j=1}^8 \left(\frac{1}{D_j}\right)^x} \quad (3.5)$$

y_i : degré d'appartenance dans la classe i .

x : paramètre de fuzzification (toujours supérieur à 0).

Pour chaque classe d'intérêt, nous traçons la variation du degré d'appartenance (DA) en fonction des différentes valeurs de x . Les figures suivantes montrent les courbes obtenues pour les trois classes de changement (les graphiques des autres classes sont donnés dans l'annexe E).

➤ La classe *végétation* → *sol nu* :

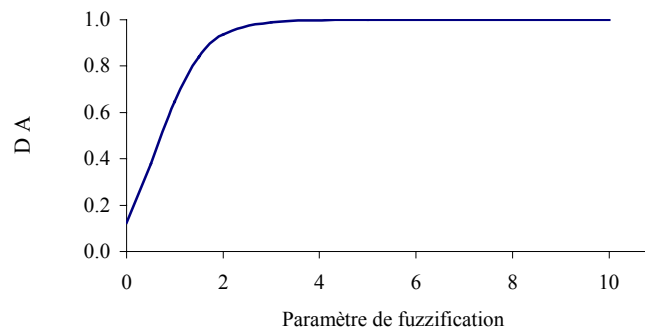


Figure 3.1. Evolution du degré d'appartenance dans la classe *végétation* → *sol nu* en fonction du paramètre de fuzzification

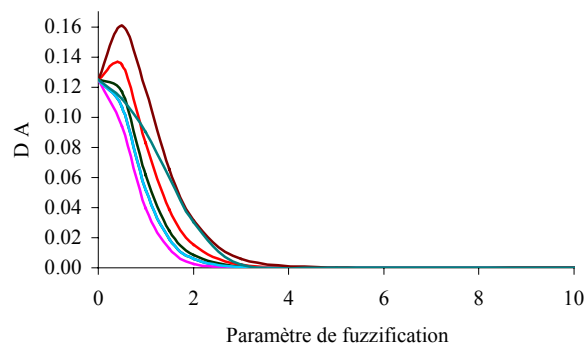


Figure 3.2. Evolution du degré d'appartenance dans les autres classes en fonction du paramètre de fuzzification

➤ La classe *construction* → *sol nu* :

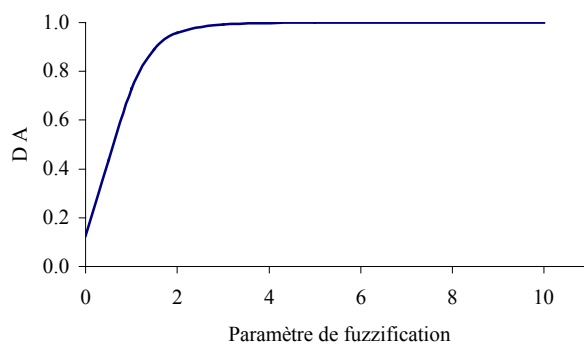


Figure 3.3. Evolution du degré d'appartenance dans la classe *construction* → *sol nu* en fonction du paramètre de fuzzification

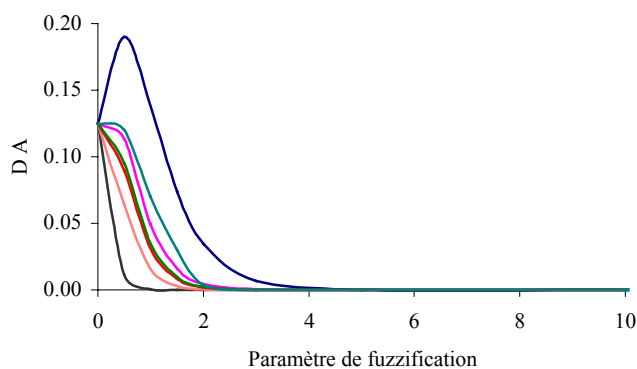


Figure 3.4. Evolution du degré d'appartenance dans les autres classes en fonction du paramètre de fuzzification

➤ La classe *eau* → *sol nu* :

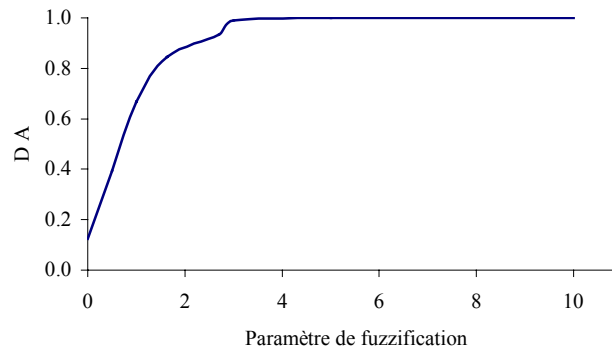


Figure 3.5. Evolution du degré d'appartenance dans la classe *eau* → *sol nu* en fonction du paramètre de fuzzification

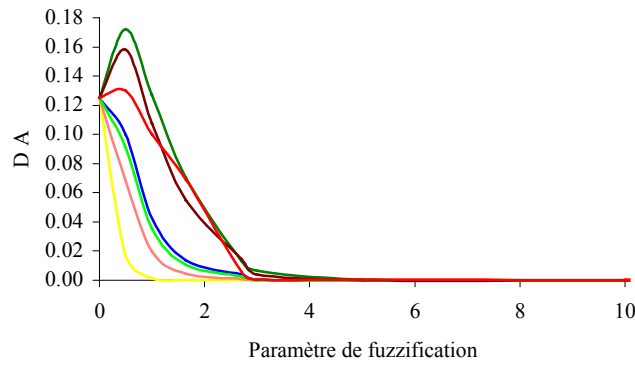


Figure 3.6. Evolution du degré d'appartenance dans les autres classes en fonction du paramètre de fuzzification

Nous remarquons d'après les graphiques obtenus pour les différentes classes qu'à partir d'une valeur de 1.5, les pixels présentent des degrés d'appartenance maximaux dans leurs classes et des degrés pratiquement négligeables dans les autres classes. Ainsi, nous avons choisi dans nos tests la valeur 2 comme paramètre de fuzzification.

Dans la comparaison par post classification, nous avons utilisé la forme décrite par l'équation (3.3) pour l'extraction des classes. Cependant, pour la classe $X \rightarrow Nuages$ le degré d'appartenance est calculé à l'aide de l'équation suivante :

$$MIN(MAX(MAX(MAX(\mu_1(x, t_1), \mu_2(x, t_1)), \mu_3(x, t_1)), \mu_4(x, t_1)), \mu_5(x, t_2))) \quad (3.6)$$

3.5.2 Etude comparative des approches *AS* et *AC* floues

3.5.2.1 Critères d'évaluation

La théorie des ensembles flous fournit plusieurs moyens pour estimer la précision. Nous distinguons en particulier, les opérateurs flous, les distances floues, l'entropie de Shannon ainsi que l'entropie floue, etc. Cependant, bien que ces techniques sont diversifiées, dans la plupart des cas on ne possède pas une vérité terrain adéquate aux valeurs du degré d'appartenance. Dans notre application nous utilisons le degré d'appartenance moyen calculé pour chaque classe pour estimer les différences entre les deux approches. Par ailleurs, l'évaluation visuelle est particulière. Dans la classification *Hard*, chaque classe est représentée par une couleur spécifique dans l'image résultante. Une telle représentation ne convient pas à la classification floue car elle ne permet pas de montrer l'intensité du degré d'appartenance. Une solution alternative consiste à illustrer l'appartenance des pixels dans chaque classe en niveau de gris où le 255 représente le 0 (la non appartenance absolue) et le 0 représente le 1 (l'appartenance absolue).

3.5.2.2 Evaluation quantitative

Pour estimer les différences entre les deux types d'analyse, une étude comparative a été menée en se basant respectivement sur les valeurs du degré d'appartenance global (DAG) et par classe.

Classes	<i>AC</i> (%)	<i>AS</i> (%)
1	98.38	89.06
2	67.54	73.00
3	78.52	75.52
4	83.91	75.94
5	61.84	82.61
6	77.21	80.90
7	17.65	49.69
8	75.90	66.78
DAG	70.56	74.43

Tableau 3.4. Degrés d'appartenance global et par classe calculés pour chaque approche

Les résultats du tableau 3.4 indiquent que les deux approches ont des capacités rapprochées. Leurs précisions diffèrent d'une classe à l'autre mais globalement l'approche *AS* donne la meilleure précision. Cette dernière fournit les taux de précision les plus élevés dans les trois classes de changement.

3.5.2.3 Evaluation qualitative

Nous avons évalué dans ce test le degré d'appartenance dans chaque classe d'intérêt. Nous introduisons dans ce qui suit les résultats obtenus à partir des deux approches représentés sous forme d'images.



Figure 3.7. Degré d'appartenance dans la classe
 $eau \rightarrow eau$ obtenu par AS



Figure 3.8. Degré d'appartenance dans la classe
 $eau \rightarrow eau$ obtenu par la AC



Figure 3.9. Degré d'appartenance dans la classe *construction* → *construction* obtenu par *AS*



Figure 3.10. Degré d'appartenance dans la classe *construction* → *construction* obtenu par *AC*



Figure 3.11. Degré d'appartenance dans la classe
végétation $\rightarrow$ *végétation* obtenu par *AS*



Figure 3.12. Degré d'appartenance dans la classe
végétation $\rightarrow$ *végétation* obtenu par *AC*



Figure 3.13. Degré d'appartenance dans la classe
sol nu $\rightarrow$ *sol nu* obtenu par AS



Figure 3.14. Degré d'appartenance dans la classe
sol nu $\rightarrow$ *sol nu* obtenu par AC



Figure 3.15. Degré d'appartenance dans la classe
végétation $\rightarrow$ sol nu obtenu par AS



Figure 3.16. Degré d'appartenance dans la classe
végétation $\rightarrow$ sol nu obtenu par AC

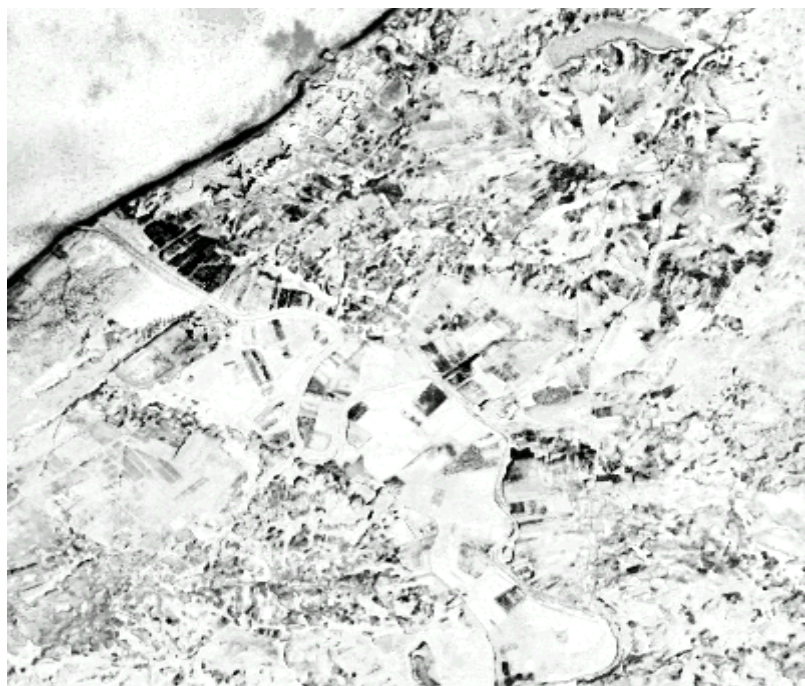


Figure 3.17. Degré d'appartenance dans la classe *construction* → *sol nu* obtenu par *AS*

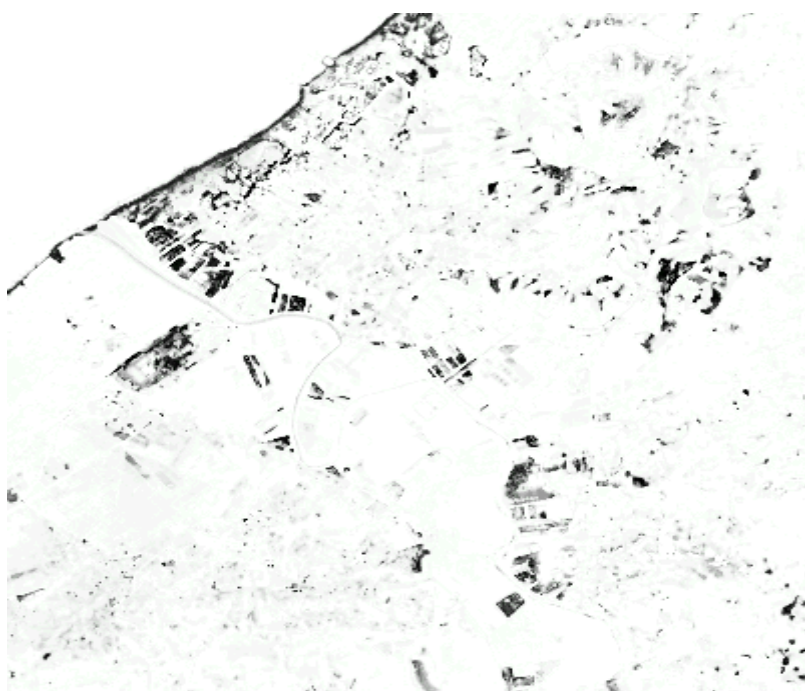


Figure 3.18. Degré d'appartenance dans la classe *construction* → *sol nu* obtenu par *AC*



Figure 3.19. Degré d'appartenance dans la classe
eau $\rightarrow$ sol nu obtenu par AS



Figure 3.20. Degré d'appartenance dans la classe
eau $\rightarrow$ sol nu obtenu par AC



Figure 3.21. Degré d'appartenance dans la classe
 $X \rightarrow \text{nuages}$ obtenu par AS



Figure 3.22. Degré d'appartenance dans la classe
 $X \rightarrow \text{nuages}$ obtenu par AC

Nous représentons également les compositions colorées des degrés d'appartenance dans les trois classes de changement (Figures 3.23 et 3.24).

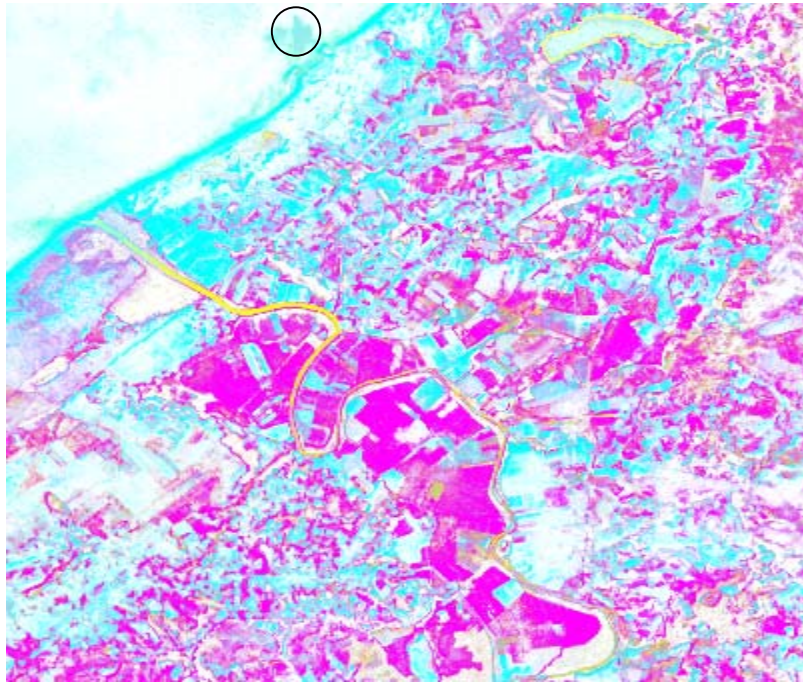


Figure 3.23. Composition colorée des degrés d'appartenance dans les trois classes de changement (*AS*)

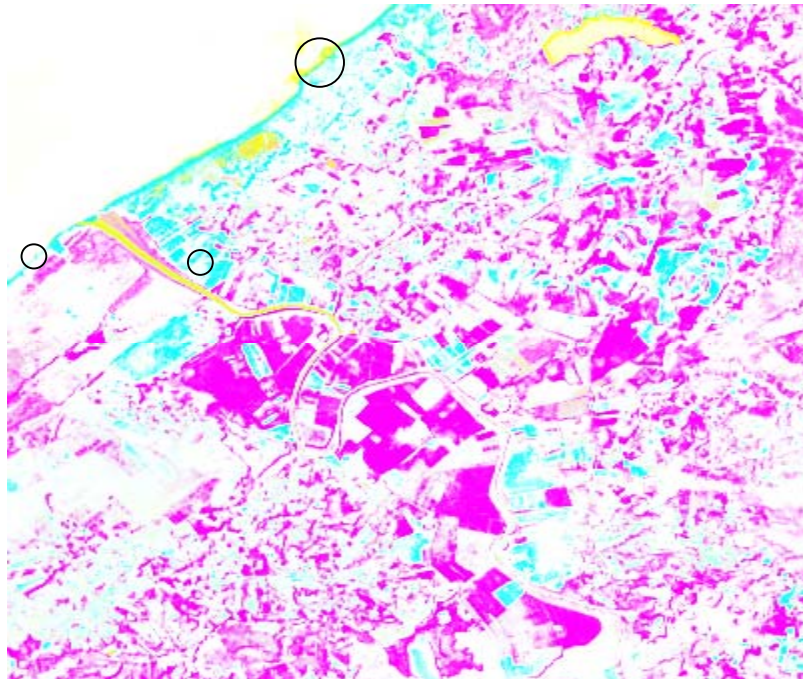


Figure 3.24. Composition colorée des degrés d'appartenance dans les trois classes de changement (*AC*)

Les images en niveau de gris et les compositions colorées révèlent que les deux approches fournissent globalement une précision satisfaisante. Néanmoins, nous notons quelques erreurs de classification. L'approche *AS* produit des commissions dans la classe *construction* $\rightarrow$ *sol nu* (Cercle dans la figure 3.23) alors que l'*AC* produit des omissions dans cette classe (Cercles dans la figure 3.24). Par ailleurs, les deux approches détectent moyennement la classe de changement *eau* $\rightarrow$ *sol nu*. Notons que l'analyse simultanée n'a pas détecté le lac par contre, l'analyse comparative a négligé une partie de la rivière.

Ainsi, visuellement parlant, les analyses simultanée et comparative ont des précisions qui diffèrent d'une classe à l'autre mais globalement l'utilisation de l'analyse simultanée permet de fournir l'image la plus précise.

3.5.3 Etude comparative des méthodes floue et neuronale

En comparant les résultats obtenus par la *CPC* floue à ceux de la *CPC* Hard appliquée dans le chapitre précédent, on voit clairement que l'approche floue a amélioré d'une façon notable la précision de la détection (ce constat concerne particulièrement la classe *eau* $\rightarrow$ *sol nu* qui est presque non détectée par la *CPC* Hard). Par ailleurs, les deux méthodes floue et neuronale favorisent l'analyse simultanée comme étant l'approche la plus efficace pour détecter les changements. Ainsi, pour comparer quantitativement les performances des méthodes floue et neuronale opérant par analyse simultanée, nous avons exploité l'idée de crisper la sortie du classifieur flou proposée dans Bárdossy et Samaniego (2002).

Les valeurs du taux de bonne reconnaissance fournies par l'approche *AS* floue crispée sont reportées dans le tableau 3.5 .

Classes	Seuil = 0	Seuil = 0.9
1	98.00	68.40
2	86.92	38.46
3	88.51	44.44
4	92.40	24.40
5	90.71	55.00
6	93.84	51.15
7	60.00	16.08
8	72.60	36.96
TMBR	85.91	42.41
Khat	83.26	39.08

Tableau 3.5. Taux de bonne reconnaissance fournis par l'approche *AS* floue crispée (valeurs en %)

Nous remarquons que l'utilisation du seuil dans la méthode floue dégrade énormément le résultat en sortie. Ceci est confirmé par la figure 3.25 où l'aire sous la courbe *ROC* est presque nulle indiquant que cette méthode discrimine très faiblement entre les changements et les non changements.

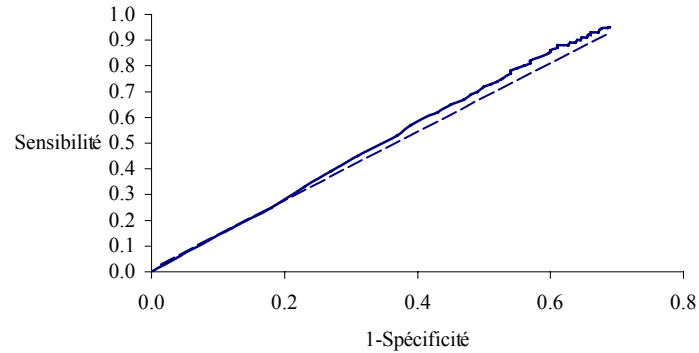


Figure 3.25. Courbe ROC du processus flou crispé

Bien que les résultats semblent assez satisfaisants, la distance de Mahalanobis utilisée dans le calcul du degré d'appartenance manque a priori de précision. Pour améliorer la performance de la détection de changement, nous proposons de remplacer la distance de Mahalanobis par un réseau neuronal afin de calculer les degrés d'appartenance.

3.6 Utilisation des combinaisons neurofloues pour la détection de changement

A mesure que la complexité d'un processus en voie de modélisation augmente la difficulté de développer les règles d'inférence floues augmente aussi. Alors, les combinaisons neurofloues ont été introduites pour optimiser l'extraction de ces règles (Fuller, 1995). Takagi (1997) a rapporté que parmi les combinaisons possibles entre un réseau de neurones (RN) et un système flou (SF) quatre combinaisons sont les plus utilisées (Figure 3.26).

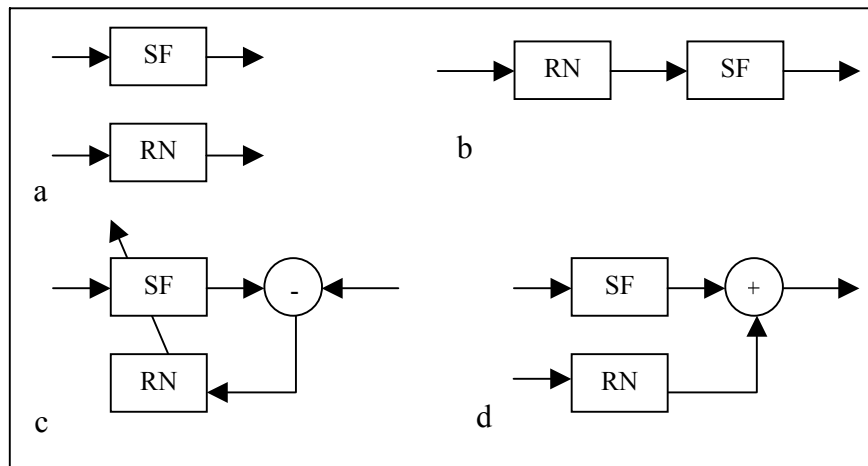


Figure 3.26. Combinaisons neurofloues

Un premier travail en détection de changement utilisant les combinaisons neurofloues a été effectué par Abulgassim et Woodcock en 1999. Les auteurs ont utilisé un réseau à architecture évolutive *ART* (*Adaptive Resonance Theory*) entraîné par l'algorithme *ARTMAP* en utilisant l'approche d'analyse simultanée (Kaufmann, 1999). Ils ont noté que le réseau de neurones adaptatif flou est plus efficace que les méthodes conventionnelles. Par ailleurs, nous n'avons pas trouvé une référence utilisant les combinaisons neurofloues du perceptron

multicouche dans ce contexte. Ainsi, comme notre but consiste à remplacer seulement la distance de Mahalanobis par un réseau de neurones, nous avons adopté la combinaison en cascade représentée par la figure 3.26(b). Dans ce modèle la combinaison est de type $RN \rightarrow SF$ ou $SF \rightarrow RN$. Par conséquent, nous proposons la formule suivante pour établir le degré d'appartenance dans les différentes classes :

$$d_i = \frac{(y_i^2)^x}{\sum_{j=1}^8 (y_j^2)^x} \quad (3.7)$$

y_i : est la valeur de sortie du réseau dans la classe i .

3.6.1 Comparaison entre le modèle neuroflou et le réseau de neurones

Nous comparons dans ce test les performances de la combinaison neurofloue à celles du réseau conventionnel. Pour cela, nous utilisons des réseaux comportant 15 nœuds dans chaque couche cachée. Le seuil de détection est fixé à 0.9 et chaque combinaison neurofloue attribue les pixels à la classe qui correspond au degré d'appartenance ayant la valeur maximale. Le tableau 3.6 résume les résultats obtenus.

Classes	RP : 1 CC (%)		RP : 2 CC (%)		FK : 1 CC (%)		FK : 2 CC (%)	
	RN	CNF*	RN	CNF	RN	CNF	RN	CNF
1	99.20	99.60	99.60	99.60	99.20	99.60	99.20	99.20
2	78.84	91.92	93.46	96.15	82.30	92.30	94.61	94.61
3	81.11	91.85	72.22	78.51	85.92	90.37	90.00	90.74
4	45.60	78.40	73.20	85.60	64.00	91.60	76.40	85.60
5	64.28	85.00	77.21	80.71	64.28	66.42	73.64	79.64
6	49.61	82.30	56.92	93.07	76.15	81.15	81.92	86.15
7	60.86	76.95	64.78	73.91	55.65	66.95	70.43	76.52
8	80.43	96.08	50.43	86.95	57.82	85.65	78.26	78.26
TMBR	69.95	87.78	73.79	86.84	73.54	84.23	83.34	86.50

Tableau 3.6. Taux de bonne reconnaissance des réseaux de neurones et des combinaisons neurofloues

Classes	RP : 1 CC (%)		RP : 2 CC (%)		FK : 1 CC (%)		FK : 2 CC (%)	
	RN	CNF	RN	CNF	RN	CNF	RN	CNF
1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
2	0.22	0.07	0.05	0.01	0.17	0.07	0.04	0.04
3	0.15	0.01	0.24	0.17	0.08	0.04	0.01	0.01
4	0.42	0.07	0.12	0.05	0.26	0.10	0.06	0.07
5	0.34	0.12	0.22	0.17	0.36	0.33	0.23	0.16
6	0.47	0.04	0.37	0.20	0.19	0.11	0.11	0.01
7	0.34	0.15	0.31	0.19	0.40	0.31	0.24	0.16
8	0.18	0.02	0.45	0.07	0.41	0.11	0.20	0.19

Tableau 3.7. Taux de surestimation des réseaux de neurones et des combinaisons neurofloues

* CNF : combinaison neurofloue.

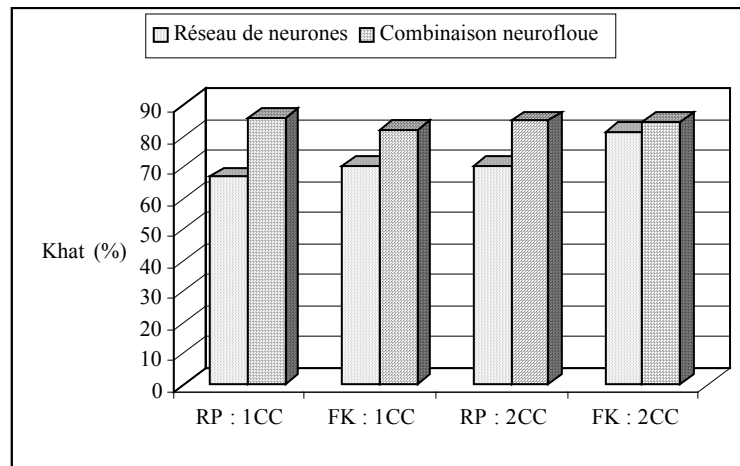


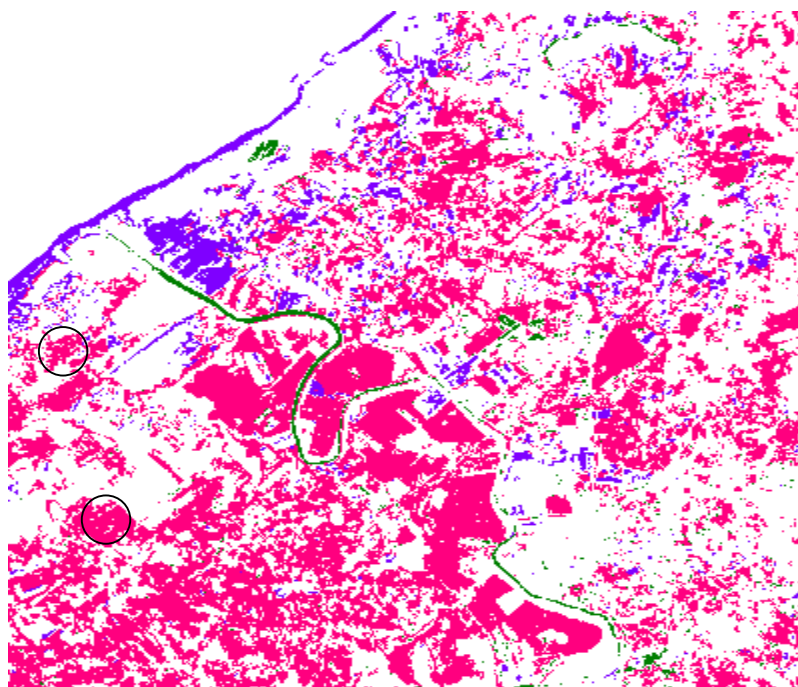
Figure 3.27. Comparaison entre les réseaux de neurones et les combinaisons neurofloues selon la valeur du *Khat*

Les résultats ainsi obtenus permettent de faire une double interprétation. D'une part, comparativement aux résultats donnés dans le tableau 3.5, nous constatons que tous les modèles neuroflous fournissent une meilleure précision que celle du modèle fondé sur la distance de Mahalanobis dans toutes les classes d'intérêt. D'autre part, comparativement aux réseaux de neurones, le tableau 3.6 montre que la combinaison neurofloue permet une augmentation significative dans la valeur du taux de bonne reconnaissance pratiquement dans toutes les classes. Par conséquent, les taux de surestimation sont globalement réduits (Tableau 3.7) et les valeurs du *Khat* ont augmenté (Figure 3.27).

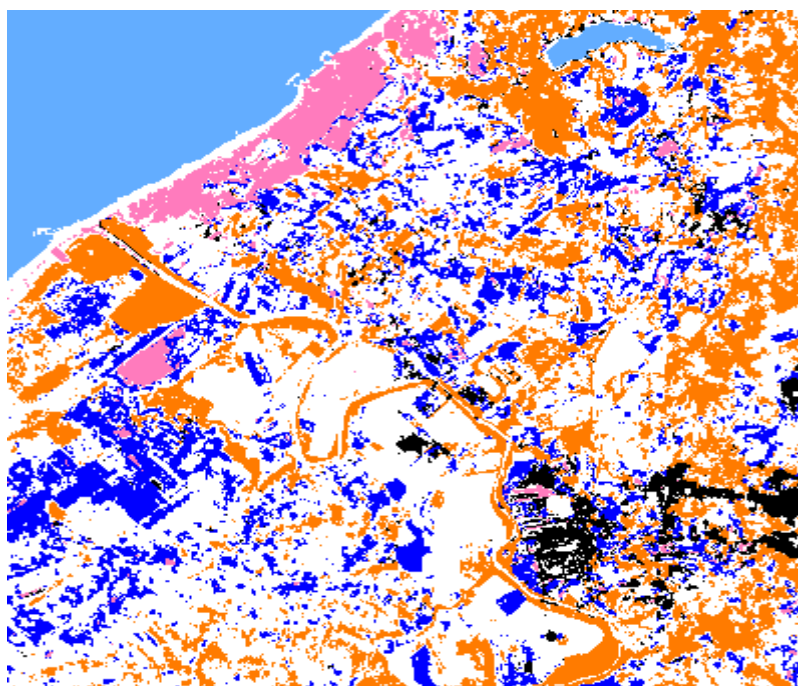
Ces constats sont également confirmés par les tests de simulation (Voir l'annexe D : tableau D.5). Ainsi, nous pouvons conclure que la combinaison neurofloue est plus efficace que les systèmes individuels (réseau de neurones et système flou) pour la détection de changement.

3.6.2 Appréciation visuelle

Pour voir l'apport visuel de la combinaison neurofloue, nous montrons sur les figures suivantes les images obtenues à partir des combinaisons utilisées dans le test précédent.



(a)

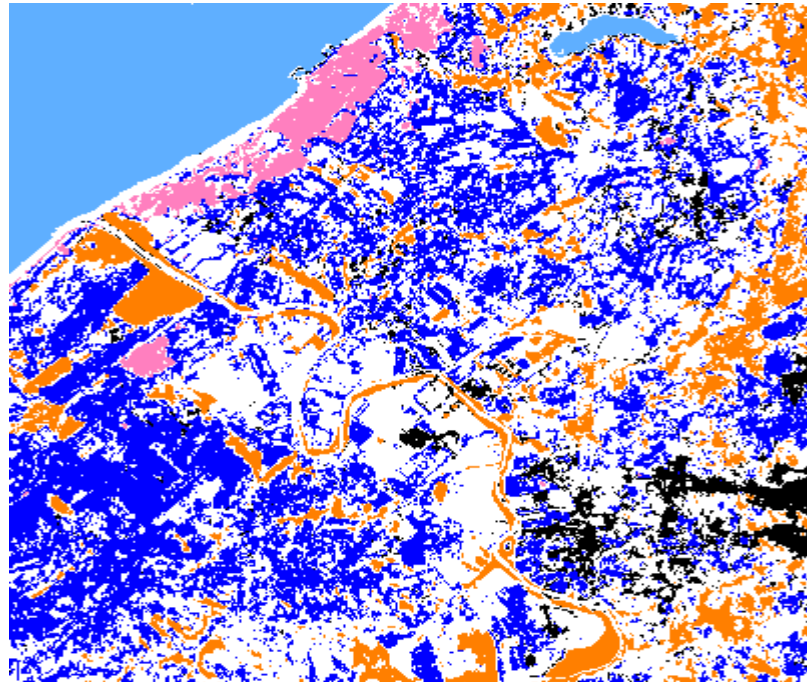


(b)

Figure 3.28. Image obtenue par la combinaison neurofloue du réseau à une couche cachée entraîné par l'algorithme RP (a : classes de changement, b : classes de non changement)



(a)



(b)

Figure 3.29. Image obtenue par la combinaison neurofloue du réseau à une couche cachée entraîné par l'algorithme FK (a: classes de changement, b: classes de non changement)

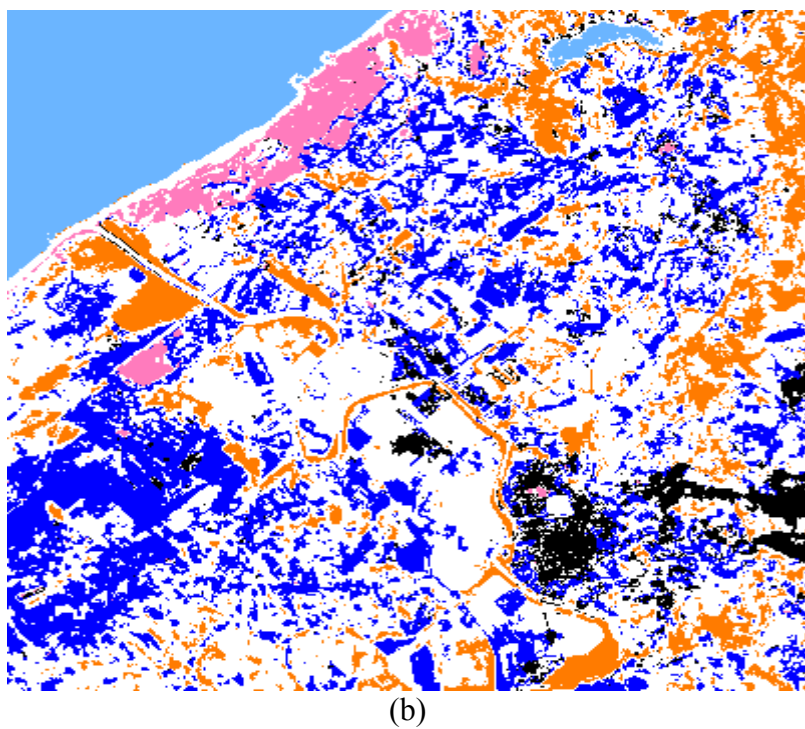
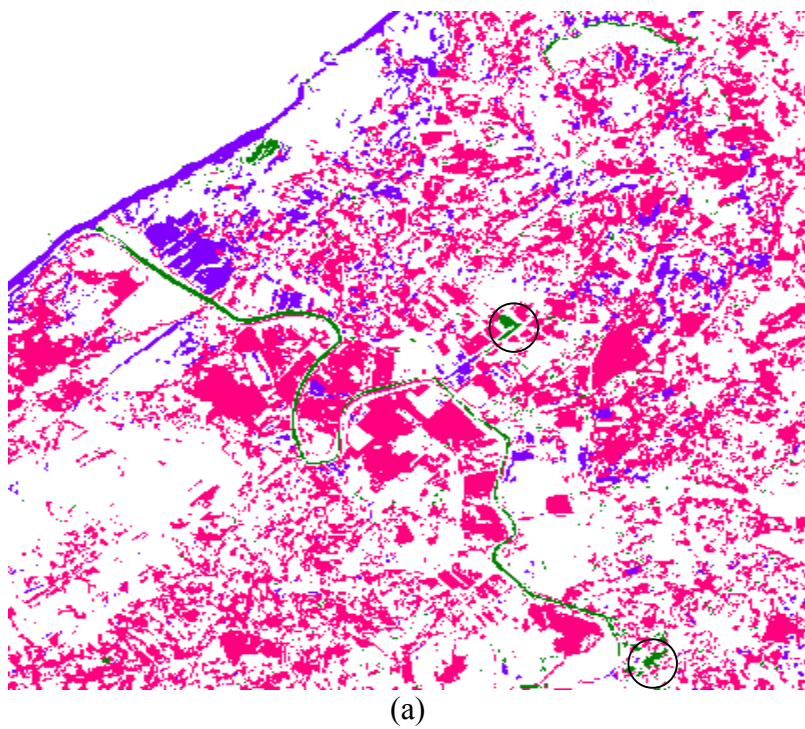
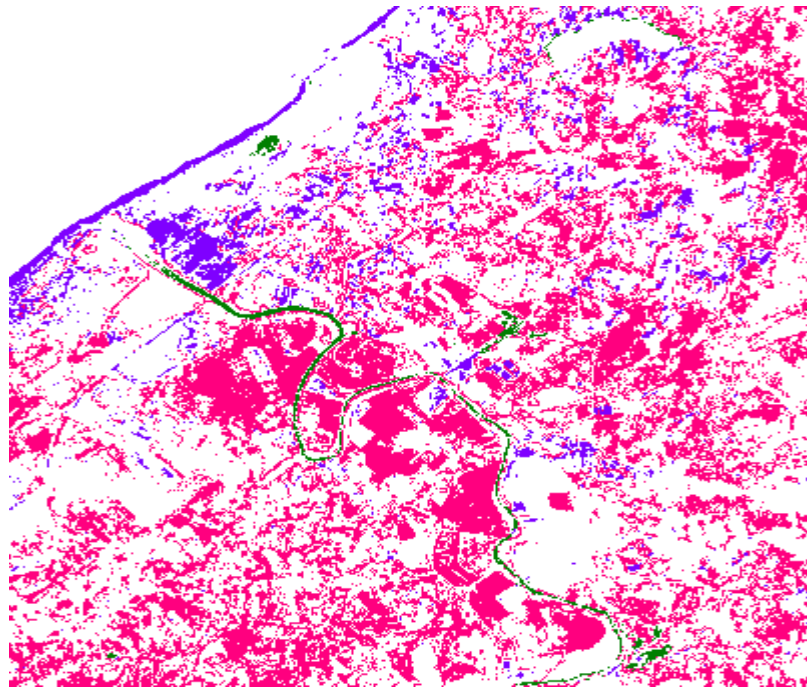
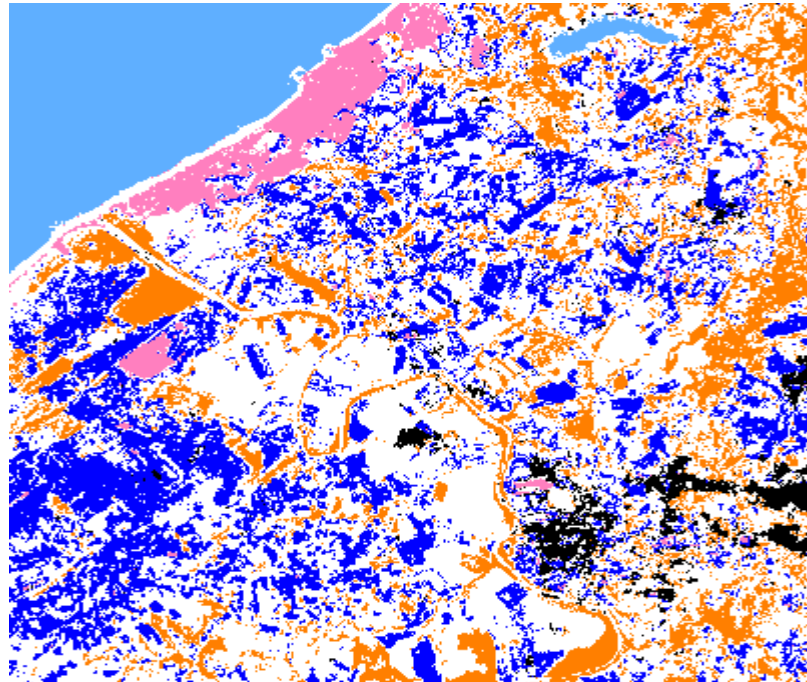


Figure 3.30. Image obtenue par la combinaison neurofloue du réseau à deux couches cachées entraîné par l'algorithme RP (a: classes de changement, b: classes de non changement)



(a)



(b)

Figure 3.31. Image obtenue par la combinaison neurofloue du réseau à deux couches cachées entraîné par l'algorithme FK (a: classes de changement, b: classes de non changement)

Les images obtenues à partir des différentes combinaisons montrent une amélioration importante comparativement aux réseaux de neurones conventionnels (Chapitre 2 : figures 2.12, 2.13, 2.14, 2.15). Les classes sont assez correctement reconnues mais nous notons quand même des confusions dans la classe *eau* $\rightarrow$ *sol nu* qui persistent encore (par exemple les cercles dans la figure 3.30.(a)). Par ailleurs, nous notons que la combinaison du réseau à une couche cachée entraîné par l'algorithme RP, produit une confusion assez forte entre les classes *végétation* $\rightarrow$ *sol nu* et *sol nu* $\rightarrow$ *sol nu* (voir les cercles dans la figure 3.28.(a)).

3.7 Conclusion

L'objectif de ce chapitre est la présentation de l'utilité des ensembles flous pour la détection de changement. Les différents tests réalisés nous ont permis de mettre en évidence que la classification floue munie des normes triangulaires améliore la détection par l'analyse comparative. Par ailleurs, l'approche opérant par analyse simultanée reste globalement plus précise.

Le modèle fourni par la combinaison neurofloue s'est avéré robuste pour le calcul du degré d'appartenance en permettant à la fois, l'amélioration de la précision du réseau de neurones et de la méthode floue.

Chapitre 4

Amélioration de la Détection de Changement par Combinaison de Classifieurs

Résumé : Nous proposons dans ce chapitre une amélioration de la précision de la détection de changement en combinant plusieurs classifieurs. Dans un premier test, nous effectuons une combinaison des réseaux de neurones opérant par analyse simultanée et entraînés par l'algorithme de filtrage de Kalman en utilisant les intégrales floues. Le second test porte sur la combinaison des analyses comparative et simultanée fondées sur la classification floue.

4.1 Introduction

Les résultats obtenus dans les chapitres précédents indiquent qu'il n'est pratiquement pas possible de trouver un détecteur qui donne les précisions les plus élevées dans toutes les classes. Cette variation dans les taux de bonne reconnaissance rend la comparaison des différentes méthodes de détection de changement une tâche qui n'est pas maîtrisable. Dans le domaine de la classification, le fait que tout processus fonctionne différemment d'une classe à l'autre était à l'origine de l'idée de combiner (ou fusionner) les sorties des classifieurs individuels pour construire un consensus de décision et, améliorer ainsi, la caractérisation du sol. Bien que ce concept ne concerne pas spécifiquement la détection de changement, nous nous sommes intéressés à son utilisation. Pour illustrer les performances de la combinaison de classifieurs, le premier test combine des réseaux de neurones procédant par analyse simultanée. Puis, pour avoir une idée plus globale sur son apport, nous effectuons la combinaison des analyses comparative et simultanée floues en exploitant une méthode de combinaison basée sur les mesures floues.

4.2 Combinaison de classifieurs : état de l'art

Le principe de cette approche consiste à faire plusieurs classifications de la scène d'étude et, établir par la suite, une décision collective en utilisant une méthode de combinaison. Généralement, la combinaison est effectuée sans tenir compte de la structure interne de chaque classifieur ni de la méthode utilisée. En revanche, elle se base sur certains critères. D'abord, il existe deux façons de traitement concernant les données en entrée. Dans la première, tous les classifieurs utilisent la même base de données alors que dans la seconde, chaque classifieur utilise une base contenant des caractéristiques spécifiques de ces données (Cho et Kim, 1995 ; Kittler et al., 1998). Par ailleurs, Kittler et al. (1998) ont rapporté que les classifieurs peuvent être conditionnellement dépendants ou indépendants. Dans le cas de l'indépendance conditionnelle des classifieurs, la sortie de chacun d'eux est prise directement. Cette stratégie n'est pas forcément la meilleure puisque les sorties des classifieurs peuvent être de nature différente. Dans ce cas, Chitroub (2003) a noté la difficulté que l'on rencontre dans la phase de combinaison. Il a cité entre autres les différents types de sorties à savoir le type classe, rang, ensemble de classes et le type mesure. Cependant, même dans le cas

contraire où les sorties sont de la même nature cela influe sur le choix de la méthode de combinaison. Si on possède seulement les étiquettes des pixels (type classe), c'est une technique de vote qui est utilisée. Par contre si les sorties des classifieurs sont des probabilités a posteriori, on peut utiliser la moyenne par exemple. En outre, il existe plusieurs approches de combinaison telles que l'approche séquentielle, parallèle, série-parallèle, etc (Chitroub, 2003). Le choix d'une telle ou telle approche dépend en premier lieu de l'utilisateur. Néanmoins, l'approche parallèle où le processus de combinaison ne voit que les sorties des différents classifieurs est la plus utilisée (Figure 4.1). C'est l'approche adoptée dans notre application.

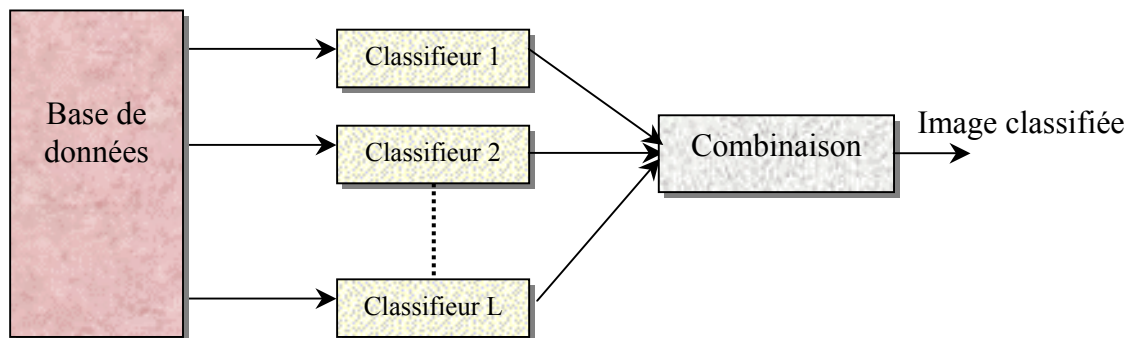


Figure 4.1. Approche de combinaison parallèle des classifieurs

4.3 Méthodes de combinaison

La littérature comporte un nombre important de travaux indiquant l'utilité des procédures de combinaison dans le domaine de la classification. Les méthodes utilisées sont nombreuses, nous citons quelques unes dans ce qui suit.

- *Le vote par majorité* : la classe correcte est celle qui est choisie par la majorité des classifieurs. Il faut que plus de la moitié des classifieurs choisissent cette classe sinon le pixel sera rejeté (Cho, 1995).
- *Le compte Borda* : une autre technique de vote qui consiste à calculer pour chaque classe, le nombre de classes qui sont rangées au dessous d'elle (c'est à dire les classes dont la valeur de sortie est inférieure à celle de la classe en question) pour un classifieur donné. Ainsi, le compte *Borda* est la somme des résultats obtenus à partir des différents classifieurs pour chaque classe. Le pixel est affecté à la classe qui présente le plus grand compte *Borda* (Verikas et al., 1999).
- *La moyenne* : cette technique calcule la moyenne simple des sorties individuelles des classifieurs pour chaque classe. La classe correspondant à la moyenne maximale est prise comme étant la classe correcte du pixel.
- *La combinaison Bayésienne* : Le pixel est affecté à la classe qui correspond à la croyance maximale calculée en utilisant les probabilités conditionnelles de l'appartenance des pixels (Verikas et al., 1999 ; Chitroub, 2003).

Par ailleurs, d'autres méthodes existent telles que la combinaison par l'opérateur de compensation de Zimmermann et les techniques de la théorie de l'évidence (Plusieurs méthodes sont appliquées et évaluées dans: Cho, 1995 ; Autret et Thepaut, 1997 ; Kittler et al., 1998 ; Verikas et al., 1999 ; Lipnikas, 2000 ; Lipnikas et al., 2000 ; Lipnikas, 2001 ; Ho, 2001 ; Cho, 2002 ; Chitroub, 2003).

Généralement, ces méthodes partagent le principe d'utilisation des poids qui indiquent l'importance (ou la compétence) d'un classifieur à reconnaître les pixels d'une classe donnée. Ces poids peuvent être attribués subjectivement par un expert ou calculés à partir d'une base d'entraînement (Cho et Kim, 1995; Cho, 1995). En pratique, un classifieur peut avoir le même poids pour toutes les classes ou un poids pour chaque classe. Ces critères sont plutôt liés à l'utilisateur. Cependant, on peut toujours avoir des pixels qui ne possèdent pas typiquement les caractéristiques de leurs classes. Ainsi, les importances déjà calculées ne conviennent pas vraiment à ces pixels. Pour remédier à ce problème, la méthode de combinaison par les intégrales floues recalcule les importances lors de l'affectation de chaque pixel en tenant compte des réponses des différents classifieurs.

4.4 Combinaison par les intégrales floues

Cette méthode a été proposée par Cho et Kim (1995) pour combiner les sorties de différents réseaux de neurones pour la reconnaissance des caractères écrits à la main. Elle possède la particularité de prendre en considération les changements dans la performance des classifieurs au cours de la combinaison (Cho et Kim 1995 ; Cho, 2000; Verikas et al., 2000 ; Liu et al., 2001 ; Lipnikas, 2001; Cho, 2002). Pour présenter cette méthode, nous rappelons dans la section suivante quelques concepts de la théorie des mesures floues.

4.4.1 Définitions

4.4.1.1 Définition de la mesure floue

Soit Z un ensemble fini non vide.

Une application $g : 2^Z \rightarrow [0, 1]$ est une mesure floue si :

1. $g(\emptyset) = 0, g(Z) = 1$,
2. Si $A, B \subset 2^Z$ et $A \subset B$ alors : $g(A) \leq g(B)$.
3. Si $A_n \subset 2^Z$ pour $1 \leq n < \infty$ et la séquence $\{A_n\}$ est monotone au sens d'inclusion, alors :

$$\lim_{n \rightarrow \infty} g(A_n) = g\left(\lim_{n \rightarrow \infty} A_n\right)$$

La mesure floue de l'union de deux ensembles disjoints ne peut pas être directement calculée à partir des mesures floues ordinaires des sous ensembles. En 1977, Sugeno a introduit la mesure floue λ pour enlever cette limitation (Santos et al., 2000). Ensuite, d'autres mesures ont été proposées telles que la mesure dépendante de la cardinalité de l'ensemble, la mesure floue additive du second ordre, etc.

4.4.1.2 Mesure floue λ

Dans ce cas la mesure de l'union est calculée comme suit :

$$g(A \cup B) = g(A) + g(B) + \lambda g(A)g(B) \quad (4.1)$$

pour tout $A, B \subset Z$ et $A \cap B = \emptyset$.

Pour déterminer λ , on doit résoudre une équation du degré $L-1$ telle que :

$$\prod_{i=1}^L [1 + \lambda g_i] = 1 + \lambda \quad (4.2)$$

L : nombre de sources.

Sugeno a montré qu'il n'existe qu'une seule valeur de λ non nulle appartenant à l'intervalle $]-1, +\infty[$ et vérifiant l'équation (4.2) (Lipnikas et al., 2000).

4.4.1.3 Intégrale floue de Sugeno

En utilisant le concept de la mesure floue λ , l'intégrale floue a été introduite par Sugeno comme étant une opération non linéaire définie sur des ensembles mesurables (Santos et al., 2000).

Soit Z un ensemble fini et $h : Z \rightarrow [0, 1]$ une fonction.

L'intégrale floue sur Z de la fonction h par rapport à une mesure floue g est définie par :

$$e = \int h(x) \circ g = \max_{Z_i \subseteq Z} \left[\min \left(\min_{x \in Z_i} (h(x)), g(A_i) \right) \right] \quad (4.3)$$

où les sources (Classifieurs) Z_i sont permutées pour former la séquence $A_i = \{Z_1, \dots, Z_L\}$ telle que:

$$0 \leq h(Z_L) \leq \dots \leq h(Z_1) \leq 1.$$

Ainsi, les importances sont recalculées comme suit :

- $g(A_1) = g(Z_1) = g_1$.
- $g(A_i) = g_i + g(A_{i-1}) + \lambda g_i g(A_{i-1})$.

Avec :

$g_i = g(Z_i)$ appelée aussi densité de la mesure floue λ .

Le MIN et le MAX sont évalués en utilisant respectivement les normes et les conormes triangulaires. Dans le cas de la classification, le $MIN(h(x))$ revient souvent à $h(x)$ car il correspond aux réponses de la classe provenant d'une seule source. Cette valeur est donc fournie par un classifieur à une classe particulière et elle est unique.

Le calcul de l'intégrale floue par rapport à la mesure floue nécessite seulement la connaissance de la fonction de densité où la g^i est l'importance de la source Z_i dans l'évaluation finale. La valeur obtenue par la comparaison de l'évidence et de l'importance en utilisant l'opérateur MIN est interprétée comme étant le grade d'agrément entre les possibilités réelles $h(x)$ et les prédictions g . L'intégration floue est donc interprétée comme étant la recherche du grade d'agrément maximal entre l'évidence objective et la prédiction.

4.4.1.4 Intégrale discrète de Choquet

En utilisant toujours la notion des mesures floues, Murofushi et Sugeno ont proposé une autre forme de l'intégrale floue appelée "Intégrale discrète de Choquet". Cette intégrale est définie comme suit (Yager, 1999 ; Chang et Greenberg, 2002) :

Si g est une mesure floue sur Z , l'intégrale de Choquet de la fonction $h : Z \rightarrow R^+$ respectivement à la mesure g est définie par :

$$C_g = \{h(Z_1), \dots, h(Z_L)\} = \sum_{i=1}^L \{h(Z_i) - h(Z_{i-1})\} \cdot g(A_i) \quad (4.4)$$

où les indices i ont été permutés pour que : $0 \leq h(Z_1) \leq \dots \leq h(Z_L) \leq 1$.

Avec :

$$h(Z_0) = 0$$

$A_i = \{Z_1, \dots, Z_L\}$ est la séquence de sources obtenue après la permutation.

4.4.2 Méthode de combinaison

La combinaison est décrite par l'algorithme suivant (Cho et Kim, 1995):

Début de la combinaison

```

Pour chaque classe  $t_k$ 
  Pour chaque classifieur  $Z_i$ 
    Déterminer  $g_k(Z_i)$ .
  Fin pour
  Calculer  $\lambda$ .
Fin pour
Pour chaque pixel
  Pour chaque classe  $t_k$ 
    Pour chaque classifieur  $Z_i$ 
      Lire  $h_k(Z_i)$ .
    Fin pour
    Calculer l'intégrale  $e_k$ .
  Fin pour
Fin pour

```

Le pixel est affecté à la classe qui correspond à l'intégrale maximale.

FIN.

Pour définir les importances des classifieurs, plusieurs techniques sont mises en œuvre. Généralement, elles sont soit calculées à partir de la base d'entraînement ou d'une base de validation soit attribuées par un expert. Dans notre application, nous avons sélectionné une base de validation contenant 80 pixels de chaque classe pour calculer les importances des différents détecteurs de changement.

Ainsi, pour chaque classe t_k et pour chaque classifieur Y_i , l'importance est calculée selon l'équation (4.5).

$$g_K(Y_i) = \frac{R_K}{R_K + R_C + R_O + R_R} \quad (4.5)$$

R_K : taux de bonne reconnaissance dans la classe t_k .

R_C : taux de commission dans la classe t_k .

R_O : taux d'omission dans la classe t_k .

R_R : taux de rejet dans la classe t_k .

4.5 Application à la combinaison des classifieurs neuronaux

Les réseaux de neurones sont des processus non linéaires de décision. Ils présentent quelques difficultés dans la pratique car on ne possède jamais des critères qui peuvent aider pour choisir la taille optimale du réseau ainsi que les valeurs des paramètres utilisés dans la phase d'apprentissage. Alors pour surmonter ces limitations on utilise souvent les méthodes d'optimisation telles que les algorithmes génétiques et les algorithmes de Metropolis (Cho, 1995). Une alternative à ces méthodes est la combinaison des réponses des réseaux de neurones. Dans ce sens, plusieurs travaux sur la fusion des réseaux de neurones ont montré qu'un résultat satisfaisant est obtenu à partir de plusieurs réseaux qui sont non seulement imprécis mais aussi, divers dans le sens qu'ils produisent leurs erreurs dans différentes régions de l'espace d'entrée (Lipnikas et al., 2000). Par ailleurs, Cho et Kim (1995) indiquent que si on veut combiner des réseaux de précisions différentes, il est évident qu'il n'est pas adéquat de prendre directement leur simple moyenne ou de faire un simple vote. Ils proposent comme solution l'utilisation de la combinaison par les intégrales floues qui prend en considération la différence des performances des réseaux.

Pour évaluer l'apport de la combinaison des réseaux de neurones pour la détection de changement, nous effectuons la combinaison de trois réseaux de neurones entraînés par l'algorithme de filtrage de Kalman. Les réseaux choisis comportent 10, 15 et 20 nœuds dans chaque couche cachée.

Les paramètres d'apprentissage sont fixés comme suit :

Nombre d'itérations = 400.

Seuil de détection = 0.9.

Pas d'apprentissage = 300.

Facteur d'oubli = 0.995.

4.5.1 Combinaison par l'intégrale de Sugeno

4.5.1.1 Evaluation statistique

Les valeurs des taux de bonne reconnaissance sont fournies dans les tableaux 4.1 et 4.2 pour une combinaison des réseaux à une couche cachée et à deux couches cachées respectivement.

Classes	Réseaux individuels (%)			Combinaison (%)
	10 nœuds	15 nœuds	20 nœuds	
1	97.20	99.20	97.60	100
2	81.15	82.30	82.69	94.23
3	91.85	85.92	83.33	95.55
4	5.10	64.00	62.80	92.40
5	60.00	64.28	47.85	85.71
6	76.92	76.15	71.15	87.30
7	47.39	55.65	55.65	72.60
8	62.17	57.82	58.26	93.91
TMBR	65.12	73.54	70.05	90.34

Tableau 4.1. Taux de bonne reconnaissance des réseaux à une couche cachée et du processus de combinaison

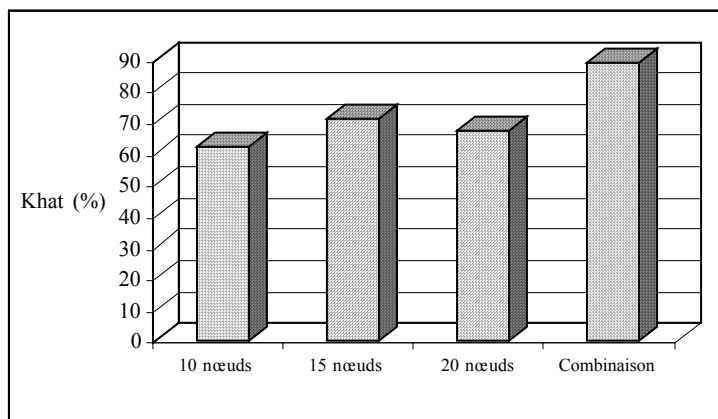


Figure 4.2. Khat des réseaux à une couche cachée et du processus de combinaison

Classes	Réseaux individuels (%)			Combinaison (%)
	10 nœuds	15 nœuds	20 nœuds	
1	99.60	99.20	100	100
2	91.15	94.61	90.38	93.07
3	63.33	90.00	94.81	96.66
4	74.80	76.40	76.00	90.40
5	69.28	73.64	72.50	84.28
6	78.07	81.92	71.92	90.38
7	65.65	70.43	67.39	78.26
8	86.08	78.26	81.17	90.43
TMBR	78.35	83.34	82.02	90.54

Tableau 4.2. Taux de bonne reconnaissance des réseaux à deux couches cachées et du processus de combinaison

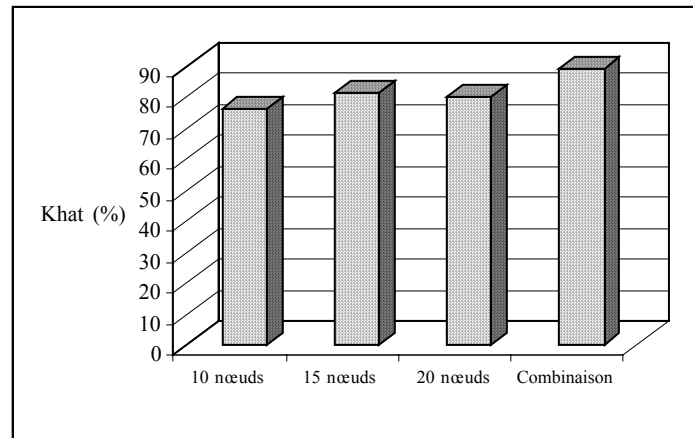


Figure 4.3. Khat des réseaux à deux couches cachées et du processus de combinaison

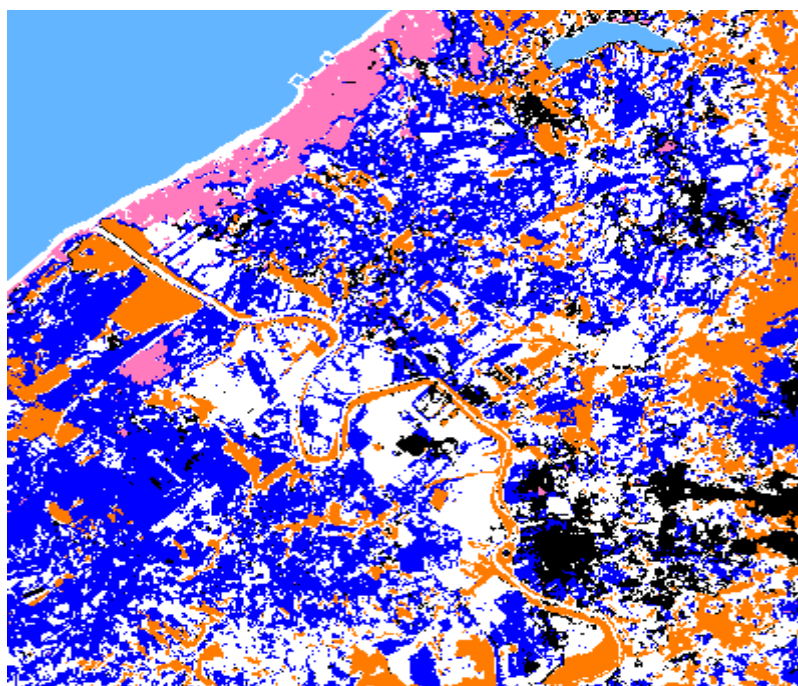
L'examen des résultats montre que la combinaison fournit les taux de bonne reconnaissance les plus élevés pratiquement pour toutes les classes. Nous remarquons aussi que cette approche tend à égaliser les taux de bonne reconnaissance dans les différentes classes.

4.5.1.2 Appréciation visuelle

Pour apprécier visuellement l'apport de la combinaison nous fournissons dans cette section les images produites par la combinaison des réseaux de neurones à une couche cachée et à deux couches cachées.

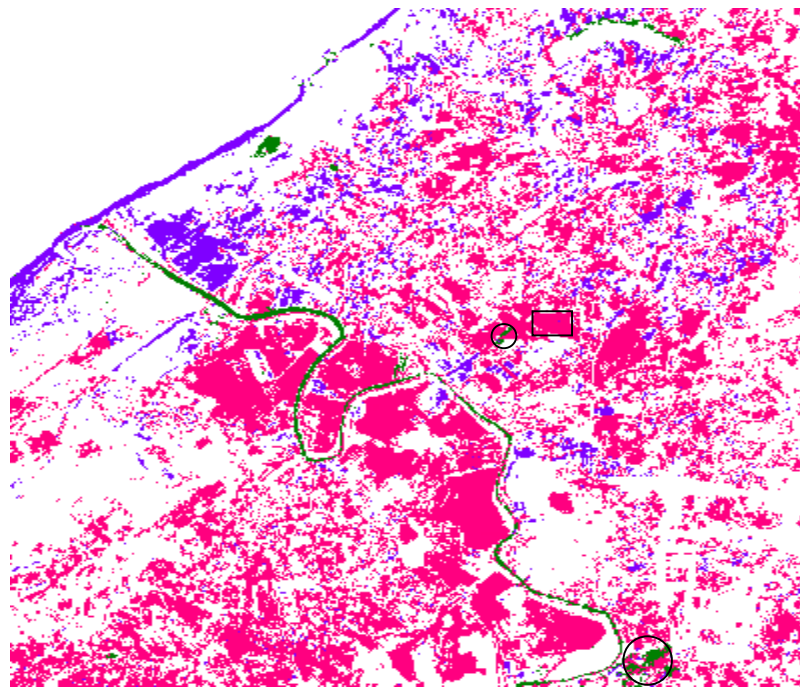


(a)

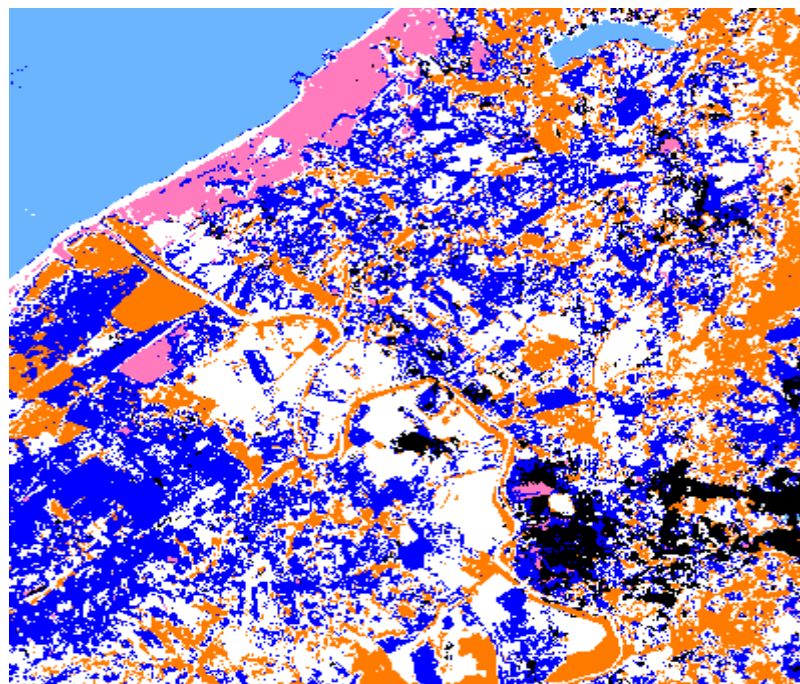


(b)

Figure 4.4. Image produite par la combinaison des réseaux de neurones à une couche cachée
(a : classes de changement, b : classes de non changement)



(a)



(b)

Figure 4.5. Image produite par la combinaison des réseaux de neurones à deux couches cachées
(a : classes de changement, b : classes de non changement)

L'évaluation sur la scène entière montre une amélioration importante dans la qualité de classification dans les classes de changement ainsi que dans les classes de non changement comparativement aux réseaux individuels. Cependant, nous notons des commissions dans la classe *végétation* $\rightarrow$ *sol nu* (dus à des apparitions de végétation : rectangles dans les figures 4.4(a) et 4.5(a)) et dans la classe *eau* $\rightarrow$ *sol nu* (cercles dans les figures 4.4(a) et 4.5(a)).

4.5.2 Comparaison entre l'intégrale de Sugeno et l'intégrale de Choquet

Pour évaluer les différences pouvant exister entre l'intégrale floue de Sugeno et l'intégrale discrète de Choquet, nous avons refait les tests de combinaison en utilisant l'intégrale de Choquet. Ainsi, le tableau 4.3 fournit les taux de bonne reconnaissance (*TBR*) obtenus par les deux intégrales.

Classes	Réseaux 1 cc (%)		Réseaux 2 cc (%)	
	IS	IC	IS	IC
1	100	100	100	100
2	94.23	95.76	93.07	93.07
3	95.55	95.55	96.66	95.18
4	92.40	40.00	90.40	89.60
5	85.71	85.71	84.28	84.64
6	87.30	88.07	90.38	63.46
7	72.60	69.13	78.26	80.86
8	93.91	95.21	90.43	88.69
TMBR	90.34	83.94	90.54	86.94

Tableau 4.3. *TBR* des processus de combinaison par les intégrales de Sugeno (IS) et de Choquet (IC)

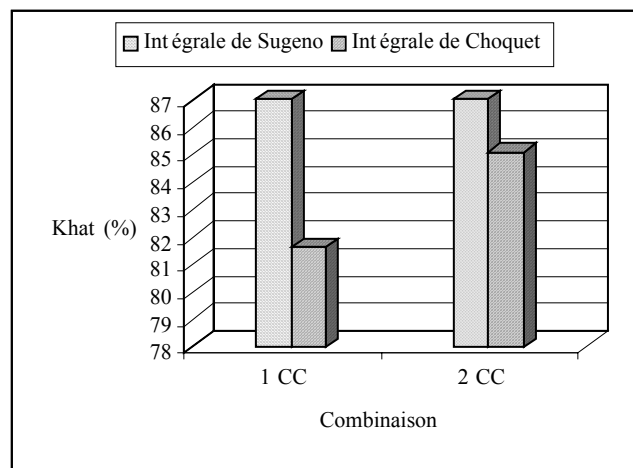


Figure 4.6. *Khat* obtenu par les intégrales de Sugeno et de Choquet

Les résultats ainsi obtenus montrent que l'intégrale de Sugeno et l'intégrale de Choquet ont des performances très rapprochées. Cependant, l'intégrale de Sugeno fournit globalement une meilleure précision. Nous remarquons également qu'au niveau de chaque classe le processus qui donne le *TBR* le plus élevé n'est pas toujours le même. Par conséquent, nous pouvons conclure que ces combinaisons répondent différemment au niveau de chaque classe mais présentent globalement une performance meilleure que celles des réseaux individuels.

4.6 Application à la combinaison des approches *AS* et *AC* floues

4.6.1 Schéma de Combinaison

Dans la partie précédente, nous avons vu l'utilité de la combinaison pour améliorer la précision de la détection de changement. Néanmoins, cette technique n'a pas permis d'éliminer tous les compromis de classification. Pour cela, nous avons pensé qu'il fallait mieux combiner les deux approches d'analyse dont le fonctionnement n'est préalablement pas le même. Nous exploitons ainsi, la méthode de combinaison par les intégrales floues pour combiner les réponses des approches *AS* et *AC* floues utilisant le degré d'appartenance fondé sur la distance de Mahalanobis. Le schéma de combinaison est décrit par la figure suivante :

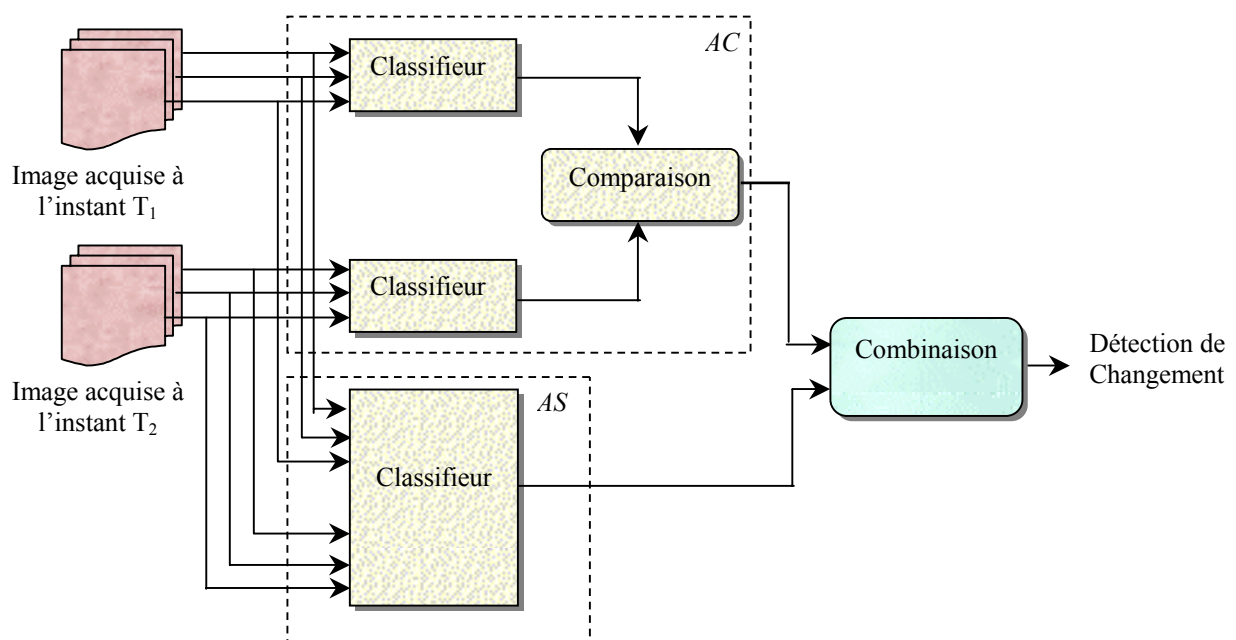


Figure 4.7. Schéma de combinaison des approches *AS* et *AC*

Pour mesurer les importances, nous avons utilisé les *TBR* obtenus en crispant la sortie de chaque processus. La combinaison est réalisée par l'intégrale floue de Sugeno munie de la mesure λ .

4.6.2 Présentation des résultats

4.6.2.1 Evaluation statistique

Puisque chaque pixel est affecté à la classe qui correspond à l'intégrale maximale, les sorties du processus de combinaison ne peuvent pas être interprétées comme étant des degrés d'appartenance. C'est pourquoi, nous comparons les taux de bonne reconnaissance produits par la combinaison avec les taux de bonne reconnaissance des processus *AS* et *AC* flous crispés. Le seuil de détection est fixé à zéro car il n'a pas été pris en considération dans la phase de combinaison. Le tableau 4.4 montre les résultats obtenus.

Classes	<i>AS</i>	<i>AC</i>	Combinaison
1	98.00	100	100
2	86.92	81.15	87.30
3	88.51	92.96	90.74
4	92.40	93.20	97.20
5	90.71	73.21	86.78
6	93.84	83.46	88.84
7	60.00	24.78	46.08
8	72.60	86.52	87.39
TMBR	85.91	79.95	86.56

Tableau 4.4. TBR du processus de combinaison et des approches *AS* et *AC* flous crispés

4.6.2.2 Appréciation visuelle

L'image produite par la combinaison est illustrée sur la figure 4.8.

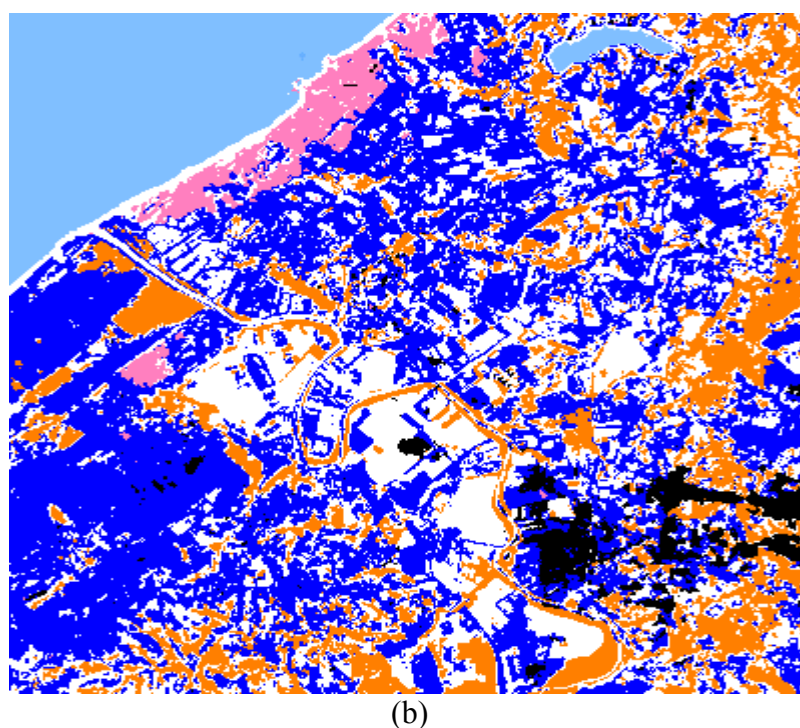
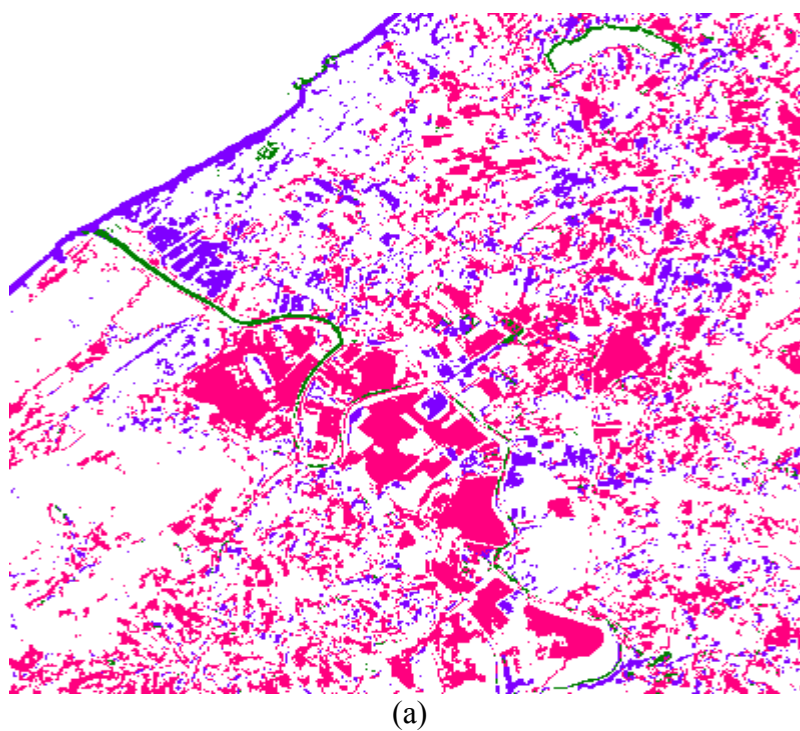


Figure 4.8. Image produite par la combinaison des approches *AS* et *AC* floues
(a: classes de changement, b: classes de non changement)

Quantitativement, la combinaison n'a pas fourni une grande amélioration dans le *TMBR*. Ceci provient du fait que la combinaison se fait uniquement avec deux processus de détection de changement. Cependant, visuellement parlant, l'image obtenue en sortie révèle que la combinaison a permis la réduction des erreurs de classification des processus individuels (Chapitre 3 : figures 3.23 et 3.24).

4.7 Conclusion

Les tests effectués dans ce chapitre confirment l'utilité de l'approche de combinaison par les intégrales floues pour l'amélioration de la détection de changement. Ceci a été prouvé pour différentes combinaisons de trois et de deux réseaux de neurones aussi bien pour les images de simulation que pour les images réelles (Voir l'annexe *D*, tableaux : *D.6* et *D.7*, figure *D.13*). Cependant, ces résultats ne garantissent pas d'avoir une amélioration au niveau de toutes les classes pour toutes les combinaisons. Ce qui est assuré par cette méthode est l'amélioration globale. Cela peut être expliqué par le fait que le résultat est lié à plusieurs facteurs à savoir le nombre de processus combinés, la mesure floue utilisée et la façon selon laquelle on attribue les importances des différents détecteurs de changement.

Conclusion Générale

L'objectif du travail présenté dans cette thèse était l'étude et l'implantation de quelques méthodes de détection de changement dans les images multispectrales de télédétection. Cette fonction s'avère importante pour plusieurs applications comme la surveillance de territoires, la planification de l'utilisation des ressources naturelles ainsi que le contrôle de l'évolution du couvert végétal. Etant donné la diversité des techniques, notre travail s'est focalisé sur les méthodes fondées sur la classification. Ce choix s'est motivé par le fait que ces méthodes permettent non seulement de détecter le changement mais aussi de fournir sa nature précise. Ainsi, nous nous sommes intéressés à la comparaison des approches d'analyse simultanée et comparative des données multitemporelles en utilisant les procédures de classification.

Après avoir décrit brièvement quelques méthodes de détection de changement, la première partie de notre travail a porté sur la comparaison des analyses simultanée et comparative en utilisant la technique de classification par les réseaux de neurones. Nous avons commencé par l'amélioration de la phase d'apprentissage où la performance de l'algorithme de rétropropagation du gradient était comparée à celle de l'algorithme de filtrage de Kalman. Les tests effectués sur les images simulées ainsi que sur les images multitemporelles ont montré que l'algorithme de filtrage de Kalman permet non seulement d'accélérer la convergence du réseau mais aussi, il est plus efficace pour la détection de changement en permettant d'éliminer les confusions produites par la rétropropagation. Par ailleurs, en comparant les deux approches, il s'est révélé que l'analyse simultanée est plus robuste en terme de détection. En effet, dans l'analyse comparative, les taux de confusion et de rejet dans les images classifiées se manifestent par des non détections ou par des fausses alarmes dans l'image résultante et, dégradent ainsi, la précision de la détection.

Pour améliorer les performances de l'analyse comparative, nous avons opté pour l'utilisation des techniques découlant de la théorie des ensembles flous. Dans un premier temps, nous avons rappelé quelques notions sur les ensembles flous et le contexte d'utilisation du degré d'appartenance. L'application de ces notions a permis l'élaboration d'une méthode de classification utilisant la distance de Mahalanobis pour la mesure du degré d'appartenance. L'examen des différents tests effectués, confirme que la classification floue augmente la précision de la comparaison par post classification. Ensuite, comme l'analyse simultanée s'est montrée encore plus performante, nous nous sommes intéressés à la comparaison des méthodes neuronale et floue en utilisant cette approche. Pour cela, nous avons été amenés à crispifier la sortie du processus flou. Cette procédure a conduit néanmoins à une forte altération de la précision. Ainsi, nous avons proposé une modélisation du degré d'appartenance dans laquelle la distance de Mahalanobis est remplacée par un réseau neuronal. Les expériences effectuées sur cette modélisation ont montré qu'elle apporte une amélioration importante de la détection de changement. Elle s'avère ainsi, plus efficace que les systèmes individuels (réseaux de neurones et système flou). Dans ce contexte, nous pensons que l'applicabilité des autres combinaisons neurofloues pour établir le modèle du degré d'appartenance mérite d'être étudiée.

Bien que les tests étaient diversifiés et assez nombreux, nous ne sommes pas parvenus à un détecteur qui permet de produire les taux de précision les plus élevés dans chacune des classes étudiées. Pour améliorer la précision de reconnaissance de chaque classe, nous nous sommes

proposés d'utiliser une méthode fondée sur la combinaison des classifieurs afin d'évaluer son apport dans la détection de changement. Ce thème a fait l'objet de la dernière partie de ce travail dans laquelle nous avons présenté globalement les approches et les méthodes de combinaison de classifieurs. La méthode retenue dans notre étude est basée sur les mesures floues. Celle-ci a été utilisée d'une part, dans la combinaison de trois réseaux de neurones opérant par analyse simultanée et entraînés par l'algorithme de filtrage de Kalman et, d'autre part, dans la combinaison des analyses comparative et simultanée floues. Les tests réalisés confirment que la combinaison de classifieurs permet une augmentation notable dans la précision de la détection de changement. En outre, elle tend à égaliser les taux de bonne reconnaissance dans les différentes classes tout en améliorant la précision globale. Nous pensons ainsi, qu'il serait intéressant de tester l'apport de cette approche en utilisant d'autres opérateurs d'agrégation floue.

Durant ce travail de thèse, nous avons remarqué que la détection de changement demeure une tâche difficile en particulier dans l'extraction des classes de changement. Une autre difficulté à relever est liée à l'utilité des données terrains pour valider efficacement chacune des méthodes. Bien que les tests réalisés sur des images de simulation confirment la robustesse des différentes méthodes appliquées, nous pensons que ce travail peut être complété par une validation avec les données terrains.

Abréviations

<i>ACP:</i>	Analyse en Composantes Principales.
<i>RGB:</i>	Red, Green, Blue: composition colorée.
<i>CPC:</i>	Comparaison par Post Classification.
<i>CCA:</i>	Cross Correlation Analysis : analyse de l'intercorrelation.
<i>UTM:</i>	Universal Transverse Mercator.
<i>AMG:</i>	Australian Map Grid.
<i>MNT:</i>	Modèle Numérique de Terrain.
<i>ROC:</i>	Receiver Operating Characteristic.
<i>TP:</i>	Taux Producteur.
<i>TU :</i>	Taux Utilisateur.
<i>RP :</i>	Rétropropagation.
<i>DF :</i>	Différence.
<i>1cc :</i>	Réseau à une couche cachée.
<i>2cc :</i>	Réseau à deux couches cachées.
<i>FK :</i>	Filtrage de Kalman.
<i>T-normes :</i>	Normes Triangulaires.
<i>T-conormes :</i>	Conormes Triangulaires.
<i>TMBR :</i>	Taux Moyen de Bonne Reconnaissance.
<i>TBR :</i>	Taux de Bonne Reconnaissance.
<i>AS:</i>	Analyse Simultanée.
<i>AC :</i>	Analyse Comparative.
<i>SF:</i>	Système Flou.
<i>RN:</i>	Réseau de Neurones.
<i>ART:</i>	Adaptive Resonance Theory.
<i>IS:</i>	Intégrale de Sugeno.
<i>IC:</i>	Intégrale de Choquet.
<i>TM :</i>	Thematic Mapper.
<i>AVC:</i>	Analyse du Vecteur de Changement.
<i>DA :</i>	Degré d'appartenance.
<i>NDVI :</i>	Normalized Difference of Vegetation Index (Indice de Végétation normalisé).
<i>DIFF-NDVI:</i>	Différence des NDVI.

Annexe A

Algorithmes d'apprentissage du Perceptron Multicouche

Nous introduisons dans cette annexe le calcul détaillé des algorithmes utilisés pour l'apprentissage du perceptron multicouche.

A.1 Algorithme de rétropropagation du gradient

L'algorithme de rétropropagation du gradient est utilisé principalement pour l'apprentissage du perceptron multicouche. Il détermine itérativement les poids du réseau en minimisant l'erreur entre la sortie courante et la sortie désirée.

A.1.1 Calcul détaillé (Hérault et Jutton, 1994)

Dans le perceptron multicouche, les nœuds d'une couche donnée sont connectés à tous les nœuds de la couche suivante. En revanche, il n'existe pas de connexions entre les nœuds d'une même couche. Soit le réseau représenté par la figure A.1 (l'indice de la couche est mis entre parenthèses).

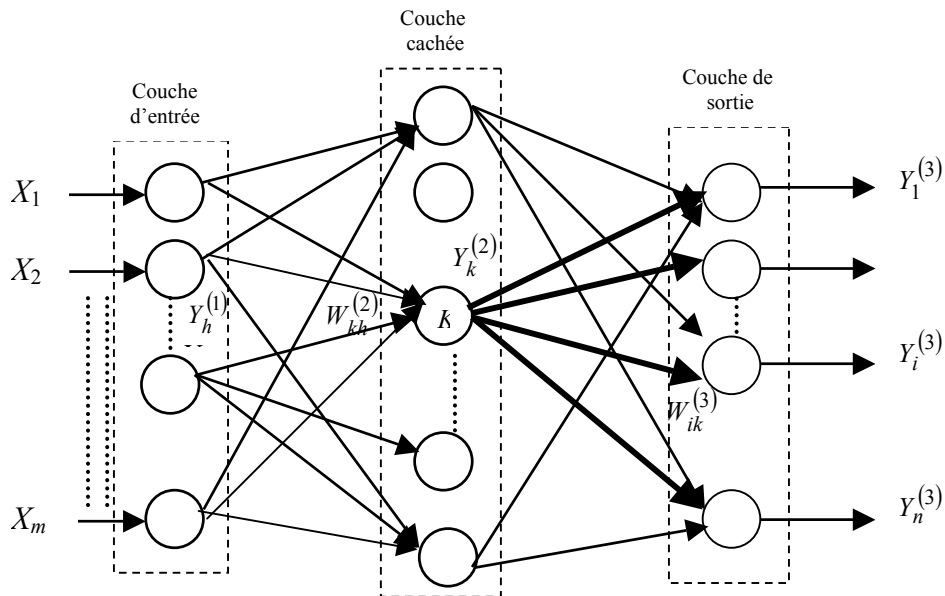


Figure A.1. Perceptron avec une seule couche cachée

On considère que le réseau reçoit en entrée un vecteur de m composantes X_k distribuées sur tous les neurones de la couche d'entrée. La sortie d'un neurone i de cette couche est calculée par :

$$Y_i = \sigma(P_i - \theta) = \sigma \left[\sum_{j=1}^m W_{ij} \cdot X_j - \theta \right] \quad (\text{A.1})$$

où σ : fonction d'activation, θ : seuil constant au niveau d'un nœud, P_i : potentiel calculé au niveau du neurone i , Y_i : sortie du neurone i .

Chaque vecteur d'entrée est associé à un vecteur de sortie désirée Y_d . Puisque les poids W_{ik} ont des valeurs aléatoires, la sortie calculée est a priori différente de Y_d . On calcule l'erreur quadratique moyenne de la différence par :

$$E = \frac{1}{2} \sum_{j=1}^n (Y_{d_j} - Y_j)^2 \quad (\text{A.2})$$

Le gradient de cette erreur par rapport au poids W_{ik} est donné par :

$$\frac{\partial E}{\partial W_{ik}} = \sum_{j=1}^n (Y_{d_j} - Y_j) \frac{\partial (Y_{d_j} - Y_j)}{\partial W_{ik}} \quad (\text{A.3})$$

La dérivée partielle du membre de droite est nulle pour tout $j \neq i$, car seule la sortie Y_i est fonction du poids W_{ik} . De plus, la sortie désirée ne dépend d'aucun poids. Nous obtenons ainsi :

$$\frac{\partial E}{\partial W_{ik}} = -(Y_{d_i} - Y_i) \cdot \frac{\partial Y_i}{\partial W_{ik}} = -(Y_{d_i} - Y_i) \cdot X_k \cdot \sigma'(P_i - \theta) \quad (\text{A.4})$$

En posant : $\delta_i = (Y_{d_i} - Y_i) \cdot \sigma'(P_i - \theta)$, le gradient de l'erreur quadratique moyenne E s'écrit comme suit :

$$\frac{\partial E}{\partial W_{ik}} = -\delta_i \cdot X_k \quad (\text{A.5})$$

La mise à jour des poids selon le principe du gradient s'écrit :

$$\Delta W_{ik} = a \cdot \delta_i \cdot X_k \quad (\text{A.6})$$

a : étant un gain d'adaptation (ou pas d'apprentissage) positif compris entre 0 et 1.

Ainsi, la règle générale de modification des poids de la couche de sortie à la n^{eme} itération est donnée par :

$$W_{ik}(n) = W_{ik}(n-1) + \Delta W_{ik} \quad (\text{A.7})$$

A.1.2 Adaptation des poids dans les couches cachées

Pour établir la règle d'adaptation ou de mise à jour des poids des couches cachées nous reprenons l'équation (A.2) de l'erreur quadratique moyenne à la sortie du réseau. Nous calculons le gradient de cette erreur par rapport au poids W_{kh} associé à la connexion entre le neurone h de la couche j et le neurone k de la couche $j+1$ (l'indice de la couche est mis entre parenthèses).

$$\frac{\partial E}{\partial W_{kh}^{(2)}} = \sum \frac{\partial E}{\partial P_i^{(3)}} \cdot \frac{\partial P_i^{(3)}}{\partial W_{kh}^{(2)}} \quad (\text{A.8})$$

Le premier terme du membre de droite peut être exprimé par :

$$\frac{\partial E}{\partial P_i} = -[Y_{d_i}^{(3)} - Y_i] \cdot \sigma' [P_i^{(3)}] \quad (\text{A.9})$$

Et le second par :

$$\frac{\partial P_i^{(3)}}{\partial W_{kh}^{(2)}} = \frac{\partial}{\partial W_{kh}^{(2)}} \left[\sum_{j \in \text{couche2}} W_{ij}^{(3)} \cdot Y_j^{(2)} \right] \quad (\text{A.10})$$

Nous remarquons que le seul terme qui dépend de $W_{kh}^{(2)}$ est la sortie $Y_k^{(2)}$ obtenue pour $j = k$. Ainsi, l'équation (A.10) se réduit à :

$$\frac{\partial P_i^{(3)}}{\partial W_{kh}^{(2)}} = W_{ik}^{(3)} \cdot \frac{\partial}{\partial W_{kh}^{(2)}} \left[\sigma \left(\sum_{m \in \text{couche1}} W_{km}^{(2)} \cdot Y_m^{(1)} \right) \right] \quad (\text{A.11})$$

Le seul terme de la somme sur m dont la dérivée par rapport à $W_{kh}^{(2)}$ n'est pas nulle est le terme obtenu pour $m = h$. On écrit alors :

$$\frac{\partial P_i^{(3)}}{\partial W_{kh}^{(2)}} = W_{ik}^{(3)} \cdot \sigma' [P_k^{(2)}] \cdot Y_h^{(1)} \quad (\text{A.12})$$

Le gradient de l'erreur E s'écrit donc :

$$\frac{\partial E}{\partial W_{kh}^{(2)}} = - \sum_{i \in \text{couche3}} (Y_{d_i}^{(3)} - Y_i) \cdot \sigma' (P_i^{(3)}) \cdot W_{ik}^{(3)} \cdot \sigma' (P_k^{(2)}) \cdot Y_h^{(1)} \quad (\text{A.13})$$

On pose : $\delta_i^{(3)} = (Y_{d_i}^{(3)} - Y_i) \cdot \sigma' (P_i^{(3)})$ l'équation (A.13) devient :

$$\frac{\partial E}{\partial W_{kh}^{(2)}} = - \left[\sum_{i \in \text{couche3}} \delta_i^{(3)} \cdot W_{ik}^{(3)} \right] \cdot \sigma' (P_k^{(2)}) \cdot Y_h^{(1)} \quad (\text{A.14})$$

On pose aussi :

$$\delta_k^{(2)} = \left[\sum_{i \in \text{couche3}} \delta_i^{(3)} \cdot W_{ik}^{(3)} \right] \cdot \sigma' (P_k^{(2)}) \quad (\text{A.15})$$

L'équation (A.15) montre que l'erreur à la sortie du neurone k de la couche (2) est égale à la somme des erreurs des neurones de la couche suivante (3).

En remplaçant (A.15) dans (A.14) on obtient :

$$\Delta W_{kh}^{(2)} = a \cdot \delta_k^{(2)} \cdot Y_h^{(1)} \quad (\text{A.16})$$

D'une manière générale, dans n'importe quelle couche cachée du réseau, l'incrément d'un poids $W_{kh}^{(j)}$ d'une couche j est le produit de l'entrée $Y_h^{(j-1)}$ provenant de la couche précédente $j-1$ par l'erreur attribuée au neurone k . L'erreur $\delta_k^{(j)}$ du neurone k est la somme des erreurs

$\delta_i^{(j+1)}$ associées aux neurones i de la couche suivante $j+1$ pondérées par les poids synaptiques $W_{ik}^{(j+1)}$ qui connectent les neurones i avec le neurone k . On a alors :

$$\Delta W_{kh}^{(j)} = a \cdot \delta_k^{(j)} \cdot Y_h^{(j-1)} \quad (\text{A.17})$$

avec : $\delta_k = Y_{d_k} - Y_k$ pour la couche de sortie ,

Et pour les couches cachées :

$$\delta_k^{(j)} = \left[\sum_{i \in \text{couche}(j+1)} \delta_i^{(j+1)} \cdot W_{ik}^{(j+1)} \right] \cdot \sigma'(P_k^{(j)}) \quad (\text{A.18})$$

La règle d'adaptation des poids s'écrit finalement à l'itération $t+1$:

$$W_{ik}(t+1) = W_{ik}(t) + a \cdot \delta_i \cdot Y_k^{(j-1)}(t) \quad (\text{A.19})$$

A.1.3 Lissage de la règle d'adaptation

La forme d'adaptation des poids donnée dans l'équation (A.19) est rarement utilisée telle quelle en pratique. Le fait que l'algorithme soit fondé sur la minimisation de l'erreur instantanée et non pas l'erreur calculée sur toute la base d'apprentissage, rend la convergence assez lente car la modification des poids à chaque mise à jour n'est pas importante. De plus, lors du choix du pas d'apprentissage il existe un compromis entre la vitesse de convergence et les instabilités (ou les minimums locaux). C'est pourquoi il est préférable d'utiliser une valeur assez petite du pas d'apprentissage pour que les oscillations de l'erreur soient négligeables. En revanche, pour augmenter la vitesse de convergence du réseau, on a introduit un facteur de filtrage γ sur les incréments d'adaptation connu sous le nom *Momentum*. Ce facteur apporte une pondération exponentielle sur l'erreur comme l'indique l'équation (A.20).

$$E(t) = \frac{1}{2} \sum_{i=1}^n \gamma^{(n-i)} \cdot (Y_d - Y)^2 \quad (\text{A.20})$$

La règle d'adaptation finale est donnée par :

$$W_{jk}(t+1) = W_{jk}(t) + a \cdot \delta_j(t) \cdot X_k(t) + \gamma \cdot \Delta W_{jk}(t) \quad (\text{A.21})$$

(Le calcul qui fait aboutir à l'équation (A.21) est donné dans (Hérault et Jutton, 1994)).

L'algorithme de rétropropagation du gradient se résume comme suit :

1. *Choix de la taille du réseau :*
 - Nombre de couches et de nœuds cachés.
 - Initialisation aléatoire des poids.
 - Choix de la valeur du seuil.
2. *Choix d'une paire (X, Y_d) dans la base d'apprentissage.*
3. *Calcul des sorties des différentes couches pour l'entrée X par l'équation (A.1).*

4. Mise à jour des poids :

- Mise à jour des poids de la dernière couche selon l'équation (A.6),
- Mise à jour des poids de couches cachées selon l'équation (A.21).

5. Si le test d'arrêt n'est pas satisfait on revient à (A.2).

A.2 Algorithme du filtrage de Kalman

Cet algorithme utilise une forme modifiée de la rétropropagation pour réduire l'erreur quadratique entre la sortie actuelle et la sortie désirée du réseau. Il se base sur l'hypothèse que la présence de la fonction non linéaire entre les poids et l'erreur est la raison principale pour laquelle l'algorithme de rétropropagation est lent. Alors, dans cette alternative on ne prend en considération que la partie linéaire de chaque nœud. La détermination des poids devient un problème linéaire car l'erreur est minimisée au niveau des entrées des non-linéarités (Scalero et Tepedelenlioglu, 1992).

A.2.1 Principe

Dans le réseau multicouche on ne connaît que les entrées de la première couche et les sorties désirées de la dernière couche. Notons que les sommes désirées des neurones de la couche de sortie sont connues (par l'inversion de la fonction d'activation). Le reste doit être calculé. En se référant à la figure A.2, nous remarquons que si les entrées $X_{j-1, k}$ de tous les nœuds et si les sorties de tous les sommateurs Y_{jk} sont spécifiées, nous aboutissons à un système d'équations linéaires reliant ces entrées et ces sorties avec les poids W_{jk} . Ce système d'équations linéaires est obtenu en calculant les dérivées partielles de l'erreur par rapport aux poids et est résolu par l'utilisation du filtre de Kalman.

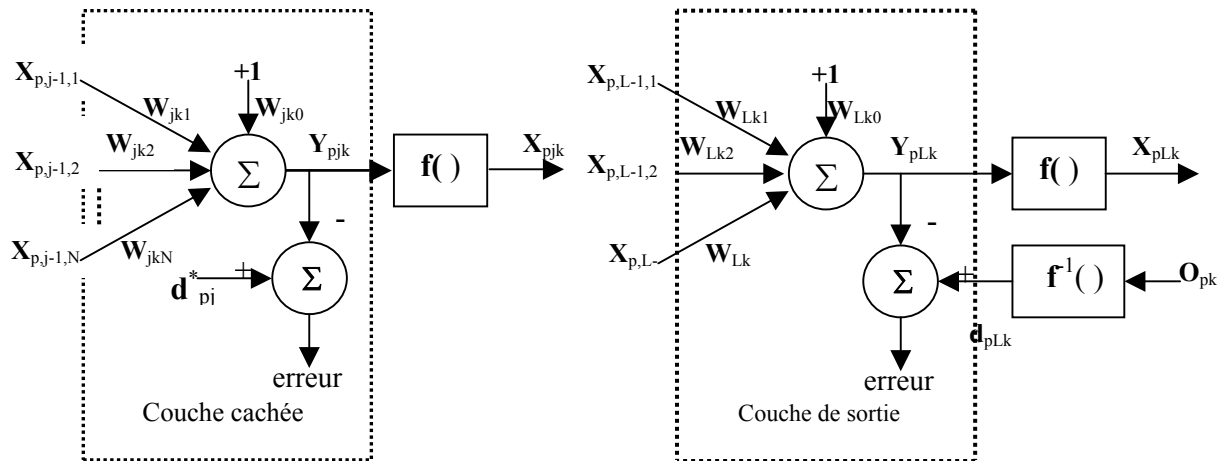


Figure A.2. Portion linéaire d'un neurone dans une couche cachée et dans la couche de sortie

Pour établir les équations de l'algorithme nous prenons en considération un seul neurone. Les indices identifiant les autres neurones sont négligés et seule la partie linéaire est prise en considération dans le calcul.

Le but est de minimiser l'erreur quadratique moyenne E donnée par :

$$E = \sum_{p=1}^M (d_p - y_p)^2 \quad (\text{A.22})$$

M : le nombre d'échantillons d'apprentissage,

y_p : la sortie courante du sommateur,

d_p : la sortie désirée du sommateur.

Pour minimiser E , on annule les dérivées partielles de l'erreur par rapport aux poids W_{jk} . Ainsi, on doit résoudre le système d'équations linéaires suivant :

$$\frac{\partial E}{\partial W_n} = -2 \sum_{p=1}^M (d_p - y_p) \frac{\partial y_p}{\partial W_n} = 0 \quad (\text{A.23})$$

Le résultat est obtenu par la résolution de $N+1$ équations linéaires ($N+1$ étant le nombre de poids dans le neurone).

Au niveau de chaque nœud de $n = 0$, jusqu'à N nous avons :

$$y_p = \sum_{i=0}^N W_i X_{pi} \quad \text{ainsi :} \quad \frac{\partial y_p}{\partial W_n} = X_{pn}$$

L'équation (A.23) s'écrit :

$$\frac{\partial E}{\partial W_n} = - \sum_{p=1}^M \left(d_p - \sum_{i=0}^N W_i \cdot X_{pi} \right) X_{pn} = 0 \quad (\text{A.24})$$

Ou encore :

$$\sum_{p=1}^M d_p X_{pn} = \sum_{p=1}^M \sum_{i=0}^N W_i \cdot X_{pi} \cdot X_{pn} \quad (\text{A.25})$$

On peut écrire aussi :

$$\sum_{p=1}^M \sum_{i=0}^N W_i X_{pi} X_{pn} = \sum_{p=1}^M X_{pn} \sum_{i=0}^N W_i X_{pi} \quad (\text{A.26})$$

En changeant la sommation du membre de droite en une multiplication vectorielle on obtient de $n=0$ jusqu'à N :

$$\sum_{p=1}^M X_{pn} W^T X_p = \left[\sum_{p=1}^M X_{pn} X_p^T W \right] \quad (\text{A.27})$$

A partir des équations (A.25) et (A.27) on définit :

- La matrice de corrélation des échantillons d'apprentissage :

$$R = \sum_{p=1}^M X_p X_p^T \quad (\text{A.28})$$

- Le vecteur d'intercorrélation entre les échantillons d'apprentissage et les réponses désirées correspondantes :

$$P = R \cdot W \quad (\text{A.29})$$

Les poids W sont obtenus par la résolution de l'équation :

$$W = R^{-1}P \quad (\text{A.30})$$

Durant la phase d'apprentissage l'équation (A.30) doit être résolue à chaque nœud du réseau pour réajuster les poids. A chaque itération t les équations (A.28) et (A.29) s'écrivent comme suit :

$$R(t) = \sum_{k=1}^t X(k) \cdot X(k)^T \quad (\text{A.31})$$

$$P(t) = \sum_{k=1}^t d(k) \cdot X(k) \quad (\text{A.32})$$

On remarque que les estimations d'une couche donnée sont basées sur les estimations de la couche précédente. Lorsque cette dernière subit un mauvais traitement, les estimations de la couche courante ne seront pas bonnes. Ce fait allonge la durée de l'apprentissage. Pour remédier à ce problème on a inséré un facteur dit facteur d'oubli b compris entre 0 et 1 dans les équations de calcul de la matrice de corrélation et du vecteur d'intercorrélation tel que :

$$R(t) = \sum_{k=1}^t b^{(t-k)} \cdot X(k) \cdot X(k)^T \quad (\text{A.33})$$

$$P(t) = \sum_{k=1}^t b^{(t-k)} d(k) \cdot X(k) \quad (\text{A.34})$$

Le facteur d'oubli permet à la nouvelle information (la dernière information calculée) de dominer quand les estimations de corrélation ont un effet négligeable sur les estimations courantes. On aboutit enfin au système d'équations suivant :

$$R(t) = b \cdot R(t-1) + X(t)X(t)^T \quad (\text{A.35})$$

$$P(t) = b \cdot P(t-1) + d(t) \cdot X(t) \quad (\text{A.36})$$

$$W(t) = R^{-1}(t) \cdot P(t) \quad (\text{A.37})$$

La forme récursive des équations ci-dessus pose un problème au calcul. Le but de l'apprentissage est d'adapter le traitement de l'erreur E à la solution donnée dans l'équation (A.37) d'une manière itérative. Plusieurs techniques peuvent être utilisées pour résoudre ces équations. Dans ce cas, on propose l'utilisation du filtre de Kalman.

A.2.2 Filtre de Kalman

Pour la j^{eme} couche on donne l'équation du gain de Kalman à l'itération t :

$$K_j(t) = \frac{R_j^{-1}(t-1) \cdot X_{j-1}(t)}{b_j + X_{j-1}^T(t) \cdot R_j^{-1}(t-1) \cdot X_{j-1}(t)} \quad (\text{A.38})$$

L'équation de mise à jour de la matrice inverse est donnée par :

$$R_j^{-1}(t) = [R_j^{-1}(t-1) - K_j(t) \cdot X_{j-1}^T(t) \cdot R_j^{-1}(t-1)] \cdot b_j^{-1} \quad (A.39)$$

Les poids de la couche de sortie sont adaptés selon l'équation suivante :

$$W_{LK}(t) = W_{LK}(t-1) + K_L(t)(d_k - y_k) \quad (A.40)$$

et ceux des couches cachées par :

$$W_{jk}(t) = W_{jk}(t-1) + K_j(t)e_{jk}(t) \cdot \mu_j \quad (A.41)$$

μ_j : pas d'apprentissage.

$X_j(t)$: vecteur d'entrée de la couche j à l'itération t .

b_j : facteur d'oubli de la couche j .

A.2.3 Algorithme

L'algorithme d'apprentissage peut être résumé comme suit :

1. *Initialisation : pour toutes les couches*

- Initialiser les entrées $X_{(j-1), 0}$ de chaque nœud ;
- Initialiser les poids du réseau par des valeurs aléatoires ;
- Initialiser la matrice inverse R^{-1} .

2. *Sélectionner les échantillons de l'apprentissage en spécifiant les paires (X_0, O) ou (entrée, sortie).*

3. *Propager un échantillon vers la sortie du réseau et pour chaque couche j jusqu'à L calculer :*

- La sortie de la somme au niveau de chaque nœud : $Y_{jk} = \sum_{i=0}^N X_{j-1,i} W_{jki}$
- Et la sortie du nœud par :

$$X_{jk} = f(Y_{jk}) = \frac{1 - \exp(-a \cdot Y_{jk})}{1 + \exp(-a \cdot Y_{jk})} \quad (A.42)$$

N : étant le nombre des entrées.

4. *Calculer les équations du filtre de Kalman pour chaque couche de $j = 1$ jusqu'à L :*

- Le gain de Kalman par l'équation (A.38).
- Et la mise à jour de la matrice inverse par l'équation (A.39).

5. *Calculer les erreurs telles que :*

- Calculer la dérivée de $f(y_{jk})$ en utilisant :

$$f'(y_{jk}) = \frac{2a[\exp(-ay_{jk})]}{[1 + \exp(-ay_{jk})]^2} \quad (\text{A.43})$$

- L'erreur de la couche de sortie ($j = L$) :

$$e_{Lk} = f'(Y_{Lk}) \cdot (O_k - X_{Lk}) \quad (\text{A.44})$$

- Et pour les couches cachées de $j = L-1$ jusqu'à 1 :

$$e_{jk} = f'(Y_{jk}) \cdot \sum_i (e_{j+1,i} \cdot W_{j+1,i,k}) \quad (\text{A.45})$$

- Pour chaque nœud de la couche de sortie calculer le potentiel désiré par :

$$d_k = \frac{1}{a} \cdot \log \left[\frac{1+O_k}{1-O_k} \right] \quad (\text{A.46})$$

- Mise à jour des poids :
- Pour la couche de sortie par l'équation (A.40).
- Et pour les couches cachées par l'équation (A.41).

NB :

- W_{jk} : la matrice poids liant la couche j à la couche précédente k .
- a : la constante de la sigmoïde.

6. Test d'arrêt : à cette étape soit on utilise l'erreur quadratique moyenne de la sortie du réseau pour tester la convergence et retourner à 2 si le test n'est pas vérifié, soit on fixe un nombre d'itérations.

Annexe B

Calcul de la Matrice de Confusion et du Coefficient *Kappa*

B.1 Matrice de confusion

La matrice de confusion (ou matrice d'erreur) est un excellent critère d'évaluation de la classification. Cette matrice permet d'estimer non seulement les taux de bonne classification (ou de bonne reconnaissance) mais également les erreurs d'inclusion (ou commission) et les erreurs d'exclusion (ou omission). Il existe deux types de normalisation de la matrice de confusion (Mangolini, 1994):

- *Normalisation en lignes* : elle se fait relativement aux nombres totaux des pixels de chaque catégorie de référence. La matrice ainsi obtenue indique la probabilité qu'un pixel de référence soit classé dans telle ou telle classe. Ces valeurs sont appelées le taux producteur car elles indiquent au producteur de la classification la façon selon laquelle chaque catégorie est classifiée. Pour une ligne correspondant à une classe i on a :

$$TP_i = \frac{X_{ii}}{\sum_j X_{ij}} \quad (B.1)$$

- *Normalisation en colonnes* : une normalisation en colonnes se fait relativement aux nombres totaux des pixels classés dans chaque catégorie et fournit les taux utilisateurs. Ces taux indiquent à l'utilisateur la probabilité qu'un pixel classé dans une classe donnée y appartienne réellement.

$$TU_j = \frac{X_{jj}}{\sum_i X_{ij}} \quad (B.2)$$

Nous avons choisi de mettre les classes de référence en lignes et en colonnes le résultat de la classification.

La différence entre les valeurs du taux producteur et du taux utilisateur d'une classe donnée indique le taux de surestimation dans cette classe (Bárdossy et Samaniego, 2002).

Si le nombre de classes est égal à r et si le nombre total des pixels est N , les différents taux sont calculés comme suit :

$$\text{Le taux moyen de bonne reconnaissance : } TMBC = \frac{\sum_{i=1}^r X_{ii}}{N} \times 100 \quad (B.3)$$

$$\text{Le taux de bonne reconnaissance d'une classe } i : TBR = \frac{X_{ii}}{N_i} \times 100 \quad (B.4)$$

$$\text{Le taux de confusion ou d'erreur d'une classe } i : TC = \frac{\sum_j X_{ij}}{N_i} \times 100 \quad (\text{B.5})$$

Si le nombre de pixel rejetés de la classe i est de R_i le taux de rejet est égal à :

$$TR = \frac{R_i}{N_i} \times 100 \quad (\text{B.6})$$

B.2 Coefficient Kappa

Les techniques statistiques multivariées (ou multivariationnelles) discrètes sont un outil très important pour la mesure de précision. L'analyse *Kappa* est l'une de ces techniques et a été développée par Cohen en 1960 (Banko, 1998). On l'utilise souvent pour mesurer la différence entre la concordance observée et la concordance aléatoire pour donner l'intensité de la concordance réelle. L'application de cette analyse fournit le coefficient *Kappa* dont l'estimation la plus répandue est la statistique *Khat* $\hat{K}$. Cette statistique est calculée à partir de la matrice de confusion telle que :

$$\hat{K} = \frac{N \sum_{i=1}^r X_{ii} - \sum_{i=1}^r X_{i+} X_{+i}}{N^2 - \sum_{i=1}^r X_{i+} X_{+i}} \quad (\text{B.7})$$

r : nombre de lignes et de colonnes dans la matrice de confusion.

N : nombre total d'observations (ou de pixels).

X_{ii} : concordance observée dans la ligne i et la colonne i (ou le taux de bonne classification).

X_{i+} : marginale d'une ligne.

X_{+i} : marginale d'une colonne.

$\sum X_{i+} X_{+i}$: concordance aléatoire.

Pour mieux interpréter la formule du *Khat*, on utilise souvent l'écriture suivante :

$$\hat{K} = \frac{P_0 - P_e}{1 - P_e} \quad (\text{B.8})$$

$$P_0 : \text{précision de la concordance observée, } \frac{\sum X_{ii}}{N} \quad (\text{B.9})$$

$$P_e : \text{estimation de la concordance aléatoire, } \frac{\sum X_{i+} X_{+i}}{N^2} \quad (\text{B.10})$$

Cette technique a été intégrée au traitement de données de télédétection par Congalton et Mead en 1983 qui ont utilisé le coefficient *Kappa* pour mesurer le taux de concordance entre la classification et la référence dans la matrice de confusion (Banko, 1998). Par la suite, cette technique est devenue très utilisée pour mesurer la précision de la classification et même recommandée comme étant la mesure standard de la précision. La concordance est d'autant plus élevée que la valeur du *Khat* est proche de 1, Alors qu'elle est nulle si la concordance observée est égale à la concordance aléatoire.

Annexe C

La Courbe Receiver Operating Characteristic : *ROC*

La courbe *ROC* est une méthode robuste pour évaluer la sensibilité au coût des classifieurs (Dietterich, 2001). Plusieurs représentations de cette courbe existent, la plus générale est décrite comme suit:

Un test de détection de changement cherche à discriminer entre deux classes de changement (*CH*) et sans changement (*SCH*). Le résultat de la méthode appliquée sera décrit par la matrice d'erreur. La validité du test est mesurée par les valeurs conjointes de sa sensibilité S_n et de sa spécificité S_p . La sensibilité du test correspond à sa capacité de se révéler positif lorsque l'anomalie est présente (dans notre cas détecter les bons changements). La spécificité correspond à sa capacité de se révéler négatif lorsque l'anomalie est absente (dans notre cas détecter les bons pixels inchangés). Si l'ensemble *CH* contient M_1 pixels et l'ensemble *SCH* contient M_0 pixels, nous obtenons la matrice d'erreur suivante :

	CH	SCH	Total
CH	a	b	N_1
SCH	c	d	N_0
Total	M_1	M_0	N

On calcule la sensibilité par :

$$S_n = \frac{a}{M_1} \quad (\text{C.1})$$

Et la spécificité par :

$$S_p = \frac{d}{M_0} \quad (\text{C.2})$$

La capacité discriminante de la méthode est mesurée par le rapport de côtes qui est donné par :

$$RC = \frac{ad}{bc} = \frac{S_n \times S_p}{(1 - S_n) \times (1 - S_p)} \quad (\text{C.3})$$

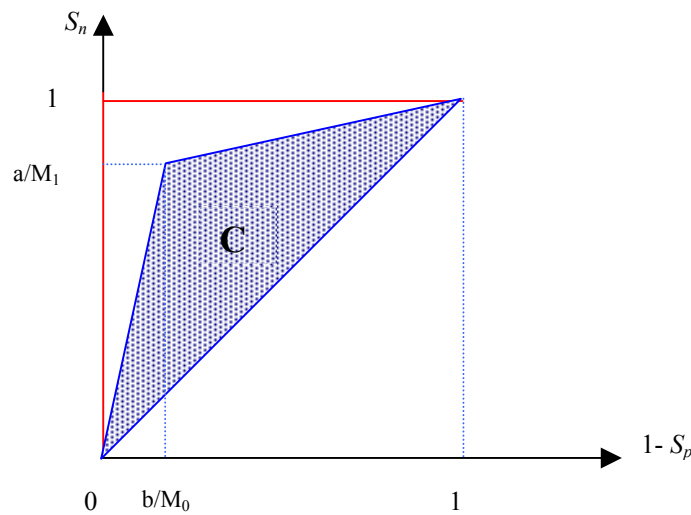
Une méthode dont la capacité discriminante est nulle présente un rapport de cotes égal à 1. Par ailleurs, si elle discrimine bien les pixels de changement des pixels sans changement, son rapport de côtes sera très grand. La capacité discriminante du test peut être parfaitement traduite par la mesure d'un certain coefficient C . Ce coefficient, toujours compris entre 0 et 1, a la propriété d'être égal à 1 quand il y a une séparation parfaite, égal à 0,5 quand la séparation

entre les deux classes est nulle et inférieur à 0,5 lorsque la séparation est inversée (situation où $S_n + S_p < 1$). Il correspond ainsi, à la probabilité que les pixels de changement et les pixels sans changement sont correctement classés.

Il est calculé par l'équation suivante :

$$C = \frac{2ad + ab + cd}{2M_1M_0} \quad (C.4)$$

Géométriquement, ce coefficient correspond à l'aire sous la courbe S_n en fonction de $1 - S_p$. La sensibilité et la spécificité sont calculées selon un seuil compris entre 0 et 1 (ou entre le minimum et le maximum des valeurs de sortie obtenues par la méthode appliquée). La courbe obtenue est appelée la courbe ROC. La figure suivante présente un exemple de la courbe ROC.



Quand le test a une discrimination parfaite, le coefficient C correspond à la surface du triangle tracé en rouge et est égal à 1. En pratique plus l'aire sous la courbe est grande plus la méthode est plus discriminante et donc plus efficace.

Annexe D

Présentation des Résultats des Tests de Simulation

Nous présentons dans cette annexe les résultats obtenus par les tests de simulation. Pour cela, nous avons utilisé une image *SPOT* couvrant une région de l'Algérie à partir de laquelle nous avons généré une autre image contenant les changements. Pour évaluer les performances de chaque méthode de détection de changement, nous avons sélectionné cinq classes de non changement et quatre classes de changement (Figure D.1).

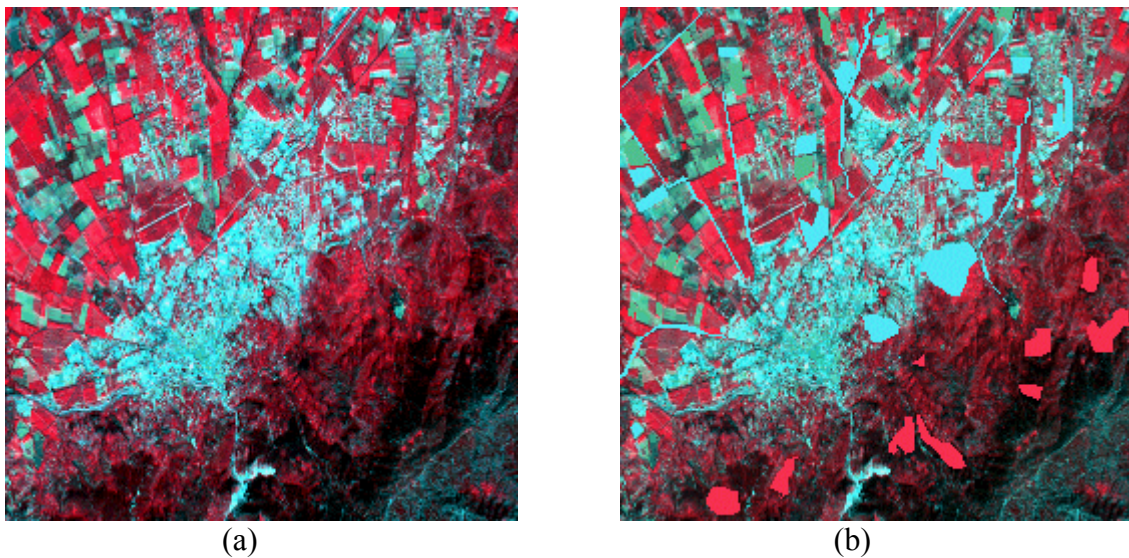


Figure D.1. Composition colorée des images multispectrales
(a : scène réelle, b : scène modifiée)

La base d'entraînement utilisée comprend les thèmes suivants :

Classes	Description	Données	
		Apprentissage	Test
1	<i>Végétation</i> → <i>végétation</i>	800	1000
2	<i>Végétation dense</i> → <i>végétation dense</i>	800	1000
3	<i>Urbain</i> → <i>urbain</i>	800	1020
4	<i>Sol nu</i> → <i>sol nu</i>	800	1000
5	<i>Sol sec</i> → <i>sol sec</i>	800	1000
6	<i>Végétation</i> → <i>urbain</i>	800	1000
7	<i>Végétation</i> → <i>sol nu</i>	800	1000
8	<i>Sol nu</i> → <i>urbain</i>	800	1050
9	<i>Végétation dense</i> → <i>végétation</i>	800	750

Tableau D.1. Données de la base d'entraînement

D.1 Utilisation des Réseaux de Neurones pour la Détection de Changement

Paramètres de l'algorithme RP:

- ✓ Momentum = 0.995.
- ✓ Pas d'apprentissage = 0.08.

Paramètres de l'algorithme FK:

- ✓ Pas d'apprentissage = 40.
- ✓ Facteur d'oubli = 0.995.

D.1.1 Analyse simultanée

D.1.1.1 Vitesse de convergence

Nous présentons graphiquement l'évolution de l'erreur quadratique moyenne au cours du temps pour un réseau contenant 20 nœuds dans chaque couche cachée.

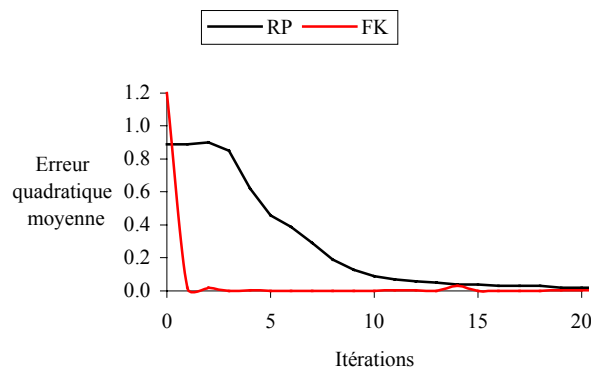


Figure D.2. Evolution de l'erreur quadratique moyenne au cours du temps pour un réseau à une couche cachée

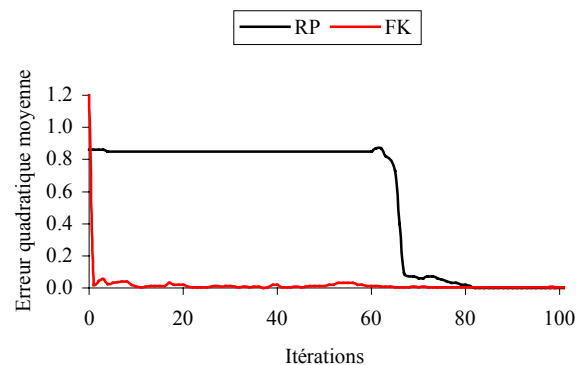


Figure D.3. Evolution de l'erreur quadratique moyenne au cours du temps pour un réseau à deux couches cachées

D.1.1.2 Influence du nombre de nœuds

Nous présentons graphiquement l'influence du nombre de nœuds sur la vitesse de convergence. Dans ce test, nous cherchons le nombre d'itérations nécessaires pour atteindre une erreur fixée à 0.01.

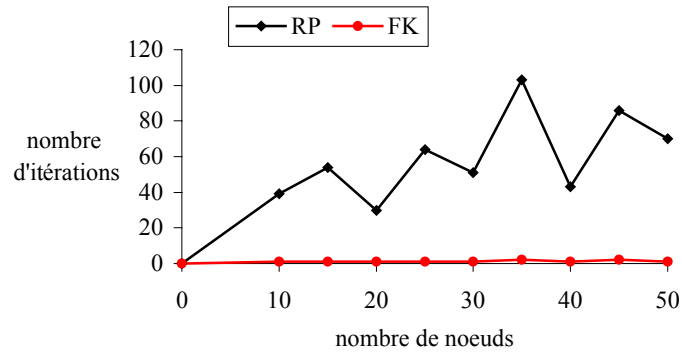


Figure D.4. Influence du nombre de nœuds pour un réseau à une couche cachée

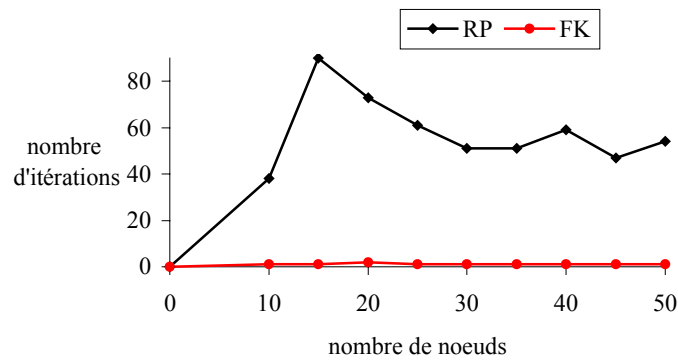


Figure D.5. Influence du nombre de nœuds pour un réseau à deux couches cachées

D.1.1.3 Influence du nombre d'itérations

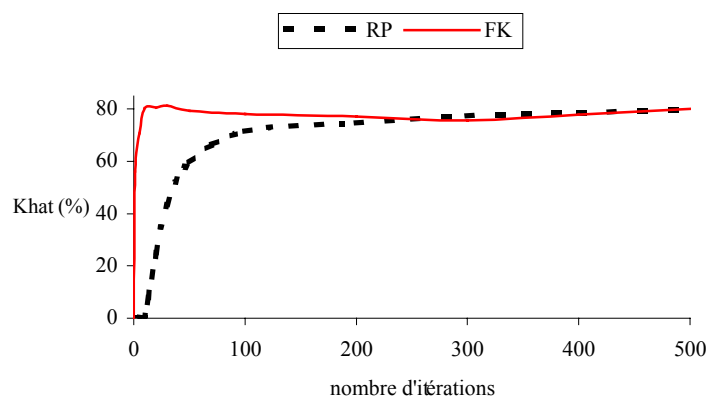


Figure D.6. Influence du nombre d'itérations (Réseau 1 cc)

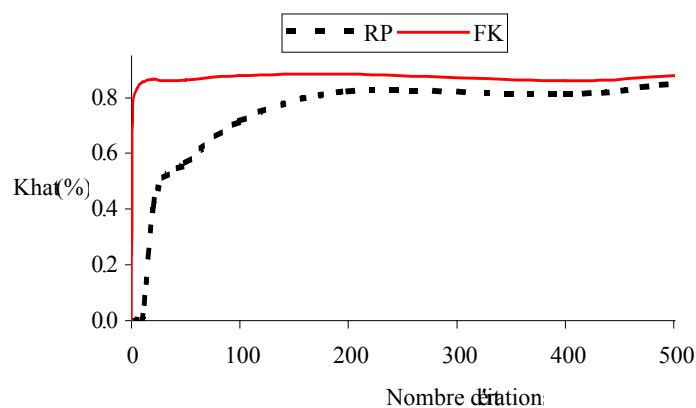


Figure D.7. Influence du nombre d'itérations (Réseau 2 cc)

D.1.1.4 Test de détection de changement

Réseau	$Khat$ (%)
RP. 1 cc	74.19
FK. 1 cc	75.09
RP. 2 cc	81.15
FK . 2 cc	87.63

Tableau D.2. Valeurs du $Khat$ pour les différents réseaux (15 nœuds dans chaque couche cachée)

D.1.1.5 Courbes ROC

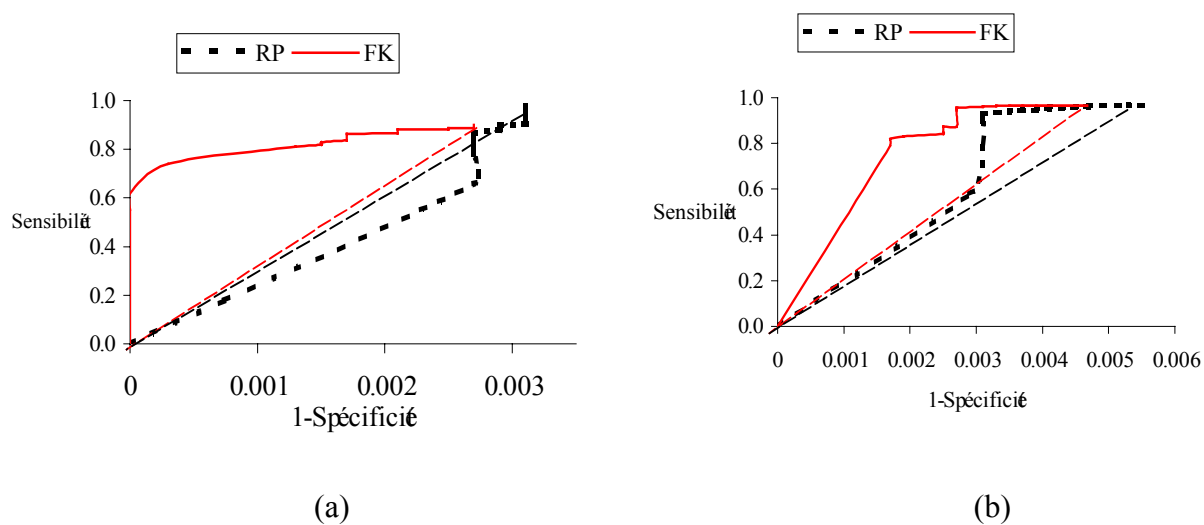


Figure D.8. Courbes ROC (a : réseau à une couche cachée, b : réseau à deux couches cachées)

D.1.2 Etude comparative des approches *AS* et *AC*

Nous présentons les résultats obtenus pour le réseau à une couche cachée car il a été suffisant pour montrer l'efficacité de l'algorithme FK par rapport à l'algorithme RP. Le tableau D.3 présente les *TMBR* obtenus pour différents réseaux.

Nombre de nœuds	<i>AC</i> (%)		<i>AS</i> (%)	
	RP: 1cc	FK: 1cc	RP :1cc	FK :1cc
10	45.25	51.77	75.7	77.97
15	39.57	58.81	76.43	77.27
20	49.14	58.42	76.31	77.13
25	48.47	58.97	73.47	79.35
30	49.43	58.13	74.21	79.34
35	48.50	59.85	78.03	78.96
40	49.84	61.56	75.48	79.76
45	51.34	61.05	75.79	79.17
50	51.39	59.98	74.23	80.36

Tableau D.3. *TMBR* des approches *AS* et *AC*

D.2 Détection de Changement par les Opérateurs de la Logique Floue

D.2.1 Etude comparative des approches *AS* et *AC* floues

Le tableau D.4 présente les degrés d'appartenance (donnés en %) dans chaque classe, obtenus en utilisant les approches *AS* et *AC*.

Classes	<i>AC</i> (%)	<i>AS</i> (%)
1	97.07	89.06
2	93.33	27.10
3	49.23	75.03
4	70.29	93.26
5	87.44	44.38
6	63.30	86.30
7	62.25	76.49
8	3.16	96.19
9	84.95	93.93
DAG	66.96	75.26

Tableau D.4. Degrés d'appartenance par classe et global des approches *AS* et *AC* floues

D.2.2 Etude comparative des méthodes neuronale et neurofloue

Les réseaux utilisés contiennent 20 nœuds dans chaque couche cachée.

Classes	RP : 1 CC (%)		RP : 2 CC (%)		FK : 1 CC (%)		FK : 2 CC (%)	
	RN	CNF	RN	CNF	RN	CNF	RN	CNF
1	98.30	99.90	100	100	79.30	99.80	99.50	99.90
2	51.80	89.80	85.60	92.20	55.90	80.80	79.20	88.09
3	14.11	26.27	2.35	29.41	22.35	55.86	41.17	68.13
4	69.40	92.90	94.40	95.70	66.80	86.60	88.60	92.30
5	91.30	98.09	94.50	98.80	80.80	91.90	94.30	95.30
6	96.90	98.80	98.90	98.90	96.30	97.20	98.50	98.59
7	89.80	97.09	95.40	96.70	94.20	96.00	95.90	93.60
8	93.23	99.33	99.33	99.61	99.90	99.71	98.95	98.28
9	94.38	96.43	95.34	96.43	93.15	94.15	92.87	94.52
TMBR	76.31	88.42	84.67	89.46	77.13	88.97	87.46	91.98

Tableau D.5. TBR obtenus par les méthodes neuronale et neurofloue

D.3 Amélioration de la Détection de Changement par Combinaison de Classifieurs

D.3.1 Combinaison des réseaux de neurones

D.3.1.1 Combinaison des réseaux à une couche cachée

Classes	10 nœuds	15 nœuds	20 nœuds	Combinaison
1	95.90	95.80	79.30	100
2	55.40	60.60	55.90	89.60
3	13.03	24.60	22.35	72.15
4	71.90	66.90	66.80	95.09
5	79.80	68.90	80.80	96.60
6	98.50	97.80	96.30	99.00
7	96.00	89.70	94.20	96.60
8	99.90	99.90	99.90	99.90
9	96.57	96.30	93.15	96.98
TMBR	77.97	77.27	77.13	93.88
Khat	75.84	75.09	74.93	91.81

Tableau D.6. Performance du processus de combinaison comparativement aux réseaux individuels (les valeurs sont données en %)

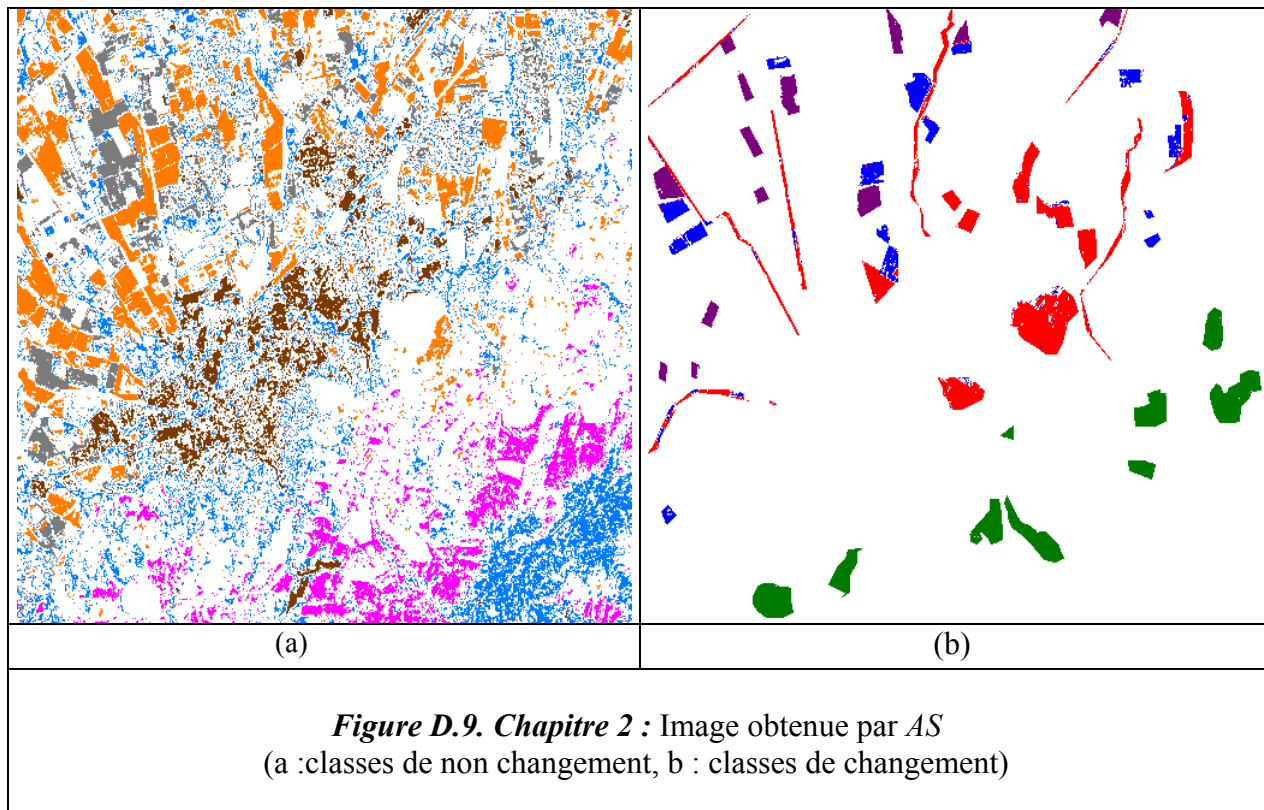
D.3.1.2 Combinaison des réseaux à deux couches cachées

Classes	10 nœuds	15 nœuds	20 nœuds	Combinaison
1	100	100	99.50	100
2	81.70	80.80	79.20	93.60
3	42.54	55.58	41.17	77.45
4	86.50	90.60	88.60	95.30
5	87.50	89.50	94.30	94.50
6	97.20	96.80	98.50	98.70
7	97.00	92.50	95.90	97.40
8	99.90	99.71	98.95	99.90
9	96.98	96.84	92.87	96.98
TMBR	87.38	88.89	87.46	94.79
Khat	85.96	87.63	86.07	94.13

Tableau D.7. Performance du processus de combinaison comparativement aux réseaux individuels (les valeurs sont données en %)

D.4 Evaluation sur la scène entière

Les images obtenues à partir des différents tests en utilisant un réseau à une couche cachée comportant 15 nœuds et entraîné par l'algorithme FK, sont illustrées sur les figures suivantes.



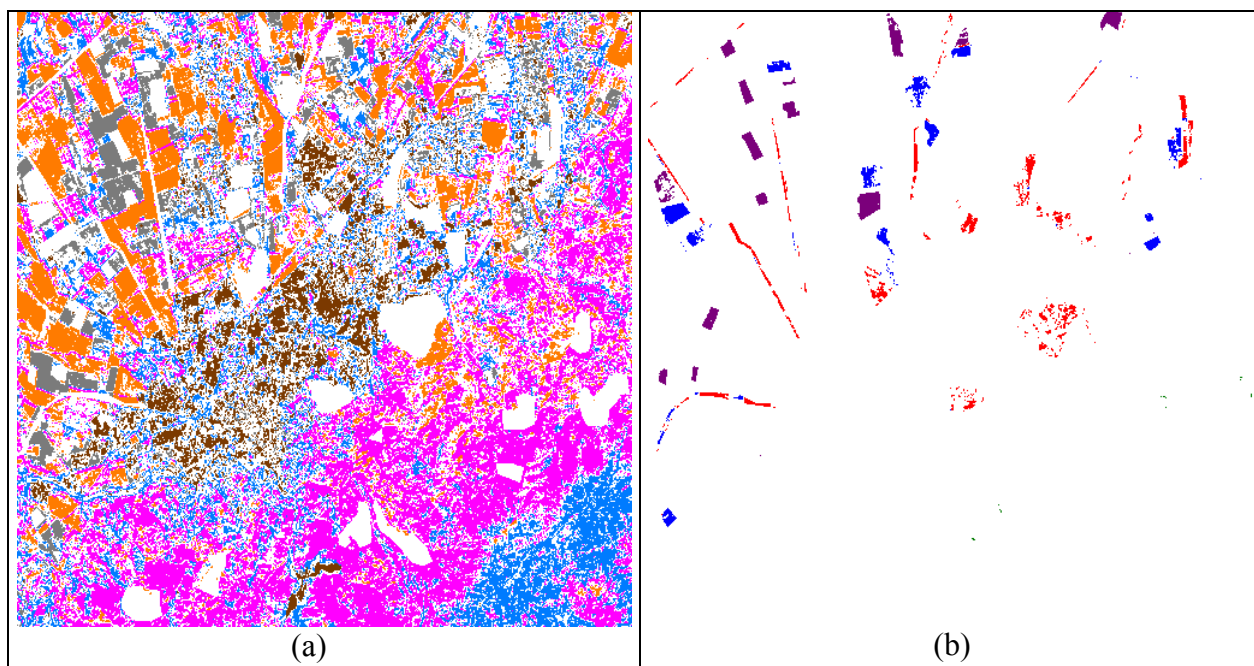


Figure D.10. Chapitre 2 : Image obtenue par AC
(a : classes de non changement, b : classes de changement)

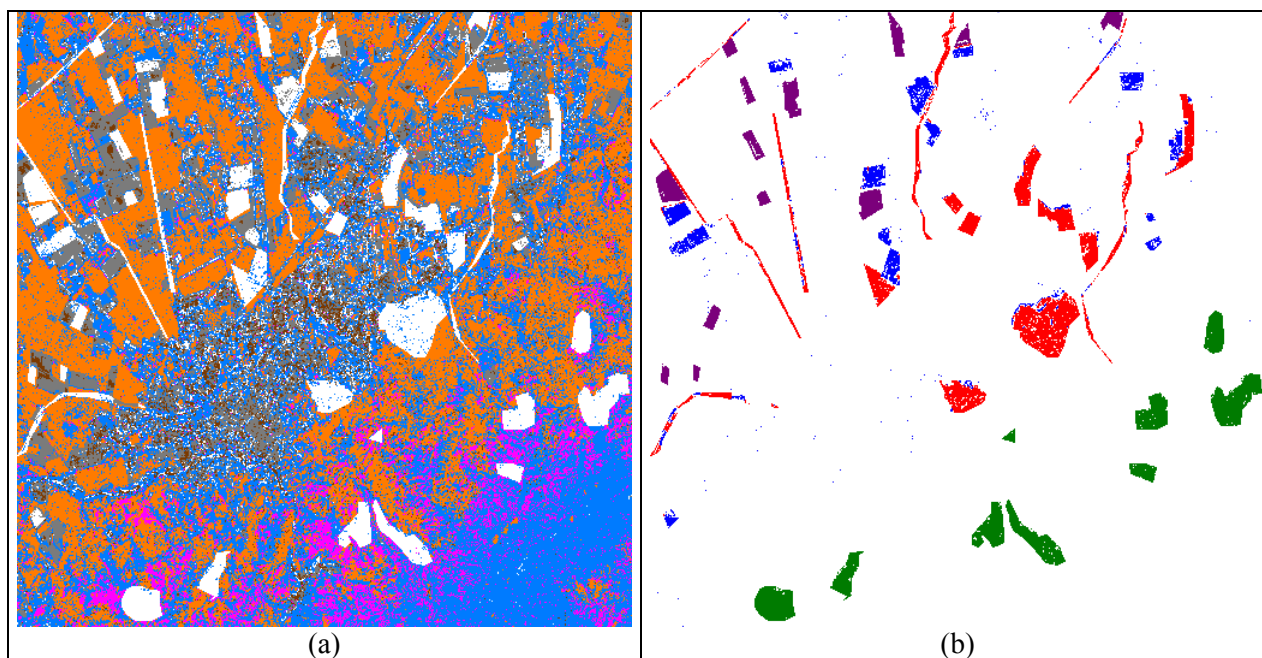


Figure D.11. Chapitre 3 : Image obtenue par le processus flou crispé utilisant la distance de Mahalanobis (a: classes de non changement, b : classes de changement)

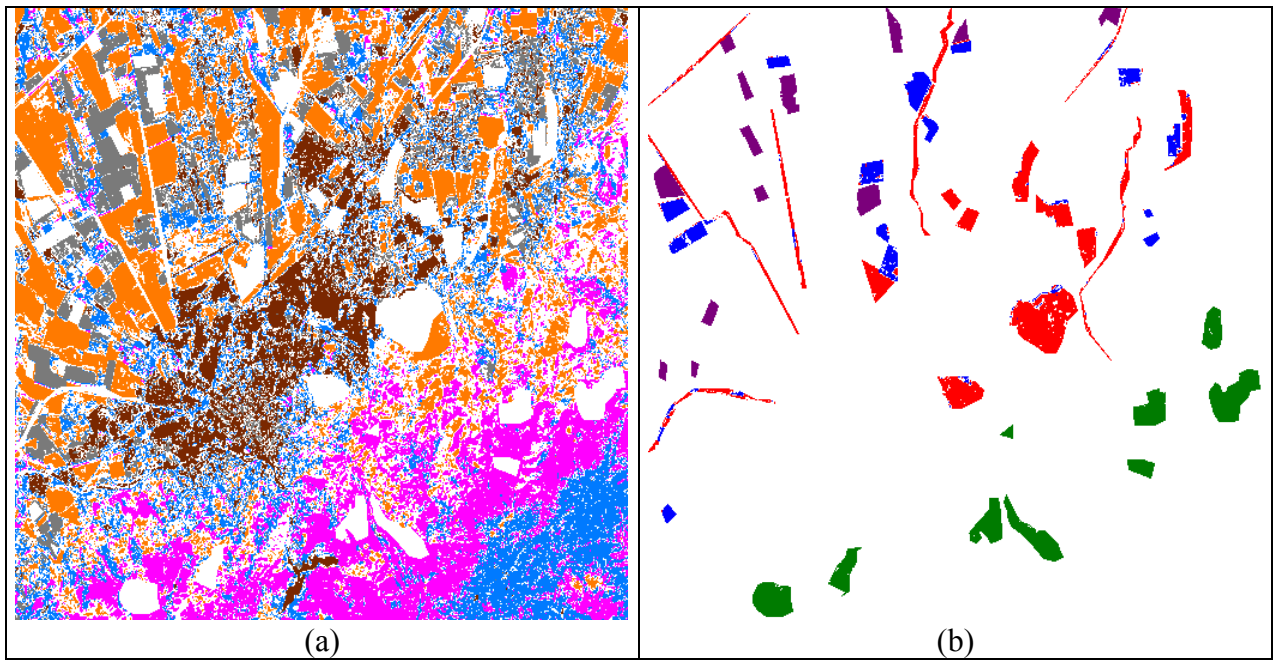


Figure D.12. Chapitre 3 : Image obtenue par le modèle neuroflou
(a : classes de non changement, b : classes de changement)

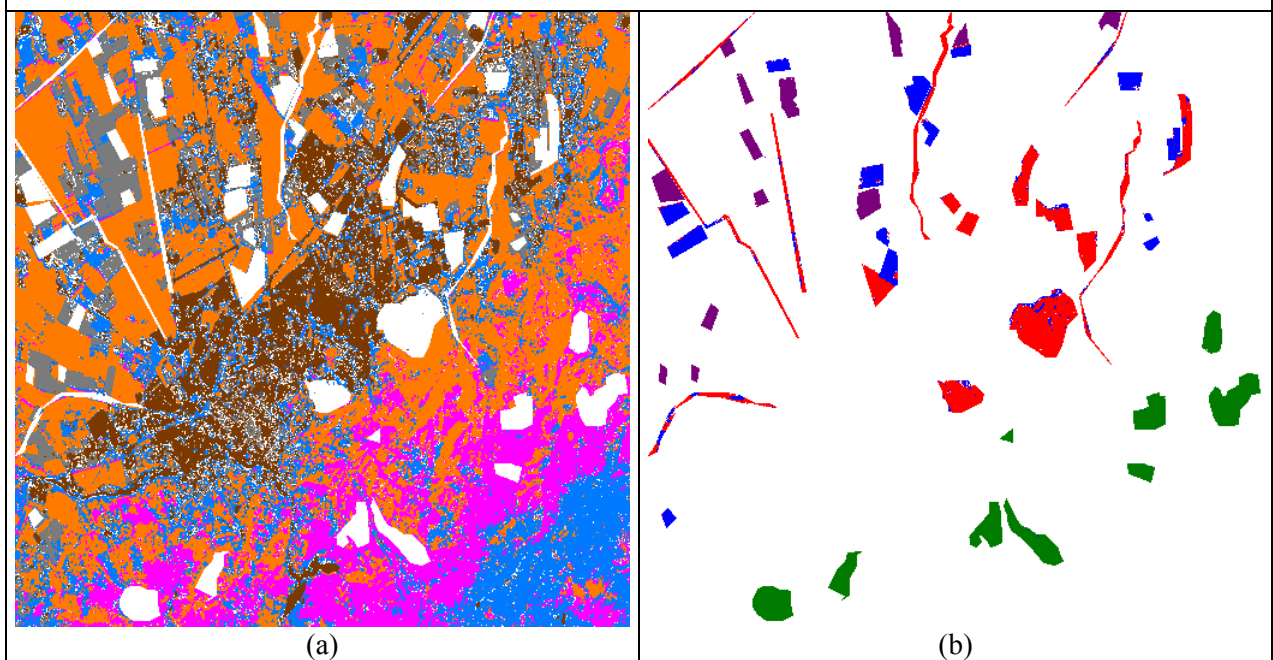


Figure D.13. Chapitre 4 : Image obtenue par la combinaison de trois réseaux de neurones entraînés par l'algorithme FK et comportant respectivement 10, 15, 20 nœuds dans la couche cachée (a : classes de non changement, b : classes de changement)

Annexe E

Statistiques du Chapitre 3

E.1 Statistiques de l'approche AS

E.1. Matrice inverse de covariance de la classe *eau* $\rightarrow$ *eau* :

	$XS_1(1989)$	$XS_2(1989)$	$XS_3(1989)$	$XS_1(1991)$	$XS_2(1991)$	$XS_3(1991)$
$XS_1(1989)$	0.010	-0.018	-0.009	-0.001	0.005	-0.003
$XS_2(1989)$	-0.018	0.063	-0.057	0.005	-0.012	0.004
$XS_3(1989)$	-0.009	-0.057	0.738	0.012	0.017	-0.097
$XS_1(1991)$	-0.001	0.005	0.012	0.006	-0.013	0.004
$XS_2(1991)$	0.005	-0.012	0.017	-0.013	0.043	-0.032
$XS_3(1991)$	-0.003	0.004	-0.097	-0.004	-0.032	0.044

E.2. Matrice inverse de covariance de la classe *construction* $\rightarrow$ *construction* :

	$XS_1(1989)$	$XS_2(1989)$	$XS_3(1989)$	$XS_1(1991)$	$XS_2(1991)$	$XS_3(1991)$
$XS_1(1989)$	0.0150	-0.0173	0.0017	-0.0028	0.0034	-0.0002
$XS_2(1989)$	-0.0170	0.0220	-0.0033	0.0027	-0.0056	0.0014
$XS_3(1989)$	0.0010	-0.0033	0.0043	-0.0001	0.0010	-0.0031
$XS_1(1991)$	-0.0020	0.0027	-0.0001	0.0071	-0.0068	0.0012
$XS_2(1991)$	0.0030	-0.0056	0.0010	-0.0068	0.0114	-0.0032
$XS_3(1991)$	-0.0002	0.0014	-0.0031	0.0012	-0.0032	0.0064

E.3. Matrice inverse de covariance de la classe *végétation* $\rightarrow$ *végétation* :

	$XS_1(1989)$	$XS_2(1989)$	$XS_3(1989)$	$XS_1(1991)$	$XS_2(1991)$	$XS_3(1991)$
$XS_1(1989)$	0.0277	-0.0288	-0.0032	-0.0016	0.0006	-0.0009
$XS_2(1989)$	-0.0288	0.0347	0.0038	0.0026	-0.0035	0.0014
$XS_3(1989)$	-0.0032	0.0038	0.0033	-0.0006	0.0001	-0.0019
$XS_1(1991)$	-0.0016	0.0026	-0.0006	0.0064	-0.0049	0.0012
$XS_2(1991)$	0.0006	-0.0035	0.0001	-0.0049	0.0067	-0.0008
$XS_3(1991)$	-0.0009	0.0014	-0.0019	0.0012	-0.0008	0.0029

E.4. Matrice inverse de covariance de la classe *sol nu* $\rightarrow$ *sol nu* :

	$XS_1(1989)$	$XS_2(1989)$	$XS_3(1989)$	$XS_1(1991)$	$XS_2(1991)$	$XS_3(1991)$
$XS_1(1989)$	0.0054	-0.0053	-0.0015	-0.0026	0.0031	0.0007
$XS_2(1989)$	-0.0053	0.0061	0.0007	0.0028	-0.0037	-0.0007
$XS_3(1989)$	-0.0015	0.0007	0.0033	0.0011	-0.0012	-0.0005
$XS_1(1991)$	-0.0026	0.0028	0.0011	0.0038	-0.0044	-0.0005
$XS_2(1991)$	0.0031	-0.0037	-0.0012	-0.0044	0.0065	-0.0011
$XS_3(1991)$	0.0007	-0.0007	-0.0005	-0.0005	-0.0011	0.0029

E.5. Matrice inverse de covariance de la classe *végétation* $\rightarrow$ *sol nu* :

	$XS_1(1989)$	$XS_2(1989)$	$XS_3(1989)$	$XS_1(1991)$	$XS_2(1991)$	$XS_3(1991)$
$XS_1(1989)$	0.0090	-0.0100	-0.0017	-0.0029	0.0036	0.0015
$XS_2(1989)$	-0.0100	0.0110	0.0015	0.0028	-0.0038	-0.0017
$XS_3(1989)$	-0.0017	0.0015	0.0030	0.0012	-0.0017	-0.0009
$XS_1(1991)$	-0.0029	0.0028	0.0012	0.0033	-0.0031	0.0003
$XS_2(1991)$	0.0035	-0.0038	-0.0017	-0.0031	0.0039	-0.0007
$XS_3(1991)$	0.0015	-0.0017	-0.0009	0.0003	-0.0007	0.0012

E.6. Matrice inverse de covariance de la classe *construction* $\rightarrow$ *sol nu* :

	$XS_1(1989)$	$XS_2(1989)$	$XS_3(1989)$	$XS_1(1991)$	$XS_2(1991)$	$XS_3(1991)$
$XS_1(1989)$	0.0330	-0.0340	-0.0065	-0.0005	0.0005	-0.0008
$XS_2(1989)$	-0.0330	0.0360	0.0077	0.0005	-0.0001	0.0012
$XS_3(1989)$	-0.0053	0.0065	0.0028	-0.00004	-0.0001	0.0011
$XS_1(1991)$	-0.0089	0.0100	0.0044	0.0022	-0.0026	0.0018
$XS_2(1991)$	0.0097	-0.011	-0.0051	-0.0024	0.0039	-0.0027
$XS_3(1991)$	-0.0003	0.0008	0.0011	0.00007	-0.0007	0.0038

E.7. Matrice inverse de covariance de la classe *eau* $\rightarrow$ *sol nu* :

	$XS_1(1989)$	$XS_2(1989)$	$XS_3(1989)$	$XS_1(1991)$	$XS_2(1991)$	$XS_3(1991)$
$XS_1(1989)$	0.0395	-0.0430	0.0044	0.0008	0.0074	0.0088
$XS_2(1989)$	-0.0438	0.0580	-0.0013	0.0008	-0.0100	-0.011
$XS_3(1989)$	0.0044	-0.0013	0.0027	0.0017	-0.0011	0.0006
$XS_1(1991)$	0.0008	0.0008	0.0017	0.0068	-0.0062	0.0016
$XS_2(1991)$	0.0074	-0.0100	-0.0011	-0.0062	0.0089	0.0013
$XS_3(1991)$	0.0088	-0.0110	0.0006	0.0016	0.0013	0.0042

E.8. Matrice inverse de covariance de la classe *X* $\rightarrow$ *nuages* :

	$XS_1(1989)$	$XS_2(1989)$	$XS_3(1989)$	$XS_1(1991)$	$XS_2(1991)$	$XS_3(1991)$
$XS_1(1989)$	0.0258	-0.0270	-0.0059	0.0003	0.0009	0.0005
$XS_2(1989)$	-0.0276	0.0300	0.0071	0.0001	-0.0011	-0.0016
$XS_3(1989)$	-0.0059	0.0070	0.0043	0.0009	-0.0001	-0.0044
$XS_1(1991)$	0.0003	0.0015	0.0009	0.0150	-0.009	-0.0022
$XS_2(1991)$	0.0009	-0.0011	-0.0001	-0.0090	0.0110	-0.0033
$XS_3(1991)$	0.0005	-0.0016	-0.0044	-0.0022	-0.0033	0.0160

E.9. Vecteurs moyennes :

	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	Classe 7	Classe 8
$XS_1(1989)$	28.95	236.14	56.61	162.51	216.88	70.83	128.35	109.53
$XS_2(1989)$	11.87	236.34	61.90	186.13	216.59	76.65	123.53	115.73
$XS_3(1989)$	5.35	151.01	195.31	155.90	147.17	205.96	79.83	198.18
$XS_1(1991)$	102.37	241.37	38.26	163.88	141.04	126.87	136.69	252.22
$XS_2(1991)$	32.64	238.22	73.28	202.94	138.69	181.67	160.16	245.84
$XS_3(1991)$	19.36	162.80	223.26	201.76	150.16	199.54	181.92	247.68

E.2 Statistiques de l'approche AC

- Données de la scène acquise en 1989 :

E.10. Matrice inverse de covariance de la classe *eau* :

	XS_1	XS_2	XS_3
XS_1	0.0700	-0.0083	0.0016
XS_2	-0.0083	0.1080	-0.0032
XS_3	0.0016	-0.0032	0.0028

E.11. Matrice inverse de covariance de la classe *construction* :

	XS_1	XS_2	XS_3
XS_1	0.0100	-0.0130	0.0030
XS_2	-0.0130	0.0170	-0.0045
XS_3	0.0030	-0.0045	0.0034

E.12. Matrice inverse de covariance de la classe *végétation* :

	XS_1	XS_2	XS_3
XS_1	0.0290	-0.0280	-0.0033
XS_2	-0.0280	0.0300	0.0032
XS_3	-0.0033	0.0032	0.0012

E.13. Matrice inverse de covariance de la classe *sol nu* :

	XS_1	XS_2	XS_3
XS_1	0.0034	-0.0032	-0.0001
XS_2	-0.0032	0.0036	-0.0002
XS_3	-0.0002	-0.0002	0.0017

E.14. Vecteurs moyennes :

	Classe 1	Classe 2	Classe 3	Classe 4
XS_1	37.04	228.98	64.49	155.78
XS_2	29.01	228.93	66.67	175.96
XS_3	15.52	143.98	226.18	151.94

- Données de la scène acquise en 1991 :

E.15. Matrice inverse de covariance de la classe *eau* :

	XS_1	XS_2	XS_3
XS_1	0.0043	-0.0095	0.0057
XS_2	-0.0095	0.0280	-0.0200
XS_3	0.0057	-0.0200	0.0170

E.16. Matrice inverse de covariance de la classe *construction* :

	XS_1	XS_2	XS_3
XS_1	0.0068	-0.0072	0.0015
XS_2	-0.0072	0.0121	-0.0028
XS_3	0.0015	-0.0028	0.0032

E.17. Matrice inverse de covariance de la classe *végétation* :

	XS_1	XS_2	XS_3
XS_1	0.0038	-0.0031	-0.0013
XS_2	-0.0031	0.0410	0.0006
XS_3	-0.0013	0.0006	0.0053

E.18. Matrice inverse de covariance de la classe *sol nu*:

	XS_1	XS_2	XS_3
XS_1	0.0024	-0.0029	0.0020
XS_2	-0.0029	0.0050	-0.0034
XS_3	0.0020	-0.0034	0.0035

E.19. Matrice inverse de covariance de la classe *nuages*:

	XS_1	XS_2	XS_3
XS_1	0.0120	-0.0075	-0.0026
XS_2	-0.0075	0.0080	0.0017
XS_3	-0.0026	0.0017	0.0120

E.20. Vecteurs moyennes :

	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5
XS_1	108.09	242.26	41.06	156.62	250.37
XS_2	37.49	240.65	80.54	194.65	243.61
XS_3	23.13	160.54	235.56	184.52	249.46

E.3 Variations du degré d'appartenance en fonction de la valeur du paramètre de fuzzification

- La classe *eau* $\rightarrow$ *eau* :

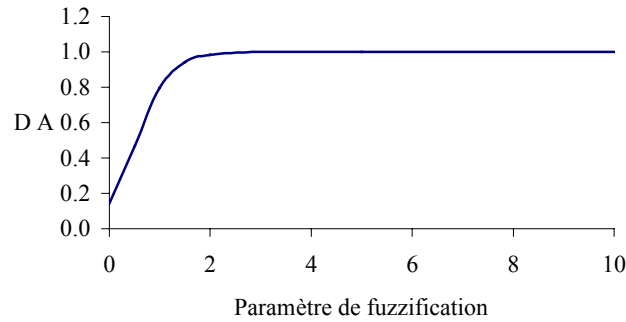


Figure E.1. Evolution du degré d'appartenance en fonction du paramètre de fuzzification dans la classe *eau* $\rightarrow$ *eau*

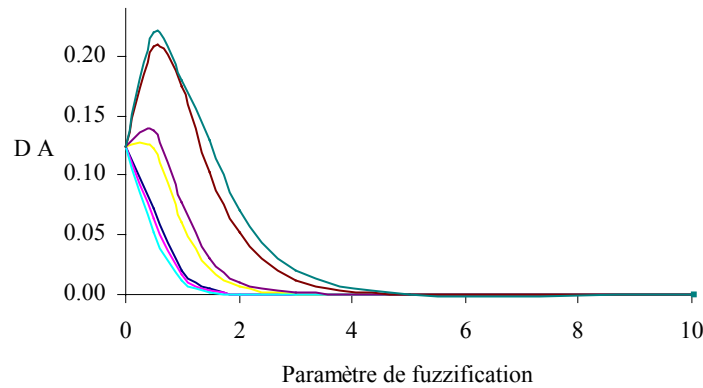


Figure E.2. Evolution du degré d'appartenance en fonction du paramètre de fuzzification dans les autres classes

- La classe *construction* $\rightarrow$ *construction* :

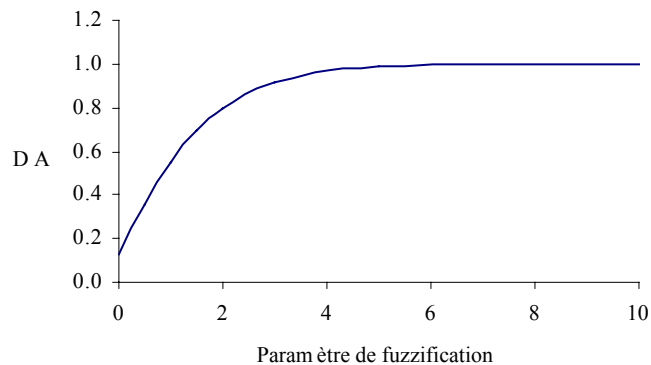


Figure E.3. Evolution du degré d'appartenance en fonction du paramètre de fuzzification dans la classe *construction* $\rightarrow$ *construction*

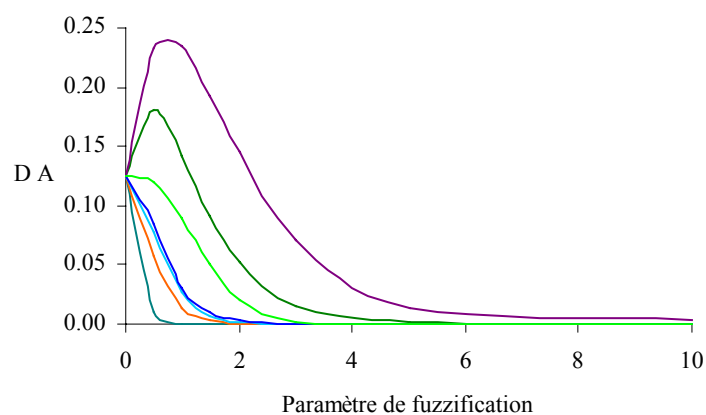


Figure E.4. Evolution du degré d'appartenance en fonction du paramètre de fuzzification dans les autres classes

- La classe *végétation* $\rightarrow$ *végétation* :

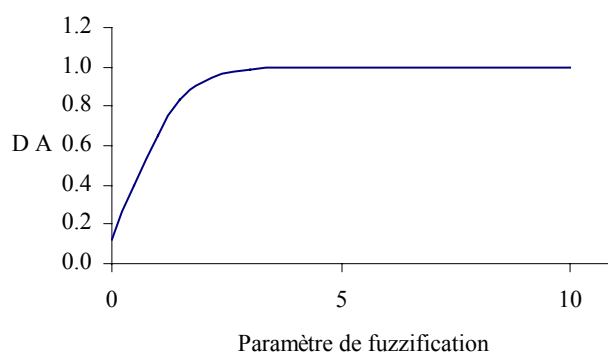


Figure E.5. Evolution du degré d'appartenance en fonction du paramètre de fuzzification dans la classe *végétation* $\rightarrow$ *végétation*

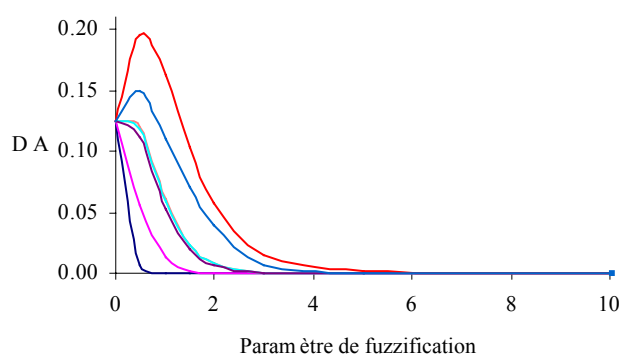


Figure E.6. Evolution du degré d'appartenance en fonction du paramètre de fuzzification dans les autres classes

- La classe $sol\ nu \rightarrow sol\ nu$:

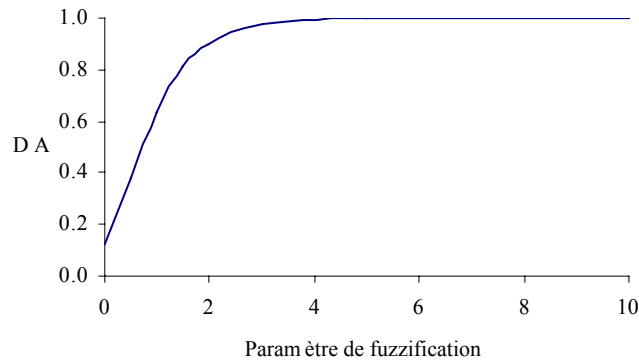


Figure E.7. Evolution du degré d'appartenance en fonction du paramètre de fuzzification dans la classe $sol\ nu \rightarrow sol\ nu$

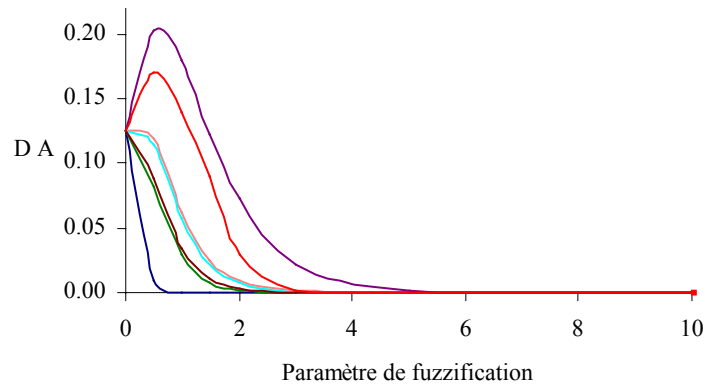


Figure E.8. Evolution du degré d'appartenance en fonction du paramètre de fuzzification dans les autres classes

- La classe $X \rightarrow nuages$:

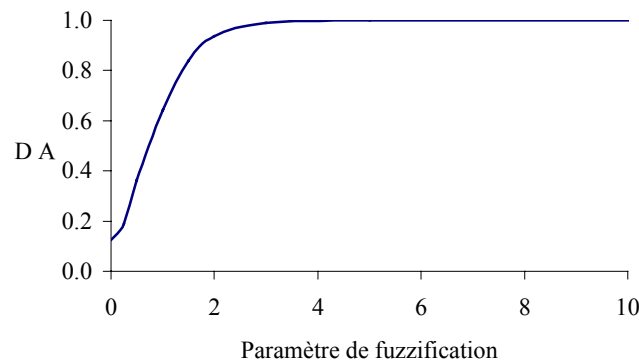


Figure E.9. Evolution du degré d'appartenance en fonction du paramètre de fuzzification dans la classe $X \rightarrow nuages$

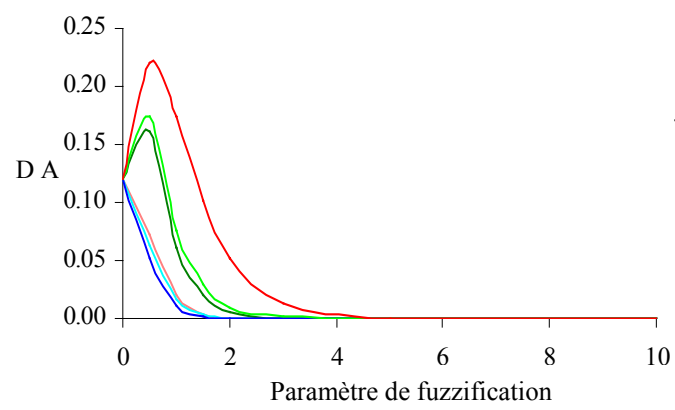


Figure E.10. Evolution du degré d'appartenance en fonction du paramètre de fuzzification dans les autres classes

Références Bibliographiques

- Almutairi, A., 2000, "Monitoring land cover change detection in an arid urban environment: a comparison of change detection techniques", *thesis of Master of arts, university of West Virginia*, USA, 88 pages.
- Autret, Y., et Thepaut, A., 1997, "Neural Network Cooperation for Handwritten Digit Recognition : A Comparison of four Methods", *ESANN'1997 proceedings - European Symposium on Artificial Neural Networks Bruges* 16-18 April 1997, Belgium, D-Facto public., ISBN 2-9600049-7-3, pp. 7-12.
- Banko, G., 1998, "A Review of Assessing the Accuracy of Classifications of Remotely Sensed Data and of Methods Including Remote Sensing Data in Forest Inventory", *report, International Institute for Applied Systems Analysis*, A-2361 Luxembourg, Austria.
- Bárdossy, A., et Samaniego, L., 2002, "Fuzzy Rule-Based Classification of Remotely Sensed Imagery", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 40, No. 2, pp. 362-374.
- Bromiley, P. A., Courtney, P., et Thaker, N. A., 2001, "A case study in the use of ROC curves for algorithm design", *Imaging Science and Biomedical Engineering, Stopford Building, report, University of Manchester, Oxford Road, Manchester, M13 9PT*, U. K.
- Bruzzone, L., et Serpico, S. B., 1997, "Detection of changes in remotely-sensed images by the selective use of multi-temporal information", *International journal of remote sensing*, Vol 18, No. 18, pp. 3883-3888.
- Byeungwoo, J., et Landgrebe, D. A., 1992, "Classification with Spatio-Temporal Interpixel Class Dependency Contexts", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 30, No. 4, pp. 663-672.
- Cacdac, J., 1998, "Application of Change Detection Algorithms for Mine Environment Monitoring", *report, GIS development*.
- Canty, M., Niemeyer, I., et Klaus, D., 1997, "Possibilities and limits of remote sensing for the verification of International Agreements : Algorithms to detect changes at nuclear plants", *report, Centre de recherche de Juelich*, Juelich, Germany.
- Canty, M., Niemeyer, I., et Klaus, D., 1998, "Change detection using multitemporal satellite images for nuclear safeguards puposes", *report, Centre de recherche de Juelich*, Juelich, Germany.
- Canty, M., et Niemeyer, I., 1999, "Change detection techniques for nuclear safeguards puposes", *quatrième Conférence internationale de télédétection*, 21-24, Juin 1999, Ottawa, Ontario, Canada.
- Canty, M., et Niemeyer, I., 2001, "Knowledge-based analysis of change images by object-oriented post-classification", *report, Centre de recherche de Juelich*, Juelich, Germany.
- Carlotto, M. J., 1997, "Detection and analysis of change in remotely sensed imagery with application to wide area surveillance", *IEEE Transactions on Image Processing*, Vol. 6, No. 1, pp. 189-202.
- Carroll, R. W., 2000, "Detecting Building changes through imagery and automatic feature processing", *report, Hitachi software global technology, Westminster, Colorado, USA*.
- Chan, J. C. W., et Defries, R. S., 2000, "Texture features for land cover change detection at 250M resolution an application of machine learning feature subset selection", *IEEE conference on Geoscience and Remote Sensing Symposium*.

- Chang, S., et Greenberg, S., 2002, "Application of Fuzzy Integration Based Multiple Information Aggregation in Automatic Speech Recognition", *report, University of California, USA*.
- Chauvin, S., 1995, "Evaluation des Théories de la Décision Appliquées à la Fusion de Capteurs en Imagerie Satellitaire", *Thèse de Doctorat, Département Images, Ecole nationale supérieure de Télécommunication, Paris, France*, 222 pages.
- Chibani, Y., 2001, "Méthodes et Algorithmes de Fusion d'Images Satellitaires dédiés à l'Amélioration de l'Interprétation", *Thèse de Doctorat d'état, Université des Sciences et de la Technologie Houari Boumediene, Alger, Algérie*, 165 pages.
- Chibani, Y., et Nemmour, H., 2003, "Kalman Filtering as a Multilayer Perceptron Training Algorithm for Detecting Changes in Remotely Sensed Imagery", *IEEE International Geoscience and Remote Sensing Symposium*, July 21-25, Toulouse, France. pp. 4101-4103.
- Chitroub, S., 2003, "Modélisation, Analyse, Classification et Fusion d'Images de Télédétection Multisources Optiques et Radar", *Thèse de Doctorat d'état, Université des Sciences et de la Technologie Houari Boumediene, Alger, Algérie*, 237 pages.
- Cho, S. B., et Kim, J. H., 1995, "Combining Multiple Neural Networks by Fuzzy Integrals for Robust Classification", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, No. 2, pp. 380-384.
- Cho, S. B., 1995, "Fuzzy Aggregation of Modular Neural Networks With Ordered Weighted Averaging Operators", *International journal of approximate reasoning*, Vol.13, pp. 359-375.
- Cho, S. H., 2000, "Digital change detection by Post-Classification Comparison of Multitemporal Remotely Sensed Data", *Journal of the Korean Society of Remote Sensing*, Vol.16, No. 4, pp. 367-373.
- Cho, S. B., 2002, "Fusion of Neural Networks with Fuzzy Logic and Genetic Algorithm", *Integrated Computer-Aided Engineering, IOS Press*, Vol. 9. pp. 363-372.
- Civco, D.L., Hurd, J.D., et Prisloe, S., 2000, "Characterization of suburban sprawl and forest fragmentation through remote sensing applications", *report, University of Connecticut, USA*.
- Civco, D.L., Hurd, J.D., et Wilson, E. H., 2002, "A comparison of land use and land cover change detection methods", *Asprs-acsm annual conference and fig xxii congress*, Avril 22-26, 2002.
- Clifton, C., 2001, "Change detection in overhead imagery using neural networks", *submitted to the international journal of applied intelligence*, 2001.
- Cohen, W. B., Kushla, J. D., et Ripple, W. J., 1996, "An introduction to digital methods in remote sensing of forested ecosystems: Focus on the pacific northwest, USA", *Environmental management*, Vol. 20, No. 3, pp. 421-435.
- Congalton, R. G., 2001, "Accuracy assessment and validation of remotely sensed and other spatial information", *International Journal of Wildland Fire*, Vol.10, pp. 321-328.
- Dai, X., et Khorram, S., 1997, "Development of a New Automated Land Cover Change Detection System from Remotely Sensed Imagery Based on Artificial Neural Networks", *IEEE Conference*, pp. 1029-1031.
- Deer, P., et Eklund, P. W., 1995, "On the Fusion of Image Features", *report, Department of computer science, Adelaide University, Australia*.
- Deer, P., 1998, "Digital Change Detection in Remotely Sensed Imagery Using Fuzzy Set Theory", *PHD thesis, Department of Geography and Department of Computer Science, University of Adelaide, Australia*, 185 pages.
- Deer, P., et Eklund, P., 2000, "Fuzzy logic for change detection in classified images", *report, Université de Griffith, Australia*.

- Dietterich, T. G., 2001, "Methods for Cost Sensitive Learning", *PHD thesis in computer science, Oregon state University, USA*, 102 pages.
- Dubois, D., et Prade, H., 1994, a, "La fusion d'informations imprécises", *GRETSI*, Vol. 11, No. 6, pp. 447-458.
- Dubois, D., et Prade, H., 1994, b, "Possibility Theory and Data Fusion in Poorly Informed Environments", *Control eng. Practice*, Vol. 2, No. 5, pp. 811-823.
- Dubois, D., et Prade, H., 1994, c, "Gradual Rules and the Approximation of Control Law", *Theoretical Aspects of fuzzy control edition : John Wiley & Sons, Inc*, 1994.
- Dubois, D., et Prade, H., 1994, d, "Fuzzy Sets and Automated Reasoning: Problems and Methods", *rapport, Institut de recherche en en informatique de Toulouse*, Toulouse, France.
- Fahlman, S. E., et Lebiere, C., 1991, "The cascade-correlation learning algorithm", *report, School of computer science Carnegie Mellon University*, Pittsburgh, 1991.
- Fisher, C. S., et Levien, L. M., 2001, A, "Monitoring California's Hardwood Range lands using remotely sensed data", *report*.
- Fisher, C. S., et Levien, L. M., 2001, B, "Toward operational monitoring of forest cover change in California using multitemporal remote sensing data", *report*.
- Foody, G.M., McCulloch, B., et Yates, W. B., 1995, "Classification of Remotely Sensed Data by an Artificial Neural Network : Issues Related to Training Data Characteristics", *Photogrammetric Engineering & Remote Sensing*, Vol. 61, No. 4, pp. 391-401.
- Fullér, R., 1995, "Neural Fuzzy Systems", *Notes de cours*, 253 pages, Åbo Academy, Sweden.
- Hayes, D.J., et Sader, S.A., 1999, "Change detection techniques for monitoring forest clearing and regrowth in a tropical moist forest", *report, California university*, USA.
- Herauld, J., et Jutton, C., 1994, "Réseaux neuronaux et traitement du signal", *édition Hermès*, Paris, France.
- Ho, T. K., 2001, "Multiple Classifier Combination : Lessons and Next Steps", *To appear in, A. Kindel, H. Brunke, (eds.), Hybrid Methods in Pattern Recognition, World Scientific*.
- Huang, C., et al., 1998, "Land cover change detection with change vector in the red and near-infrared reflectance space", *IEEE conference Geoscience and Remote Sensing Symposium*.
- Huang, C., et Townshend, J. R. G., 2000, "Detecting land cover changes based on their trajectories in the spectral space", *report, Raytheon ITSS, USGS/EROS Data Center, Sioux Falls, SD 57198*.
- Kaufmann, R. K., 1999, "Modeling and Forecasting Effects of Land-Use Change in China Based on Socioeconomic Drivers", *report, Department of Geography, Boston University, USA*.
- Keller, J. M., et Gader, P. D., 2000, "Soft Fusion Algorithms", *report, Computer Engineering and Computer Science Department, University of Missouri- Columbia, USA*.
- Kim, J.H., et Lee, K.S., 2000, "Change analysis of forest area and canopy conditions in Kaesung , north Korea using landsat, Spot and Kompsat data". *Journal of the Korean society of remote sensing*, Vol 16, No. 4, pp. 327-338.
- Kittler, J., Hatef, M., Robert, P. W., et Matas, J., 1998, "On Combining Classifiers", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3, pp. 226-239.
- Lambin, E. F., et Strahler, A. H., 1994, "Change vector analysis in multitemporal space : a tool to detect and categorize land cover change processes using high temporal resolution satellite data", *Remote Sensing. Environ*, Vol. 48, pp. 231-244.
- Levien, L. M., Fisher, C. S., Roffers, P. D., et Maurizi, B. A., 1990, "Statewide change detection using multitemporal remote sensing data", *report, University of California*, USA.
- Lipnickas, A., 2000, "An Empirical Comparison of two Sampling Techniques for Training Neural Network Committees", *report, ISSN 1392 – 124X Informacin Technologijos ir Valdymas*, 2000, Vol. 2(15).

- Lipnickas, A., Malmqvist, K., et Verikas, A., 2000, "Fuzzy Measures in Neural Networks Fusion", *ISSN 1392-124X Informacins Technologijos*, Vol. 2, pp. 7-16.
- Lipnickas, A., 2001, "Classifiers Fusion with Data Dependent Aggregation Schemes", *7 th International Conference on Information Networks, Systems and Technologies*, pp. 147-153.
- Liu, Q. Z., et al., 2001, "Dynamic Image Sequence Analysis Using Fuzzy Measures", *IEEE transactions on systems, man, and cybernetics—Part B: Cybernetics*, Vol. 31, No. 4, pp. 557- 572.
- Mangolini, M., 1994, "Apport de la Fusion d'Images Satellitaires Multicapteurs au Niveau Pixel en Télédétection et Photointerprétation", *Thèse de Doctorat, université de Nice, France*, 174 pages.
- Martinez-Rios, J. J., et Curtis Monger, H., 2002, "Soil Classification in Arid Lands with Thematic Mapper Data", *TERRA*, Vol.20, No. 2, pp. 89-100.
- Murakami, H., et Nakagawa, k., 1999, "Change detection of buildings using an airborne laser scanner", *ISPRS Journal of Photogrammetry & Remote Sensing*, Vol, 54, pp. 148-152.
- Nemmour, H., et Chibani, Y., 2003, "Comparison Between Object and Pixel Level Approaches for Change Detection in Multispectral Images by Using Neural Networks", *SPIE conference*, September 9-12, 2003, Barcelona, Spain, pp.
- Paola, J. D., et Schowengerdt, R. A., 1995, "A Review and Analysis of Backpropagation Neural Networks for Classification of Remotely Sensed Multispectral Imagery", *Remote Sensing*, Vol. 16, No. 16, pp. 3035-3056.
- Richards, J. A., 1986, "*Remote Sensing Digital Image Analysis, an introduction*", édition ISBN, Germany.
- Rignot, E. J. M., et Zyl, J. J. V., 1993, "Change detection techniques for ERS-1 SAR data", *IEEE 1993*, Vol. 31, No. 4, pp. 896-906.
- Rossi, F., 1996, "Calcul de différentielles dans les réseaux de neurones généralisés : algorithmes, complexités, implantation logicielle et applications", *thèse de doctorat, université de Paris IX dauphine, France*, 197 pages.
- Scalero, R. S., et Tepedelenlioglu, N., 1992, "A Fast New Algorithm for Training Feedforward Neural Networks", *IEEE Transactions on signal processing*, Vol. 40, No. 1, pp. 202-210.
- Simpson, J. J., 1998, "The technique MAD", *report, physical oceanography of Oregon State University, USA*.
- Santos, R.V., Vellasco, M.R., et Feitosa, R. Q., 2000, "An Application of Combined Neural Networks to Remotely Sensed Images", *report, Department of electrical and department of computer engineering, catholic university of Rio de Janeiro, Brazil*.
- Sung-Ryong, H., et Byung, W. A., 2001, "Utilization of multi-temporal Kompsat-1 EOC data for extracting agricultural information", *report, Université de Chungbuk, Chungbuk, Koréa*.
- Sung-Ryong, H., et al., 2002, "Change detection of land cover from multi-temporal Kompsat-1 EOC imageries", *Korean journal of remote sensing*, Vol.18, No. 1, pp. 13-23.
- Takagi, H., 1997, "Introduction to Fuzzy Systems, Neural Networks, and Genetic Algorithms", *Intelligent Systemes: Fuzzy Logic, Neural Networks, and Genetic Algorithms, CH.1, D. Ruan,, Kluwer Academic Publishers (Norwell, Massachusetts, USA)*, Spetember, 1997, pp.1-33.
- Touzet, C., 1992, "Les réseaux de neurones artificiels introduction au connexionisme".
- Verikas, A., Lipnikas, A., Malmqvist, K., Bacauskiene, M., Gelzinis, A., 1999, "Soft Combination of Neural Classifiers: A Comparative Study", *Pattern recognition letters, Elsevier Science*, Vol. 2, pp. 429-443.
- Wiemker, R., 1997, "An iterative spectral-spatial bayesian labeling approach for unsupervised robust change detection on remotely sensed multispectral imagery", *Proceedings of the*

- 7th International Conference on Computer Analysis of Images and Patterns, Springer.LNCS 1296 (ISBN 3-540-63460-6), pp. 263-270.*
- Wiemker, R., Kulbach, D., et Speck, A., 1997, "Unsupervised robust change detection on multispectral imagery using spectral and spatial features", *report, Université de Hamburg, Germany.*
- Willhauck, G., 2000, "Comparison of object oriented classification techniques and standard image analysis for the use of change detection between SPOT multispectral satellite images and aerial photos", *ISPRS, Vol. XXXIII, Amsterdam, Holland.*
- Yager, R. R., 1999, "Criteria Aggregations Functions Using Fuzzy Measures and the Choquet Integral", *International Journal of Fuzzy Systems, Vol. 1, No. 2, pp. 99-112.*
- Yager, R. R., 2001, "Characterizing Fuzzy Measures Used in Uncertainty Representation", *2nd International Symposium on Imprecise Probabilities and Their Applications, Ithaca, New York, 2001.*
- Yang, X, M., 1999, "Change Detection Based on Remote Sensing Information Model and its Application on Coastal Line of Yellow River Delta", *report, GIS development, 1999.*