RAPPORT TECHNIQUE

Implémentation d'un système multi-étages pour la classification de coraux

Présenté à :

Robert Sabourin

Pour :

SYS843 – Réseaux de neurones et systèmes flous

Par :

Jonathan Bouchard

Université du Québec
École de technologie supérieure

Montréal, le 3 avril 2009

Enregistrement 2, 2009-04-03 11:04:55

TABLE DES MATIÈRES

LISTE DES TABLEAUX ET FIGURES

INTRODUCTION

Au fil des dernières années, la protection de l'environnement et des écosystèmes naturels est devenue une préoccupation grandissante pour l'humanité. Différentes études sont en cours aux quatre coins du monde pour tenter d'évaluer l'impact de l'activité humaine sur l'équilibre des écosystèmes. L'évolution de la biodiversité est un des sujets d'étude permettant de constater les conséquences des actes de l'homme sur l'environnement. Dans cette optique, une équipe de chercheurs de Manille, aux Philippines, travaillent en collaboration avec M. Jacques-André Landry en mettant à jour chaque année une vaste base de données de photographies sous-marines de récifs coralliens.

Le corail est une forme de vie très sensible aux perturbations humaines, il est un bon indicateur de la présence de polluants, des changements climatiques et par conséquent de la santé de la planète. Plusieurs études montrent que l'intégrité écologique des récifs coralliens est actuellement menacée. Le projet d'étude présenté au cours des prochaines pages vise l'amélioration d'un outil d'analyse facilitant la tâche des biologistes dans le suivi de l'intégrité et de la biodiversité des bancs de coraux.

Problématique

Après plus de dix années de capture d'images, la gestion et l'analyse des données photographiques des récifs est devenue très fastidieuse. Pour faciliter la tâche des chercheurs, le développement d'un ensemble de méthodes automatisées capables de classifier et de comptabiliser les différentes espèces de coraux présentes dans un espace donné est envisagé. Ainsi, l'équipe de chercheurs possèdera un outil apte à traiter objectivement les images et à en extraire les statistiques utiles à l'étude de la biodiversité. Cette classification est réalisée avec diverses méthodes de reconnaissance de formes dans des images numériques en mettant en application la théorie sur la vision par ordinateur, les algorithmes évolutifs et l'intelligence artificielle. Des experts en la matière ont fourni une base de données d'images où les différentes espèces de coraux sont déjà identifiées. Cette base est utilisée pour entraîner les classificateurs qui permettront par la suite de traiter en masse toutes les autres images.

Toutefois, l'implémentation actuelle du système ne permet pas de résoudre le problème. En effet, les taux d'erreurs du système en généralisation atteignent jusqu'à 50% sur les meilleurs classificateurs utilisés jusqu'ici. (Bouchard, 2008a; 2008b) Cette grande difficulté qu'ont les classificateurs à séparer les classes est attribuable à la qualité de la base de données et des caractéristiques utilisées. La quantité de données disponibles pour chacune des espèces varie grandement. Pour près de la moitié des classes à prendre en considération moins de cent échantillons sont disponibles. De plus, la base de données contient neuf classes qui ne sont pas d'intérêt pour l'étude la biodiversité. Tous ces phénomènes complexifient énormément le problème et expliquent les performances médiocres atteintes jusqu'ici.

Objectifs

Le présent projet d'étude vise à palier aux différents problèmes cités précédemment avec l'implantation d'un mécanisme de détection et de rejet au sein du système. Le mécanisme de détection est le premier étage du système. Il permet d'éliminer les images qui ne sont pas des coraux. Ensuite, le rejet permet de réduire l'ensemble des données traitées en envoyant à la classification manuelle les spécimens qui ne présentent pas un grand taux de confiance sur la décision. Voici la liste des différents objectifs poursuivis :

- Effectuer une analyse statistique sur les données;
- Implanter un système de détection :
    - Sélectionner les classes d'images pertinentes;
    - Entraîner un détecteur à séparer les coraux des non-coraux;
- Implanter un mécanisme de rejet :
    - Optimiser le ou les seuils de rejet selon :
        - La méthode proposée par Chow;
        - La méthode proposée par Fumera;
- Assembler le système
    - Calculer les taux d'erreurs et de rejets optimaux;
    - Comparer les solutions.

Au cours des prochaines pages, les différentes informations seront présentées au cours de quatre sections : le sommaire des approches, la méthodologie expérimentale, les résultats de simulation et la conclusion.

SOMMAIRE DES APPROCHES

Cette section présente les différentes approches et théories mises à profit dans l'implémentation du système de classification des coraux avec détecteur et mécanisme de rejet. Les différents systèmes mis à l'essai y seront présentées dans un ordre de complexité croissante.

LE SYSTÈME DE BASE

Au cours des tests effectués dans le passé, le perceptron multicouche s'est avéré le meilleur algorithme de classification. Les tests précédents démontrent qu'il est capable de généraliser sur le problème des coraux mieux que tous les autres algorithmes testés, notons le SVM, l'arbre C4.5, le kPPV et le classificateur naïf de Bayes. Le perceptron multicouche sera donc utilisé comme système de référence. L'architecture proposée pour ce système accepte les 43 caractéristiques disponibles en entrée et doit séparer les coraux entre les 32 classes de la base de données. La figure 1 illustre ce système.

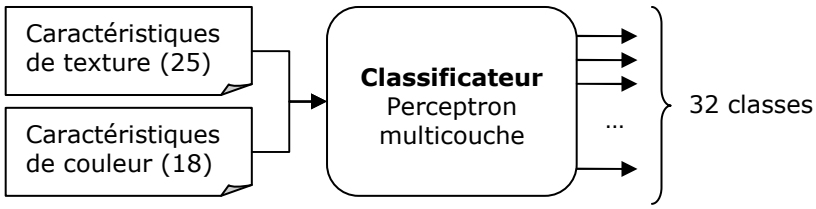

Figure 1 : Le système de base

Les investigations précédentes ont démontré que le système présenté plus haut atteint une performance de classification en test de 41,6%. Le tout lors d'un entraînement sur 66% des données et d'un test sur 33% des données. (Avec séparation stratifiée, où chaque classe possède une probabilité à priori similaire dans l'une ou l'autre des bases de données.) (Bouchard, 2008a)

Lors de la présente expérimentation tous perceptron multicouche seront entraîné avec les mêmes méta-paramètres que pour les expérimentations précédentes. Le taux d'apprentissage par rétro propagation de l'erreur a été fixé à 0,3. Cette valeur est celle spécifiée par défaut par l'environnement Weka (Witten et Frank, 2005). Le momentum est fixé à 0,2, encore une fois il s'agit d'une valeur par défaut. De plus, l'architecture du réseau sera configurée selon la règle euristique

bien connue dans le domaine des perceptrons multicouches : la couche cachée contiendra *n* neurones où *n* est la somme du nombre d'entrées et du nombre de sorties divisée par deux. Ces paramètres ne seront pas optimisés puisque le but de l'expérimentation est d'améliorer les performances via un mécanisme de rejet en se comparant aux expérimentations précédentes. Un total représentant 15% des données a été utilisé pour valider l'arrêt de l'apprentissage. Enfin, cet arrêt survient lorsque 20 itérations successives entraînent une augmentation de l'erreur globale de classification.

LE SYSTÈME DÉTECTEUR

Comme mentionné en introduction, plusieurs classes de la base de données des coraux ne sont pas d'intérêt pour l'étude de la biodiversité. Toutefois, dans un contexte d'échantillonnage réel, les images appartenant à ces classes doivent être traitées par le système. Présentement neuf classes décrivent ces données non pertinentes : gravier, sable, sédiments, eau, aucune données, indéterminé, animal, inconnu et une classe qui est simplement identifiée par l'acronyme GO.

Puisque l'on ne s'intéresse pas aux statistiques de ces classes, elles peuvent tout simplement être regroupées. La classe résultante de ce regroupement sera nommée « non-corail ». Toutes les autres classes, celles qui sont pertinentes à l'analyse, seront regroupées sous l'étiquette « corail » et seront traitées en détail dans une deuxième étape du processus. Le classificateur résultant qui implémente le système détecteur possède donc deux classes de sorties. C'est une extrême simplification du problème qui entraîne des taux de classifications beaucoup plus encourageants. En effet, les premiers tests démontrent que cet étage seul permet de classifier correctement 78,35% des données. La figure 2 montre l'architecture de ce sous système.

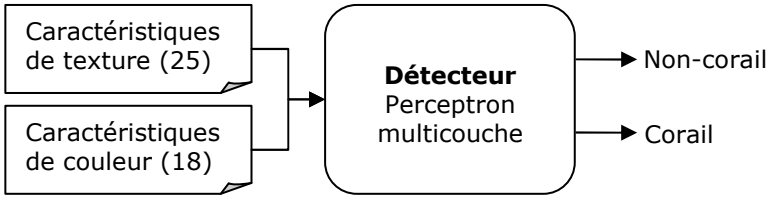

Figure 2 : Le système de détection

Le perceptron multicouche du détecteur est entraîné en utilisant les mêmes paramètres présentés précédemment.

LE SYSTÈME DÉTECTEUR AVEC MÉCANISME DE REJET

Une deuxième variante du système de détection a été mise à l'essai. Cette version du système inclut un mécanisme de rejet sur la décision. Les mécanismes de rejets sont très simples. Les classificateurs tels le perceptron multicouche émettent un niveau de confiance sur la décision. À partir de cette information, il est possible de rejeter vers la classification manuelle les spécimens qui ne sont pas classifiés avec un taux de confiance suffisant. Ce système est illustré par la figure suivante :

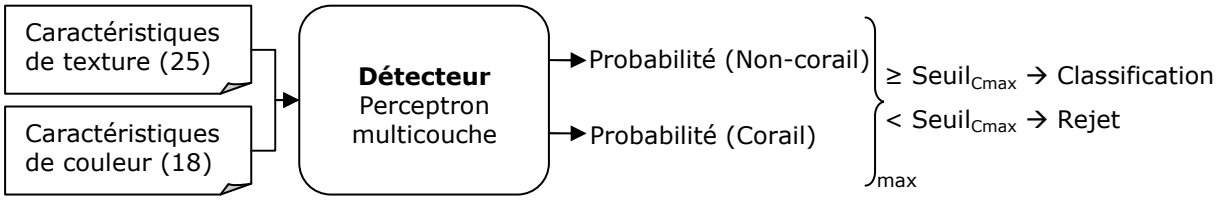

Figure 3 : Le système de détection avec rejet

Toutefois, la performance du système devient intimement liée aux choix des seuils. Deux méthodes sont décrites par la littérature pour faire la sélection des seuils. Il est possible d'appliquer un seuil de rejet unique, identique pour toutes les classes du problème ou d'utiliser un seuil de décision différent pour chacune des classes considérées.

La première méthode, celle utilisant un seuil unique est proposée par (Chow, 1970). Les écrits de cet auteur démontrent qu'il est possible d'atteindre un compromis idéal entre le taux de rejet et l'erreur en utilisant un seuil de décision unique pour le rejet. Cette méthode a toutefois été outrepassée par les expérimentations de (Fumera, Roli et Giacinto, 2000) qui démontrent qu'il est possible de créer un système encore plus performant. Pour un même taux de rejet, il est possible de diminuer d'avantage l'erreur de classification en optimisant un ensemble de seuils différents appliqués pour chacune des classes. Ce phénomène est expliqué par la figure suivante :

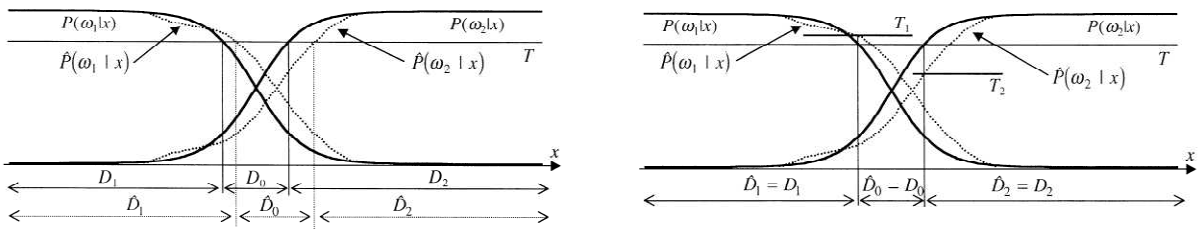

Figure 4 : Les seuils multiples de Fumera (Fumera, Roli et Giacinto, 2000)

La figure 4 présente par son graphe de gauche un rejet basé sur un seuil unique T. Les lignes pleines représentent la distribution des deux classes tandis que les lignes pointillées la distribution réelle des classes. On remarque que le déphasage entre l'estimation et la réalité peut favoriser l'acceptation des individus d'une classe au détriment de l'autre. Dans d'autres mots, la position de la borne de décision établie à l'apprentissage pourrait se retrouver plus près de l'une des deux bornes de rejets lorsque le ce mécanisme est appliqué en généralisation. Ce qui entraîne une probabilité de rejet non uniforme pour toutes les classes. Afin d'uniformiser cette probabilité de rejet, un seuil différent pour chacune des classes (T_1, T_2) peut être optimisé. C'est ce qui est représenté par la partie droite de la figure 4. Ainsi, la zone de rejet est bien centrée sur la borne de décision et l'optimalité du compromis erreur-rejet peut être atteint. Tout réside donc dans la méthode d'optimisation de ces seuils, elle sera présentée au cours de la méthodologie expérimentale.

LE SYSTÈME DE CLASSIFICATION AVEC MÉCANISME DE REJET

Le système de classification est dérivé du système de base. En fait il y est en tout points identique si on ne considère pas le mécanisme de rejet. La figure 5 montre le système avec rejet. Il faut toutefois comprendre que le seuil est appliqué uniquement sur la probabilité de sortie la plus forte. Dans le cas de la règle de Chow, le seuil est le même peut importe la classe gagnante. Au contraire, avec Fumera, le seuil relatif à la classe gagnante est appliqué à la probabilité maximale. D'où la notation « $Seuil_{Cmax}$ » utilisée par les figures.

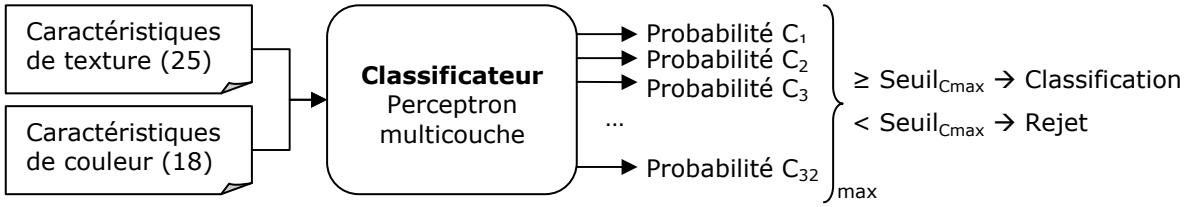

Figure 5 : Le classificateur avec rejet

Note : Le système de classification sans seuillage n'a pas été présenté précédemment puisqu'il est totalement identique au système de base utilisé.

LE SYSTÈME DE CLASSIFICATION DE GROUPE BENTHIQUE

Afin de pousser un peu plus loin l'analyse un deuxième classificateur a été mis à l'essai. Depuis peu, une information supplémentaire est disponible sur les 32 classes de la base de données : elles appartiennent à divers groupe benthique. Les groupes sont au nombre de 5, ce qui réduit évidement la complexité nécessaire pour la séparation des classes. De plus, en considérant le problème de la séparation de groupes, la base de données possède répartition plus uniforme du nombre d'individus par classe que lorsque que l'on considère les 32 classes initiales.

Afin de tester l'approche de détection et de rejet sur un problème plus simpliste, un classificateur perceptron multicouche a aussi été entraîné sur ce problème. L'architecture est présentée par la figure suivante :

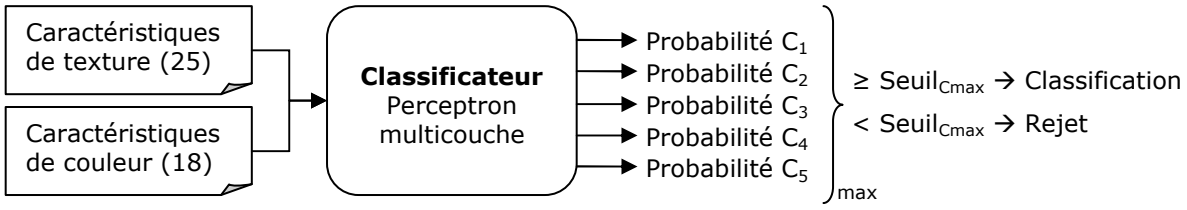

Figure 6 : Le classificateur de groupe avec rejet

LE SYSTÈME COMPLET

Enfin, voici une description du système complet. Lorsque l'optimisation de chacune des parties est terminée, il suffit d'assembler le détecteur et le classificateur afin d'obtenir le système à deux étages. Plusieurs variantes de cet assemblage seront mises à l'essai, soit avec un mécanisme de rejet au premier niveau seulement et aux deux niveaux simultanément. Toutefois, seule la deuxième architecture sera présentée par la figure 7, soit la plus complexe.

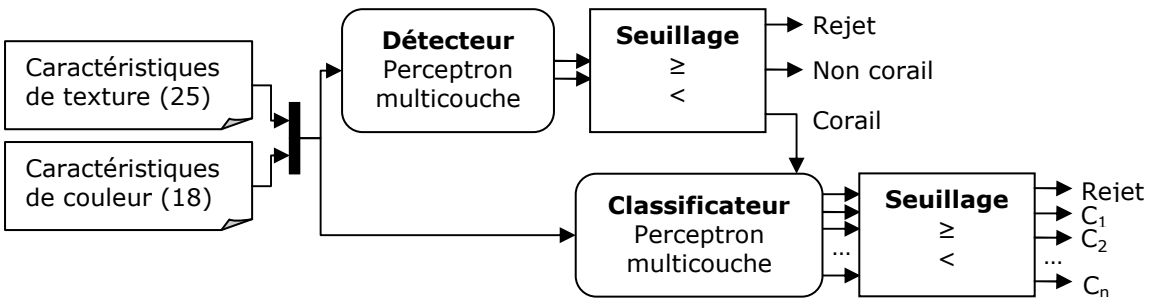

Figure 7 : Le système complet

MÉTHODOLOGIE EXPÉRIMENTALE

Au cours de cette section, la méthodologie expérimentale sera décrite en deux points soit : la base de données et le protocole expérimental.

LA BASE DE DONNÉES

La base de données utilisée comprend 1951 images numériques du récif de Sulu Sulawesi aux Philippines. Ces images possèdent dix zones d'intérêts marquées par des croix rouge. (Voir figure 8) Les marqueurs représentent l'échantillonnage effectué sur les images. Chaque marqueur est donc étiqueté par un expert selon l'une des 32 classes d'objets. L'information de classification est stockée dans un fichier Excel qui rend disponible l'information sur les 19 510 échantillons.

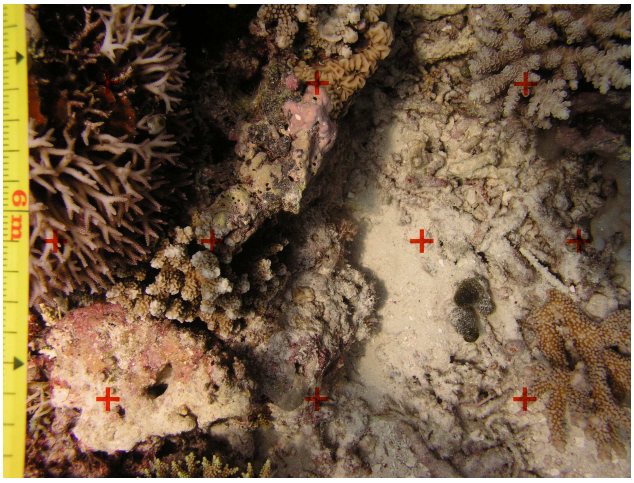

Figure 8 : Exemple d'image

Le nombre d'individus pour chacune des classes est proportionnel à la réalité. Malheureusement cette réalité rend la représentation de classes très hétérogène. Comme dans tout système vivant il y a certaines espèces qui prédominent l'espace. C'est un phénomène très visible dans la base de données.

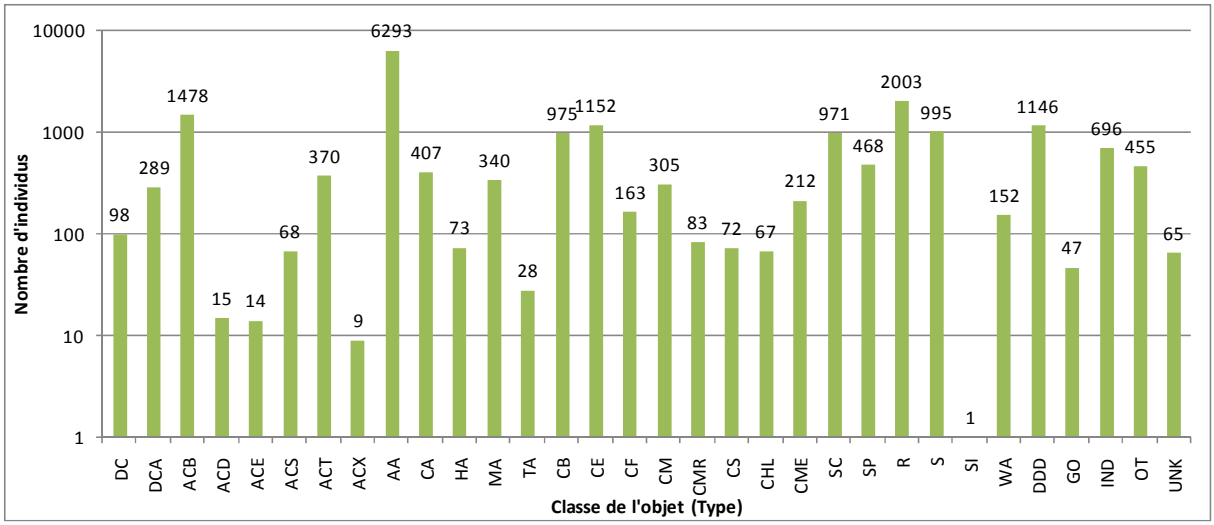

Figure 9 : Répartition des classes

Le graphe de la figure 9 présente la répartition très hétérogène des classes. En effet 13 des 32 classes possèdent moins de 100 individus, ce qui rends l'apprentissage très difficile. Au contraire, la classe AA possède plus de 6000 individus, soit le tiers des données disponibles, ce qui en fait une classe très prédominante. Le tableau suivant dresse un portrait complet de la situation :

Tableau 1 : Répartition des classes

Catégorie	Compte	Groupe benthique	Compte	Type	Description*	Compte	Ratio
Corail	13950	Abiotique	387	DC	Dead coral	98	0,5%
	71,5%		2,0%	DCA	Algae covered coral	289	1,5%
		Acropora	1954	ACB	Branching	1478	7,6%
			10,0%	ACD	Digitate	15	0,1%
				ACE	Encrusting	14	0,1%
				ACS	Submassive	68	0,3%
				ACT	Tabulate	370	1,9%
				ACX	Bottlebrush	9	0,0%
		Algue	7141	AA	Algal Assemblage	6293	32,3%
			36,6%	CA	Coralline Algae	407	2,1%
				HA	Halimeda sp	73	0,4%
				MA	Macro algae	340	1,7%
				TA	Turf Algae	28	0,1%
		Non-acropora	2750	CB	Branching	975	5,0%
			14,1%	CE	Encrusting	1152	5,9%
				CF	Foliaceous	163	0,8%
				CM	Massive	305	1,6%
				CMR	Mushroom Coral	83	0,4%
				CS	Submassive	72	0,4%
		Autres	1718	CHL	Heliopora (blue coral)	67	0,3%
			8,8%	CME	Millepora (fire coral)	212	1,1%
				SC	Soft Coral	971	5,0%
				SP	Sponge	468	2,4%
Non-corail	5560	Abiotique	3151	R	Rubble	2003	10,3%
	28,5%		16,2%	S	Sand	995	5,1%
				SI	Silt	1	0,0%
				WA	Water	152	0,8%
		Autres	2409	DDD	Non-data points	1146	5,9%
			12,3%	GO	(Desc. inconnue)	47	0,2%
				IND	Inderterminate	696	3,6%
				OT	Animal	455	2,3%
				UNK	Unknown	65	0,3%
Total						**19510**	**100,0%**

* La description des classes a volontairement été laissée en anglais pour qu'elle concorde avec l'abbréviation choisie par l'auteur de la base de données.

Voici maintenant une description des bases de données secondaires. La première d'entre elle est présentée à gauche dans la figure 10. C'est la répartition des groupes benthiques. Comme mentionné précédemment, ces groupes seront

utilisés pour entraîner un classificateur de coraux sur un problème moins difficile que celui à 32 classes. Enfin, le graphe de droite présente la proportion des données qui sont identifiés comme étant des coraux. Il est intéressant de noter que notre base de données fournit un échantillon suffisant concernant la distribution des non-coraux. Alors contrairement à ce qui est suggéré dans la littérature sur les systèmes multi-étages avec détecteur, il n'est pas impossible d'implémenter un classificateur à deux classes pour résoudre ce problème. (Landgrebe et al., 2005)

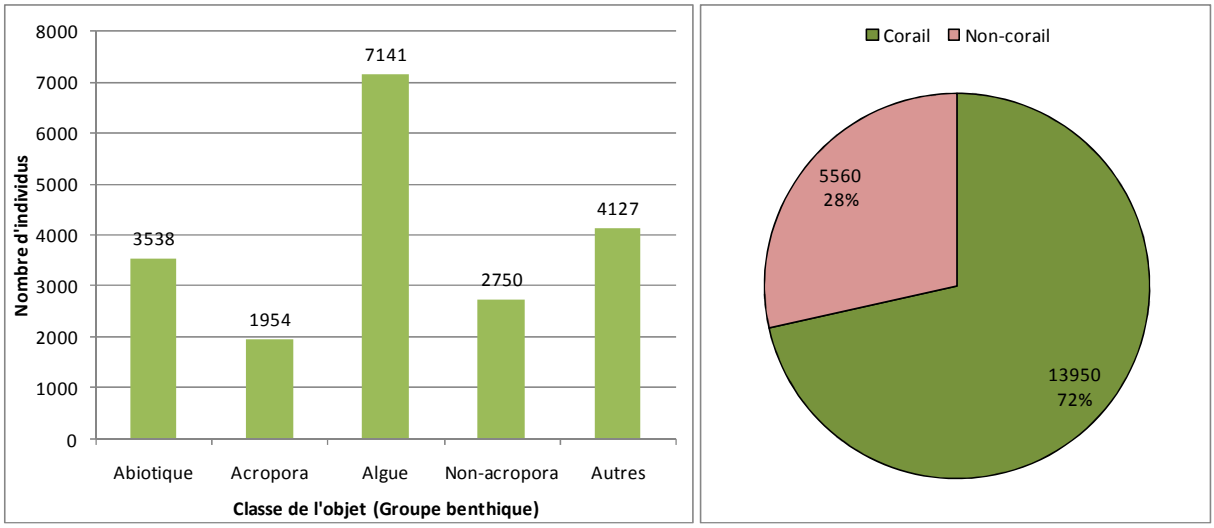

Figure 10 : Répartition des groupes et des catégories

Tableau 2 : Les caractéristiques

Caractéristiques de texture (25)	
Nom	**Description**
Graymean	Moyenne
StdDev	Écart type
MomentR	Moment - valeur R
Skewness	Asymétrie
Kurtosis	Coefficient d'aplatissement
Uniformity	Uniformité
EntropyGray	Entropie du niveau de gris
HContrast	Contraste horizontal
VContrast	Contraste vertical
DContrast	Contraste diagonal
HCorrelation	Corrélation horizontale
VCorrelation	Corrélation verticale
DCorrelation	Corrélation diagonale
HEnergy	Énergie horizontale
VEnergy	Énergie verticale
DEnergy	Énergie diagonale
HHomogeneity	Homogénéité horizontale
VHomogeneity	Homogénéité verticale
DHomogeneity	Homogénéité diagonale
LbpMomentR	Moment R de la forme binaire locale
LbpSkewness	Asymétrie de la forme binaire locale
LbpKurtosis	Coefficient d'aplatissement de la forme binaire locale
FourierOrigin	Origine de la droite pour transformée de Fourier
FourierSlope	Pente de la droite pour transformée de Fourier
FourierCorrelation	Corrélation de la droite pour transformée

Caractéristiques de couleur (18)	
Nom	**Description**
Rhistomax	Maximum de l'histogramme pour le canal rouge
Ghistomax	Maximum de l'histogramme pour le canal vert
Bhistomax	Maximum de l'histogramme pour le canal bleu
Rmean	Moyenne du canal rouge
Gmean	Moyenne du canal vert
Bmean	Moyenne du canal bleu
Rmax	Maximum du canal rouge
Gmax	Maximum du canal vert
Bmax	Maximum du canal bleu
Rmin	Minimum du canal rouge
Gmin	Minimum du canal vert
Bmin	Minimum du canal bleu
Rsobel	Somme du résultat d'un filtre Sobel sur le canal rouge
Gsobel	Somme du résultat d'un filtre Sobel sur le canal vert
Bsobel	Somme du résultat d'un filtre Sobel sur le canal bleu
Rentropy	Entropie du canal rouge
Gentropy	Entropie du canal vert
Bentropy	Entropie du canal bleu

Les caractéristiques présentées à la page précédente sont extraites à partir des zones d'intérêts de 96x72 pixels centrées sur les croix. Ces caractéristiques sont un sous ensemble des descripteurs d'objets biologiques suggérés par les travaux de Yan Levasseur. (Levasseur, 2008) Il s'agit d'un sous ensemble puisque les coraux sont impossible à segmenter. Ainsi, aucune des caractéristiques morphologiques proposées peuvent être extraites.

Enfin, voici une petite analyse séparabilité des classes exécutée à l'aide des outils de l'environnement RapidMiner. (Mierswa et al., 2006) Dans le but de visualiser la complexité du problème, les différents échantillons disponibles ont été placés dans un système à deux dimensions via une analyse en composantes principales. La première figure illustre la répartition des 32 classes d'objets dans un espace considérant deux composantes principales. La figure 12 présente ensuite ces mêmes composantes appliquées sur le problème de la séparation de la catégorie de l'objet. La catégorie indique si l'objet est un corail ou non.

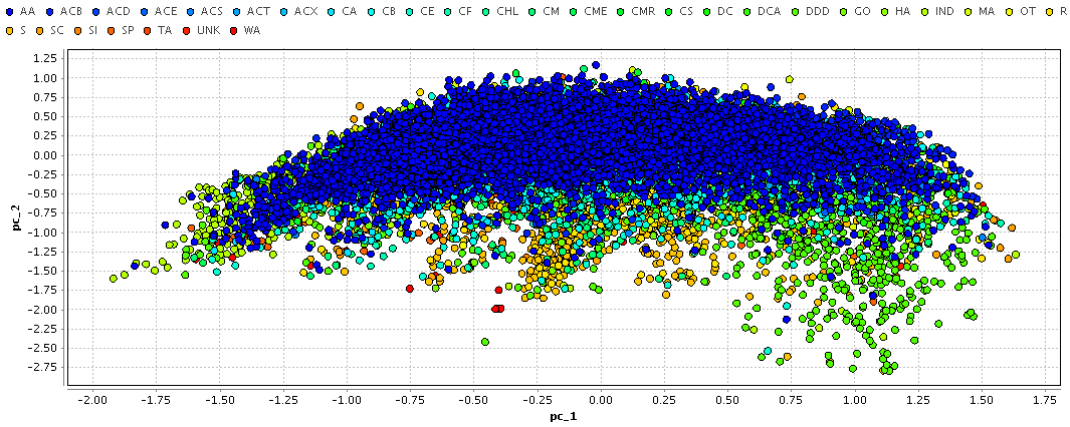

Figure 11 : Analyse par composantes principales sur le type d'objet

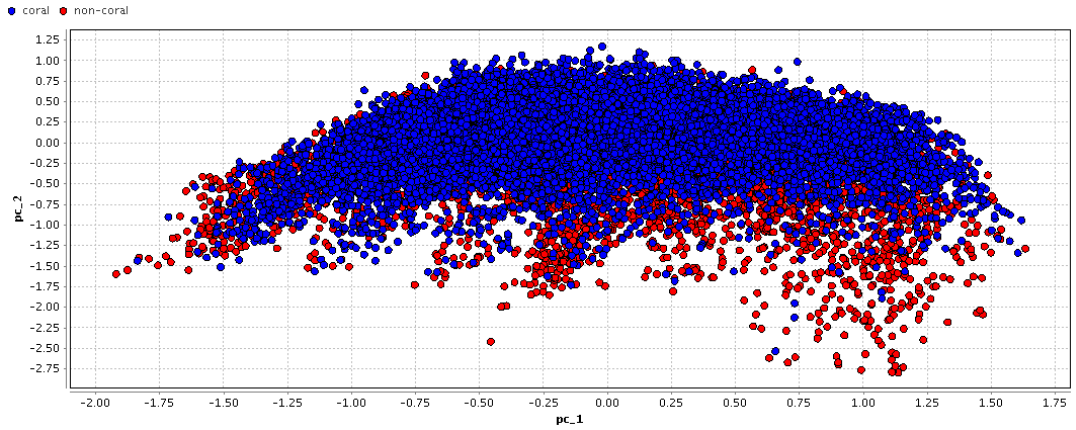

Figure 12 : Analyse par composantes principales sur la catégorie d'objet

Il est évident de constater que les données de chacune des classes sont très confondues dans l'espace. Même en considérant les deux ou trois axes qui expriment le maximum de variation dans les données, il est impossible de séparer clairement les classes. Et ce, peu importe si l'on considère le problème à 32 classes ou la représentation du détecteur à 2 classes. Toutefois, la classe des « non corail » est beaucoup plus répartie dans l'espace. Ce comportement est typique de la classe négative d'un système de détection puisqu'elle comporte, par définition, tout sauf le sujet d'intérêt.

PROTOCOLE EXPÉRIMENTAL

Dans cette section, le protocole expérimental sera présenté en détails. Les étapes seront présentées dans l'ordre chronologique en commençant par la création des bases de données. Ensuite, l'entraînement des classificateurs, l'extraction des prédictions, l'optimisation des seuils de rejet et l'assemblage du système complet seront explicités.

La création des bases de données

La base de données initiale a été générée par les algorithmes tirés de (Levasseur, 2008). Puisque ce dernier utilisait l'environnement de Weka (Witten et Frank, 2005) pour expérimenter, les données sont stockées sous un format texte spécifique à cette application. (ARFF) La première étape est donc d'extraire les données de ce format de fichier. Toutefois l'environnement de RapidMiner (Mierswa et al., 2006) utilisé pour expérimenter propose déjà un filtre d'importation adéquat. Ensuite, les données ont été normalisées afin de limiter l'impact de la plage de variation des caractéristiques. Pour chaque caractéristique, le minimum observé a été ramené à 0 et le maximum observé à 1. Le tout selon une mise à l'échelle linéaire.

Enfin, les données ont été permutées aléatoirement afin d'obtenir un ordre de présentation idéal. Initialement, les données étaient trillées selon leur classe dans la base de données. De plus, deux bases de données indépendantes ont été créées à partir des échantillons permutés aléatoirement. Une base d'apprentissage et de validation comprenant 80% des données a été créée. Finalement, le reste des donnés a été concilié dans une base réservée pour les

tests en généralisation. Le processus de sélection pour les deux bases de données a été effectué avec stratification. Un nombre d'échantillons proportionnel à la distribution initiale a donc été placé dans chacune des bases.

L'entraînement des classificateurs

L'apprentissage des perceptrons multicouches a été décrit brièvement précédemment. Cette section ajoutera plusieurs détails à cette présentation. Puisqu'un faible nombre de données est disponible, l'entraînement a été effectué via un mécanisme de validation croisée à 10 blocs. Encore une fois, les blocs de données ont été extraits avec stratification afin d'informer correctement les classificateurs sur les probabilités à priori de chacune des classes. Cette séparation stratifiée justifie l'utilisation d'une base d'apprentissage contenant 80% des données. En fait, la quantité de données d'apprentissage a été optimisée pour maximiser le nombre de classes possédant plus de dix échantillons; afin que chaque bloc de validation croisée possède au moins un échantillon des classes à faible nombre.

Les perceptrons ont été entraînés avec l'algorithme de Weka (Witten et Frank, 2005) suivant un taux d'apprentissage de 0,3, un momentum de 0,2 et un nombre de neurones cachés égal à la somme du nombre d'entrées et de sorties divisée par deux. Ces paramètres ont été utilisés tels quels pour permettre une comparaison avec les études passées. Enfin, pour chacun des blocs de validation croisée appris par les perceptrons, 15% des données ont été utilisées pour valider l'arrêt de l'apprentissage lorsque 20 itérations successives entraînent l'augmentation de l'erreur. Malheureusement, ce paramètre est intrinsèque à l'algorithme et ne permet pas une sélection stratifiée pour la validation. L'algorithme utilise tout simplement les quinze premiers pourcents du bloc de données en validation. Toutefois, la permutation aléatoire des données élimine une partie du biais imposé par ce mécanisme. Voici un tableau résumant les divers taux de classification et temps d'apprentissage pour les classificateurs :

Tableau 3 : Statistiques sur l'apprentissage des classificateurs

Classificateur	Taux de classification	Variance sur les 10 blocs	Temps d'apprentissage*	Nombre de nœuds cachés
Détecteur (catégorie)	77,68 %	0,71 %	1h 27m 56s	23
Classificateur (groupe)	49,79 %	1,47 %	1h 04m 27s	28
Classificateur (type)	42,65 %	0,69 %	2h 11m 53s	68

* Les temps d'apprentissages sont fournis pour comparaison uniquement. (Algorithme de Weka 3.5.8 sur Mac OS X 10.4, Java 1.5.0_16, CPU quad 2,66 GHz, 5 Go RAM ECC)

L'extraction des prédictions

L'extraction des prédictions est une étape très simple. Il suffit d'utiliser les modèles entraînés des perceptrons multicouches pour généraliser sur la base d'apprentissage et de test. Cette opération permet d'obtenir la prédiction discrète de chaque algorithme de classification et d'évaluer sa performance. De plus, c'est au cours de cette étape que la métrique de confiance est calculée sur la décision. Tout comme pour la création des bases de données et l'entraînement, l'extraction des prédictions a été exécutée à même l'environnement de RapidMiner (Mierswa et al., 2006) en utilisant le perceptron multicouche de Weka. (Witten et Frank, 2005)

Suivant l'exécution du modèle, une probabilité est ajoutée pour chacune des classes et pour chacun des échantillons. Ces probabilités ont une somme égale à un lorsqu'on considère un seul échantillon. La classe possédant la plus forte probabilité est donc sélectionnée comme étant la prédiction du classificateur. Toutes ces données ont été stockées sur disque afin de les conserver. Le reste de l'expérimentation a été complétée à l'aide de MatLab puisque l'environnement de RapidMiner n'est pas assez flexible pour couvrir les manipulations sur les données.

L'optimisation des seuils de rejet

L'optimisation des seuils est sans doute la tâche la plus critique du processus. Étant donné la grande quantité de solutions possibles il vaut mieux déterminer une règle unique permettant de spécifier le meilleur compromis entre l'erreur et le rejet à priori. Cette règle a donc été formalisée par la somme du taux d'erreur et du taux de rejet pour un seuil donné. Les expérimentations font systématiquement varier les taux de rejets entre 0 et 1 et pendant ce temps, le taux d'erreur varie sur une plage moins importante. Considérant ce fait, les taux d'erreurs observés ont été systématiquement normalisés entre 0 et 1 afin que la valeur de rejet ne soit trop défavorisée au cours du processus de minimisation.

La première méthode implémentée est celle de Chow avec seulement un seuil pour toutes les classes. Cette méthode est très simple, elle s'exécute en trois étapes. Premièrement, un seuil est choisi dans un processus itératif. Pour l'expérimentation, 1001 seuils entre 0 et 1 ont été employés. Ensuite, les taux

d'erreur et de rejet reliés au seuil courant sont calculés et stockés. Les taux sont calculés à partir de la base de données d'apprentissage. Cette information est conservée afin de tracer une courbe d'erreur en fonction du taux de rejet t. Enfin, le point d'opération est sélectionné par minimisation de la fonction de coût $f_c = e_n(t) + r(t)$. Voici le pseudo code de cet algorithme :

1) Pour chaque seuil t entre 0 et 1 avec incréments de 0,001
    a. Calculer le taux de rejet $r(t)$ sur la base d'apprentissage
    b. Calculer le taux d'erreur $e(t)$ sur la base d'apprentissage
2) Calculer l'erreur normalisée $e_n(t)$
3) Trouver le seuil qui minimise la fonction de coût $f_c = e_n(t) + r(t)$.

Figure 13 : Algorithme pour le seuillage de Chow

Pour implanter la méthode de Fumera, l'algorithme est plus complexe. En fait, l'utilisation d'un seuil de rejet par classe génère un espace de paramètres très vaste. Considérant l'exploration de 1001 seuils différents pour chacune des classes, le cas du classificateur à 32 classes accepte 1001^{32} possibilités de seuillage, soit approximativement 1,0325 x 10^{96} solutions uniques. L'algorithme choisi pour l'ajustement des seuils est donc très critique. C'est pourquoi deux différents algorithmes ont été mis à l'essai.

En premier lieu, une approche d'optimisation locale a été testée. Au cours de cet algorithme, chaque seuil a été optimisé individuellement. Ce processus s'exécute aussi en trois étapes. Premièrement, une classe parmi le lot des classes à traiter est isolée. L'erreur et le taux de rejet sont ensuite calculés sur cette parcelle des données pour les différents seuils explorés. Ces taux sont uniquement relatifs à la classe courante. La troisième étape consiste à trouver le seuil qui minimise la fonction de coût pour la classe en cours de traitement. Enfin, le processus réitère jusqu'à avoir trouvé l'ensemble des seuils optimaux. Voici sous forme de pseudo-code ce premier algorithme mis à l'essai pour Fumera :

1) Pour chaque classe c
    a. Pour chaque seuil t_c entre 0 et 1 avec incréments de 0,001
        i. Calculer le taux de rejet $r(t_c)$ sur la base d'apprentissage
        ii. Calculer le taux d'erreur $e(t_c)$ sur la base d'apprentissage
    b. Calculer l'erreur normalisée $e_n(t_c)$
    c. Trouver le seuil qui minimise la fonction de coût $f_c = e_n(t_c) + r(t_c)$.
2) Calculer les taux d'erreur et de rejet globaux pour la solution trouvée

Figure 14 : Algorithme 1 pour le seuillage de Fumera

Enfin, un deuxième algorithme a été mis à l'essai pour explorer les seuils de Fumera. Cet algorithme est directement suggéré par (Fumera, Roli et Giacinto, 2000). La littérature présente cet algorithme au cours d'une expérimentation qui démontre que les seuils multiples sont plus performants que l'approche de Chow sur un problème de classification réel. Cet algorithme est aussi très simple. Il utilise toutefois un paradigme d'itération différent. La première étape consiste à itérer sur un ensemble de taux de rejets. Dans l'implémentation choisie, 1001 taux de rejets sont testés entre 0 et 100%. Pour un taux de rejet donné, les seuils associés à chacune des classes sont incrémentés successivement jusqu'à atteindre le rejet désiré. Lorsque cette condition est établie, un nouveau taux de rejet est émis et le processus recommence pour un rejet plus élevé. Cet algorithme a l'avantage de fournir les données nécessaires au traçage de la courbe erreur-rejet, contrairement à l'algorithme 1 présenté plus tôt qui permet seulement d'isoler une solution unique. Voici le pseudo-code :

1) Initialiser le vecteur $\vec{t}_c$ avec les plus hauts seuils qui résultent en un taux de rejet nul
2) Pour chaque taux de rejet r entre 0 et 1 avec incréments de 0,001
    a. Initialiser $c = 1$
    b. Tant que le taux de rejet r n'est pas atteint
        i. Augmenter le seuil t_c de la classe c de 0,001
        ii. Calculer le taux de rejet global $r(\vec{t}_c)$ sur la base d'apprentissage
        iii. Changer de classe pour la prochaine itération de « b. » : $c = c + 1$
        iv. Si $c \geq nombre\ de\ classse$ Alors $c = 1$
    c. Calculer l'erreur $e(\vec{t}_c)$ pour ce point d'opération
3) Calculer l'erreur normalisée $e_n(\vec{t}_c)$ pour tous les niveaux de rejets testés
4) Trouver l'ensemble de seuils qui minimise la fonction de coût $f_c = e_n(\vec{t}_c) + r(\vec{t}_c)$.

Figure 15 : Algorithme 2 pour le seuillage de Fumera

L'assemblage du système à deux étages

Les deux étages du système, ainsi que les seuils de décision étant bien appris, il est temps d'assembler le tout. Cet assemblage est simple, il suffit d'éliminer les spécimens identifiés comme étant les « non-coraux », d'éliminer les spécimens rejetés au premier étage et de concilier les données restantes dans une structure de données. Cette structure est ensuite analysée par le deuxième étage. Il s'agit ensuite d'évaluer le taux d'erreur et le taux de rejet de l'architecture complète. La section suivante présente ces résultats, plusieurs assemblages y sont testés.

RÉSULTATS DE SIMULATION

Les résultats seront présentés en deux parties. La première concerne la sélection des seuils à partir des données d'entraînement, tandis que la deuxième montre l'impact des systèmes de type détecteur-classificateur sur les données de test. Voici tout d'abord les trois courbes erreur-rejet sur la base d'apprentissage pour le détecteur, le classificateur de groupe benthique et le classificateur de type à 32 classes.

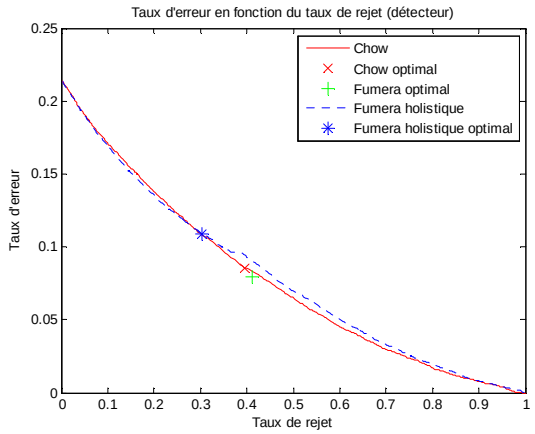

Figure 18 : Courbe d'erreur-rejet (détecteur)

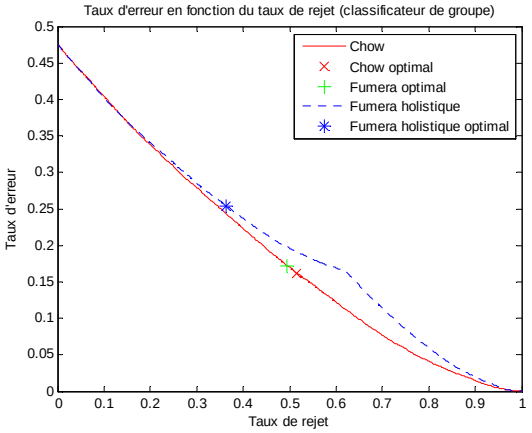

Figure 17 : Courbe d'erreur-rejet (classificateur de groupe)

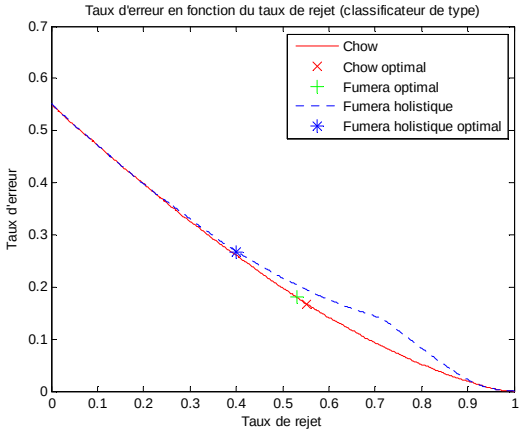

Figure 16 : Courbe d'erreur-rejet (classificateur de type)

Les courbes des figures 16, 17 et 18 possèdent toutes la même anomalie. Selon la théorie de Fumera (Fumera, Roli et Giacinto, 2000), l'optimisation de seuils multiple résulte en une courbe d'erreur rejet qui est aussi bonne ou sinon meilleure que la courbe de Chow. Les courbes bleues en pointillés devraient donc se situer sous les courbes rouges en ligne continues. Ce n'est manifestement pas le cas. L'algorithme 2 pour Fumera, ou l'algorithme de Fumera holistique tel qu'il est identifié sur les graphes, n'est donc pas capable de trouver une solution qui outrepasse celle de Chow. Au cours des expérimentations, il a été remarqué que l'ordre de modification des seuils impacte sur la forme de la courbe obtenue. Les courbes présentées utilisent un

ordre séquentiel pour la modification des seuils, tel que spécifié par l'algorithme. Toutefois, aucun des autres ordres testés n'a surpassé la méthode de Chow.

Du coté du seuillage optimisé par l'algorithme 1 de Fumera, soit les symboles « + » verts sur les graphes, on remarque que la théorie de Fumera est respectée. (Les points identifiés sur les graphes sont ceux qui minimisent la fonction de coût présentée dans la description des algorithmes.) Dans le cas du détecteur, le point d'opération sélectionné est meilleur de celui de Chow, il se situe sous la courbe. Pour les autres sous-systèmes, le point d'opération sélectionné par l'algorithme 1 se situe exactement sur la courbe d'erreur-rejet de Chow. Ce qui prouve que le théorème peut être respecté. Toutefois, tout dépend de l'algorithme d'optimisation des seuils. Une solution optimale sera bien évidemment supérieure à Chow. Toutefois, comme les écrits de Fumera le dictent, même l'algorithme de sélection des seuils qu'il propose est sous optimal. (Soit l'algorithme 2) Pour trouver un seuillage idéal il faut réellement exécuter un algorithme holistique qui explore tout l'espace des seuils à chaque itération. Les algorithmes simples testés ici explorent une solution locale et une solution un peu plus holistique mais qui fait varier les seuils dans un ordre qui influence le résultat. C'est pourquoi l'optimalité n'a pas été atteinte.

Enfin, malgré que la théorie de Fumera n'a pas été validée, les résultats obtenus sont intéressant. On observe une diminution significative de l'erreur en appliquant la méthode seuillage sur les probabilités. Les graphes et la table de la page suivante présentent les résultats des diverses combinaisons de seuillage. La figure 19 présente les performances obtenues pour les points d'opération sélectionnés pour le classificateur de groupe selon huit groupes de barres. La barre de gauche en bleu indique le taux de rejet appliqué, la barre de droite en rouge, le taux d'erreur associé à cette configuration de rejet. Les groupes de barres présentent la performance du système *de base* avec un seul étage pour la classification, la performance du système *sans rejet* avec détecteur et classificateurs combinés, la performance du système avec rejet de *Chow*, *Fumera* et *Fumera holistique* appliqué au premier étage et enfin la performance du système de *Chow X2*, *Fumera X2* et *Fumera Holistique X2* pour un rejet appliqué aux deux étages successivement. Il en va de même pour le deuxième graphe qui présente les mêmes résultats pour le classificateur de type.

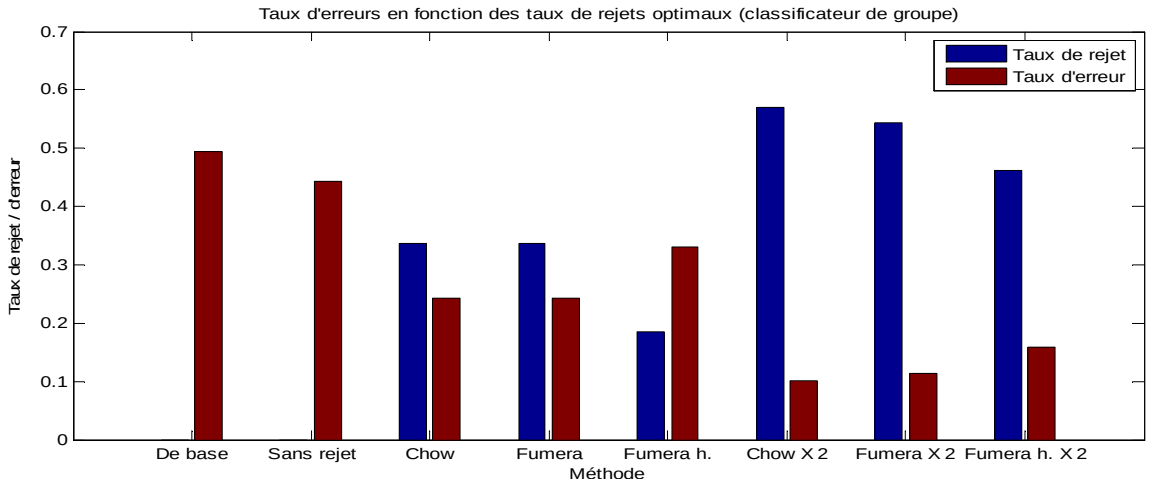

Figure 19 : Performances pour le classificateur de groupe

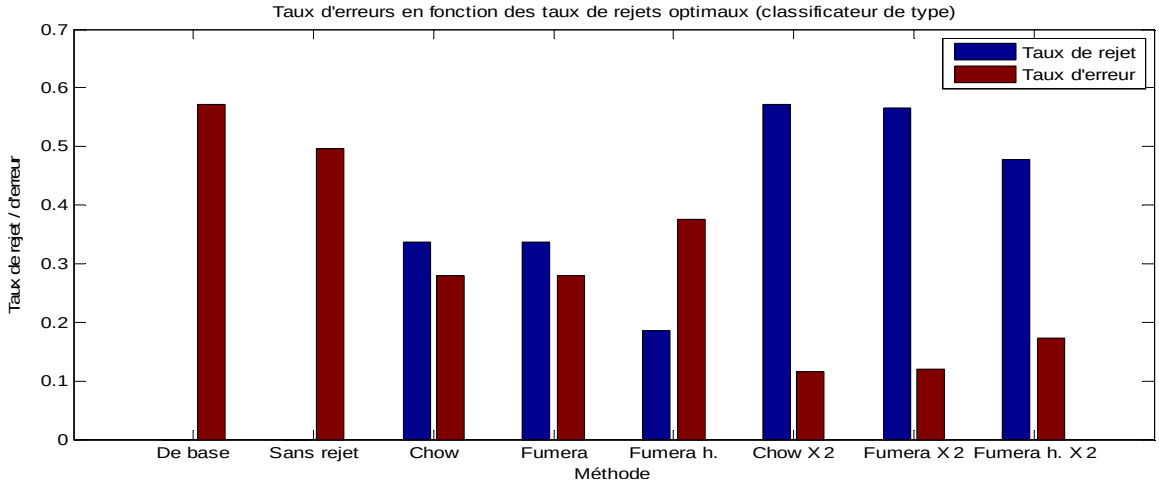

Figure 20 : Performances pour le classificateur de type

Tableau 4 : Résultats de classification sur la base de test

Systèmes simples	**Taux de classification**	**Taux de rejet**
De base sans validation croisée (Bouchard, 2008a)	41,6 %	0,00 %
Détecteur	78,61 %	0,00 %
Classificateur de groupe	50,47 %	0,00 %
Classificateur de type	42,74 %	0,00 %
Systèmes à deux étages	**Taux de classification**	**Taux de rejet**
Dét. et class. de groupe sans rejet	55,75 %	0,00 %
Dét. et class. de groupe avec un rejet Chow	75,69 %	33,64 %
Dét. et class. de groupe avec un rejet Fumera	75,69 %	33,64 %
Dét. et class. de groupe avec un rejet Fumera H.	67,03 %	18,47 %
Dét. et class. de groupe avec deux rejets Chow	89,88 %	57,01 %
Dét. et class. de groupe avec deux rejets Fumera	88,65 %	54,29 %
Dét. et class. de groupe avec deux rejets Fumera H.	84,04 %	46,12 %
Dét. et class. de type sans rejet	50,29 %	0,00 %
Dét. et class. de type avec un rejet Chow	72,07 %	33,64 %
Dét. et class. de type avec un rejet Fumera	72,07 %	33,64 %
Dét. et class. de type avec un rejet Fumera H.	62,39 %	18,47 %
Dét. et class. de type avec deux rejets Chow	88,47 %	57,11 %
Dét. et class. de type avec deux rejets Fumera	88,11 %	56,50 %
Dét. et class. de type avec deux rejets Fumera H.	82,60 %	47,83 %

On remarque que pour augmenter la précision des classificateurs on doit payer un fort coût en classification manuelle. Les meilleurs résultats obtenus amènent les taux de classification au voisinage de 90%, toutefois pour arriver à ce taux plus de la moitié de la base de donnée a été rejetée et envoyée à la classification manuelle. Enfin, c'est sans considérer que le système de détection élimine déjà 17,17% des données, ce qui ramène le taux de données non considérées à 74,28% pour le système détecteur et classificateur avec rejets de Chow sur chaque étage.

Le rejet appliqué à deux étages successifs est quelque peu excessif. Toutefois, étant donné la grande complexité du problème et la mauvaise qualité de certains échantillons de la base de données, c'est ce gendre d'approche qui résulte en un taux d'erreur acceptable. Il est tout de même possible d'améliorer de près de 20% les taux de classification en ne considérant qu'un rejet de 18,47% avec l'algorithme de Fumera holistique. Il en va de même pour l'autre extrême, une amélioration de 45,73% de la précision est possible avec le rejet de 57,11% des coraux. L'amélioration de 1% de la classification est donc attribuable à un rejet semblable de l'ordre du 1%. Évidement, les coûts reliés aux rejets pourraient être amoindris en utilisant un algorithme de sélection plus évolué pour Fumera.

Une dernière constatation intéressante est que le problème à cinq classes pour le classificateur de groupe n'est pas significativement plus performant que l'approche avec 32 classes. La classe AA contribue le plus aux bons taux de classification puisque c'est la plus populeuse de l'échantillonnage disponible. Cette tendance se maintient pour le problème à 5 classes, c'est le groupe benthique « algue » qui englobe la classe « AA » qui est le mieux classifié. Le débalancement des classes dans les deux échantillonnages est probablement ce qui cause les erreurs de classification similaires observées pour le problème à 5 et 32 classes.

CONCLUSION

Au cours de cette expérimentation, un système multi étage a été implémenté. Le détecteur et le classificateur de ce système ont ensuite été utilisés conjointement à un mécanisme de rejet. Une amélioration significative des performances du système a été observée. Toutefois, cette amélioration est sujette à un coût non négligeable en classification manuelle. En effet, il existe une relation qui relève pratiquement du un pour un en ce qui concerne l'augmentation des taux de rejets et l'amélioration des taux de classification observée. L'approche de Fumera s'est avérée plus efficace que celle de Chow dans un des cas, celui du système de classification à deux classes pour le détecteur. Dans les autres cas, l'approche de Chow est prédominante puisque nos algorithmes d'optimisations des seuils sont sous optimaux.

Afin d'explorer d'avantage l'ensemble de seuils optimaux de Fumera, il serait intéressant d'utiliser un algorithme qui prends en charge toute la complexité du problème. Un algorithme holistique inspiré de la programmation génétique pourrait être utilisé. La littérature présente un cas où un algorithme de type PSO (Particle Swarm Optimisation) a été utilisée pour solutionner ce problème. (Oliveira, Britto et Sabourin, 2005)

De plus, le protocole proposé par Fumera prévoit uniquement une base de données d'apprentissage et une base de test. Pour s'assurer d'optimiser les seuils de décisions sur des données indépendantes, il serait intéressant de reprendre l'expérience avec une base de données de validation. Toutefois, la base de données du récif corallien étudié est très limitée. L'implémentation de l'optimisation des seuils au cours du processus de validation croisée serait donc à privilégier. Pour ce faire, une moyenne des dix seuils obtenus en validation croisée pourrait être comptabilisée. Si la variance des seuils sur les divers blocs de donnés utilisés est trop grande, une approche itérative pourrait surpasser la moyenne. Par exemple, la solution trouvée lors d'une première itération de validation croisée pourrait être conservée comme solution initiale de l'itération suivante. Ainsi les seuils convergeront vers une solution unique optimale et acceptable en termes d'indépendance. Le tout en réservant un bloc de validation au sein du processus de validation croisée, en plus d'un bloc de test.

Il serait aussi très intéressant de consulter l'image originale des échantillons rejetés et des échantillons conservés par le processus de rejet. Cette observation aiderait à comprendre quels facteurs influent sur la qualité d'un échantillon d'image ou qu'est ce qui impacte les taux de confiances sur la décision des classificateurs. L'extraction des images originales a été tentée au cours de ce travail. Toutefois, un problème de programmation hors de notre contrôle a fait en sorte que l'outil RapidMiner a corrompu le champ qui identifie de façon unique les données. Ce problème a rendu impossible une telle analyse.

Enfin, la sélection des points d'opérations en fonction de divers taux d'erreur désirés serait aussi intéressante. Plutôt que de tenter d'isoler le meilleur compromis entre l'erreur et le rejet, cette méthode permet de fixer l'erreur à un taux acceptable et se concentrer sur la diminution du taux de rejet associé. De plus, le système utilise le classificateur de base comme deuxième étage de classification. Ce classificateur est entraîné à séparer 32 classes distinctes, dont les 9 classes qui correspondent à la distribution des « non coraux » dans le paradigme de détection. Utiliser un classificateur entraîné uniquement à séparer les coraux serait une des premières approches à tester pour diminuer les rejets pour un taux d'erreur prédéterminé.

En somme, plusieurs tests restent à être exécutés. L'approche de classification avec rejet est une technique très intéressante qui sera certainement incluse dans la solution finale pour le problème sur les coraux. Ce mécanisme garantit la durée de vie d'un système de classification pour des données à classifier qui évoluent dans le temps.

RÉFÉRENCES

Bouchard, Jonathan. 2008a. *Préparation des données pour la classification d'images de coraux*. Exigence du cours SYS821. Montréal: École de technologie supérieure, 20 p.

Bouchard, Jonathan. 2008b. *Rapport final : Optimisation des méthodes de classification*. Exigence du cours GPA792. Montréal: École de technologie supérieure, 22 p.

Chow, C. 1970. « On optimum recognition error and reject tradeoff ». *Information Theory, IEEE Transactions on,* vol. 16, n° 1, p. 41-46.

Fumera, Giorgio, Fabio Roli et Giorgio Giacinto. 2000. « Reject option with multiple thresholds ». *Pattern Recognition,* vol. 33, p. 2099-2101.

Landgrebe, Thomas, Pacl, Pavel k, David M. J. Tax et Robert P. W. Duin. 2005. « Optimising two-stage recognition systems ». In. Vol. 3541, p. 206-215. Coll. « Lecture Notes in Computer Science ». Seaside, CA., United states: Springer Verlag.

Levasseur, Yan. 2008. « Techniques de l'intelligence artificielle pour la classification d'objets biologiques dans des images bidimensionnelles ». Montréal, École de technologie supérieure, 172 p.

Mierswa, Ingo, Michael Wurst, Ralf Klinkenberg, Martin Scholz et Timm Euler. 2006. « YALE: Rapid Prototyping for Complex Data Mining Tasks ». *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 935-940

Oliveira, L. S., A. S. Britto, Jr. et R. Sabourin. 2005. « Optimizing class-related thresholds with particle swarm optimization ». In *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on*. Vol. 3, p. 1511-1516 vol. 3.

Witten, Ian H., et Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*, 2. San Francisco: Morgan Kaufmann, 525 p.