

Ministre de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie d'Oran « Mohamed Boudiaf »

Faculté des Sciences

Département d'Informatique

Spécialité : Informatique

**Option : Télédétection, Analyse et Traitement
Informatique des Données Spatiales**

Mémoire présenté par M^{elle}. BENMOSTEFA Soumia

Pour l'obtention du diplôme de Magister

Thème :

**Conception de la Technique des Chauves-souris
pour la Classification Automatique des Images**

2011/2012

Remerciements

Je tiens à exprimer mes profonds remerciements à mon encadreur Mme FIZAZI ABETENE .H, responsable de la post graduation « Télédétection, Analyse et traitement informatique des données spatiales » (TATIDS) pour l'aide précieuse qu'elle m'a apporté, pour sa patience et ses encouragements; elle s'est toujours montrée à l'écoute et très disponible tout au long de la réalisation de ce mémoire, pour l'inspiration, l'aide et le temps qu'elle a bien voulu me consacrer et sans qui ce mémoire n'aurait jamais vu le jour.

On remercie le président de jury xxxxxxxx, ainsi que les membres : xxxxxxxx, Madame xxxxxxxx d'avoir accepté de valoriser ce travail.

J'exprime aussi mes sincères reconnaissances à tous mes enseignants pour leurs efforts fournis durant toute la période d'étude ainsi qu'à tous ceux qui ont collaboré d'une façon ou d'une autre à l'élaboration de ce travail.

Je remercie les membres du laboratoire pour le bon accueil, la bonne humeur, l'alliance parfaite entre la détente et la recherche.

Mes remerciements s'adressent également à Mr Alexandre Laurent d'avoir gardé un œil sur l'avancement de ce mémoire, qui a relu mon travail à mesure que j'écrivais. Qu'il trouve ici l'expression de ma reconnaissance et de mon respect.

Ma famille,

Aucune des expressions ne saurait exprimer l'immense amour que je vous porte.

Je vais donc me contenter de vous dédier ce modeste travail

en guise de ma reconnaissance éternelle

Conception de la Technique des Chauves-souris pour la Classification Automatique des Images

-Résumé-

La classification des images joue un rôle prépondérant dans le traitement d'images, l'aide à la décision, la télédétection, et les applications cliniques de diagnostic médical. La classification d'image conventionnelle consiste principalement à partitionner l'image en différentes régions ou groupes de manière à optimiser un certain critère. Les progrès technologiques ont rendu possible la retranscription du problème de classification en un problème d'optimisation. Dans ce mémoire, nous proposons un nouvel algorithme d'optimisation, dit Algorithme de chauves-souris pour résoudre le problème de classification d'images, où chaque pixel est assigné à un seul groupe. Les résultats obtenus après une étude expérimentale et comparative illustrent une amélioration significative dans la classification d'images par rapport aux méthodes classiques de classification et d'autres techniques récentes. De surcroît, les expérimentations menées ainsi que les résultats prouvent que l'utilisation de l'algorithme des chauves-souris peut améliorer la structure de et l'efficacité de recherche des groupes optimales.

Mots clés— *Algorithme des chauves-souris; Segmentation des images; Classification non-supervisée des images ; Classification automatique d'images.*

Conception of the Bat Algorithm for Automatic Image Classification

-Abstract-

Image classification plays a major role in image analysis, computer aided detection and diagnostics for clinical applications. Conventional image classification is primarily concerned with the partitioning of data points into different clusters so as to optimize a certain criterion. Rapid advances in technology have made possible to address classification problems via optimization theory. In this thesis, we propose a novel Bat Optimization Algorithm to solve the image classification problem, where each data point is to be assigned to one cluster. The results obtained by experimental and comparative study demonstrated a significant improvement in image classification over the classical clustering techniques and some other techniques. Moreover, the results validate and suggest that using a new Bat Optimization Algorithm can improve the structure research and effectiveness.

Keywords-*Unsupervised Image classification; Bat Algorithm; Image clustering; Automatic Image Classification;*

Table des matières

Introduction générale	2
------------------------------------	---

Chapitre 1 : Interprétation des images

1. Introduction.....	6
2. Chaîne de traitement d'images.....	7
2.1. Acquisition d'images.....	7
2.2. Prétraitement.....	11
2.3. Interprétation des images.....	12
2.4. Post-traitement.....	12
3. Algorithmes d'interprétation des images.....	12
3.1. Les algorithmes non supervisés de localisation.....	13
3.2. Les algorithmes de reconnaissance et de catégorisation.....	14
3.3. Les algorithmes supervisés de localisation.....	17
3.4. Les algorithmes de localisation multi-classes.....	19
3.5. Les autres algorithmes d'interprétation des images.....	21
4. Conclusion.....	22

Chapitre2 : Classification et méthodes de classification

1. Introduction.....	25
2. Définition formelle de la classification.....	25
3. Mesures de ressemblances.....	26
4. Techniques classiques de la classification.....	26
4.1. Algorithmes de classification hiérarchique.....	27
4.2. Techniques de classification partitionnaires.....	28
4.3. Les algorithmes de classification basés sur la densité.....	34
4.4. Les algorithmes de classification basés sur les grilles.....	35
4.5. Techniques basées sur la population.....	36
5. Classification des images.....	38
6. Conclusion.....	40

Chapitre 3 : Algorithme des chauves-souris

1. Introduction.....	42
2. Echolocation des chauves-souris.....	42
2.1. Comportement des microchiroptères.....	42
2.2. L'acoustique d'écholocation.....	43
3. Optimisation par algorithme des chauves-souris.....	44
4. Conclusion.....	47

Chapitre 4 : Classification des images par algorithme des chauves-souris

1. Introduction.....	49
2. Formulation du problème.....	49
2.1. Formulation du problème de la classification des images.....	49
2.2. Indices de validités utilisés.....	50
2.3. Mesures de similarité utilisées.....	52
3. Classification des images avec l'algorithme des chauves-souris.....	52
4. Environnement de travail.....	53
5. Les images de test.....	54
5.1. Images synthétiques.....	54
5.2. Images satellitaires.....	55
6. Schéma de l'application.....	55
7. Résultats expérimentaux et discussion.....	56
7.1. Influence des paramètres.....	56
8. Conclusion.....	59
Conclusion générale et perspectives.....	62
Bibliographies.....	65

Liste des Figures

Figure 1. Types d'images selon la nature de leurs systèmes d'acquisition.....	7
Figure 2. Les étapes du processus de Télédétection.....	9
Figure 3. Une chaîne de traitement d'une image (images du ©LASTI).....	11
Figure 4. Localisation par estimation de l'avant plan.....	14
Figure 5. Exemple de reconnaissance utilisant des patches de l'objet.....	17
Figure 6. Localisation de véhicules comme Exemple de résultats de localisation supervisée.	18
Figure 7. Exemples de localisation par reconnaissance de points d'intérêt.....	19
Figure 8. Exemple d'interprétation d'images.....	21
Figure 9. Exemple de suivi d'un véhicule dans un parking.....	22
Figure 10. Exemple illustrant la classification fractionnelle et agglomérative.....	27
Figure 11. Algorithme de classification partitionnaire.....	29
Figure 12. (a) : l'image avant classification, (b) : image après classification.....	39
Figure 13. Echolocation.....	43
Figure 14. Pseudo code de l'algorithme des chauves-souris.....	46
Figure 15. Algorithme des chauves-souris adapté à la classification d'images.....	53
Figure 16. Base de données d'images de test.....	54
Figure 17. Images synthétiques	55
Figure 18. Zone d'étude.....	55
Figure 19. L'image satellitaire étiquetée.....	55
Figure 20. Schéma général de l'application.....	57
Figure 21. La valeur de la fitness en fonction de la taille de population.....	58
Figure 22. La valeur de la fitness en fonction du nombre de population.....	58

Liste des Tableaux

Table 1. Image synthétique classifiée par algorithme des chauves-souris.....	59
Table 2. Image satellitaire classifiée par algorithme des chauves-souris.....	60

Introduction générale

Introduction générale

Positionnement de la problématique

Durant ces dernières années, l'évolution des moyens informatiques a permis la manipulation et le stockage de grandes quantités d'images de hautes qualités et très riches en informations. Ces informations sont potentiellement utiles pour un large éventail d'applications. Parmi celles-ci nous pouvons compter les applications militaires, la prévention du crime, les systèmes d'information géographique (SIG), la télédétection et les diagnostics médicaux. Le défi majeur pour tous ces domaines est l'extraction et l'exploitation des informations afin d'en tirer le maximum.

Lorsqu'un être humain observe une image naturelle, il divise l'image en objets ou segments, puis les identifie afin de pouvoir l'interpréter. Cependant, cette interprétation visuelle atteint actuellement ses limites, vu que les images sont devenues très complexes et formées de nombreux éléments. De ce fait est né le besoin de mettre en œuvre des techniques et des méthodes automatiques permettant d'extraire le maximum d'informations possibles à partir des images. Ces méthodes s'inscrivent dans le cadre du traitement d'images. L'opération la plus importante dans ce traitement, permettant une meilleure interprétation, est la classification. Il s'agit donc d'une étape-clé et sa qualité conditionne fortement la prise de décision dans des domaines très variés.

La classification, dite aussi segmentation, consiste à localiser et à délimiter les entités présentes dans une image. Elle est souvent réalisée avant les étapes d'analyse et de prise de décision. On peut distinguer deux approches principales de classification d'images : supervisée et non supervisée. Dans l'approche supervisée, le nombre et la signification des classes sont connus au préalable (par un analyste) et sont utilisés dans l'étape d'apprentissage. Quant à l'approche non supervisée, appelée aussi classification automatique ou segmentation, elle regroupe les pixels similaires selon un certain critère d'homogénéité sans connaître à l'avance ni le nombre de régions ni leurs significations. L'approche non supervisée est plus avantageuse que l'approche supervisée, du fait qu'aucune connaissance préalable d'un analyste n'est nécessaire.

De nombreuses méthodes ont été conçues pour résoudre le problème de la classification automatique d'images. Parmi les plus populaires : K-means, ISO-DATA, Parallélépipède et le maximum de vraisemblance. Cependant ces approches classiques présentent quelques inconvénients, dont les principaux sont : leur grande sensibilité à la configuration initiale ainsi que la convergence prématurée vers un optimum local. De surcroît, plusieurs classifications d'une même image sont possibles. Par conséquent, les recherches ont adaptés le problème de classification en un problème d'optimisation. En effet, les méthodes d'optimisation présentent l'avantage de donner en sortie un nombre optimal de groupes et classifie l'image avec la meilleure partition selon une fonction objective. Ce point de vue a permis d'appliquer la catégorie la plus puissante des méthodes d'optimisation dite métaheuristique à la classification des images. La majorité des métaheuristicques s'inspirent des phénomènes biologiques et physiques de la nature, telle que l'optimisation par essaim particulaires (PSO) basée sur le comportement d'essaim d'oiseaux ou de poissons, les algorithmes génétiques dérivés de la théorie d'évolution de la nature et le recuit simulé fondé sur un processus de métallurgie.

Objectifs

Ce mémoire porte principalement sur l'étude et la conception de l'algorithme des chauves-souris pour la classification automatique des images. L'algorithme des chauves-souris a été proposé par Xin-She.Yang en 2010, c'est une métaheuristique qui tente de modéliser le comportement d'écholocation chez les chauves-souris. Ces dernières émettent des pulsations sonores très fortes. Dès qu'un ultrason rencontre un obstacle (proie, végétation...), il rebondit vers la chauve-souris. Celle-ci capte l'écho grâce à ses oreilles. Son cerveau va alors calculer la distance, la vitesse, la position et la forme de l'objet détecté.

Formellement, l'algorithme des chauves-souris considère chaque chauve-souris comme une classification possible de l'image. À chaque itération la chauve-souris se déplace selon trois équations correspondantes à la fréquence, la position et la vélocité. Chaque déplacement donne lieu à une classification possible de l'image. Cette dernière est évaluée et comparée avec les classifications obtenues par la chauve-souris elle-même, puis avec toutes les autres chauves-souris de la volée. La classification retenue est celle qui est jugée meilleure selon une fonction objective.

La nouvelle approche de classification automatique basée sur l'algorithme des chauves-souris est implémentée et appliquée au préalable sur des images synthétiques dont les régions sont distinguables à l'œil nu, cela permet de tester l'applicabilité et l'efficacité de l'approche. Puis, sur des images satellitaires, particulièrement sur une image de la région d'ORAN ouest en ALGERIE. Cette zone d'étude a été choisie pour notre maîtrise thématique de la région ainsi que pour son paysage varié pouvant présenter un intérêt pour l'évaluation de notre approche. Après une étude expérimentale et comparative, il s'est avéré que l'approche proposée s'est montrée plus performante que d'autres algorithmes dédiés à la classification selon certains critères, tels que l'absence de confusions entre les classes, le taux de classification et le temps de classification.

Finalement, notons que l'algorithme des chauves-souris, comme tout algorithme de classification, est influencé par des paramètres tels que la taille de la population, le nombre d'itérations, le critère d'arrêt, etc. Le choix de ses paramètres influence fortement sur la qualité de la classification.

Organisation du mémoire

Dans le premier chapitre nous évoquerons brièvement les principes et les opérations de base du traitement d'images. Dans un deuxième chapitre, nous détaillerons et nous formulerons le problème de classification automatique, puis nous présenterons ses techniques de base. Dans le troisième chapitre nous énoncerons et expliquerons l'algorithme des chauves-souris. La suite du mémoire sera consacrée notre contribution : dans le quatrième chapitre nous détaillerons nos travaux sur la classification des images et sur l'application de l'algorithme des chauves-souris à la classification automatique des images. Finalement nous conclurons à partir des résultats de nos travaux et discuterons des perspectives et des extensions ultérieures de notre approche.

Chapitre 1 :
Interprétation des images

1. Introduction

Grâce aux récentes avancées de la technologie, en particulier dans le domaine du multimédia et de l'informatique, l'information numérique est devenu le cœur de tous les secteurs d'activités : dans le domaine industriel, médical, scientifique, juridique, géographique, etc. Ces progrès se sont accompagnés d'une baisse des coûts des équipements informatiques qui a facilité la diffusion et l'échange de données multimédia numérisées vers le grand public. Cette masse de données n'aurait aucun intérêt si l'on ne pouvait pas facilement retrouver les informations concernant un intérêt particulier. Cela a suscité un besoin en développement de techniques de recherche d'information multimédia, et en particulier de recherche d'images. Ces techniques s'inscrivent dans le cadre du traitement ou de l'interprétation des images.

Le traitement d'images est un domaine vaste. De nombreux algorithmes permettent de traiter les images ou les vidéos depuis leur acquisition par un capteur (appareil photographique, webcam, satellite, échographe...) jusqu'à l'extraction de l'information utile pour l'application (détection d'un objet particulier, mesure quantitative...). L'extraction d'un maximum d'informations sur les objets composant une scène, le plus automatiquement possible, est l'objectif principal de l'interprétation d'images. Les algorithmes d'interprétation sont donc des algorithmes de haut niveau et occupent une place importante dans le processus de traitement.

Les algorithmes d'interprétation d'images peuvent être utilisés pour de nombreuses applications : détection de la présence d'objets, comptage, contrôle qualité, localisation dans l'image... Pour chacune de ces applications spécifiques, il existe plusieurs algorithmes permettant de traiter le problème. Il est alors nécessaire de définir clairement leurs principes de fonctionnement, notamment les données qu'ils utilisent en entrée ou qu'ils renvoient, afin de pouvoir les évaluer correctement.

Dans ce chapitre, nous présentons la chaîne du traitement d'images ainsi que les principaux types d'algorithmes d'interprétation d'images et quelques exemples. Une discussion sur les algorithmes d'interprétation d'images est enfin proposée.

2. Chaîne de traitement d'images

2.1. Acquisition des images

Les images constituent une source importante d'observation de phénomènes physiques. Elle correspond à la représentation d'une scène réelle sur un support. Ce support peut être une toile ou bien du papier lorsque l'on s'intéresse à la peinture ou à la photographie. Cependant, dans le cadre du traitement automatisé des images, c'est-à-dire dans le domaine informatique, ce support prend généralement la forme d'une matrice dont les valeurs correspondent à la numérisation d'un signal acquis par un capteur. Ce capteur est sensible à un rayonnement physique et peut renvoyer une ou plusieurs données correspondant chacune à une longueur d'onde (lumière visible, infrarouge, rayons X...).

Les capteurs permettant d'acquérir des images numériques étant de différentes natures, les images que nous traitons le sont également. Nous pouvons obtenir des images dans le domaine visible prises depuis un satellite ou bien au sol, mais nous pouvons également obtenir des images dans des domaines non visibles par l'homme, tels que les infrarouges ou les rayons X. Quelques images, provenant de différents systèmes d'acquisition, sont présentées ci-dessous.



(a) Image satellitaire



(b) Image médicale



(c) Image naturelle



(d) Image aérienne

Figure 1. Types d'images selon la nature de leurs systèmes d'acquisition

Les techniques d'acquisition d'images sont nombreuses et varient selon la nature de l'information recherchée, la nature du milieu concerné ainsi que de son étendu. On peut citer :

- la photographie pour le grand public fournissant des images naturelles;
- l'imagerie médicale;
- l'imagerie aérienne;
- l'imagerie satellitaire.

Les moyens et les techniques d'acquisition d'images permettent de caractériser leurs natures. Par conséquent, les images peuvent être classées comme suit :

Image satellitaire

A l'heure actuelle, on voit des images satellitaires partout: dans les informations météo à la TV, dans les journaux et même dans les publicités. Les images satellitaires sont des images prises à partir d'une 'lune artificielle' (un satellite) qui orbite au-dessus de la Terre. Elles résultent d'une suite d'étapes qui s'inscrivent dans un processus appelé Télédétection.

La Télédétection est la technique qui, par l'acquisition d'images, permet d'obtenir des informations sur la surface de la Terre sans contact direct avec celle-ci. Elle englobe tout le processus qui consiste à capter et à enregistrer l'énergie d'un rayonnement électromagnétique émis ou réfléchi, à traiter et à analyser l'information ; pour ensuite mettre en application cette information. »

La Télédétection a plusieurs applications telles que ; la prévention du temps, l'étude de l'évolution des climats, la mesure et la surveillance du trou dans la couche d'ozone, la détection et le suivi des feux de forêt ou des nappes de pétrole, créer des cartes, etc.

Le processus de télédétection comporte les sept étapes que nous élaborons ci-après :

1. *Source d'énergie ou d'illumination (A)* - À l'origine de tout processus de télédétection se trouve nécessairement une source d'énergie pour illuminer la cible.
2. *Rayonnement et atmosphère (B)* – Durant son parcours entre la source d'énergie et la cible, le rayonnement interagit avec l'atmosphère. Une seconde interaction se produit lors du trajet entre la cible et le capteur.

3. *Interaction avec la cible (C)* - Une fois parvenue à la cible, l'énergie interagit avec la surface de celle-ci. La nature de cette interaction dépend des caractéristiques du rayonnement et des propriétés de la surface.
4. *Enregistrement de l'énergie par le capteur (D)* - Une fois l'énergie diffusée ou émise par la cible, elle doit être captée à distance (par un capteur qui n'est pas en contact avec la cible) pour être enfin enregistrée.
5. *Transmission, réception et traitement (E)* - L'énergie enregistrée par le capteur est transmise, par des moyens électroniques, à une station de réception où l'information est transformée en images (numériques ou photographiques).
6. *Interprétation et analyse (F)* - Une interprétation visuelle et/ou numérique de l'image traitée est ensuite nécessaire pour extraire l'information que l'on désire obtenir sur la cible
7. *Application (G)* - La dernière étape du processus consiste à utiliser l'information extraite de l'image pour mieux comprendre la cible, pour découvrir de nouveaux aspects ou pour aider à résoudre un problème particulier.

Ces sept étapes couvrent le processus de la télédétection, du début à la fin.

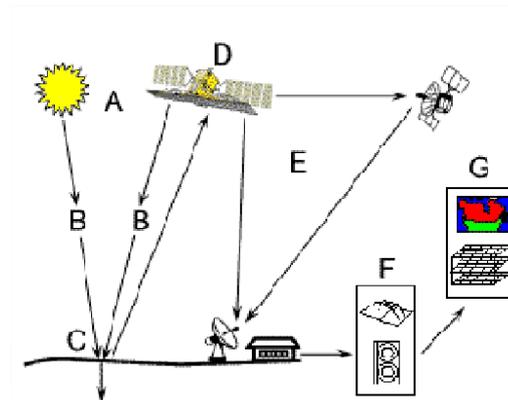


Figure 2. Les étapes du processus de Télédétection

Image aérienne

La photographie aérienne permet d'enregistrer les entités anthropiques et naturelles en constante évolution qui se trouvent à la surface de la Terre. Elle montre les entités comme les montagnes, les canyons et les basses plaines, les cours d'eau, de la source à l'embouchure; elle

révèle les ressources terrestres, comme les lacs, les forêts et la végétation; elle permet enfin de reconnaître les densités de population. Les photographies aériennes ont de nombreuses applications pratiques telles que la production de cartes, la planification urbaine et rurale, l'étude des impacts sur l'environnement, les actions civiles, l'évaluation des biens immobiliers et même la décoration murale.

En termes génériques, une image aérienne est une image prise des airs. Normalement, les images aériennes sont prises suivant la verticale, à bord d'un aéronef, à l'aide d'un appareil de prise de vues hautement précis.

Contrairement à une carte, les entités qui figurent sur une photographie aérienne ne sont pas généralisées ni exprimées sous forme de symboles. Les photographies aériennes enregistrent toutes les entités visibles à la surface de la Terre telles que vues d'en haut. Bien que les entités soient visibles, elles ne sont pas toujours faciles à identifier. On appelle photo-interprétation le processus d'étude et de collecte des renseignements requis pour identifier les diverses entités anthropiques et naturelles. Lorsque correctement interprétées, les photographies aériennes constituent une excellente source de données spatiales pour l'étude de l'environnement de la Terre.

Image médicale

Des millions d'images médicales sont produites chaque année dans le monde pour établir un diagnostic ou contrôler une action thérapeutique. Ces images proviennent principalement du tomodensitomètre à rayons X ou scanner, de l'imagerie par résonance magnétique ou IRM, de l'imagerie ultrasonore ou échographie, ou bien de l'imagerie de médecine nucléaire ou scintigraphie.

Les images médicales fournissent des informations sur la forme et le fonctionnement des organes du corps humain. Malheureusement, ces informations sont extrêmement difficiles à exploiter de manière quantitative et objective.

La conception d'algorithmes dédiés à l'analyse d'images médicales doit permettre d'optimiser leur exploitation, pour le plus grand bénéfice du patient et du médecin.

2.2. Prétraitements

La chaîne de traitement d'images, présentée dans la figure ci-dessous, regroupe les différents traitements qui peuvent être appliqués à une image avant de pouvoir l'interpréter. Nous pouvons voir qu'après la phase d'acquisition suit une phase de prétraitement. Cette phase sert à traiter les défauts de l'image due à l'acquisition. Par exemple, un temps d'exposition trop court peut être corrigé en rehaussant le contraste de l'image. Cette phase n'est pas toujours nécessaire si les conditions d'acquisitions de l'image sont bonnes.

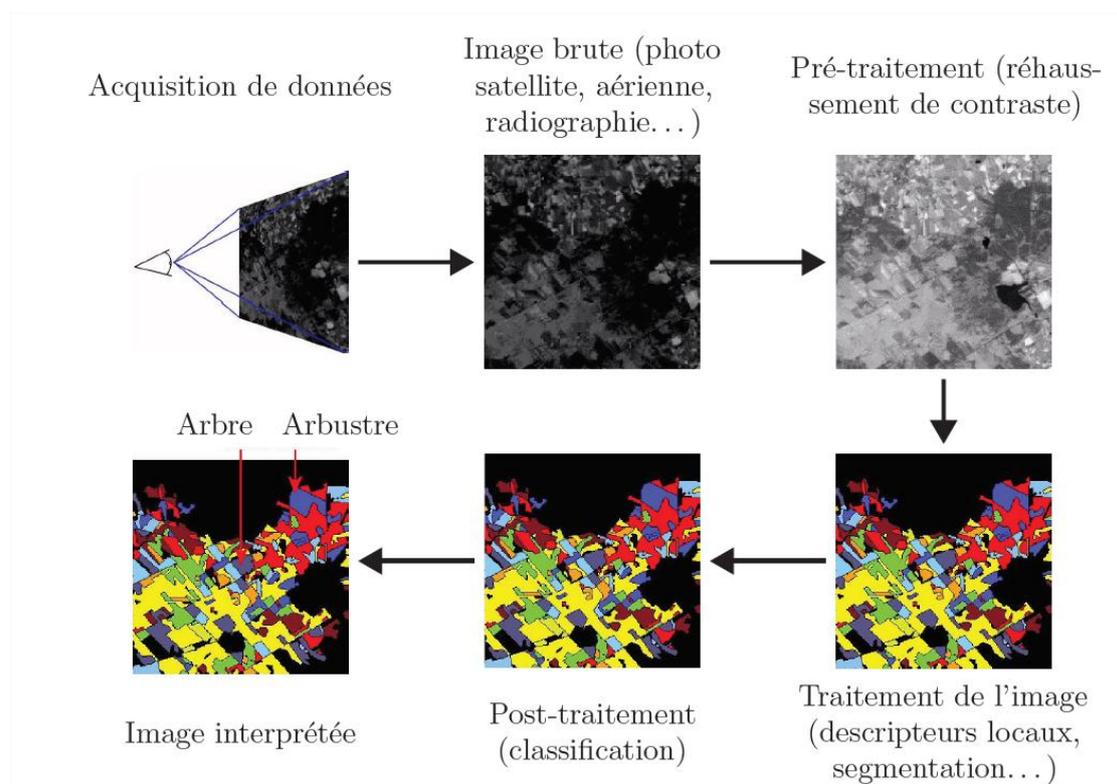


Figure 3. Une chaîne de traitement d'une image (images du ©LASTI).

En raison de la multitude de sources de bruits, et surtout de la multitude d'effets de ces bruits sur une image, il n'existe pas de technique de prétraitement générale, adaptée à toutes les situations. Il existe de nombreuses recherches permettant d'approcher ce prétraitement automatisé.

En effet, ces prétraitements peuvent parfois détériorer l'image plus qu'ils ne l'améliorent. Ainsi, les artefacts de l'image, tels que le bruit ou la variation de luminosité, correspondent à des problèmes auxquels les algorithmes d'interprétation doivent être robustes, au lieu de reposer sur un prétraitement qui corrigerait ces problèmes (1) (2) (3) (4).

2.3. Interprétation des images

Suit alors une phase de traitement de l'image qui va permettre de la rendre interprétable. Cette phase comporte, selon l'algorithme d'interprétation utilisé, une phase de segmentation ou bien une extraction de données caractéristiques calculées à partir de descripteurs locaux.

2.4. Post-traitement

Vient ensuite une phase de post-traitement qui va permettre d'obtenir l'image interprétée, c'est à dire une image comportant l'information sur les objets de l'image. Cette phase utilise généralement des méthodes de classification ou d'apprentissage afin d'interpréter les données extraites précédemment.

L'interprétation d'images est donc la combinaison du traitement et du post-traitement des images, ce qui en fait le maillon final de la chaîne de traitement d'images.

Un second point de vue de l'interprétation d'images, considère l'interprétation d'images comme un traitement à part entière. Ainsi, les images en entrée d'algorithmes ne subissent aucun prétraitement.

3. Algorithmes d'interprétation d'images

Parmi les différents algorithmes d'interprétation d'images, on peut distinguer trois types principaux. Certains algorithmes présentent un résultat sous forme d'image. C'est principalement le cas des algorithmes de localisation dont le résultat est le plus souvent une image contenant la frontière des objets localisés. D'autres algorithmes renvoient une donnée textuelle ou alphanumérique (position d'un objet dans l'image, liste d'éléments présents, valeurs numériques, vecteurs...).

C'est le cas des algorithmes dont le but est de prédire la classe d'un objet ou bien de compter le nombre d'objets présents dans une image. Enfin, des algorithmes hybrides vont renvoyer une image contenant des informations textuelles. C'est le cas des algorithmes qui sont capables de localiser un objet dans une image tout en prédisant sa classe. La valeur des pixels de l'image renvoyée par ce genre d'algorithme va contenir les deux types d'informations : la localisation de l'objet et sa classe.

Si les principaux objectifs des algorithmes d'interprétation d'images sont la reconnaissance et la localisation d'objets d'intérêt dans une image, il existe d'autres

algorithmes d'interprétation d'images tels que les algorithmes de détection, de suivi de cible ou bien ceux visant à faire du contrôle qualité sur des chaînes de fabrication.

Ces algorithmes sont développés à partir des algorithmes de reconnaissance et de localisation et sont souvent définis spécifiquement pour une application.

Cette partie vise à décrire et à formaliser le fonctionnement de plusieurs algorithmes classiquement utilisés en interprétation d'images (5). Des exemples d'algorithmes sont également présentés afin d'illustrer le propos par quelques cas concrets.

3.1. Les algorithmes non supervisés de localisation

Les algorithmes non supervisés de localisation vont permettre de localiser des objets dans une image ou une vidéo sans avoir de connaissance a priori sur ces objets d'intérêt. Dans ce cas, l'algorithme n'a pas recours à l'utilisation d'une base de connaissances afin d'apprendre à détecter les objets. L'avantage est que l'algorithme va pouvoir localiser tout type d'objet dans les images, mais en contrepartie, il n'a aucune information sur le type d'objet qui a été localisé. Ces algorithmes sont essentiellement basés sur la segmentation ou la détection de mouvement dans des vidéos.

Formalisation du problème

L'objectif des algorithmes de localisation non supervisés est de donner une localisation d'un ou plusieurs objets comme résultat en ne prenant en compte qu'une ou plusieurs images comme entrée de l'algorithme.

$$\text{localisation: } I \rightarrow Z_i \quad (1)$$

Avec, I l'image d'entrée et Z_i le résultat de localisation

Exemples

De nombreuses méthodes existent dans la littérature pour la localisation non supervisée, notamment d'objets en mouvement dans des vidéos (6) (7). Une méthode simple consiste à faire une différence temporelle, c'est-à-dire à soustraire deux images prises à des temps différents. Cette méthode pose cependant quelques problèmes lorsqu'il s'agit de détecter des objets non texturés.

Il est aussi possible de faire une estimation de l'avant-plan, c'est-à-dire le plan contenant les objets que l'on souhaite détecter, en soustrayant l'arrière-plan, avec une image

prise avant la détection (ne contenant pas d'objets) (8). Une binarisation de seuil permet alors de faire ressortir les objets de l'avant plan comme on peut le voir dans la figure ci-dessous. Suite à cette binarisation, il est possible de détecter des objets présents dans l'avant-plan et de les localiser facilement. Ce procédé est souvent utilisé dans le traitement de vidéos de scènes d'intérieur ainsi qu'en contrôle qualité par vision, car l'environnement y est contrôlé.

Discussion

On peut voir que les algorithmes de localisation peuvent fonctionner sans avoir recours à des informations a priori sur les objets à localiser mais ils ne peuvent évidemment pas permettre dans ce cas de cibler la détection d'un seul type d'objet.

Afin de pouvoir localiser certains objets d'intérêt, il faut avoir recours à une base de données permettant d'apprendre la nature de ces objets. Dans ce cas, on parle d'algorithme supervisé.

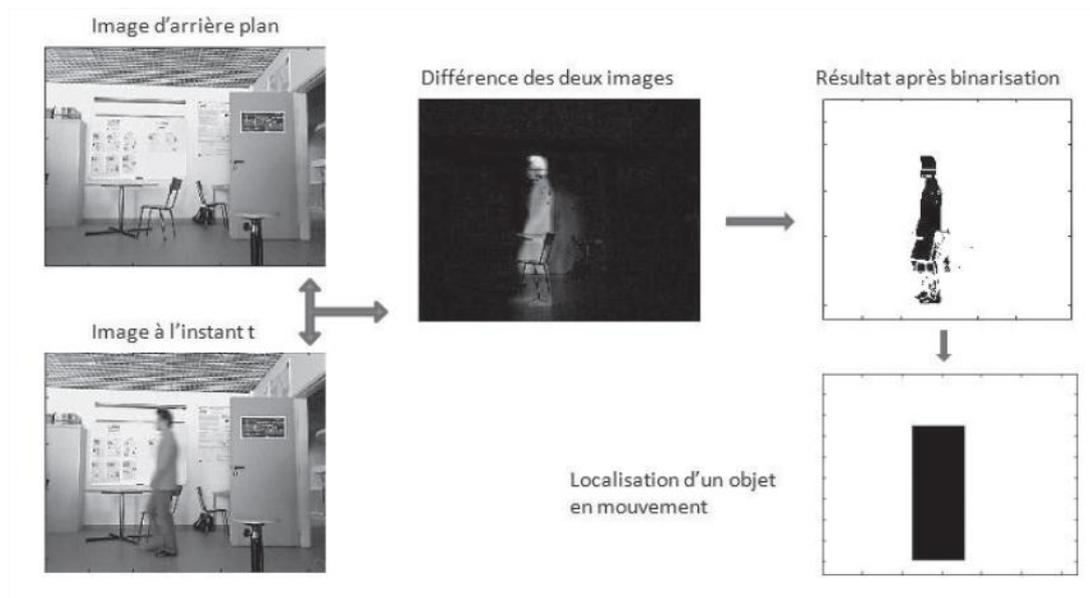


Figure 4. Localisation par estimation de l'avant plan

3.2. Les algorithmes de reconnaissance et de catégorisation

Les algorithmes de reconnaissance répondent au problème de la prédiction de la classe d'un ou plusieurs objets présents dans une image. La différence entre les algorithmes de reconnaissance et de catégorisation est minime. Elle se situe au niveau de la précision souhaitée dans les résultats. Ainsi, un algorithme de reconnaissance fera la différence entre deux voitures de marques différentes alors qu'un algorithme de catégorisation les affectera dans la même catégorie. En pratique, la formalisation du problème est identique pour la

reconnaissance et la catégorisation. Ces algorithmes ayant besoin d'une base de données pour fonctionner sont par nature supervisés.

Formalisation du problème

Les algorithmes de reconnaissance doivent, à partir d'une image contenant un objet, lui assigner une classe $y \in Y$. L'ensemble Y est défini à partir de la base de données B et chacune des classes Y est apprise empiriquement à partir des exemples. On peut donc résumer cela ainsi :

$$\text{Reconnaissance: } B, I \rightarrow y \quad (2)$$

Ces algorithmes ne sont pas capables d'identifier plusieurs objets à la fois. Si l'on souhaite reconnaître plusieurs objets dans une même image, il va falloir les reconnaître les uns après les autres. Pour cela, il faut donner en entrée un ensemble de localisations (provenant d'une localisation non supervisée préalable) en plus de l'image elle-même. On a alors ceci :

$$\text{Reconnaissance: } B, I, Z_i \rightarrow y_i \quad (3)$$

Les algorithmes de reconnaissance doivent faire face à deux challenges majeurs. D'une part, ils doivent être capables de faire la différence entre les classes apprises. D'autre part, ils doivent être capables de différencier les objets appartenant à l'ensemble d'apprentissage et les objets inconnus. Les sorties possibles de ce type d'algorithme sont alors les suivantes :

- $y \in Y$ quand l'algorithme reconnaît une des classes apprises,
- une décision *Autre*, quand l'algorithme reconnaît un objet n'appartenant pas à l'ensemble d'apprentissage,
- une décision *Ambigu*, quand l'algorithme reconnaît une des classes, mais n'arrive pas à décider laquelle.

$$Y^+ = Y \cup \textit{autre} \cup \textit{Ambigu} \quad (4)$$

Afin d'avoir plus ou moins confiance dans les résultats fournis par l'algorithme de reconnaissance, un paramètre μ peut accompagner la réponse de l'algorithme.

Ce paramètre est généralement compris dans l'intervalle $[0; 1]$, 1 représentant la confiance maximale dans les résultats de l'algorithme. De même, un paramètre λ peut être mis en entrée

de l'algorithme. Ce paramètre fonctionne alors comme un seuil et l'algorithme renvoie une réponse *Ambigu* si la confiance accordée au résultat est inférieure . Finalement, on a :

$$\text{Reconnaissance: } B, I, Z_i \rightarrow y_i, \mu_i \quad (5)$$

Ou

$$\text{Reconnaissance: } B, I, Z_i, \lambda \rightarrow y_i \quad (6)$$

L'algorithme peut également fournir un vecteur de résultats $\mu_{i_c} = \{\mu_c\}_{c \in Y}$, avec μ_c la probabilité d'appartenance de l'objet i à la classe c . Cela laisse ainsi la possibilité de choisir la classe ultérieurement, en prenant par exemple la classe ayant la probabilité maximum.

$$\text{Reconnaissance: } B, I, Z_i \rightarrow \mu_{i_c} \quad (7)$$

Exemples

Une méthode fréquemment utilisée en reconnaissance d'objets consiste à effectuer une reconnaissance sur des données locales dans l'image (9) (10), en prenant en compte leur répartition dans l'image. Pour cela, un détecteur de points d'intérêt est utilisé afin de localiser les zones de l'image qui vont être décrites. Suite à cette détection de points d'intérêt, les voisinages de ces points sont décrits au moyen d'un descripteur invariant. Un classifieur est ensuite utilisé sur les descripteurs dans le cadre des algorithmes de reconnaissance.

Afin de pouvoir effectuer une catégorisation, on peut également utiliser un « dictionnaire visuel » et une représentation en « patches » (11) (12). Pour cette méthode, on définit tout d'abord lors de l'apprentissage un dictionnaire de « mots visuels » obtenus par des méthodes de regroupement des vecteurs de descripteurs. On assigne ensuite chaque vecteur de l'image à un mot visuel, puis on étudie la répartition de ces mots visuels au moyen d'un histogramme indiquant le nombre de fois où les mots visuels apparaissent dans l'image. On peut voir le principe de l'utilisation de sacs de mots dans la figure ci-dessous. Enfin, un classifieur est utilisé sur la répartition des mots visuels afin d'obtenir la catégorisation.

Discussion

Comme nous avons pu le voir, les algorithmes de reconnaissance et de catégorisation peuvent utiliser de nombreuses étapes: détection de points d'intérêt, descripteurs, apprentissage... Ces algorithmes sont donc plus complexes que les algorithmes de localisation non supervisés vus

précédemment. Il est alors indispensable de pouvoir évaluer les algorithmes de reconnaissance afin de vérifier que l'enchaînement de ces différents composants soit correct.

De plus, le problème des algorithmes de reconnaissance est que leur utilisation se limite à un seul objet à reconnaître dans l'image. Dans le cas où plusieurs objets sont présents dans l'image, il faut alors faire appel aux algorithmes de localisation afin d'isoler les différents objets du reste de la scène présentée dans l'image.

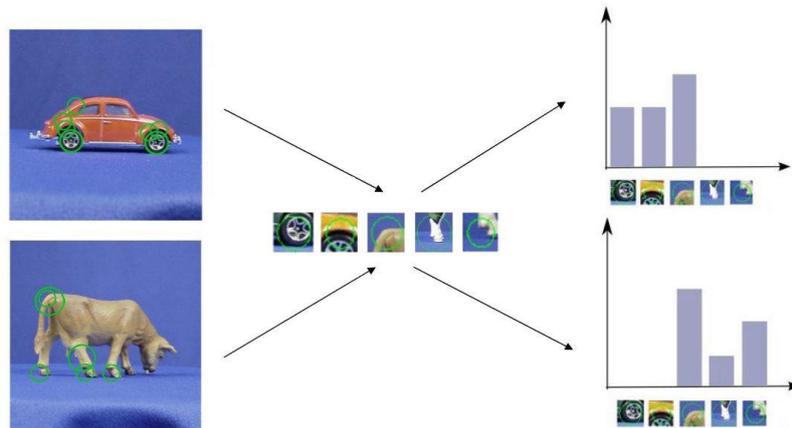


Figure 5. Exemple de reconnaissance utilisant des patches de l'objet (11) : chaque point d'intérêt est associé à un mot visuel, puis un histogramme indiquant la répartition des mots visuels dans l'image est calculé. La classe de l'objet est déduite de l'histogramme.

3.3. Les algorithmes supervisés de localisation

Comme nous l'avons vu auparavant, le but des algorithmes de localisation est de déterminer l'emplacement d'un ou de plusieurs objets dans une image. Les algorithmes supervisés se limitent généralement à la localisation d'un seul type de l'objet et vont utiliser une base de données disposant d'images du type d'objet à localiser. On peut cependant chercher à localiser plusieurs objets de la même classe dans une seule image comme on peut le voir dans la figure ci-dessous.



Figure 6. Localisation de véhicules comme Exemple de résultats de localisation supervisée

Formalisation du problème

Les algorithmes supervisés de localisation fournissent, à partir d'une image I , une liste de localisations d'objets $\{z_i\}$. Les objets que l'on souhaite localiser sont appris au préalable au moyen d'une base d'apprentissage B contenant des exemples d'images avec les objets d'intérêt.

$$\text{localisation: } B, I \rightarrow Z_i \quad (8)$$

L'algorithme peut renvoyer la confiance accordée dans la localisation de l'objet au moyen d'un paramètre. La décision peut alors être effectuée en dehors de l'algorithme. L'algorithme peut également utiliser un critère λ en entrée, permettant un seuillage de la fonction de décision afin de savoir si un objet est présent. Le nombre d'objets localisés dans l'image I dépend de cette fonction de décision et est donc lié au seuil λ . On peut donc résumer cela ainsi :

$$\text{localisation: } B, I \rightarrow \{Z_i, \mu_i\} \quad (9)$$

Ou

$$\text{localisation: } B, I, \lambda \rightarrow \{Z_i\} \quad (10)$$

Exemples

Un exemple d'algorithme supervisé de localisation d'objets dans une image est l'algorithme utilisé dans (13) (14) (15), dans lequel une fenêtre de taille fixe parcourt l'ensemble de l'image mise à différentes échelles. Une classification est ensuite effectuée afin de dire si un objet de la classe apprise est présent dans la fenêtre ou non.

Ainsi, dans (13) (14), la classification a lieu avec un réseau de neurones tandis que dans (15), on utilise un histogramme des gradients orientés, similaire au descripteur SIFT, combiné avec un séparateur à vaste marge (SVM). Toutes les fenêtres où l'on a trouvé un objet sont ensuite fusionnées afin d'obtenir une localisation de tous les objets dans l'image par des boîtes englobantes.

Une autre possibilité est de déterminer la localisation par reconnaissance de points d'intérêt comme le font Jurie et coll. Dans (16). Ainsi, des points d'intérêt sont calculés sur l'ensemble de l'image. Au voisinage de chaque point d'intérêt, on essaye de reconnaître un objet préalablement appris. Une boîte englobante entourant les points d'intérêt correctement reconnus est calculée. Nous pouvons voir en haut de la figure ci-dessous, l'image apprise de

l'objet à localiser et son contour; Dans la partie inférieure sont présentés les points d'intérêt extraits d'une image test et la localisation des objets qui en découle.

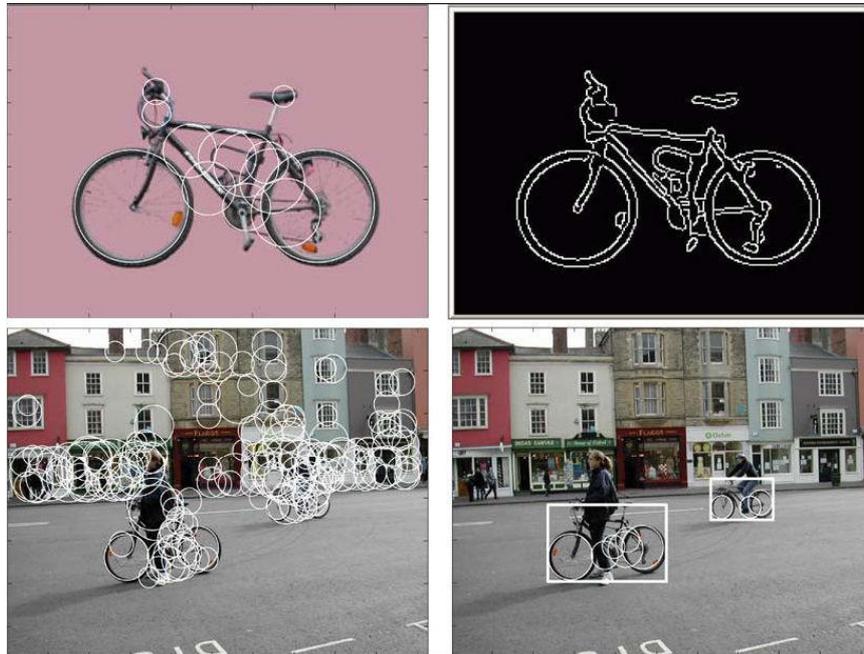


Figure 7. Exemples de localisation par reconnaissance de points d'intérêt (image tirée de (16)).

Discussion

Nous avons vu que les algorithmes supervisés de localisation apportent plus d'informations que les algorithmes de reconnaissance. En effet, on y retrouve des techniques assez proches de celles utilisées en reconnaissance: l'utilisation de descripteurs et la classification en sont des étapes essentielles. Cependant, afin d'effectuer une localisation, d'autres étapes sont ajoutées comme le parcours d'une fenêtre dans l'image ou l'étude de l'agencement spatial des points d'intérêt (16) (17) (18) (19).

Comme nous venons de le voir, les algorithmes supervisés de localisation sont des algorithmes complexes, utilisant différents composants.

3.4. Les algorithmes de localisation multi-classes

On qualifiera d'algorithme de localisation multi-classes un algorithme capable à la fois de localiser et de reconnaître un ou plusieurs objets de classes différentes présents dans une image.

Formalisation du problème

Le but d'un algorithme d'interprétation d'image est de fournir à la fois la localisation et la classe d'un ou plusieurs objets dans une image I . Cet algorithme est forcément supervisé et utilise donc une base de données B pour l'apprentissage des classes des objets à localiser dans l'image. On a alors :

$$\textit{localisation multi - classes: } B, I \rightarrow \{y_i, z_i\} \quad (11)$$

L'algorithme peut également fournir un paramètre pour chaque objet localisé, indiquant la confiance que l'on a dans le résultat obtenu pour cet objet. On peut également fournir en entrée un paramètre λ qui fonctionnera comme un seuil sur la fonction de décision. Au final, on a donc ceci :

$$\textit{localisation multi - classes: } B, I \rightarrow \{y_i, z_i, \mu_i\} \quad (12)$$

Ou

$$\textit{localisation multi - classes: } B, I, \lambda \rightarrow \{y_i, z_i\} \quad (13)$$

Exemple

Une méthode simple consiste à utiliser plusieurs fois un algorithme supervisé de localisation, avec une classe apprise différente à chaque fois. Il est également possible d'utiliser un classifieur qui va apprendre directement plusieurs classes. Ceci est réalisé, par exemple, par Torralba et coll. dans (20). Un algorithme de boosting est utilisé afin de faire une localisation robuste. Pour pouvoir reconnaître et localiser plusieurs classes tout en réduisant le nombre de classifieurs nécessaires, un apprentissage est fait en partageant les données issues des différentes classes. Cela permet de faire de l'interprétation d'images avec différentes classes d'objets, mais également avec différentes vues d'un même objet. On peut voir sur la figure ci-dessous des images correctement interprétées.

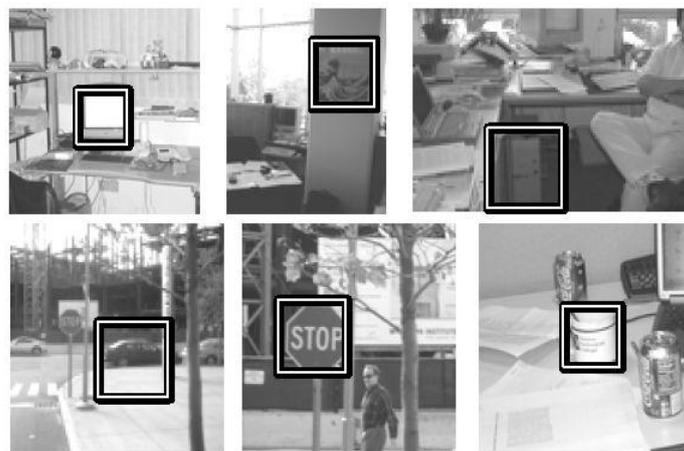


Figure 8. Exemple d'interprétation d'images (images tirées de (20))

Discussion

Les algorithmes de localisation multi-classes sont les algorithmes les plus complets. Ils sont capables de fournir à la fois la localisation et la classe de plusieurs objets dans une image. C'est à l'évaluation de tels algorithmes que nous nous sommes intéressés, ainsi qu'à l'évaluation d'algorithmes plus simples de localisation ou de reconnaissance qui en sont des briques de bases.

3.5. Les autres algorithmes d'interprétation d'images

Les algorithmes que nous avons présentés jusqu'à présent sont les algorithmes les plus usuels en interprétation d'images. Cependant, il en existe d'autres qui vont extraire des informations sur les objets contenus dans une image. Ces algorithmes sont généralement dérivés des algorithmes présentés précédemment.

Détection

Nous appellerons algorithmes de détection, un algorithme dont le but est de déterminer la présence ou l'absence d'un objet dans une image. Ces algorithmes peuvent être supervisés ou non, multi-classes ou non.

Dans le cas non supervisé, l'algorithme de détection donne en résultat une réponse de type *Vrai* ou *Faux* indiquant si un objet est présent dans l'image.

$$Detection: I \rightarrow (vrai|faux) \quad (14)$$

Dans le cas supervisé, l'algorithme de détection ne doit renvoyer *Vrai* que si un objet d'une classe apprise, à partir des images de la base d'apprentissage B , est présent dans l'image. De plus, l'algorithme peut utiliser un paramètre λ ou μ , comme précédemment, afin d'indiquer le degré de confiance que l'on a dans le résultat. On peut donc résumer cela ainsi :

$$Dectection: B, I \rightarrow (vrai|faux), \mu \quad (15)$$

Ou

$$Dectection: B, I, \lambda \rightarrow (vrai|faux|Ambigu) \quad (16)$$

Ce genre d'algorithme peut, par exemple, être utilisé afin de détecter la présence d'un individu afin de réguler l'éclairage ou le chauffage dans une pièce. Pour certains algorithmes,

il n'est pas nécessaire de connaître la position de l'individu. Pour ce faire, on peut alors utiliser un algorithme de reconnaissance et renvoyer le résultat *Vrai* uniquement si un individu est reconnu dans l'image.

Suivi

Les algorithmes de suivi sont des algorithmes dérivés des algorithmes de localisation. En effet, leur but va être de localiser un ou plusieurs objets dans une vidéo et de le suivre tout au long de son déplacement (voir la Figure ci-dessous). Le suivi est généralement plus robuste que la localisation. En effet, le fait de disposer d'une vidéo permet d'avoir plus d'informations qu'une image seule. Ainsi, on peut augmenter la confiance que l'on a à propos d'un résultat de localisation s'il est proche du résultat sur les images précédentes.



Figure 9. Exemple de suivi d'un véhicule dans un parking

Le suivi d'objets permet entre autres d'étudier le comportement d'individus ou des trajectoires anormales d'objets. Par exemple, le suivi embarqué dans un véhicule va permettre d'alerter le conducteur d'un comportement anormal d'un objet (un autre véhicule) dans son champ de vision.

4. Conclusion

Nous avons vu que l'interprétation regroupe un grand nombre de méthodes répondant chacune à une problématique différente. En effet, le terme même d'interprétation d'image est très mal défini par la communauté. Nous avons donc essayé de proposer une définition générale de l'interprétation. Nous avons également proposé une classification de plusieurs algorithmes d'interprétation d'images particuliers, notamment la localisation et la reconnaissance : les algorithmes de reconnaissance cherchent à savoir quels types d'objets

sont présents dans une image, alors que les algorithmes de localisation cherchent à savoir où ils se situent.

Devant ce foisonnement, le problème de l'évaluation de l'interprétation devient primordial et soulève de nombreuses questions. Il faut en effet se demander dans quelle mesure nous sommes, par exemple, capables de mettre en évidence qu'une méthode de localisation ou de reconnaissance est meilleure qu'une autre.

Le chapitre suivant, présente un état de l'art des méthodes de classification. Dans la suite de ce mémoire, le problème de classification est restreint en un problème d'optimisation.

Chapitre 2
Classification et méthodes de
classification

1. Introduction

La classification des données consiste à identifier les groupes homogènes dans un ensemble de données multidimensionnelles en se basant sur une mesure de similarité. Elle joue un rôle prépondérant dans la reconnaissance des formes et l'apprentissage automatique (16) et constitue l'étape principale dans plusieurs processus en intelligence artificielle (17). Ces algorithmes sont utilisés dans multiples applications, telles que la classification des images (18) (19), la quantification des images couleurs (20), la fouille de données (21), la compression (22), etc.

Ce chapitre étudie en détail le problème de classification. Dans un premier lieu nous allons définir et discuter la problématique, les concepts et les approches de classification. Dans un second lieu nous définirons la classification des images. Finalement, nous citons les techniques des plus populaires de classification des images.

2. Définition formelle de la classification

Dans cette section nous définissons les termes et les concepts basiques de classification, qui seront abordés au cours de ce projet.

- **Donnée:** ou un point, c'est une structure abstraite ou physique d'objets, qui devront être groupés par un algorithme de classification.
- **Caractéristique:** coordonnée ou attribut, est un composant individuel d'un point.
- **Groupe:** est un ensemble de points ayant des caractéristiques similaires tel que, deux points appartenant à deux groupes différents sont dissimilaires (23).
- **Mesure de distance:** est une métrique permettant d'évaluer la similarité entre les données (23). Formellement, le problème de classification peut être défini comme suit (24):

Soit l'ensemble des données $Z = \{Z_1, Z_2, \dots, Z_{N_p}\}$ tel que Z_p est un point dans l'espace multidimensionnel de caractéristiques, et N_p est le nombre de points dans Z , d'où la classification de Z en K groupes $\{C_1, C_2, \dots, C_K\}$ doivent satisfaire les conditions suivantes :

- Chaque point devrait être assigné à un groupe, i.e.

$$\bigcup_{k=1}^K C_k = Z \quad (17)$$

- Chaque groupe contient au moins un pixel, i.e.

$$C_k \cap C_{kk} = \emptyset \text{ Ou } k \neq kk \quad (18)$$

3. Mesures de ressemblance

Tout système ayant pour but d'analyser ou d'organiser automatiquement un ensemble de données doit utiliser, sous une forme ou une autre, un opérateur capable d'évaluer précisément les ressemblances ou les dissemblances qui existent entre ces données.

La notion de ressemblance (ou proximité) a fait l'objet d'importantes recherches dans des domaines extrêmement divers. Pour qualifier cet opérateur, plusieurs notions comme la similarité, la dissimilarité ou la distance peuvent être utilisées. Dans le cas d'une similarité, le lien entre deux individus sera d'autant plus fort que sa valeur est grande. Quant à la dissimilarité, le lien sera d'autant plus fort que sa valeur est petite.

Tel qu'il a été mentionné, la classification consiste à identifier un regroupement, ou des groupes dans un espace de données multidimensionnel en se basant sur une mesure de similarité. D'où, les mesures de similarité sont des composantes fondamentales dans la plus part des algorithmes de classification (23).

Le choix le plus populaire d'une métrique permettant d'évaluer la similarité est l'utilisation des mesures de distance. La *distance Euclidienne* est la plus employée, elle est définie comme suit

$$d(z_u, z_w) = \sqrt{\sum_{j=1}^{N_d} (z_{u,j} - z_{w,j})^2} = \|z_u - z_w\| \quad (19)$$

La distance Euclidienne est considérée comme étant un cas particulier (d'où $\alpha = 2$) de la métrique de Minkowski (23) qui est définie comme suit

$$d^\alpha(z_u, z_w) = (\sum_{j=1}^{N_d} (z_{u,j} - z_{w,j})^\alpha)^{1/\alpha} = \|z_u - z_w\|^\alpha \quad (20)$$

Dans le cas où $\alpha = 1$ elle est dite Distance de Manhattan (17).

4. Techniques classiques de la classification

Une bibliographie de référence dans le domaine de la vision par ordinateur propose une classification des méthodes de classification. Nous nous en inspirons pour présenter les

deux grandes familles de techniques de classification les plus populaires connues par la classification partitionnaires et hiérarchique (25) (26). Dans ce qui suit une vue d'ensembles des deux techniques est présentée avec une discussion sur les algorithmes hiérarchiques et partitionnaires les plus populaires.

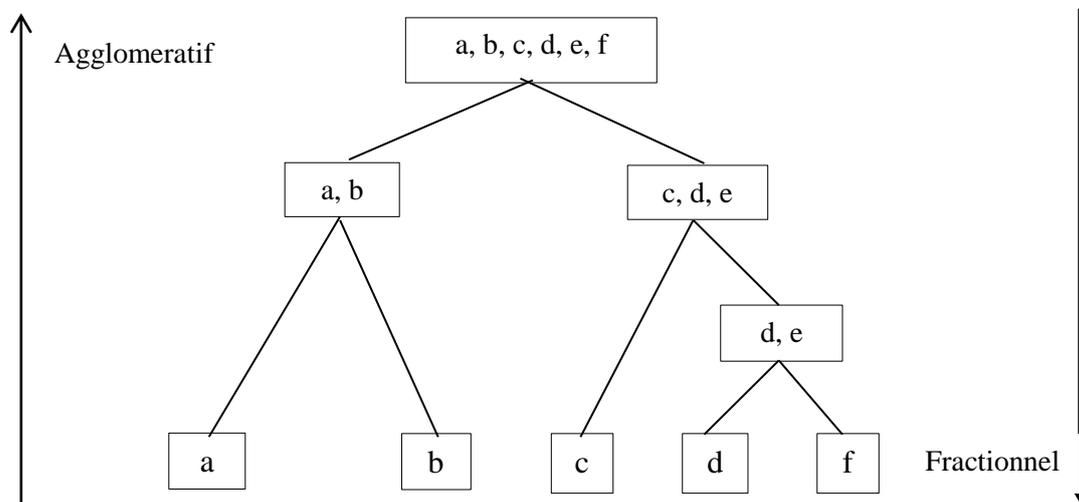


Figure 10. Exemple illustrant la classification fractionnelle et agglomérative

4.1. Algorithmes de classification hiérarchiques

Les algorithmes de cette catégorie permettent de générer un arbre de classification (ou dendrogramme) en utilisant des heuristiques ou des techniques de fusion et de division (17). Un arbre de classification est défini par « un arbre indiquant une séquence de regroupements tel que chaque regroupement est une partition de l'ensemble de données » (26). Les algorithmes utilisant la division pour générer l'arbre de classification sont appelés *Fractionnels*. D'un autre côté, les algorithmes utilisant la fusion pour générer l'arbre de classification sont appelés *Agglomératifs*. Les algorithmes hiérarchiques fractionnels supposent initialement que tous les points appartiennent à un seul groupe. Puis, une division est appliquée aux groupes à chaque étape jusqu'à atteindre la meilleure classification. Contrairement aux algorithmes hiérarchiques fractionnels, les algorithmes hiérarchiques agglomératifs supposent à l'étape initiale que chaque point est assigné à une classe. Ensuite tous les groupes similaires seront fusionnés dans un seul groupe. Cette étape est répétée jusqu'à ce que tous les points soient assignés à un seule groupe (27). Plusieurs algorithmes hiérarchiques agglomératifs ont été proposés dans la littérature qui se diffère par la façon avec laquelle les deux groupes les plus similaires sont calculés. Les deux catégories les plus

populaires sont les algorithmes *Single link* (28) et *Complete link* (29). Les algorithmes *Single Link* fusionnent les groupes dont la distance entre leurs points les plus proches est minimale. Tandis que les algorithmes *Complete Link* fusionnent les groupes dont la distance entre leurs points les plus distants est minimale (27). En général, les algorithmes *Complete Link* génèrent des groupes compacts, tandis que les algorithmes *Single link* génèrent des groupes étirés. Ainsi, les algorithmes *Complete Link* sont généralement beaucoup plus performants que les algorithmes *Single link* (23). Il existe une autre catégorie populaire d'algorithmes hiérarchiques agglomératifs appelée *centroïd method* (29). Ces derniers fusionnent les groupes dont la distance entre leurs centres (centroïdes) est minimale. Un inconvénient de cette catégorie est que les *Centroids algorithms* considèrent qu'un groupe trop petit est perdu lorsqu'il est fusionné avec un groupe très large (27). D'autres détails sur les algorithmes hiérarchiques de segmentation peuvent être trouvés dans (30).

En général, les algorithmes de segmentation hiérarchique présentent les inconvénients suivants (31) :

- Le nombre de groupes n'est pas forcément défini a priori, et
- Ils sont indépendants des conditions initiales.

Néanmoins, les algorithmes de segmentation hiérarchique souffrent des inconvénients suivants :

- Ils sont coûteux en termes de complexité (le temps de complexité est de $O(N_p^2 \log N_p)$ et l'espace de complexité est de $O(N_p^2)$ (27)). Ce qui est dispendieux en cas où on est confronté à un nombre élevé de points.
- Ils sont statiques, i.e. les points sont assignés à un groupe ne peuvent pas être assignés à d'autres groupes.
- Ils peuvent se montrer incapables de séparer les groupes qui se chevauchent en cas de manque d'informations sur la forme globale ou la taille des groupes.

4.2. Techniques de classification partitionnaire

Les algorithmes de classification partitionnaire divisent l'ensemble de données en un certain nombre de groupes. Le but de ces algorithmes est de minimiser un critère spécifique (telle que la fonction d'erreur quadratique), d'où ils peuvent être traités de même qu'un

problème d'optimisation. Ces problèmes sont le plus souvent combinatoires de type NP-Complet (26). Les avantages des algorithmes hiérarchiques sont des inconvénients des algorithmes partitionnaires et vice versa. En raison de leurs avantages, les algorithmes partitionnaires sont plus populaires que les techniques hiérarchiques en reconnaissance des formes (32), en particulier, ce mémoire s'intéresse aux techniques partitionnaires. Ces derniers sont généralement des algorithmes itératifs qui convergent vers un optimum local (16). En employant la forme générale la classification itérative utilisée par Harmely et Elkan en 2002, les étapes d'un algorithme de classification itératif peuvent être données comme suit :

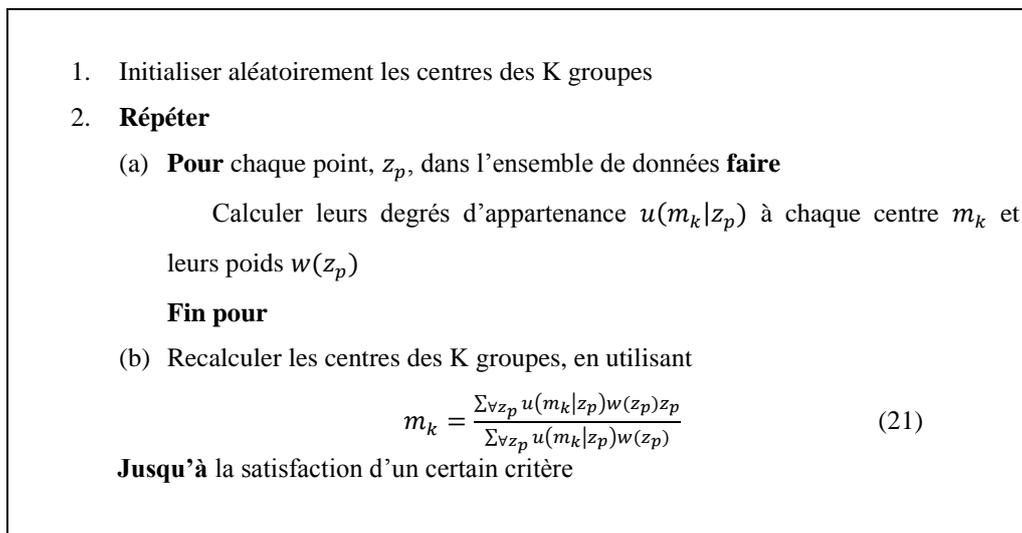


Figure 11. Algorithme de classification partitionnaire

- $u(m_k|z_p)$ est la fonction d'appartenance qui quantifie l'appartenance du point z_p au groupe k . La fonction d'appartenance $u(m_k|z_p)$, doit satisfaire les contraintes suivantes :
 - 1.** $u(m_k|z_p) \geq 0, p = 1, \dots, N_p$ et $k = 1, \dots, K$
 - 2.** $\sum_{k=1}^K u(m_k|z_p) = 1, p = 1, \dots, N_p$
- Les algorithmes déterministe de classification utilisent un degré d'appartenance déterministe (i.e. $u(m_k|z_p) \in \{0,1\}$), tandis que les algorithmes flous de segmentation utilisent une fonction flexible d'appartenance (i.e. $u(m_k|z_p) \in [0,1]$) (16).

- La fonction de poids, $w(z_p)$, dans l'équation (5) définit le degré d'influence du point z_p dans le calcul des centres dans l'itération suivante, tel que $w(z_p) > 0$ (16). La fonction du poids a été proposée par (33).
- Plusieurs critères d'arrêt peuvent être utilisés dans un algorithme itératif de classification, par exemple :
 1. Arrêter quand les valeurs des centres deviennent plus petites qu'une valeur définie à l'initialisation.
 2. Arrêter quand la quantification de l'erreur est suffisamment petite, ou
 3. Arrêter quand un nombre maximal d'itérations est excédé.

Dans ce qui suit, nous allons décrire les algorithmes de segmentation itératifs les plus populaires tout en considérant les fonctions d'appartenance et de poids de l'équation (5).

L'Algorithme K-means

L'algorithme K-means est l'un des algorithmes partitionnaires itératifs les plus populaires et les plus utilisés (34). Il se base sur l'optimisation de la fonction suivante :

$$J_{K\text{-means}} = \sum_{k=1}^K \sum_{\forall z_p \in C_k} d^2(z_p, m_k) \quad (22)$$

D'où, l'algorithme K-means minimise la distance intra-classes (16). L'algorithme K-means commence par K groupes (les valeurs initiales des groupes sont sélectionnées aléatoirement ou dérivées à partir des informations connues a priori). Puis, chaque point de l'ensemble des données est assigné au groupe auquel il est le plus proche (i.e. le centre le plus proche). Finalement, les centres sont recalculés conformément aux points récemment assignés. Ce processus est réitéré jusqu'à ce que la convergence soit atteinte.

Les fonctions de poids et d'appartenance du K-means sont définies comme par

$$u(m_k | z_p) = \begin{cases} 1 & \text{si } d^2(z_p, m_k) = \arg \min_k \{d^2(z_p, m_k)\} \\ 0 & \text{Sinon} \end{cases} \quad (23)$$

$$w(z_p) = 1 \quad (24)$$

D'où, K-means utilise une fonction déterministe d'appartenance et une fonction constante de poids, ainsi, tous les points sont d'une importance identique (16).

Les avantages majeurs de l'algorithme K-means sont les suivants (27):

- Il est très facile à implémenter, et
- Son temps est de $O(N_p)$, ce qui rends l'exécution de l'algorithme un peu lente en cas d'un nombre important de données.

Cependant, l'algorithme K-means présente quelques inconvénients (35) :

- Il dépend des données,
- Il dépend fortement des conditions initiales, et
- Le nombre de groupes doit être défini à priori.

L'algorithme K-medoids

Dans l'algorithme K-medoids (36), à chaque groupe correspond un point parmi les points représentatifs dans le groupe, ce point étant le plus proche du centre. L'algorithme (37) commence par un ensemble initial de ce qu'on appelle « medoids », et remplace un des medoids par un non-medoids, cela améliore la distance totale du regroupement résultant. Il est important de noter que cet algorithme est approprié à un ensemble petit de données (23).

L'objectif de cet algorithme est de déterminer un point représentatif (medoid) pour chaque groupe, afin de trouver les points les plus centralement localisés dans les groupes. L'algorithme commence par la sélection d'un point comme point représentatif pour chacun des k groupes. Puis, chaque point non sélectionné est groupé avec le point représentatif le plus similaire. L'algorithme permute les points représentatifs avec les points non sélectionnés jusqu'à ce que tous les points soient qualifiés comme medoid. Il est nettement clair que l'algorithme K-medoids est coûteux du fait qu'il doit comparer un point avec tous les points de l'ensemble des données (37).

L'algorithme Fuzzy C-means

La version floue du K-means, dite C-means flou ou Fuzzy C-means (FCM) (appelée aussi K-means flou ou fuzzy K-means), a été proposé par Bezdek (38) (39). L'avantage qu'offre FCM est qu'il assigne chaque point à chaque groupe avec un certain degré d'appartenance (i.e. segmentation floue). Cela est plus approprié pour des applications réelles d'où la présence de chevauchements entre les groupes dans l'ensemble de données. Le but du FCM est d'optimiser la fonction suivante :

$$J_{FCM} = \sum_{k=1}^K \sum_{p=1}^{N_p} u_{k,p}^q d^2(z_p, m_k) \quad (25)$$

Où q est connu par « fuzziness exponent », avec $q \geq 1$. L'augmentation de la valeur de q rend l'algorithme plus flou ; $u_{k,p}$ est la valeur d'appartenance pour le $p^{i\grave{e}me}$ point dans le $k^{i\grave{e}me}$ groupe en satisfaisant les contraintes suivantes :

1. $u_{k,p} \geq 0, p = 1, \dots, N_p$ et $k = 1, \dots, K$
2. $\sum_{k=1}^K u_{k,p} = 1, p = 1, \dots, N_p$

Les fonctions d'appartenance et de poids pour le FCM sont définies par (16)

$$u(m_k | z_p) = \frac{\|z_p - m_k\|^{-2/(q-1)}}{\sum_{k=1}^K \|z_p - m_k\|^{-2/(q-1)}} \quad (26)$$

$$w(z_p) = 1 \quad (27)$$

D'où, FCM utilise une fonction d'appartenance souple et une fonction constante de poids. En général, FCM est plus performant que K-means (17) et est moins affecté par la présence d'incertitudes dans l'ensemble de données (40). Cependant, tel le K-means FCM requiert l'initialisation du nombre de groupes à l'avance. De plus, il peut converger vers un optimum local (23).

Krishnapuram et Keller (41) (42) propose un « *possibilistic clustering algorithm* », appelé *possibilistic C-means*. Possibilistic clustering est similaire à la segmentation floue; la différence principale est que dans le *possibilistic clustering* les valeurs d'appartenance peuvent ne pas être additionnées en une seule (27). *Possibilistic C-means* se montre performant dans les cas où l'ensemble de données contient du bruit. Néanmoins, il a quelques inconvénients suivant (27) :

- Il est susceptible de produire des groupes coïncidents,
- Il impose à l'utilisateur de spécifier le nombre de groupes à l'avance,
- Il peut converger vers un optimum local, et
- Il dépend des conditions initiales.

L'algorithme Gaussian Expectation-Maximization

L'algorithme *Expectation-Maximisation (EM)* est un autre algorithme populaire de classification (43) (44) (45). EM est utilisé pour l'estimation des paramètres dans le cas de données manquantes ou inconnues (17). EM partitionne l'ensemble des données en groupes par la détermination d'une mixture de *Gaussians fitting* de l'ensemble de données. A chaque Gaussien correspondent deux matrices: une pour la moyenne et l'autre pour la covariance. Le critère à optimiser dans EM a été définie par (16) est

$$J_{EM} = - \sum_{p=1}^{N_p} \log(\sum_{k=1}^K p(z_p | m_k) p(m_k)) \quad (28)$$

Où $p(z_p | m_k)$ est la probabilité de z_p étant donné qu'elle est générée par une distribution Gaussienne avec le centre m_k , et $p(m_k)$ est la probabilité précédente du centre m_k .

Les fonctions d'appartenance et de poids pour EM sont définies par (16)

$$u(m_k | z_p) = \frac{p(z_p | m_k) p(m_k)}{p(z_p)} \quad (29)$$

$$w(z_p) = 1 \quad (30)$$

D'où, EM utilise une fonction souple d'appartenance et une fonction constante de poids. L'algorithme commence par une estimation initiale des paramètres. Puis, une étape de prévision est effectuée où les valeurs connues de données sont utilisées pour calculer les valeurs prévues des données inconnues (17). Cette étape est suivie par des étapes de maximisation où les valeurs connues et prévues des données sont utilisées pour générer une nouvelle estimation des paramètres. Les étapes de prévision et de maximisation sont répétées jusqu'à atteindre la convergence.

Les résultats obtenus par Veenman, Reinders et Backer (46) et Hamerly (17) proposent une étude comparative entre EM et K-means. De plus, d'autres chercheurs ont prouvé que EM est moins performant pour un ensemble de données de dimension élevée, ceci est dû aux problèmes de précision numérique. De plus, EM dépend de l'estimation initiale des paramètres (17) (27) et nécessite la connaissance du nombre de groupes à l'avance. De surcroît, EM suppose que la densité de chaque groupe est Gaussienne, ce qui n'est pas toujours vrai.

L'algorithme k-Harmonic Means

Zhang et ses collègues (47) (33) ont proposé un nouvel algorithme appelé « K-Harmonic Means » (KHM), avec des résultats prometteurs. Dans KHM la moyenne harmonique de la distance de chaque centre de groupe avec tous les points est calculée. En conséquence, les centres de groupe sont mis-à-jour. KHM doit optimiser la fonction suivante :

$$J_{KHM} = \sum_{p=1}^{N_p} \frac{K}{\sum_{k=1}^K \frac{1}{\|z_p - m_k\|^\alpha}} \quad (31)$$

Où α est un paramètre initialisé par l'utilisateur, typiquement $\alpha \geq 2$. Les fonctions d'appartenance et de poids du KHM sont (16)

$$u(m_k | z_p) = \frac{\|z_p - m_k\|^{-\alpha-2}}{\sum_{k=1}^K \|z_p - m_k\|^{-\alpha-2}} \quad (32)$$

$$w(z_p) = \frac{\sum_{k=1}^K \|z_p - m_k\|^{-\alpha-2}}{(\sum_{k=1}^K \|z_p - m_k\|^{-\alpha})^2} \quad (33)$$

Ainsi, KHM a une fonction souple d'appartenance et une fonction variable de poids. KHM assigne les poids les plus élevés aux points les plus loin de tous les centres afin d'aider ces derniers à couvrir l'ensemble entier des données (16). Contrairement au K-means, KHM est moins sensible aux conditions initiales. Les expériences menées par Zhang et ses collègues (47), Zhang (33), et Hamerly et Elkan (16) prouvent que KHM est plus performant que K-means, FCM, et EM.

4.3. Les algorithmes de classification basés sur la densité

Cette catégorie est connue le plus souvent par leurs acronyme originale : « Density-Bases clustering » considère les groupes comme des régions denses de points dans l'espace de données, ces dernières sont séparées par des régions de densité faible. Le but principal d'une approche basée sur la densité est de trouver les régions de densité faible et celle de densité élevée, de telles sortes que les régions de densités élevées doivent être séparées par des régions de densité faibles. Ces approches peuvent rendre facile la découverte des groupes arbitraires. La technique la plus répandue de cette approche consiste à diviser et à transformer un espace de grande dimension en une grille d'unités basées sur la densité. Relativement, les

unités de densités élevées correspondent aux centres de groupes, tandis que les unités de densité faible correspondent aux contours qui séparent entre les régions (48).

Il existe plusieurs algorithmes de classification basée sur la densité. Les deux algorithmes les plus populaires sont : DBSCAN (49) qui s'appuie sur la notion de base sur la densité des groupes, qui est désignée pour découvrir les groupes arbitraires et aussi ayant la capacité de traiter le bruit. La tâche principale de cet algorithme est l'identification de classe, i.e. le regroupement des points en sous-classes significatives. Dans (50) un autre algorithme de classification basé sur la densité, DENCULE, est proposé. Cet algorithme introduit une nouvelle approche de regroupement de grandes bases de données multimédia.

4.4. Les algorithmes de classification basés sur les grilles

Récemment plusieurs algorithmes ont été conçus pour des données spatiales, connus par le nom « *Grid-Bases Algorithms* ». Ces algorithmes quantifient l'espace en un nombre infini de cellules. Par conséquent, toutes les opérations seront effectuées sur l'espace quantifié. STING (Statistical Information Grid-based method) (51) est un algorithme représentatif de cette catégorie. Il divise une zone spatiale en des cellules rectangulaires en utilisant une structure hiérarchique. STING part avec un ensemble de données et calcule les paramètres statistiques (tels que la moyenne, la variance, minimum, maximum et le type de la distribution) de chaque caractéristique numérique des points à l'intérieur des cellules. Puis il génère une structure hiérarchique de la grille des cellules afin de représenter l'information de classification à différents niveaux. Se basant sur cette structure STING permet l'utilisation des informations de classification afin de chercher les incertitudes ou l'assignation efficace d'un nouveau point aux groupes. WaveCluster (52) est l'algorithme le plus récent proposé dans la littérature. Il est basé sur les techniques de traitement du signal (la transformation par ondelettes) pour convertir les données spatiales en un domaine fréquentiel. Plus spécifiquement, il réduit l'ensemble de données en imposant une structure de grille multidimensionnelle sur l'espace de données (53). Chaque grille de cellule récapitule les informations d'un groupe de points. Puis utilise la transformation par ondelettes pour transformer l'espace de caractéristiques original. Dans la transformation par ondelettes, la convolution avec une fonction appropriée permet d'obtenir un nouvel espace où les groupes naturels deviennent distinguables. Ainsi, il est possible d'identifier les groupes par la recherche de régions denses dans le domaine transformé. Aucune connaissance préalable sur le nombre de groupes n'est nécessaire dans cet algorithme de classification.

4.5. Techniques basées sur la population

Les efforts scientifiques ont rendu possible l'adaptation du problème de classification à un problème d'optimisation. Cette adaptation a offert la possibilité d'appliquer les algorithmes basés sur la population dédiés à l'optimisation en classification afin de déterminer la classification optimale de l'ensemble de données.

Dans la littérature, les méthodes d'optimisation sont réparties en deux grandes catégories: les algorithmes traditionnels exacts pour des problèmes spécifiques, et les algorithmes approximatifs applicables à une grande variété de problèmes. Dans cette section, notre intérêt va se porter sur la deuxième catégorie d'algorithmes, plus précisément sur les métaheuristiques. Plusieurs adaptations des métaheuristiques sont proposées dans la littérature. Les adaptations les plus connues sont celles basées sur les colonies de fourmis et les essaims particuliers.

Classification basée sur les colonies de fourmis

Les colonies de fourmis constituent un moyen de formuler une heuristique puissante inspirée de la nature pour résoudre le problème de classification. Parmi leurs mouvements sociaux, des chercheurs ont simulé le comportement de collaboration dans le but de regrouper les objets (tel que les corps morts) afin de garder leurs nids propres (59). Les chercheurs ont récemment découvert plusieurs espèces de fourmis qui organisent leurs cimetières dans la nature. Les chercheurs ont distribué aléatoirement des cadavres ou des parties de cadavres à proximité d'un nid de fourmis, après quelques heures les fourmis organisatrices de cimetières ont regroupé les corps morts avec un comportement similaire à l'agrégation. Si l'arène expérimentale n'est pas suffisamment large, ou si elle contient des espaces hétérogènes, les groupes seront formés tout autour des bords de l'arène, ou plus généralement, autour des bords des espaces hétérogènes. Ce phénomène d'agrégation est expliqué par la présence d'une attraction entre les éléments morts et les fourmis travailleuses: les groupes de taille petite attirent les fourmis travailleuses à déposer plus de corps. C'est cet auto-catalytique feedback qui conduit à la formation des groupes de plus en plus grands. Les chercheurs ont observé qu'avec le temps, les fourmis ont tendance à regrouper tous les cadavres dans une région spécifique de l'environnement, formant ainsi des tas de cadavres. Le tri des larves et le nettoyage des cadavres dans une région spécifique de l'environnement effectué par la fourmi ont été modélisés par Deneubourg et ses collègues, pour accomplir certaines tâches dans la robotique (60). Cela a donné naissance à un algorithme de classification basé sur les colonies

de fourmis. Lumer et Faieta ont modifié l'algorithme en utilisant une évaluation basée sur la dissemblance de la densité locale, afin de l'adapter pour la classification de données (61). Cela a introduit l'algorithme standard de classification par les fourmis (en anglais : Clusters Ant (ACA)). Il a ensuite été utilisé dans l'analyse numérique des données, l'extraction de données et la fouille de textes. De nombreux auteurs ont proposé des modifications afin d'améliorer le taux de convergence et obtenir le nombre optimal de groupes. Monmarché et ses collègues ont hybridé l'algorithme basé sur les fourmis avec le K-means (62) et l'ont comparé avec l'algorithme K-means classique après application sur plusieurs ensembles de données, et en utilisant le taux d'erreur pour l'évaluation. Cependant, les résultats obtenus par cette méthode ne sont pas applicables à la classification ordinaires basée les fourmis, car il diffère sensiblement de cette dernière.

Tel que l'algorithme standard ACO (Ant Colony Optimization), la classification basée sur les fourmis est un processus distribué qui emploie le feed-back positif. Les fourmis sont modélisées par des agents simples qui se déplacent aléatoirement dans leur environnement. L'environnement est considéré comme un espace de petite dimension, plus généralement un plan de deux dimensions telle une grille carrée. Initialement, chaque donnée correspond à un point multidimensionnel aléatoirement positionné dans l'espace de recherche 2-D. les objets dispersées au sein de l'environnement peuvent être ramassés, transportés et abandonné par les agents d'une manière probabiliste. Les opérations de ramassage et de transportation sont influencées par la similarité et la densité des objets à l'intérieur du voisinage local des fourmis. Généralement, la taille d'un voisinage est de 3x3. La probabilité de ramassage des objets est grande quand ses objets sont isolés ou entourés par des objets dissemblables. Elles ont tendance de les déposer à proximité des objets semblables. De cette façon, la classification des objets sur la grille est effectuée.

Kanade et hall (63) ont présenté une hybridation du système de fourmis avec l'algorithme classique FCM afin de déterminer automatiquement le nombre de groupes dans un espace de données. Leur algorithme hybridé doit effectuer d'abord une classification par fourmis afin de déterminer des groupes initiaux puis raffine ces groupes en utilisant l'algorithme FCM. Initialement les fourmis déplacent les objets pour former des tas. Les centres de ces tas sont considérés comme des centres de classes initiaux et l'algorithme FCM est utilisé pour les affiner. Dans la deuxième étape, les positions des objets obtenues à partir de l'algorithme FCM seront modifiées selon le critère d'appartenance maximale pour former

de nouveaux tas. Ces nouveaux tas seront parfois déplacés et fusionnés par les fourmis. Les groupes finaux formés sont ré-affinés en utilisant l'algorithme FCM.

Plusieurs modifications ont été apportées au schéma de l'algorithme de classification basique afin d'améliorer la qualité de classification, la vitesse de convergence et, en particulier, la séparation spatiale entre les groupes dans la grille, cette séparation est essentielle pour un schéma trivial de la classification.

Classification basée sur PSO

PSO (Particle Swarm Optimization) est une métaheuristique biologique proposée en 1995 par Kennedy et Eberhart (64) dans le but de modéliser le comportement de certains animaux évoluant en essaims tel que les oiseaux ou encore les poissons. Ces derniers se déplacent en créant des mouvements complexes à partir de simples règles de comportement local, et ce, en associant à chaque particule des connaissances partielles sur son environnement.

L'algorithme de classification basé sur les PSO a été introduit par Omran dans (65). Ses résultats ont prouvé que PSO est plus performant que l'algorithme K-means, FCM et quelques autres algorithmes de classification classiques.

L'application de PSO à la classification des données passe par le déplacement des particules suivant des vecteurs de position et de vitesses ce qui permet ainsi de créer des groupes de données homogènes, et par ce fait, une structuration et une classification du nuage de données. Une erreur de quantification basée sur une mesure de fitness est introduite pour juger la performance de l'algorithme de classification.

5. Classification des images

Fondamentalement, la classification des images est un processus qui consiste à découper une image en régions connexes présentant une homogénéité selon un certain critère, comme par exemple la couleur. L'union de ces régions doit redonner l'image initiale.

La classification est une étape importante pour l'extraction des informations qualitatives de l'image. Le résultat d'une classification peut être vu comme un graphe. Ce dernier fournit une description de haut-niveau : chaque région est connectée à ses voisines et chaque région porte une étiquette donnant des informations qualitatives comme sa taille, sa couleur, sa forme, son orientation. L'image se réduit donc à un graphe de nœuds étiquetés qui contient presque toute

l'information utile au système. Les arcs du graphe précisent si les deux régions connectées sont en simple contact ou si l'une est incluse dans l'autre. D'autres informations topologiques peuvent également être stockées comme par exemple le fait qu'une région est au-dessus d'une autre. Selon les techniques de classification d'image utilisées, la construction d'une image classifiée peut être plus au moins complexe.

On regroupe généralement les algorithmes de classification d'images en trois grandes catégories :

1. Classification basée sur les pixels
2. Classification basée sur les régions
3. Classification basée sur les contours

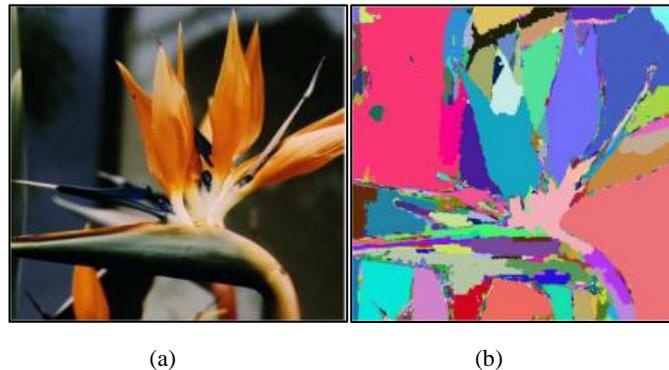


Figure 12. (a) : l'image avant classification, (b) : image après classification

Classification basée sur les pixels

Cette première catégorie travaille sur des informations de l'image. Par seuillage, clustering ou clustering flou, l'algorithme construit des classes de couleurs qui sont ensuite projetées sur l'image. La classification est implicite puisqu'on suppose que chaque cluster de l'histogramme correspond à une région dans l'image. En pratique, ce n'est pas forcément le cas, il faut séparer les régions de l'image qui sont disjointes bien qu'appartenant au même cluster de couleur. Ces algorithmes sont assez proches des algorithmes de réduction de couleur.

Classification basée sur les régions

Cette deuxième catégorie correspond aux algorithmes d'accroissement ou de découpage de région. L'accroissement de région est une méthode bottom-up : on part d'un ensemble de petites régions uniformes dans l'image (de la taille d'un ou de quelques pixels) et

on regroupe les régions adjacentes de même couleur jusqu'à ce qu'aucun regroupement ne soit plus possible. Le découpage de région est le pendant top-down des méthodes d'accroissement : on part de l'image entière que l'on va subdiviser récursivement en plus petites régions tant que ces régions ne seront pas suffisamment homogènes. Les algorithmes dit « split and merge » sont un mélange de ces deux approches.

Classification basée sur les contours

Cette troisième catégorie s'intéresse aux contours des objets dans l'image. La plus part de ces algorithmes sont locaux, c'est-à-dire fonctionnent au niveau du pixel. Des filtres détecteurs de contours sont appliqués à l'image. Le résultat est en général difficile à exploiter sauf pour des images très contrastées. Les contours extraits sont la plupart du temps morcelés et peu précis et il faut utiliser des techniques de reconstitution de contours par interpolation ou connaître à priori la forme de l'objet recherché. Formellement, ce type d'algorithme est proche des techniques d'accroissement de régions fonctionnant au niveau du pixel. Ces techniques purement locales sont en général trop limitées pour traiter des images bruitées et complexes.

6. Conclusion

Dans ce chapitre, nous avons présenté une formulation de la classification des données d'une façon classique et en tant que problème d'optimisation. Les différentes techniques et méthodes de base de classification ont été abordées. Ensuite, nous avons présenté les catégories de classification des images. Comme nous l'avons souligné dans la section 1.5, la classification peut être vue comme un problème d'optimisation, d'où la possibilité d'appliquer une métaheuristique à la classification des images. De ce fait, le prochain chapitre sera consacré à l'étude et la présentation de la métaheuristique que nous avons appliquée à la classification automatique d'images.

Chapitre 3
Algorithme de chauve-souris

1. Introduction

L'algorithme des chauves-souris, dont l'appellation d'origine est Bat Algorithm, est une métaheuristique très récente. Le premier article la concernant a été proposé en 2010 par Xin-She Yang (59). Elle est basée sur le comportement d'écholocation des chauves-souris. Spécifiquement, l'espèce des microchiroptères, qui se dirigent en vol par écholocation, peuvent trouver, discriminer les différents types d'insectes et éviter les obstacles, cela même dans l'obscurité totale (59).

Dans ce chapitre nous allons en premier lieu aborder brièvement le comportement d'écholocation des chauves-souris, dans un second sa formulation.

2. Echolocation des chauves-souris

2.1. Comportement des microchiroptères

La chauve-souris est l'un des mammifères les plus méconnus. Elle a cependant fasciné les plus grands chercheurs et intéresse aujourd'hui encore les scientifiques. Sa capacité de vol actif, unique chez les mammifères, lui permet de voler la nuit en évitant les obstacles, même à grande vitesse, grâce à son système l'écholocation très perfectionné (60). C'est l'un des animaux dont le nombre d'espèces est très élevé: il existe environ 996 espèces (61). Leurs tailles variant entre la chauve-souris bourdon dite microchiroptère ou « microbat » en anglais d'un poids d'environ 1,5g à 2g jusqu'à la chauve-souris géante, dite mégachiroptère ou « megabat » en anglais avec une envergure d'environ 2 mètres et un poids jusqu'à 1 kg. Les microchiroptères ont un corps d'une longueur variant de 2,2 à 11cm. La plupart des chauves-souris utilisent l'écholocation à un certain degré; parmi toutes les espèces, les microchiroptères disposent d'une capacité d'écholocation très avancée, dont les Mégachiroptères ne disposent pas (62).

La majorité des microchiroptères sont des insectivores. Elles émettent des pulsations sonores très fortes par la bouche ou le nez (ultrason). Dès que cet ultrason rencontre un obstacle (proie, végétation...), il rebondit vers la chauve-souris. Celle-ci capte l'écho grâce à ses oreilles. Son cerveau va alors calculer la distance, la vitesse, la position et la forme de l'objet détecté (60). Les types et les modes d'émission des ultra-sons sont très variables suivant les genres et même les espèces. Il arrive que les chauves-souris émettent des cris audibles par l'homme, il ne s'agit alors évidemment pas d'ultra-sons, mais de cris que l'on

qualifie de « cris sociaux », c'est-à-dire des cris qui sont utilisés par les chauves-souris pour communiquer entre elles (territorialité, agressivité, parade nuptiale, cri d'appel d'un jeune à sa mère, ...) (60). Intéressons-nous aux ultrasons.

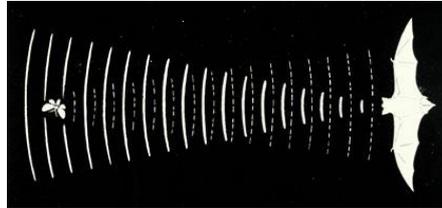


Figure 13. Echolocation

2.2. L'acoustique d'écholocation

Bien que chaque pulsation ne dure que quelques millièmes de seconde (De 8 à 10 ms), elles ont des fréquences constantes qui sont habituellement dans la région 25kHz et 100 kHz pour la plus part des espèces des chauves-souris. Toutefois, certaines espèces peuvent émettre des fréquences plus élevées allant à 150 KHz (63). Chaque cri ultrasonique peut durer moyennement 5 à 20 ms. Typiquement les microchiroptères émettent de 10 à 20 cris par seconde. En cas de chasse, le taux d'émission peut être accéléré jusqu'à environ 200 pulsations par seconde quand celles-ci volent à proximité de leurs proies. Ces cris sonores trop courts marquent la capacité étonnante du traitement puissant du signal des chauves-souris.

Comme la vitesse du son dans l'air est $v = 340$ m/s, la longueur d'onde λ des cris ultrasoniques avec une fréquence constante f est donné par

$$\lambda = \frac{v}{f} \quad (1)$$

Les longueurs d'onde sont du même ordre de grandeur que les tailles de leurs proies.

L'impulsion émise pourrait être aussi forte que 110 dB, appartenant à la région des ultrasons. L'intensité varie également de la plus forte lors de la recherche de proie et qui diminue en la poursuivant (63). Les microchiroptères peuvent éviter des obstacles même négligeables.

Les études ont prouvé que les microchiroptères utilisent le délai entre l'émission et la détection de l'écho, la différence de temps entre leurs deux oreilles, et les variations d'intensité des échos afin de construire un scénario tridimensionnel de l'environnement. Elles peuvent détecter la distance d'orientation de la cible, le type de la proie, et même la vitesse de

mouvement de la proie telles que les petits insectes. De surcroît, les études ont suggéré que les microchiroptères sont capables de discriminer leurs cibles par la variation de Doppler effect induit par le taux de wing-flutter de l'insecte cible [7].

Bien évidemment, quelques chauves-souris ont une bonne vue, et la plupart d'entre elles ont une très grande sensibilité pour l'odorat. En réalité, elles emploient ses sens en combinaison afin de maximiser le coefficient de détection de proie et la navigation minutieuse, particulièrement en obscurité totale. Cependant nous nous intéressons uniquement au comportement d'écholocation.

Le comportement d'écholocation des microchiroptères est associé à une fonction objective à optimiser nous permettant de formuler un nouveau algorithme d'optimisation appelé Algorithme des chauves-souris.

3. Optimisation par l'algorithme des chauves-souris

Comme il a été déjà cité, l'algorithme des chauves-souris tente de modéliser le comportement d'écholocation des microchiroptères. Il repose essentiellement sur le scénario suivant: des chauves-souris volent aléatoirement en vue de trouver de la nourriture, celles-ci n'ont aucune connaissance sur l'endroit où se trouvent leurs proies, cependant elles peuvent calculer la distance qui sépare chacune d'elles de la cible au moyen de l'écholocation. Leur objectif commun est non seulement d'atteindre leurs proies mais aussi de trouver la meilleure stratégie de chasse. La stratégie la plus efficace consiste à suivre la chauve-souris la plus proche de la volée. Ce comportement est similaire à résoudre un problème d'optimisation.

L'algorithme des chauves-souris est une méthode d'optimisation où toutes les chauves-souris collaborent en population dans le but d'atteindre un but collectif. Ce dernier est formulé par une fonction dite fonction objectif. Chacune des chauves-souris volantes est considérée comme une solution dans l'espace de recherche appelée solution candidate ou locale. Celle-ci sera évaluée à chaque itération de l'algorithme en calculant sa fitness par la fonction objectif et mise à jour en ajustant sa position, sa vitesse et sa fréquence, en fonction de la meilleure position qu'elle a atteinte et celle trouvée par toute la population. Pour que finalement l'algorithme maintienne la meilleure solution atteinte parmi toute la volée dite solution globale après un nombre d'itérations prédéfini.

Formellement, Les chauves-souris volent aléatoirement avec une vitesse v_i à une position x_i avec une fréquence fixée f_{min} , en variant la longueur d'onde λ et l'intensité A_0 de recherche de proie. Elles peuvent ajuster automatiquement la longueur d'onde (ou la fréquence) ainsi que le taux d'émission de pulsation $r \in [0, 1]$, selon la proximité de leurs proies.

Bien que l'intensité peut varier de plusieurs manières, l'algorithme suppose que $A \in [A_0, A_{min}]$.

L'algorithme des chauves-souris, est essentiellement composé des trois étapes suivantes, et qui seront répétées jusqu'à atteindre un certain nombre d'itérations :

1. Evaluer la fitness de chaque chauve-souris
2. Mettre à jour les meilleures solutions locales et globales.
3. Mettre à jour les positions, les vitesses et les fréquences de chacune des chauves-souris.

Les deux premières étapes sont assez simples. L'évaluation de la fitness de chacune des chauves-souris s'effectue à l'aide de la fonction objectif. Tandis que la mise à jour des meilleures solutions locales et globales s'effectue en comparant la fitness courante de chacune des chauves-souris avec la fitness de la meilleure solution qu'elle a rencontré jusqu'ici, puis de choisir la meilleure d'entre elles. Quant à la recherche de la solution globale, il suffira de comparer toutes les meilleures solutions locales et garder la meilleure.

La troisième étape est la plus cruciale, en effet, le mouvement des chauves-souris est responsable de l'efficacité de l'algorithme. Les règles définissant la nouvelle solution et mettant à jour la position et la vitesse de chacune des chauve-souris dans un espace de d dimensions sont les suivantes,

$$f_i = f_{min} + (f_{max} - f_{min})\beta, \quad (34)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i, \quad (35)$$

$$x_i^t = x_i^{t-1} + v_i^t, \quad (36)$$

Tel que,

- f_i est la fréquence d'émission de la chauve-souris x_i , et qui appartient à la gamme $[f_{min}, f_{max}]$ correspondant à la gamme de longueurs d'ondes $[\lambda_{min}, \lambda_{max}]$. Par exemple la gamme de fréquence [20KHz, 500KHz] correspond à la gamme de

longueurs d'ondes [0.7mm, 17mm] vers les plus petites distances. Afin de simplifier l'implémentation, il a été supposé que $f \in [0, f_{max}]$. Sachant que les hautes fréquences correspondent aux courtes longueurs d'ondes. Pour les chauves-souris, les gammes typiques sont de quelques mètres. Par conséquent le taux d'émission de pulsations peut être de la gamme [0, 1] ou 0 signifie qu'il n'a aucune pulsation, et 1 signifie le taux maximale d'émission de pulsations.

- $\beta \in [0,1]$ est un vecteur aléatoire tiré à partir d'une distribution uniforme.
- x_* est la meilleure location (solution) courante globale, qui sera calculée par comparaison de toutes les solutions obtenues par chacune des n chauves-souris.
- Pour la recherche locale, une fois une solution est sélectionnée parmi les meilleures solutions courantes, une nouvelle solution pour chaque chauve-souris est générée localement en utilisant le parcours aléatoire

$$x_{new} = x_{old} + \epsilon A^t, \quad (37)$$

d'où $\epsilon \in [-1,1]$ est un nombre aléatoire, tandis que $A^t = \langle A_i^t \rangle$ est la moyenne d'intensité de toutes les chauves-souris à l'instant t .

En se basant sur ces règles et ces explications, les étapes basiques de l'algorithme des chauves-souris peuvent être résumées tel qu'il est montré sur la figure suivante selon (59).

```

Définir la fonction objectif  $f(\mathbf{x})$ ,  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_d)^T$ 
Initialiser la position  $\mathbf{x}_i$  ( $i = 1, 2, \dots, n$ ), et la vitesse de chacune des chauves-souris  $\mathbf{v}_i$ 
Calculer la fréquence  $\mathbf{f}_i$  à la position  $\mathbf{x}_i$ 
Initialiser les taux d'émissions de pulsation  $\mathbf{r}_i$ , et l'intensité  $\mathbf{A}_i$ 
Tant que ( $t < \text{nbrMaxIterations}$ )
  Générer de nouvelles solutions par l'ajustement des fréquences,
  et mettre à jour les vitesses et les positions/solutions [équations (34) à (36)]
  si ( $\text{rand} > \mathbf{r}_i$ )
    Sélectionner une solution parmi les meilleures solutions
    Générer une solution locale autour de la meilleure solution sélectionnée
  fin si
  Générer une nouvelle solution en volant aléatoirement
  si ( $\text{rand} < \mathbf{A}_i$  &  $f(\mathbf{x}_i) < f(\mathbf{x}_*)$ )
    Accepter les nouvelles solutions
  Incrémenter  $\mathbf{r}_i$  et réduire  $\mathbf{A}_i$ 
  fin si
classer les chauves-souris et trouver la meilleure solution courante  $\mathbf{x}_*$ 
fin tant que
résultats et visualisations.

```

Figure 14. Pseudo code de l'algorithme des chauves-souris

Les mises-à-jour des vitesses et des positions des chauves-souris ont une certaine similarité avec la procédure standard d'optimisation par essais particuliers [8]. À un certain degré, l'algorithme des chauves-souris peut être considéré comme une combinaison d'optimisation par essais particuliers et la recherche locale contrôlée par l'intensité et le taux d'émission (59).

4. Conclusion

Ce chapitre a été consacré à la présentation de l'algorithme des chauves-souris et à son application au problème d'optimisation.

Le prochain chapitre aura pour but de présenter notre contribution pour la classification automatique des images et englobe deux étapes principales : la reformulation du problème de classification des images en un problème d'optimisation, puis appliquer l'algorithme des chauves-souris à ce problème, particulièrement à la classification des images satellitaires.

Chapitre 4 :

*Classification des images par
Algorithme de chauve-souris*

1. Introduction

Ce chapitre est consacré à la conception d'une méthode de classification automatique d'image basée sur l'optimisation par l'algorithme des chauves-souris. Cet algorithme a été déjà présenté dans le chapitre précédent. Le principe consiste à optimiser plusieurs critères de classification et de trouver la répartition qui permet d'avoir une meilleure classification.

Nous proposons dans ce chapitre une nouvelle approche de classification automatique basée sur l'algorithme des chauves-souris, cette dernière est une méthode puissante et avantageuse parmi les méthodes d'optimisation combinatoire et qui tire des avantages de plusieurs autres méthodes (Optimisation par essais particuliers et le recuit simulé). Afin d'illustrer sa vaste application, nous allons l'appliquer sur des images synthétiques, médicales et satellitaires.

Des résultats expérimentaux ont démontré que la classification automatique par l'algorithme des chauves-souris est plus performante que certaines d'autres méthodes classiques et récentes selon certain critères. Dans ce chapitre l'influence des différents paramètres et d'indices de validité sur la performance de l'algorithme sera étudiée.

Dans la première section de ce chapitre, nous formulons le problème de classification d'images par optimisation suivant l'algorithme des chauves-souris. La deuxième section est consacrée aux différents critères utilisés dans l'algorithme. Dans la troisième section, nous présentons l'algorithme de classification. Des résultats de classification des images test présentés dans la cinquième section.

2. Formulation du problème

Cette section définit la terminologie utilisée dans le reste de ce chapitre. Avant de présenter l'algorithme de classification proposé, des mesures de similarité et de quantification de la qualité de classification seront citées.

2.1. Formulation du problème de la classification d'images

Dans le cadre d'une classification des images et d'un point de vue mathématique, soit $P = \{P_1, P_2, \dots, P_m\}$ l'ensemble de 'm' pixels de dimension ' $d=3$ ', correspondant respectivement

aux couleurs Rouge, Vert et Bleu. Ce pixel peut être aussi représenté par la matrice $Z_{m \times d}$. le $i^{\text{ème}}$ vecteur \vec{Z}_i caractérise le $i^{\text{ème}}$ pixel de l'ensemble 'P' et chaque élément $z_{i,j}$ dans \vec{Z}_i correspond à la $j^{\text{ème}}$ valeur de la composante ($j = 1, 2, \dots, d$) du $i^{\text{ème}}$ pixel ($i = 1, 2, \dots, m$).

Un algorithme de classification tente de trouver une partition $C = \{C_1, C_2, \dots, C_k\}$ telle que la similarité entre les pixels du même groupe C_i est maximale et que la similarité entre les pixels de groupes différents est minimale. A partir de l'image originale plusieurs partitions peuvent être proposées, d'où la nécessité de définir une fonction d'évaluation. En effet, résoudre le problème de classification consiste à trouver une partition optimale C^* après l'avoir comparée avec toutes les autres partitions possibles $C = \{C^1, C^2, \dots, C^{N(m,k)}\}$ tel que $N(m,k)$ est le nombre de partitions possibles. Ce qui correspond à

$$\text{optimiser}_C (f(Z_{m \times d}, C)) \quad (38)$$

où ' C ' est une partition de l'ensemble ' C ' et ' f ' est une fonction objective qui doit évaluer une partition en se basant sur les mesures de similarité et dissimilarité des pixels.

2.2. Indices de validité utilisés

Les indices de validité sont généralement employés pour deux raisons. La première pour déterminer le nombre de groupes (régions), et la deuxième, pour trouver la meilleure partition (classification). Une approche traditionnelle permettant de déterminer le nombre optimale de classes, consiste à exécuter l'algorithme plusieurs fois avec des nombres de classes différents en entrée et sélectionner le partitionnement affectant la meilleure valeur à l'indice de validité.

Un indice de validité doit vérifier les critères suivants:

- *Cohésion*: Des pixels du même groupe doivent être similaires. La variance des fitness des pixels du même groupe est un indicateur de cohésion ou compacité du groupe.
- *Séparation*: Les groupes doivent être bien séparés. La distance entre les groupes, donne une indication sur la séparation entre les groupes.

En classification, plusieurs indices de validités connus sont disponibles dans la littérature, tel que "Calinski-Harabasz index", Davis-Bouldin (DB) index, PBM index, et "CS mesure", etc. Tous ces indices sont similaires à une fonction objective, leur

minimisation/maximisation mène à un partitionnement optimal. En raison de leur comportement d'optimisation, les indices de validité sont les mieux utilisés en association avec n'importe quel algorithme d'optimisation afin de trouver les classes appropriées à une image. Dans ce qui suit, nous allons présenter et étudier les indices utilisés durant notre application.

L'indice Davis-Bouldin (DB)

Nous avons utilisé l'indice DB en raison de son application très employée en télédétection spécifiquement en traitement d'images satellitaires, dont il est notre domaine d'intérêt. Une valeur minimale de l'indice correspond à une classification optimale, tentant de maximiser la distance entre deux pixels de régions différentes (Inertie interclasses), et de minimiser la distance entre les pixels de même région (Inertie intra-classe).

Premièrement, nous définissons respectivement la dispersion du *j*ème groupe et la distance entre le *i*ème et le *j*ème groupe par les équations suivantes :

$$S_i = \left[\frac{1}{N_i} \sum_{\vec{X} \in C_i} \|\vec{Z} - \vec{m}_i\|^2 \right]^{1/2} \quad (39)$$

$$d_{ij} = \|\vec{m}_i - \vec{m}_j\|_2 \quad (40)$$

Tel que, \vec{m}_i est le centre du *i*ème groupe, N_i étant le nombre de pixels appartenant au groupe '*i*' et '*C_i*' le centre du *i*ème groupe.

La valeur R_i est définie par :

$$R_i = \max_{j \in K, j \neq i} \left\{ \frac{S_i + S_j}{d_{ij}} \right\} \quad (41)$$

Finalement, nous définissons l'indice DB par :

$$DB(K) = \frac{1}{k} \sum_{i=1}^k R_i \quad (42)$$

La mesure CS

Chou et al (64) ont récemment proposé la mesure pour évaluer la validation d'une classification. Avant d'appliquer la mesure CS, le centre d'un groupe est calculé par la moyenne des pixels appartenant à ce groupe en utilisant la formule :

$$\vec{m}_i = \frac{1}{N_i} \sum_{\vec{z}_j \in C_j} \vec{z}_j \quad (43)$$

La distance entre deux pixels \vec{z}_p et \vec{z}_y est dénotée par $d(\vec{z}_p, \vec{z}_y)$. D'où la mesure CS peut être définie par,

$$\begin{aligned} CS(k) &= \frac{\frac{1}{k} \sum_{i=1}^k \left[\frac{1}{|C_i|} \sum_{\vec{z}_p \in C_i} \max_{\vec{z}_y \in C_i} \{d(\vec{z}_p, \vec{z}_y)\} \right]}{\frac{1}{k} \sum_{i=1}^k [\min_{j \in k, j \neq i} \{d(\vec{m}_i, \vec{m}_j)\}]} \\ &= \frac{\sum_{i=1}^k \left[\frac{1}{|C_i|} \sum_{\vec{z}_p \in C_i} \max_{\vec{z}_y \in C_i} \{d(\vec{z}_p, \vec{z}_y)\} \right]}{\sum_{i=1}^k [\min_{j \in k, j \neq i} \{d(\vec{m}_i, \vec{m}_j)\}]} \end{aligned} \quad (44)$$

Nous pouvons facilement apercevoir que cette mesure est une fonction du rapport entre la somme de la dispersion dans un groupe et la séparation entre les différents groupes. Ce principe est similaire à celui de l'indice DB. Selon Chou et al, cette mesure est plus efficace dans le cas où les groupes ont une densité et/ou des tailles différentes que d'autres mesures populaires. Cependant, l'augmentation de k et n peut compliquer les calculs et prolonger le temps d'exécution.

2.3. Mesure de similarité utilisés

En raison de sa popularité, son vaste domaine d'application et sa simple adaptation à tous types de problèmes, nous avons choisi la Distance Euclidienne (décrite par l'équation ci-dessous) comme mesure de similarité.

$$d(\vec{z}_u, \vec{z}_v) = \sqrt{\sum_{i=1}^d (z_{u,i} - z_{v,i})^2} = \|\vec{z}_u - \vec{z}_v\| \quad (45)$$

3. Classification d'images avec algorithme de chauve-souris

Nous présentons dans cette section le nouvel algorithme proposé pour la classification d'une image en plusieurs régions homogènes.

Afin d'appliquer l'algorithme de chauves-souris à la classification d'images, nous nous sommes inspirés de l'adaptation de l'optimisation par essaim particulière à la classification, proposée dans (65). Ainsi, nous associons à chaque chauve-souris b_i , un vecteur $b_i =$

$(x_{i,1}, x_{i,2}, \dots, x_{i,k})$ de dimension $d * k$. k étant le nombre de groupes, ou de régions présentes dans l'image. d est la dimension de l'espace de recherche, dans notre cas elle est égale à trois puisque chaque centre dispose de trois composantes: rouge, vert, et bleu. Tel que $x_{i,k}$ correspond au k_{ieme} centre du groupe de la i_{eme} chauve-souris.

Dans le cadre de ce projet, une version itérative de l'algorithme des chauves-souris pour la classification des images a été mise en œuvre dont les étapes principales peuvent se résumer dans la figure ci-dessous.

```
Initialiser la population des chauves-souris aléatoirement
Pour (t < nombre maximale d'itérations)
Pour chacune des chauves-souris
Générer des nouvelles solutions par ajustement des fréquences et mise à jour des
vélocités.
Pour chaque pixel
- calculer les distances entre le pixel et les centres des groupes
- Assigner le pixel au groupe avec une distance minimale
Si (rand > r) alors
Mettre à jour solution locale
Calculer la fitness et tester si la solution augmente ou non l'intensité
Mettre à jour la solution globale
```

Figure 15. Algorithme des chauves-souris adapté à la classification d'images

4. Environnement de travail

Notre application a été implémentée et expérimenté sur un environnement ayant les caractéristiques suivantes :

- Un microprocesseur Intel Centrino Core 2 Duo (2.00 GHz).
- Une RAM de 2 Go.
- Un disque dur de 160 Go.
- Un système d'exploitation Windows XP service pack 3.

A partir du pseudo code illustré dans la Figure ci-dessus, il est claire que l'algorithme proposé peut être implémenté par n'importe quel langage de programmation. En raison de notre maîtrise, nous l'avons programmé en JAVA, en utilisant NetBeans 6.9.1 comme EDI (Editeur de Developpement Intégré).

5. Les images de test

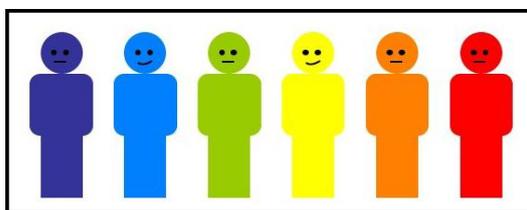
Notre algorithme de classification d'image automatique a été appliqué sur deux types d'images : synthétiques et satellitaires. Des images de chaque type ont été sélectionnées pour tester l'approche proposée.

5.1. Images synthétiques

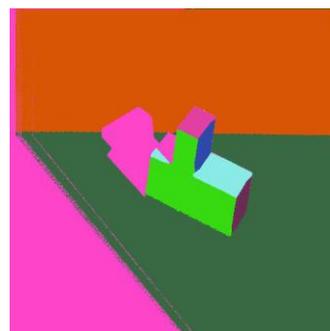
Les Figures ci-dessous représentent les images synthétiques sur lesquelles nous avons appliqué la nouvelle approche de classification proposée. (La figure (b)) a été obtenue à partir d'une base de données composée de trente images (figure 16), contenant des objets polyédriques, entre un et cinq objets sur chaque vue. Les objets et les régions figurant sur les images sont distinguables à l'œil nu, cela permet d'évaluer l'exactitude des résultats obtenus.



Figure 16. Base de données d'images de test



(a)



(b)

Figure 17. Images synthétiques

5.2. Images satellitaire

La figure 41 représente une image de la région d'Oran (l'Ouest de l'Algérie) acquise par le satellite LANDSAT5 TM (Thematic Mapper) datée du 15 mars 1993 (figure19)



Figure 18. Zone d'étude

Cette zone d'étude a été choisie pour notre maîtrise thématique de la région ainsi que son paysage varié pouvant présenter un intérêt pour l'évaluation de notre approche.



Figure 19. L'image satellitaire étiquetée

Avant d'appliquer la nouvelle approche de classification proposée sur l'image satellitaire dont nous disposons, il a été nécessaire d'effectuer les prétraitements suivants:

1. Chargement des trois images correspondantes aux trois canaux ;
2. Rehaussement de contraste ;
3. Composition colorée en associant les filtres bleu, vert et rouge aux trois canaux.

6. Schéma de l'application

Initialement nous avons trois fichiers (un fichier pour chaque canal) de type données (.IMG) contenant les valeurs radiométriques de chaque pixel de l'image. Le chargement de ces fichiers en termes de ligne et de colonne, forme les images correspondantes aux trois

canaux TM1, 3, 4. Afin d'augmenter la distinction entre les classes nous avons effectué un rehaussement de contraste par un étalement de la dynamique.

Puis, dans le but d'obtenir des classes qui se distinguent par différentes nuances de couleurs, nous avons effectué une composition colorée en associant :

- Au canal TM1 le filtre bleu
- Au canal TM3 le filtre vert
- Au canal TM4 le filtre rouge

De l'image obtenue et en utilisant nos connaissances thématiques, nous avons construit un fichier d'échantillonnage et un autre de test contenant respectivement des valeurs radiométriques et des coordonnées de pixels appartenant à différentes classes.

Une fois nos fichiers construits, nous avons appliqué une classification non supervisée qui aura pour but d'affecter aux pixels présentant des propriétés communes une étiquette de classe.

7. Résultats expérimentaux et discussion

7.1. Influence des paramètres

La performance de l'approche de classification proposée est influencée par un certain nombre de paramètres, tels que : le taux d'émission, l'intensité, les fréquences minimale et maximale, nombre d'itérations et la taille de la volée. Dans cette section nous étudierons l'influence de ces différents paramètres sur la performance de l'algorithme afin d'évaluer leur contribution à la qualité de la classification, la fonction objective et le temps d'exécution (TE), les tests et expérimentations seront effectués sur l'image synthétique de la figure 17 (a).

Taille de la population

Afin d'étudier l'effet de la taille de la population sur la performance de l'approche proposée, cette dernière a été exécutée avec des tailles de population différentes variant entre 10 et 50. Les figures ci-dessous illustrent l'effet de la taille de population sur la classification de l'image synthétique « la figure 43 » en utilisant l'algorithme de classification proposé, plus précisément sur la minimisation des deux indices utilisés DB et Jm. Sur ces figures, il est clair que plus on augmente la taille de la population plus les valeurs de DB et Jm diminuent. Néanmoins, il peut être observé qu'aucune minimisation est repérable à partir d'une taille de population variant entre 10 et 60. En général, l'augmentation du nombre de chauves-souris

augmente la diversité, limite l'effet des conditions initiales et permet d'éviter de se retrouver piégé dans des minima locaux, en explorant davantage tout l'espace des solutions, de façon à augmenter la probabilité de rencontrer le minimum optimal, c'est-à-dire le minimum global. Cependant une grande taille de population rend l'approche proposée très " gourmande " en mémoire et en temps d'exécution.

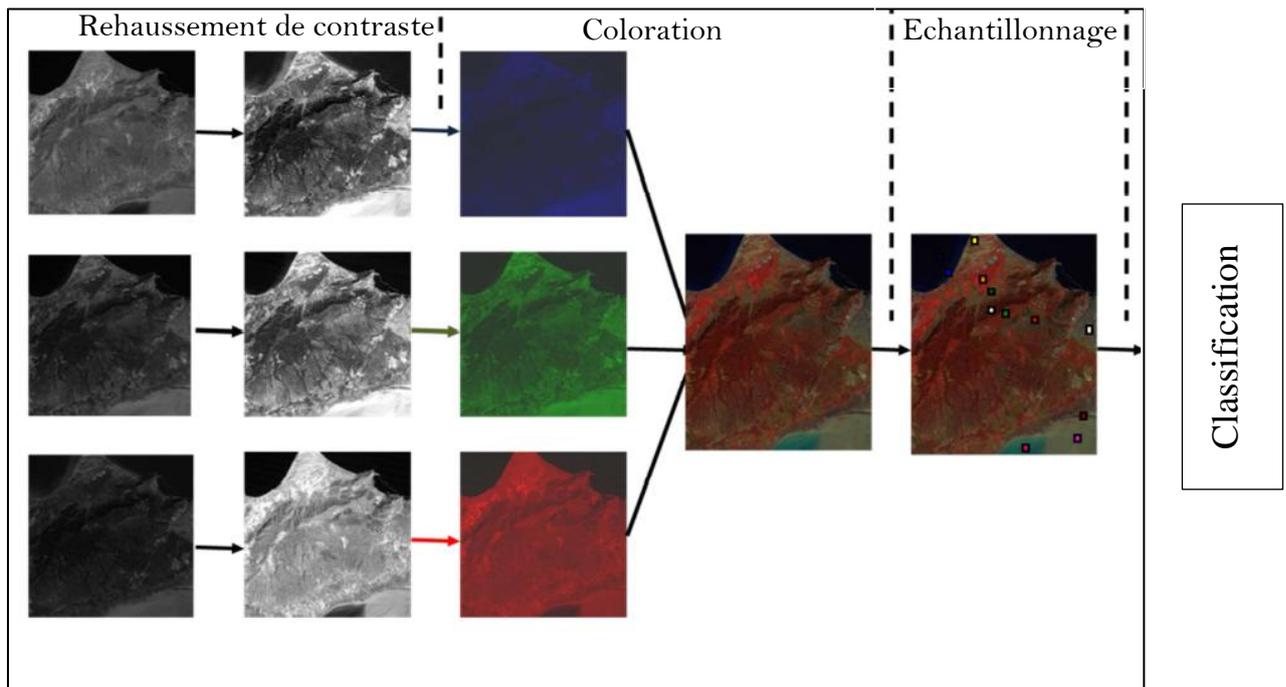


Figure 20. Schéma général de l'application

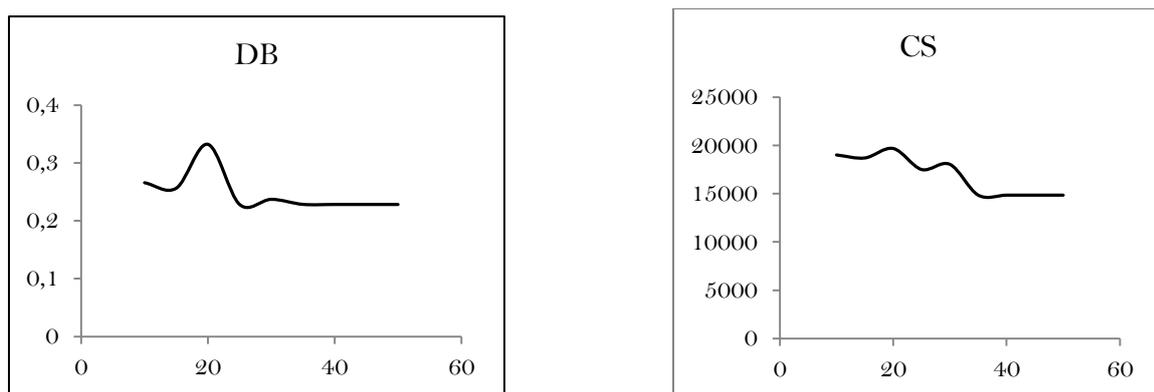


Figure 21. La valeur de la fitness en fonction de la taille de population

Nombre d'itérations

L'algorithme de classification proposée a été déroulé avec un nombre d'itérations variant entre 10 et 100. « La figure 44 » illustre que la meilleure solution est atteignable au

bout de 30 itérations, puis elle se stabilise pour des nombres d'itérations variant entre 30 et 100. Ce qui rend l'algorithme moins contraignant et moins gourmand en ressources matérielles.

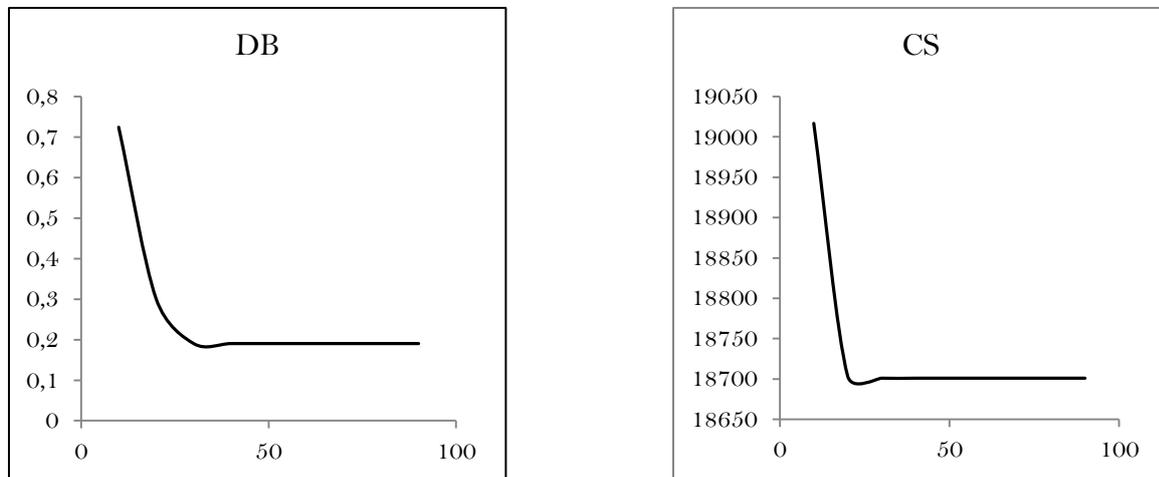


Figure 22. La valeur de la fitness en fonction du nombre d'itérations

La fréquence

Comme il a déjà été dit dans le chapitre précédent, la fréquence appartient généralement à l'intervalle $[f_{min}, f_{max}]$. Selon la littérature (66) et à titre de simplification, on peut utiliser une fréquence $f \in [0, f_{max}]$. Sachant qu'un large intervalle de fréquences permet d'avoir une recherche minutieuse, par conséquent, nous avons choisi l'intervalle $[0, 500]$ dont $f_{max}=500$ est la fréquence maximale que peut émettre une chauve-souris.

L'intensité et le taux d'émission

Nous avons choisi un taux d'émission et une intensité fixes, selon la littérature (66) $A_i = r_i = 0.25 \forall i = 1, n$.

Discussion

A partir des résultats obtenus nous avons déduit que plus on augmente le nombre de chauves-souris plus il apparaît des confusions entre les différentes classes et parfois même l'absence de certaines régions. De plus le temps d'exécution accroît. En revanche la valeur de DB reste stable. Il est aussi important de noter que nous avons fait varier le nombre

d'itérations et conclu qu'au bout de 30 itérations l'algorithme nous offre les meilleurs résultats.

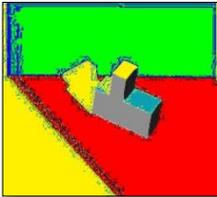
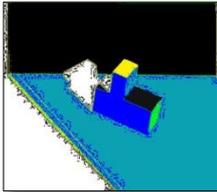
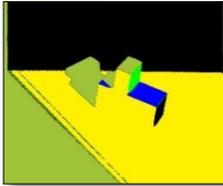
Tests	Résultats obtenus
$n = 10$	 DB = 0.2015598269315736, TE = 620766 m.s
$n = 20$	 DB = 0.14589022712166563 TE = 303969 m.s
$n = 30$	 DB = 0.3501517705672222 TE = 1052922, m.s

Table 1. Image synthétique classifiée par algorithme des chauves-souris

8. Conclusion

Nous avons présenté dans ce chapitre une nouvelle approche de classification non supervisée d'images satellitaires basée sur le comportement d'écholocation des chauves-souris. Ce dernier s'est montré performant. D'après les tests effectués, on déduit que le choix des paramètres influe fortement sur la qualité de la classification, tel que la taille de la volée, les fréquences minimales et maximales ainsi que le nombre d'itérations.

Un avantage de l'utilisation de l'algorithme des chauves-souris est que la recherche en parallèle d'une partition optimale est exécutée. Cette approche de recherche basée sur la population augmente la probabilité d'atteindre un bon partitionnement en raison de la diversité

des solutions, surtout pour des tailles de volée importantes, ce qui réduit ainsi l'effet de conditions initiales très restreintes.

Un des avantages majeurs de l'approche proposée, est qu'elle peut utiliser n'importe quel indice de validité. Par conséquent, l'utilisateur peut choisir l'indice de validité approprié à ses données. Les indices de validité utilisés dans ce chapitre sont : L'indice DB et CS.

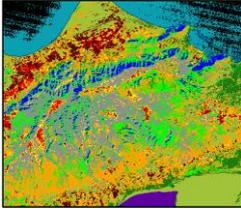
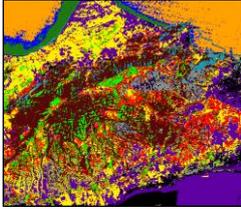
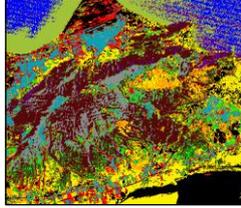
Tests	Résultats obtenus
<i>n = 10</i>	 DB = 0.216 TE = 707484 m.s
<i>n = 20</i>	 DB = 0.217 TE = 1676453 m.s
<i>n = 30</i>	 DB = 0.216 TE = 2685406 m.s

Table 2. Image satellitaire classifiée par algorithme des chauves-souris

*Conclusion générale
et perspectives*

Conclusion générale et perspectives

Dans ce mémoire, nous avons abordé la problématique de la classification des images au moyen des métaheuristiques; celles-ci présentant l'avantage de s'adresser à n'importe quel problème d'optimisation.

Parmi la multitude de métaheuristiques pouvant être utilisées, nous avons opté pour l'algorithme des chauves-souris. Cet algorithme modélise le comportement d'écholocation chez les chauves-souris.

La première étape de notre travail a porté sur l'étude de la problématique de l'interprétation d'images. Dans la deuxième étape nous nous sommes focalisés sur la classification, l'étape la plus importante en interprétation d'images. Ensuite nous avons présenté et expliqué l'optimisation par l'algorithme des chauves-souris. Finalement la dernière étape porte sur la reformulation du problème d'optimisation en un problème de classification d'images afin d'appliquer l'algorithme des chauves-souris sur cette classification.

L'approche de la classification d'images proposée présente l'avantage de donner en sortie un nombre optimal de groupes, tout en classifiant l'image avec la meilleure partition selon une fonction objective. Nous avons suggéré l'utilisation des indices de validités pour la fonction objective. Notre choix s'est porté sur les deux indices de validités DB et CS, ces derniers étant les indices les plus utilisés dans la classification d'images.

L'évaluation des résultats d'une classification est difficile, du fait que sa qualité dépend des traitements en amont, cela est dû car il peut exister plusieurs façons de classifier une image. Donc, pour avoir une idée aussi précise que possible de la performance de notre approche, nous avons opté pour une méthode "vérité terrain", basée sur le calcul de l'erreur de classification. On obtient cette erreur en comparant une image synthétique originale non bruitée et connue *a priori* à l'image classifiée d'une copie plus au moins dégradée. De plus, notre approche a aussi été testée sur des images diverses, bien qu'un accent particulier ait été mis sur des images satellitaires. L'objectif était de classifier ces images de manière à extraire correctement les régions significatives d'une image, pour ensuite évaluer l'image classifiée par rapport à une vérité du terrain.

Ce mémoire a donc proposé une nouvelle approche de classification non supervisée d'images satellitaires basée sur le comportement d'écholocation des chauves-souris. Ce dernier s'est montré performant. D'après les tests effectués, on déduit que le choix des paramètres influe fortement sur la qualité de la classification, tel que la taille de la volée, les fréquences minimales et maximales ainsi que le nombre d'itérations.

Perspectives

Les perspectives de ce travail sont nombreuses, aussi bien en pratique qu'en théorie. A titre d'exemple, il serait intéressant d'étendre l'algorithme de classification mono-objectif à un problème de classification multi-objectif.

Il est aussi possible d'améliorer les résultats de la classification en hybridant notre algorithme avec d'autres algorithmes d'optimisation.

En ce qui concerne le déroulement de l'approche, il serait intéressant que la recherche de la meilleure solution par les chauves-souris soit parallélisée, permettant ainsi de converger plus rapidement.

Enfin, une perspective intéressante reste l'application de cet algorithme de classification sur d'autres types d'images, comme des images médicales ou industrielles. Une étude plus approfondie du comportement des chauves-souris, peut apporter des extensions intéressantes à l'algorithme proposé, de surcroît l'étude des aspects de communication entre les chauves-souris, peut influencer avantageusement sur le temps d'exécution.

Ce projet a fait l'objet de communications internationales et d'une publication en cours.

Bibliographies

Bibliographies

1. **detection, Paul Viola and Michael J. Jones. Robust real-time object.** Robust real-time object detection. *International Journal of Computer Vision*. 2002. Vol. 57, 2.
2. **Schmid, Krystian Mikolajczyk and Cordelia.** A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis And Machine Intelligence (PAMI)*. 2005.
3. **Triggs, Frédéric Jurie and Bill.** Creating coefficient codebooks for visual recognition. *International Conference on Computer Vision (ICCV)*. 2005. Vol. 1.
4. **Herbert Bay, T. Tuytelaars, and L. Van Gool.** Surf : Speeded up robust features. *European Conference on Computer Vision*. 2006. Vol. 9, 5.
5. Robin challenge evaluation principles and metrics. 2006.
6. **Richard J. Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam.** Image change detection algorithms : a systematic survey. *IEEE Transactions on Image Processing*. 2005. Vol. 14, 3.
7. **Rita Cucchiara, Costantino Grana, Massimo Piccardi, and Andrea Prati.** Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*. Vol. 10, 25.
8. **Yannick Benezeth, Pierre-Marc Jodoin, Bruno Emile, and Helene Laurent.** Review and evaluation of commonly-implemented background subtraction algorithms. *International Conference on Pattern Recognition*. 2008.
9. **Thomas Deselaers, Daniel Keysers, and Hermann Ney.** Improving a discriminative approach to object recognition using image patches. *Lecture Notes In Computer Science*. 2005.
10. **Jianguo Zhang, Marcin Marsza lek, Svetlana Lazebnik, and Cordelia Schmid.** Local features and kernels for classification of texture and object categories : An in-depth study. s.l. : INRIA , 2005.
11. **Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray.** Visual categorization with bags of keypoints . *In Workshop on Statistical Learning in Computer Vision (ECCV)*. 2004.
12. **Triggs, Frédéric Jurie and Bill.** Creating efficient codebooks for visual recognition. *International Conference on Computer Vision (ICCV)*. Vol. 1.
13. **Delakis., Christophe Garcia and Manolis.** Convolutional face finder: a neural architecture for fast and robust face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*. 2004.

14. **Delakis, Christophe Garcia and Manolis.** A neural architecture for fast and robust face detection. *IEEE-IAPR International Conference on Pattern Recognition (ICPR)*. 2002.
15. **Triggs, N. Dalal and B.** Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005.
16. **Frédéric, Jurie and Cordelia Schmid.** Scale-invariant shape features for recognition of object categories. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2004.
17. **Mario Fritz, Bastian Leibe, B. Caputo, and B. Schiele.** Integrating representative and discriminant models for object category detection. In *IEEE International Conference on Computer Vision (ICCV)*. 2005. Vol. 2.
18. **Bastian Leibe, Ales Leonardis, and Bernt Schiele.** Combined object categorization and segmentation with an implicit shape model. *Workshop on Statistical Learning in Computer Vision (ECCV)*. 2004.
19. **Schmid, G. Dorko and C.** Object class recognition using discriminative local features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*. 2004.
20. **Antonio Torralba, Kevin P. Murphy, and William T. Freeman.** Sharing features :efficient boosting procedures for multiclass object detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2004.
21. **Hamerly, G. and Elkan, C.** *Alternatives to the k-means algorithm that find better clusterings*. 2002.
22. **Hamerly, G.J.** *Learning structure and concepts in data through data clustering*. 2003.
23. **Coleman, G.B. and Andrews, H.C.** *Image segmentation by clustering*. s.l. : IEEE, 1979. Vol. 67.
24. **Jain, A.K. and Dubes, R.C.** s.l. *Algorithms for clustering data*. : Prentice-Hall, Inc., 1988.
25. *A fast VQ codebook generation algorithm using codeword displacement.* **Lai, J.Z.C. and Liaw, Y.C. and Liu, J.** 1, s.l. : Elsevier, 2008, Vol. 41. *Pattern Recognition*.
26. **Judd, T. and Ehinger, K. and Durand, F. and Torralba, A.** *Learning to predict where humans look*. s.l. : IEEE, 2009.
27. *Pattern recognition and machine learning.* **Bishop, C.M. and SpringerLink (Service en ligne).** s.l. : springer New York, 2006, Vol. 4.
28. **Jain, A.K., Murty, M.N., Flynn, P.J.** Data clustering: a review. s.l. : ACM Computing, 1999. Vol. 31, 3.

29. **C. Veenman, M. Reinders and E. Backer.** A Cellular Coevolutionary Algorithm for Image Segmentation. s.l. : IEEE Transactions on Image Processing, 2003. Vol. 12, 3.
30. **Krishnapuram, H. Frigui and R.** A Robust Competitive Clustering Algorithm with. s.l. : IEEE Transactions on Pattern Analysis and, 1999. Vol. 21, 5.
31. **Y. Leung, J. Zhang and Z. Xu.** Clustering by Space-Space Filtering. s.l. : IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000. Vol. 22, 12.
32. **Turi, R.H.** Clustering-Based Colour Image Segmentation. *PhD Thesis*. s.l. : Monash University, Australia, 2001.
33. **Sokal, P. Sneath and R.** s.l. : Numerical Taxonomy, Freeman, London, UK., 1973.
34. **Anderberg., M.** Cluster Analysis for Applications. s.l. : Academic Press, New York, USA,, 1973.
35. **Everitt, B.** Cluster Analysis. s.l. : Heinemann Books, London, 1974.
36. **Krishnapuram, H. Frigui and R.** Clustering by Competitive Agglomeration. s.l. : Pattern Recognition Letters, 1997. Vol. 30, 7.
37. **A. Jain, R. Duin and J. Mao.** Statistical Pattern Recognition: A Review. s.l. : IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000. Vol. 22, 1.
38. **Zhang, B.** Generalized K-Harmonic Means - Boosting in Unsupervised Learning. *Technical Report HPL-2000-137*. s.l. : Hewlett-Packard Labs, 2000.
39. **Forgy, E.** Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification. *Biometrics*. 1965. Vol. 21.
40. **Davies, E.** Machine Vision: Theory, Algorithms, Practicalities. *Academic Press, 2nd Edition*. 1997.
41. **Theodoridis, S., Koutroumbas, K.** *Pattern Recognition, 2nd*. s.l. : Elsevier Academic Press, Amsterdam, 2003.
42. **Ng, R., Han, J.** Efficient and effective clustering methods for spatial data mining. *Proceeding's of the 20th VLDB Conference, Santiago, Chile*. 1994.
43. **Bezdek, J.** A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1980. Vol. 2.
44. **Bezdek, J.C.** Pattern Recognition with Fuzzy Objective Function Algorithms. *Plenum Press*. 1981.
45. **A. Liew, S. Leung and W. Lau.** Fuzzy Image Clustering Incorporating Spatial Continuity. *In IEE Proceedings Vision, Image and Signal Processing*. 2000. Vol. 147, 2.

46. **Keller, Krishnapuram and.** A Possibilistic Approach to Clustering. *IEEE Transactions on Fuzzy Systems*. 1993. Vol. 1, 2.
47. **Krishnapuram, R. and Keller, J.M.** The Possibilistic C-Means algorithm: Insights and Recommendations. *IEEE Transactions on Fuzzy Systems*. 1996. Vol. 4, 3.
48. **Krishnan, G. McLachlan and T.** The EM algorithm and Extensions. *John Wiley & Sons, Inc.* 1997.
49. **Walker, R. Rendner and H.** Mixture Densities, Maximum Likelihood and the EM Algorithm. *SIAM Review*. 1984. Vol. 26, 2.
50. **Bishop, C.** Neural Networks for Pattern Recognition. *Clarendon Press, Oxford*. 1995.
51. **C. Veenman, M. Reinders and E. Backer.** A Maximum Variance Cluster Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2002. Vol. 24, 9.
52. **B. Zhang, M. Hsu and U. Dayal.** K-Harmonic Means - A Data Clustering Algorithm. *Technical Report HPL-1999-124*. s.l. : Hewlett-Packard Labs, 1999.
53. **Bradley, P.S., Fayyad, U.M.** Refining Initial Points for K-Means Clustering. *ICML*. 1998.
54. **Ester, M., Kriegel, H.-P., Sander, J., Xu, X.** A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceeding of 2nd Int. Conf. on Knowledge Discovery and Data Mining*. Portland : s.n., 1996.
55. **Hinneburg, A., Keim, D.** An efficient approach to clustering in large multimedia databases. *Proceedings of KDD Conference*. 1998.
56. **Wang, W., Yang, J., Muntz, R.** STING: A statistical Information grid approach to spatial data mining. *In: Proceedings of 23rd VLDB Conference*. 1997.
57. **Sheikholeslami, C., Chatterjee, S., Zhang, A.** WaveCluster: A-multi resolution clustering approach for very large spatial database. *In: Proceedings of 24th VLDB Conference*. New York, USA : s.n., 1998.
58. **Han, J., Kamber, M.** Data Mining: Concepts and Techniques. *Morgan Kaufmann Publishers*. USA : s.n., 2001.
59. **Bonabeau, E., Dorigo, M., Theraulaz, G.** Swarm Intelligence: From Natural to Artificial Systems. *Oxford University Press Inc . Oxford* : s.n., 1999.
60. **Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., Chetien.** The dynamics of collective sorting: Robot-like ants and ant-like robots. *Proceedings of the First International Conference on Simulation of Adaptive Behaviour: From Animals to Animats I*. Cambridge : MIT Press, 1991.

61. **Handl, J., Knowles, J., Dorigo, M.** Ant-based clustering: a comparative study of its relative performance with respect to k-means, average link and 1D-SOM. Belgium : Universite Libre de Bruxelles, 2003.
62. **Monmarche, N., Slimane, M., Venturini, G.** Ant class: discovery of clusters in numeric data by a hybridization of an ant colony with the k means algorithm. 1999.
63. **Kanade, P.M., Hall, L.O.** Fuzzy ants as a clustering concept. *Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society (NAFIPS 2003)*. 2003.
64. **Jain, A.K., Dubes, R.C.** Algorithms for Clustering Data. *Prentice-Hall, Englewood Cliffs*. 1988.
65. **Omran, M., Salman, A., Engelbrecht, A.P.** Image classification using particle swarm optimization. *Conference on Simulated Evolution and Learning*. 2002.
66. **Yang, Xin-She.** A New Metaheuristic Bat-Inspired Algorithm. *math.OC*. s.l. : Department of Engineering, University of Cambridge, 2010. Vol. 1, 1004.4170.
67. **Altringham, McOwat, and Hammond.** Bats : biology and behaviour. *Oxford University Press*. USA : s.n., 1998.
68. **Maulik, Sanghamitra Bandyopadhyay and Ujjwal.** Genetic clustering for automatic evolution of clusters and application to image classification. *Pattern Recognition*. 2002. Vol. 6, 35.
69. **ligne), C.M. Bishop and SpringerLink (Service en.** Pattern recognition and machine learning. *springer*. New York : s.n., 2006. Vol. 4.
70. **Andrews, G.B. Coleman and H.C.** Image segmentation by clustering. *Proceedings of the IEEE*. 1979. Vol. 5, 65.
71. **Pakhira, M.K., Bandyopadhyay, S., Maulik, U.** A study of some fuzzy cluster validity indices, genetic clustering and application to pixel classification. *Fuzzy Sets and Systems*. 2005.
72. **Belhadj, Raz Khedam and Hde Hamdoud and Ser Miloudi and Aissa.** The algorithm of Particles Swarm Optimization applied to the clustering of satellites images. *First National Journey about Metaheuristics and their Applications JNAM07*. 2007.
73. **Comaniciu, D., Meer, P.: Mean shift.** A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24. 2002.