الشعبية الديمقراطية الجزائرية الجمهورية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

N₀ Ref : .....

Centre Universitaire Abdelhafid Boussouf Mila

Institut de Mathématique et Informatique

Mémoire préparé En vue de l'obtention du diplôme de Master en Informatique

Filière : Informatique

Spécialité : Intelligence artificielle et ses applications (I2A)

Utilisation de l'intelligence en essaim pour le traitement d'image

Présentée Par : Arar Fatima Zohra & Benmrara Marwa

Soutenue devant un jury composé de :

Encadrant :	Bekhouche maamar	MAB	C. U. Abdelhafid Boussouf, Mila
Président :	Hadjadj Abde El Halim	MAB	C. U. Abdelhafid Boussouf, Mila
Examinateur :	Guemri Oualid	MCB	C. U. Abdelhafid Boussouf, Mila

Année universitaire : 2024/2025

Résumé

Dans cette étude, nous proposons une approche de segmentation d'images fondée sur l'intelligence en essaim, en particulier à l'aide de l'algorithme *Grey Wolf Optimizer* (GWO). Dans un premier temps, une segmentation mono-seuil est réalisée à l'aide de la méthode d'Otsu, dont le résultat sert d'image de référence (vérité terrain) pour évaluer les performances de GWO. Les seuils obtenus par Otsu et GWO sont respectivement 154 et 156, avec une valeur de fitness atteignant 0,998, témoignant d'une forte similarité.

La segmentation mono-seuil ne présente pas de difficulté majeure pour les deux approches. En revanche, la segmentation multi-seuils (notamment avec trois seuils) constitue un défi significatif pour Otsu, en raison de sa méthode exhaustive par force brute, qui devient rapidement intractable sur le plan computationnel lorsque le nombre de combinaisons possibles augmente.

Notre approche est testée sur un ensemble de dix images de tailles variées. Pour chaque image, le résultat de la méthode d'Otsu est utilisé comme vérité terrain pour évaluer la segmentation obtenue par GWO. Les résultats expérimentaux démontrent les performances prometteuses de l'algorithme GWO, avec un bon compromis entre précision de segmentation et efficacité computationnelle.

Mots-clés: Algorithme d'optimisation par le loup gris, segmentation ,intelligence collective.

Abstract

In this study, we propose an image segmentation approach based on swarm intelligence, specifically using the Grey Wolf Optimizer (GWO) algorithm.

First, a single-threshold segmentation is performed using Otsu's method, and the resulting segmented image serves as a reference (ground truth) to evaluate the performance of GWO. The thresholds obtained by Otsu and GWO are 154 and 156, respectively, with a fitness value reaching 0.998, indicating a strong similarity.

Single-threshold segmentation does not present major difficulties for either approach. However, multi-threshold segmentation (particularly with three thresholds) poses a significant challenge for Otsu due to its brute-force exhaustive method, which becomes computationally intractable as the number of possible combinations increases.

Our approach is tested on a set of ten images of various sizes. For each image, the result of Otsu's method is used as ground truth to evaluate the segmentation produced by GWO. Experimental results demonstrate the promising performance of the GWO algorithm, offering a good trade-off between segmentation accuracy and computational efficiency.

Keywords: Grey Wolf Optimizer, segmentation, swarm intelligence.

ملخص

في هذه الدراسة، نقترح منهجًا لتقسيم الصور يعتمد على الذكاء الجماعي، وتحديدًا باستخدام خوارزمية محسن الذئب في البداية، يتم تنفيذ تجزئة باستخدام عتبة واحدة بواسطة طريقة أوتسو، ويُستخدم الناتج كصورة مرجعية (حقيقة أرضية) لتقييم أداء خوارزمية GWO. كانت القيم الناتجة للعتبات بواسطة أوتسو و GWOعلى التوالي 154 و156، مع قيمة لياقة (fitness) بلغت 0.998، مما يدل على تشابه كبير .

لا تمثل التجزئة ذات العتبة الواحدة صعوبة كبيرة بالنسبة للطريقتين. ولكن، تُعدّ التجزئة متعددة العتبات (وخاصة باستخدام ثلاث عتبات) تحديًا كبيرًا لطريقة أوتسو، نظرًا لاعتمادها على البحث الشامل بالقوة الغاشمة، مما يجعلها غير قابلة للتطبيق حسابيًا عند زيادة عدد التركيبات الممكنة.

لقد تم اختبار المنهجية المقترحة على مجموعة مكونة من عشر صور بأحجام مختلفة. ولتقييم أداءGWO ، تم استخدام نتيجة طريقة أوتسو كمرجع حقيقي لكل صورة. وتُظهر النتائج التجريبية الأداء الواعد لخوارزميةGWO ، من حيث تحقيق توازن جيد بين دقة التجزئة والكفاءة الحسابية.

الكلمات المفتاحية :خوارزمية التحسين بواسطة الذئاب الرمادية، تقسيم الصور، الذكاء السرب.

Remerciements

Louange à Allah qui nous a accordé le succès et a facilité l'achèvement de ce mémoire

De Master.

Nous exprimons notre profonde gratitude à notre encadrant, Monsieur **Bekhouche Maamar** pour ses orientations précieuses, sa patience dans la réponse à toutes nos interrogations, et son soutien constant qui a grandement contribué à la réalisation et à la présentation de ce travail dans sa forme actuelle.

Nous adressons également nos sincères remerciements aux membres du jury, Monsieur **:** **Hadjadj Abde EL Halim** et Monsieur **Guemri Oualid,** pour avoir accepté avec bienveillance d'évaluer ce travail, ainsi que pour leurs remarques constructives qui contribueront à améliorer nos compétences et à enrichir nos recherches futures.

Nous ne saurions oublier d'exprimer notre plus profonde reconnaissance à nos familles respectives, qui ont été notre principal soutien à toutes les étapes de notre vie, tant dans les moments de réussite que dans les périodes de difficulté. Leur encouragement constant et leur appui inébranlable ont été notre force motrice.

Nos remerciements vont également à nos enseignants, amis et collègues, qui ont été des compagnons fidèles et des soutiens tout au long de notre parcours académique.

Enfin, nous implorons Allah, le Très-Haut, d'agréer ce travail en toute sincérité pour Sa cause, et de nous accorder, ainsi qu'à vous tous, la réussite et le succès continus. Qu'Il fasse de ce travail une source d'inspiration pour les futurs étudiants.

اهداء

لم تكن الرحلة سهلة، ولم يكن الطريق مفروشًا بالورود، لكن الحمد لله الذي أنار لنا السبيل

ويسّر البدايات، وبلغنا النهايات بفضله وكرمه.

بكل فخر وامتنان، أهدي ثمرة هذا الجهد، وتاج هذا النجاح، إلى من كان النور الذي أضاء دربي، والسند الذي لم يتخلّ عني يومًا ...

*إلى من أحمل اسمه بكل اعتزاز، إلى قدوتي، إلى **أبي***

الذي علّمني أن الطموح لا يعرف المستحيل، وأن الإرادة تصنع المعجزات. هذا النجاح وعد قطعته على نفسي وها أنا أفي به، فليكن عربون وفاء وامتنان لحنانك وتضحياتك التي لا تُقدَّر

*وإلى نبع الحنان، القلب الذي لا يعرف إلا العطاء، إلى **أمي** الحبيبة*

التي كانت دعواتها النقية الحصن الحصين في كل لحظة ضعف، وكانت سندي وقت الشدة، وفرحي في كل إنجاز لك وحدك أمي، تنحني الحروف إجلالًا

*إلى أختي العزيزة **فريال**، رفيقة دربي، وسندي في لحظات الانكسار، التي كانت دومًا مثالًا للتشجيع والاحتواء*

فلك من قلبي كل الامتنان...

*إلى أخي **صهيب وباسم**، صاحب القلب الكبير، الذي كان مصدرًا للقوة والدعم في صمته واهتمامه، عنوان المحبة والمشاركة، أنتم جزء من هذا الحلم المتحقق*

وأخيرًا، أهدي هذا النجاح لنفسي... لتلك الفتاة التي آمنت بقدرتها، ورفضت أن تستسلم، وواصلت الليل بالنهار، وسقطت وقامت، فصبرت، وثابرت، وها هي اليوم تحصد ثمرة تعبها

لكم جميعًا... أهدِي هذا العمل، وهذه اللحظة التي انتظرتها طويلًا.

فاطمة الزهراء

اهداء

ما سلكنا البدايات إلا بتيسيره، وما بلغنا النهايات إلا بتوفيقه، وما حققنا الغايات إلا بفضله،

فالحمد لله الذي بنعمته تتم الصالحات، وبفضله وكرمه أُنجز هذا العمل، فله الحمد والشكر أولاً وآخراً .

إلى أمي الحبيبة،

يا من سهرتِ وتعبتِ وضحيتِ لأجلي،يا من كنتِ الحضن الدافئ والقلب الصابر،فرحتي لا تكتمل إلا برؤيتك تبتسمين ودموع الفخر والسعادة تملأ عينيك لكِ وحدكِ كل الامتنان، وكل الحب، فأنتِ النعمة الأجمل في حياتي،ونجاحي هذا ثمرة دعواتك وصبرك .

وإلى أبي العزيز،

يا من كنتَ دومًا القدوة والمُلهم، والسند الذي لا يميل، كل ما وصلتُ إليه هو بفضل دعائك، وتوجيهاتك، ودعمك الذي لم ينقطع، فلك مني كل الحب والتقدير، وأسأل الله أن يبارك فيك ويحفظك لي دائمًا .

إلى إخوتي الأعزاء

سامي، رياض ، رضوان، *كنتم السند الحقيقي، والرفاق في كل مراحل الحياة، فلكم مني كل الشكر والتقدير، حفظكم الله ووفقكم.*

إلى رفيقات المشوار اللواتي قاسمتني لحظاته رعاهم الله ووفقهم . إلى كل من كان لهم أثر في حياتي إلى كل من ساندني من قريب او بعيد لا تمام هذا العمل الذي تم بحمد الله وفضله وكرمه .

اهديكم هذا الإنجاز وثمرة نجاحي الذي لطالما تمنيته ها أنا اليوم أتممت أول ثمراته بفضل من الله عزوجل ؛

فالحمد لله على ما وهبني وأن يعينني ويجعلني مباركة أينما كنت .

مروة

Table des matières

Table des figures

Liste des tableaux

Liste des Abréviations

Optimiseur des loups gris	(Grey Wolf Optimizer)	**(GWO)**
Indice de similarité structurelle	(Structural Similarity Index)	**(SSIM)**
Coefficient de similarité de Dice	(Dice Similarity Coefficient)	**(DSC)**
Coefficient d'intersection sur union	(Intersection over Union)	**(IoU)**
L'intelligence en essaim	(Swarm Intelligence)	**(SI)**
Méthode de seuillage d'Otsu	(Otsu's thresholding method)	**(OTSU)**

Introduction générale

Le traitement d'image est devenu un outil essentiel dans le développement de solutions technologiques de pointe, notamment en médecine, la surveillance, l'industrie ou encore l'imagerie satellitaire. Parmi les étapes fondamentales de ce domaine, la segmentation d'image occupe une place essentielle : elle consiste à diviser une image en régions distinctes et homogènes, facilitant ainsi les tâches d'analyse, de reconnaissance et d'interprétation automatique des images.

Parmi les techniques classiques de segmentation, les méthodes de seuillage, et notamment la méthode d'Otsu, sont largement utilisées en raison de leur simplicité et de leur efficacité, notamment dans le cas d'images bien contrastées. Toutefois, ces méthodes présentent rapidement des limites, en particulier lorsqu'il s'agit de segmentation multi-seuils. Ce type de segmentation est nécessaire pour de nombreuses applications complexes, mais il devient coûteux en temps de calcul à cause de la croissance exponentielle du nombre de combinaisons de seuils possibles.

Dans ce contexte, les approches bio-inspirées, et plus précisément les techniques issues de l'intelligence en essaim, se présentent comme des alternatives prometteuses. Ces méthodes, inspirées des comportements collectifs observés dans la nature, permettent de résoudre efficacement des problèmes complexes d'optimisation. Parmi elles, l'algorithme Grey Wolf Optimizer (GWO), fondé sur le comportement social et de chasse des loups gris, a démontré d'excellentes performances dans plusieurs domaines, y compris le traitement d'image.

Problématique

La segmentation d'images est une étape cruciale dans de nombreuses applications de traitement d'images, telles que l'imagerie médicale, la vision par ordinateur et l'analyse de scènes. Parmi les méthodes classiques, la méthode d'Otsu est largement utilisée pour la segmentation par seuillage automatique. Cependant, cette méthode montre rapidement ses limites lorsqu'elle est appliquée à des segmentations multi-seuils, en raison de la croissance exponentielle du nombre de combinaisons à tester, rendant la recherche exhaustive peu efficace voire inapplicable pour des cas complexes ou des images de grande taille. Cela soulève la nécessité de recourir à des méthodes d'optimisation plus intelligentes, capables d'explorer efficacement l'espace de recherche tout en maintenant une bonne précision de segmentation.

Motivation de la thèse

La principale motivation de ce travail réside dans les limites des méthodes classiques de segmentation, notamment lorsqu'elles sont confrontées à des images complexes nécessitant une segmentation multi-seuils. La méthode d'Otsu, bien que fiable dans les cas simples, devient inefficace face à la complexité computationnelle liée à la recherche exhaustive de plusieurs seuils. Dans les contextes actuels, où les images sont souvent de grande taille et riches en informations, il devient essentiel de disposer d'approches rapides, précises et robustes**.**

D'autre part, les algorithmes bio-inspirés, en particulier le GWO, offrent un bon compromis entre efficacité computationnelle et qualité des résultats**.** Cependant, leur potentiel reste encore

peu exploré pour certaines tâches spécifiques de segmentation. Il est donc pertinent d'étudier, d'adapter et d'évaluer ce type d'algorithme pour des applications concrètes en segmentation d'image.

Objectifs de la thèse

L'objectif principal de cette mémoire est d'étudier l'efficacité de l'algorithme Grey Wolf Optimizer (GWO) appliqué à la segmentation d'image, en le comparant à une méthode de référence qu'est celle d'Otsu. Dans ce cadre, la recherche consiste à développer une approche de segmentation d'images basée sur GWO, capable de traiter à la fois les cas de segmentation à seuil unique et ceux à seuils multiples. La méthode d'Otsu sera utilisée comme point de référence, à la fois comme technique de segmentation et comme vérité terrain, afin d'évaluer de manière rigoureuse les performances de GWO. Les expérimentations porteront sur un ensemble varié d'images, avec pour objectif d'analyser la précision des segmentations obtenues ainsi que le coût computationnel associé à chaque méthode. Enfin, ce travail vise à mettre en évidence les atouts et les limites de l'algorithme GWO dans le contexte spécifique de la segmentation d'image, tout en suggérant des pistes d'amélioration ou d'extension pour de futures recherches.

Organisation du manuscrit

La suite de cette thèse est divisée en trois chapitres :

Chapitre 1 : Ce chapitre présente les bases du traitement d'image, en introduisant les notions fondamentales liées aux images numériques, leurs caractéristiques et types. Il détaille ensuite les étapes clés du traitement d'image, telles que le prétraitement, la segmentation et la classification.

Chapitre 2 : Dans ce chapitre nous avons introduit la notion de l'intelligence en essaim, ce chapitre est divisé en sept parties: Introduction, Définition, fonctionnement, applications, Avantages et caractéristiques de l'intelligence en essaim, dans la partie six nous avons présenté l'évolution des techniques d'intelligence en essaim, et en fin dans la partie sept nous avons présenté des travaux connexes sur le traitement d'images avec différentes méthodes.

Chapitre 3 : Ce chapitre met en valeur la contribution principale de notre travail en détaillant l'approche proposée, fondée sur l'intelligence en essaim pour la segmentation d'image, et en présentant les résultats expérimentaux obtenus.

Chapitre 1 : Traitement d'image

1.1 Introduction

Avec l'explosion des données visuelles dans des domaines aussi variés que la médecine, la sécurité, l'industrie, la robotique ou encore les réseaux sociaux, le traitement d'image est devenu un outil incontournable [1]. Il repose sur des méthodes mathématiques et algorithmiques permettant d'analyser, d'interpréter et d'exploiter automatiquement le contenu visuel [2] Ces techniques jouent un rôle crucial dans l'extraction d'informations pertinentes à partir d'images numériques. Les récentes avancées en intelligence artificielle, en particulier dans le domaine de l'apprentissage profond (deep Learning), ont profondément transformé le traitement d'image. Elles ont permis d'atteindre des performances remarquables dans des applications complexes telles que la reconnaissance faciale, l'imagerie médicale, ou encore la conduite autonome. [3]

Ce premier chapitre est consacré aux fondements du traitement d'image. Nous commencerons par définir ce qu'est une image numérique et en décrierons ses principales caractéristiques, telles que le pixel, la taille, et la résolution. Nous présenterons ensuite les processus de traitement d'image .Enfin, une section sera consacrée à la segmentation d'image, avec une attention particulière portée sur le seuillage.

1.2 Définition de l'image

L'image est une représentation d'une personne ou d'un objet, réalisée par la peinture, la sculpture, le dessin, la photographie, le film, etc. Elle constitue également un ensemble structuré d'informations ayant une signification pour l'œil humain. Cette représentation peut être exprimée sous la forme d'une fonction ***I(x, y)***, définie dans un domaine borné. Les x et y correspondent aux coordonnées spatiales d'un point de l'image, tandis qu' I représente l'intensité lumineuse et les couleurs [4]. Ce processus de numérisation permet de transformer cette information visuelle en une forme discrète, facilitant ainsi son traitement à l'aide d'algorithmes informatiques. [5]

1.3 Image numérique

L'image numérique est l'image dont la surface est divisée en éléments de taille fixe appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs. [6] La numérisation d'une image est la conversion de celle-ci de son état analogique en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques f(x,y), comme la montre la figure où :

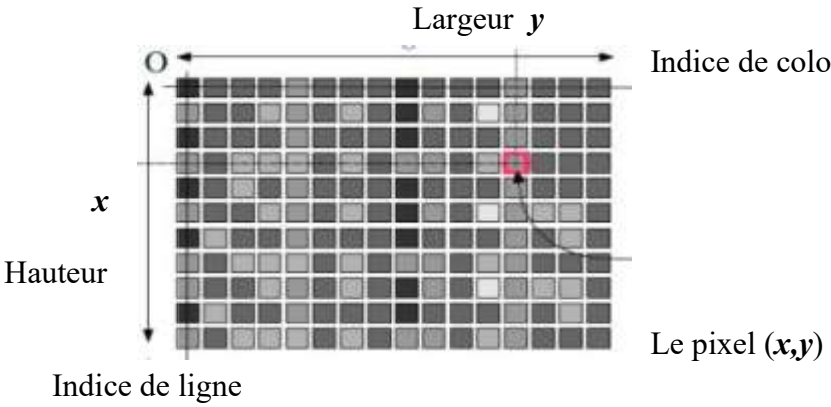


Figure 1. 1: Représentation d'image numérique.

x,y: coordonnées cartésiennes d'un point de l'image.
f (x, y) : niveau d'intensité. La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur. [4]

1.4 Caractéristiques d'une image numérique

L'image se présente comme un ensemble structuré et cohérent d'informations visuelles, organisé selon plusieurs paramètres spécifiques qui permettent de mieux comprendre sa composition et sa signification. Ces caractéristiques fondamentales jouent un rôle essentiel dans l'interprétation de l'image. Ces caractéristiques incluent notamment :

1.4.1 Pixel

Contraction de l'expression anglaise " Picture Eléments ": éléments d'image, le pixel est le plus petit point de l'image, c'est une valeur numérique représentative des intensités lumineuses. C'est le bit la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et les logiciels sur l'image. La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la figure ci-dessous Figure 1.2. [7]

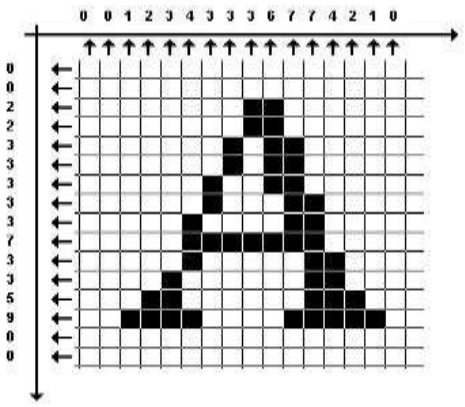


Figure 1. 2:groupe de pixel formant la lettre A

1.4.2. Dimension

La dimension d'une image fait référence à sa taille en pixels. Dans ce contexte l'image se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image. [8]

1.4.2 Taille d'une image

La taille d'une image numérique, aussi appelée poids, désigne la quantité d'espace mémoire nécessaire pour stocker cette image. Elle dépend principalement de deux éléments : le nombre total de pixels, obtenu en multipliant la largeur par la hauteur de l'image, et la profondeur de couleur, c'est-à-dire le nombre de bits utilisés pour représenter chaque pixel [5].

1.4.3 Bruit

Le bruit dans une image désigne des variations soudaines de l'intensité d'un pixel par rapport à ses voisins. Il est souvent causé par des imperfections des capteurs optiques et électroniques ou des variations de l'éclairage. La Figure 1.4 illustre une image sans bruit, tandis que la Figure 1.3 présente une image bruitée [4].

Figure 1. 3: Image Bruitée

Figure 1. 4: Image sans bruit.

1.4.4 Histogramme

L'histogramme d'une image représente la distribution de ses niveaux de gris ou de ses couleurs, indiquant la fréquence d'apparition de chaque niveau d'intensité dans l'image. Cet outil est essentiel dans diverses applications du traitement d'images. Par exemple, il est utilisé pour la correction d'éclairage, permettant d'atténuer les erreurs de quantification et d'homogénéiser la luminosité entre différentes images. De plus, il sert dans la comparaison d'images**,** en analysant des images prises sous des conditions d'éclairage variables pour en évaluer les différences. Enfin, l'histogramme est un outil clé pour l'amélioration de la qualité d'image, en ajustant le contraste et en extrayant les informations essentielles nécessaires à une meilleure interprétation de l'image [4]. La Figure 1.5 illustre une image accompagnée de son histogramme.

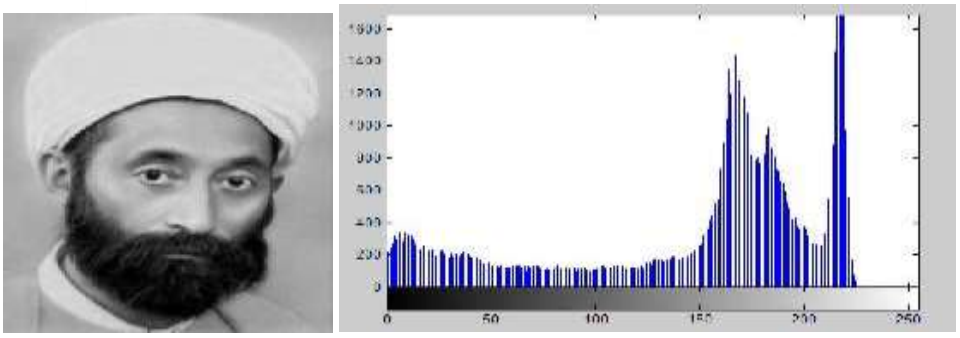


Figure 1. 5: Image avec histogramme.

1.5 Types d'image

Il existe trois types principaux d'images en traitement d'image : les images en niveaux de gris, les images binaires et les images en couleurs.

1.5.1 Images binaires

Les images binaires sont les plus simples, car chaque pixel n'a que deux valeurs possibles : 0 pour le noir et 1 pour le blanc. Ainsi, le niveau de gris est codé sur un seul bit par pixel. [9]

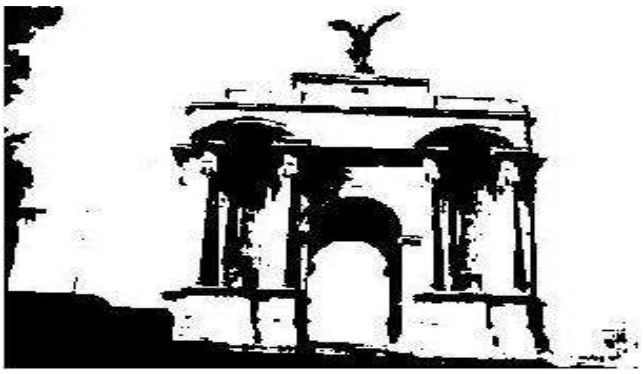


Figure 1. 6*:* Image binaire.

1.5.2 Images niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par 1 bit, mais par 1 octet. Pour cela, il faut que le matériel utilisé pour afficher l'image, soit capable de produire les différents niveaux de gris correspondant. Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la " couleur " de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux. [9]

Figure 1. 7: Images niveaux de gris

1.5.3 Images en couleurs

Les images couleur sont représentées à partir des trois composantes primaires rouge, vert et bleu (RVB), chacune codée sur 8 bits.
Un pixel est noir si R = V = B = 0 et blanc si R = V = B = 255. On peut représenter une image couleur par une combinaison des trois composantes ou par trois images séparées. [2]

Figure 1. 8: Images en couleurs

1.6 Représentation d'une image numérique

Une image numérique est représentée sous forme d'une grille de pixels (éléments d'image), chacun ayant une valeur numérique correspondant à sa couleur. Cette grille constitue une matrice où chaque case contient l'information d'un pixel. Selon le type d'image, ces valeurs peuvent représenter des niveaux de gris (en images monochromes) ou des composantes de couleur (comme les canaux rouge, vert, bleu dans une image RVB). L'image est donc traduite en données binaires, permettant son traitement, son stockage et son affichage par des dispositifs numériques.

La représentation d'une image en format numérique repose sur un ensemble de processus permettant de transformer une image continue en une image discrète, exploitable par des systèmes informatiques. [4]

1.6.1 Échantillonnage

L'échantillonnage est le procédé de discrétisation **s**patiale d'une image consistant à associer à chaque zone rectangulaire ***R(x,y)*** d'une image continue une unique valeur ***I(x,y).*** On parle de sous-échantillonnage lorsque l'image est déjà discrétisée et qu'on diminue le nombre d'échantillons. [4]

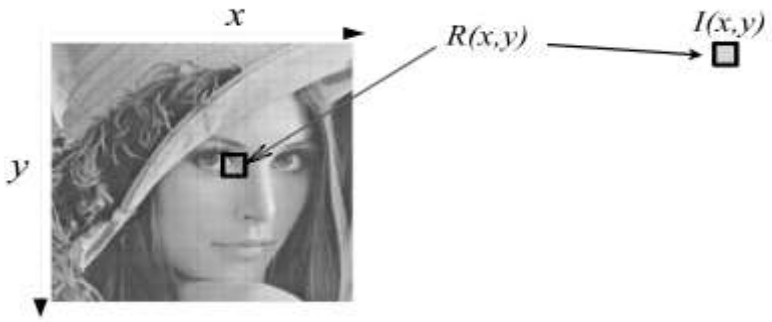


Figure 1. 9: Echantillonnage

1.6.2 Quantification

La quantification désigne la limitation du nombre de valeurs différentes que peut prendre ***I(x,y),*** [4] nombre déterminé en pratique par le nombre d'octets sur lequel on code la valeur numérique en question.

Figure 1. 10: Quantification

Ainsi, une image numérique est le résultat combiné de ces deux étapes fondamentales : l'échantillonnage et la quantification.

Figure 1. 11: Combinaison de l'échantillonnage et de la quantification

1.6.3 Résolution

La résolution d'une image désigne le nombre de pixels par unité de longueur, exprimé en dpi (dots per inch) ou ppp (points par pouce)**.** Elle détermine le niveau de détail visible : plus la résolution est élevée, plus l'image contient d'informations, ce qui augmente aussi son poids **[10] .**Contrairement à la définition**,** utilisée pour les écrans, la résolution s'applique aux images **[11].** Lors de la numérisation, elle est calculée à partir de la taille physique de l'objet (en pouces ou centimètres) et donne les dimensions de l'image en pixels via les formules :

*(X * résolution) = x pixels* *(Y * résolution) = y pixels*

X et ***Y*** sont les dimensions de l'objet**,** ***x*** et ***y*** les dimensions en pixels **[12].** Il est à noter que la résolution peut être distinguée en deux types : la résolution spatiale (capacité à discerner les petits détails) et la résolution tonale (capacité à distinguer de faibles variations d'intensité).
La conversion d'une image analogique en image numérique repose sur deux étapes fondamentales : l'échantillonnage et la quantification**.**

1.6.4 Codage des couleurs (Modèles de couleurs)

Il existe plusieurs façons de coder les couleurs. Le système RVB et le système CMJN. Le système CMJN est utilisé pour l'impression **[2]**, tandis que le système RVB est utilisé pour la lumière (écran, projecteurs, ...).

A. Le système RVB

Il existe plusieurs façons de décrire les couleurs en informatique. Le codage RVB, qui est utilisé notamment dans les formats d'image JPEG et TIFF. Les trois composantes rouge vert bleu, abrégé RVB (ou RGB de l'anglais red, green, blue), est un format de codage des couleurs. Ces trois couleurs sont les couleurs primaires en synthèse additive. Elles correspondent en fait à peu près aux trois longueurs d'ondes auxquelles répondent les trois types de cônes de l'œil humain. L'addition des trois donne du blanc pour l'œil humain. **[2]**.Elles sont utilisées en éclairage afin d'obtenir toutes les couleurs visibles par l'homme : en vidéo, pour l'affichage sur les écrans, et dans les logiciels d'imagerie. Si vous regardez un écran de télévision couleur avec une loupe, vous allez voir apparaître des groupes de trois points lumineux : un rouge, un vert et un bleu. La combinaison de ces trois points donne un point lumineux (un pixel) d'une certaine couleur. Le système RVB est une des façons de décrire une couleur en informatique. Ainsi le triplet {255, 255, 255} donnera du blanc, {255, 0, 0} un rouge pur, {100, 100, 100} un gris, etc. Le premier nombre donne la composante rouge, le deuxième la composante verte et le dernier la composante bleue.

B. Le système CMJN

La quadrichromie ou CMJN (cyan, magenta, jaune, noir ; en anglais CMYK, cyan, magenta, yellow, key) est un procédé d'imprimerie permettant de reproduire un large spectre colorimétrique à partir des trois teintes de base (le cyan, le magenta et le jaune ou yellow en anglais) auxquelles on ajoute le noir (key en anglais). L'absence de ces trois composantes donne du blanc tandis que la somme des trois donne du noir. Toutefois, le noir obtenu par l'ajout des trois couleurs Cyan, Magenta

et Jaune n'étant que partiellement noir en pratique (et coûtant cher), les imprimeurs rajoutent une composante d'encre noire. [2]

1.6.5 Format de fichier

Le format de fichier définit la manière dont l'image est stockée numériquement. Chaque format présente des caractéristiques spécifiques en termes de compression, qualité, transparence ou compatibilité. Parmi les formats les plus courants, on peut citer : JPEG, PNG, BMP, TIFF et GIF. Un format d'image désigne la manière dont une image numérique est codée, stockée et compressée dans un fichier informatique. Il détermine à la fois la structure des données (couleurs, transparence, animation, métadonnées…) et la compatibilité avec les logiciels et navigateurs.

♣ **Le format JPG (ou jpeg) :** Le format le plus communément utilisé pour toute image standard est le JPG. C'est un format qui permet de compresser considérablement les images sans trop les altérer.

♣ **Le format PNG :** Dans le format PNG, chaque pixel est codé avec trois nombres réels compris entre 0 et 1 représentant les composantes rouge, verte et bleue. Le format PNG est non destructeur, c'est-à-dire qu'il ne provoque pas de dégradation de l'image, et offre 9 niveaux de compression à l'image. Néanmoins, on l'utilise généralement lorsque l'on souhaite avoir des transparences dans l'image.

♣ **Le format GIF :** Le format GIF permet la transparence, comme le PNG, mais est plus destructeur, car il est limité à 256 couleurs, contre 16 millions pour le PNG. Son principal avantage est la possibilité de créer des images animées. Chaque pixel GIF est codé sur 4 octets : les trois premiers pour les couleurs rouges, verte et bleue, et le quatrième pour le canal alpha (transparence), dont la valeur varie de 0 (transparent) à 255 (opaque).

♣ **Le format WebP :** Le format WebP, développé par Google en 2010, offre un excellent rapport qualité/compression. Il prend en charge la transparence, les images animées et les images HDR. Grâce à un algorithme de compression par prédiction des pixels adjacents, il génère des fichiers généralement plus petits que le JPEG ou le PNG, ce qui améliore les temps de chargement des pages web.

♣ **TIFF :** Format professionnel, sans perte, souvent utilisé en imagerie médicale ou édition.

♣ **Les images en niveaux de gris :** Les images en niveaux de gris vont bien sur amener de la nuance dans le rendu. Cette nuance va être amenée par la valeur des pixels qui ne sera plus binaire mais qui évoluera dans une fourchette (par exemple de 0 a 255). Plus la valeur de pixel sera proche de zéro plus le pixel sera sombre, et inversement plus cette valeur sera proche de 255 plus il sera clair. La matrice qui représente ce type d'image est donc toujours une matrice de dimension 2 mais avec des valeurs oscillantes (par exemple entre 0 et 255).

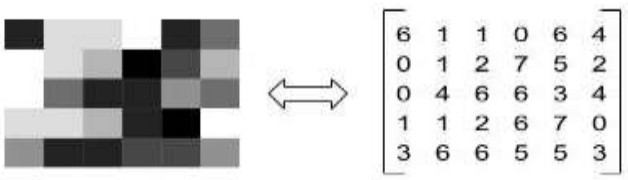


Figure 1. 12: image niveau de gris

1.7 Processus de Traitement d'Image

À l'ère du numérique où l'image occupe une place centrale dans la recherche et la communication, le traitement d'image joue un rôle fondamental dans l'extraction d'informations visuelles. Ce processus suit généralement plusieurs étapes clés :

1.7.1 Acquisition et Représentation des Images

La capture d'image est l'un des maillons les plus importants de toute chaîne de conception et de production d'image. Afin de traiter des images dans un système informatique, elles doivent d'abord être converties pour les rendre lisibles et gérables par ces moyens. La conversion de l'objet externe (image d'origine) vers la représentation interne (dans l'unité de traitement) est réalisée par un processus de numérisation. Ces systèmes d'acquisition sont appelés systèmes d'acquisition optique et peuvent être divisés en deux catégories : les appareils photo numériques et les scanners. [1]

1.7.2 Prétraitement des images (Filtrage)

Le prétraitement il a pour but la réduction des bruits induits par les capteurs afin d'améliorer la perception de certains détails. Cette étape renforce aussi la ressemblance entre pixels de même région et la dissemblance des pixels de régions différents surtout dans les cas où le contraste et la luminosité sont faibles. On se propose présenter quelques ces prétraitements.

❖ Filtrage numérique

Les images acquises à partir de capteurs ou d'appareils photo numériques sont fréquemment altérées par différents types de bruits (bruit gaussien, bruit impulsionnel, bruit de sel et poivre, etc.) qui dégradent leur qualité visuelle et réduisent la performance des algorithmes de traitement d'image. Afin de remédier à cette problématique, des techniques de filtrage numérique sont mises en œuvre pour améliorer la qualité des images et faciliter les traitements ultérieurs tels que la segmentation, la détection de contours ou la reconnaissance de formes. [2]

A. Filtres Linéaires

Les filtres linéaires sont basés sur des opérations de convolution entre l'image et un masque (ou noyau), généralement de petite taille (ex. : 3×3 ou 5×5), appliqué à chaque pixel de l'image. [2]
Ces filtres permettent de manipuler la fréquence spatiale de l'image.

♣ **Filtre passe-bas** : Ce type de filtre atténue les hautes fréquences de l'image, ce qui a pour effet de lisser les détails fins et de réduire certains types de bruit. Toutefois, il tend à produire un **effet de flou**, ce qui peut masquer certaines informations utiles.

♣ **Filtre moyen (ou moyenneur)** : Il constitue une version simplifiée du filtre passe-bas. Chaque pixel est remplacé par la moyenne arithmétique de ses voisins dans la fenêtre locale. Bien qu'il soit simple à implémenter, il est peu efficace contre les bruits impulsionnels.

♣ **Filtre gaussien** : Il s'agit d'une version plus avancée du lissage. Il applique une pondération des pixels selon une distribution gaussienne**,** caractérisée par un paramètre d'écart-type (σ). Ce filtre préserve mieux les structures que le filtre moyen tout en réduisant le bruit.

♣ **Filtre passe-haut** : Contrairement au passe-bas, ce filtre amplifie les hautes fréquences, mettant ainsi en évidence les contours et les transitions de luminosité. Il est utile pour l'extraction de bords et l'amélioration des détails.

♣ **Filtre passe-bande** : Ce filtre combine les effets des filtres passe-bas et passe-haut en ne conservant qu'une plage spécifique de fréquences. Il est utilisé pour extraire des structures particulières dans une image**.** [9]

B. Filtres Non Linéaires

Les filtres non linéaires ne reposent pas sur des opérations de convolution, mais sur des opérations statistiques appliquées aux pixels dans une fenêtre locale. Ils sont particulièrement efficaces pour traiter les bruits impulsionnels, sans provoquer de flou important.

♣ **Filtre médian** : L'un des filtres non linéaires les plus courants. Il remplace la valeur du pixel central par la **médiane** des valeurs de son voisinage. Ce filtre conserve les contours tout en éliminant efficacement les bruits de type "sel et poivre".

♣ **Filtre maximum** : Ce filtre remplace chaque pixel par la valeur maximale de son voisinage. Il est utile pour éliminer les tâches sombres sur fond clair.

♣ **Filtre minimum** : À l'inverse, ce filtre remplace chaque pixel par la valeur minimale dans la fenêtre locale. Il est adapté au traitement des points lumineux isolés**.** [9]

1.7.3 Amélioration du contraste

A. Egalisation histogramme

L'égalisation d'histogramme est une technique simple mais puissante d'amélioration du contraste, particulièrement adaptée aux images à faible dynamique de niveaux de gris. Elle consiste à redistribuer les intensités des pixels pour obtenir une répartition plus uniforme des niveaux de gris, ce qui met en valeur les détails cachés, notamment dans les zones sombres ou surexposées [13]. Techniquement, cette méthode repose sur le calcul de l'histogramme cumulé de l'image. À chaque niveau de gris x_k, on applique une transformation définie par :

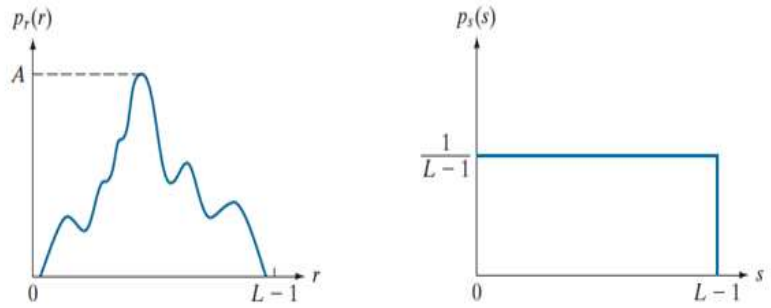


Figure 1. 13: Transformation de l'histogramme-Egalisation

$$T(x_k) = (L-1)\sum_{j=0}^{k} p_x(x_j) = \frac{(L-1)}{n}\sum_{j=0}^{k} n_j \qquad \textbf{(1)}$$

- L est le nombre total de niveaux de gris (256 pour une image en 8 bits),
- $p_x(x_j)$ est la probabilité d'occurrence du niveau x_j
- n_j est le nombre de pixels ayant l'intensité x_j
- n est le nombre total de pixels dans l'image.

Cette transformation étale l'histogramme sur toute l'échelle des niveaux de gris, ce qui améliore la perception visuelle. L'un de ses grands avantages réside dans son automatisation totale et sa capacité à s'adapter sans paramétrage préalable, ce qui la rend idéale pour les systèmes embarqués et les traitements d'images en temps réel. Elle est particulièrement efficace en cas d'éclairage faible ou non uniforme. **[13]**

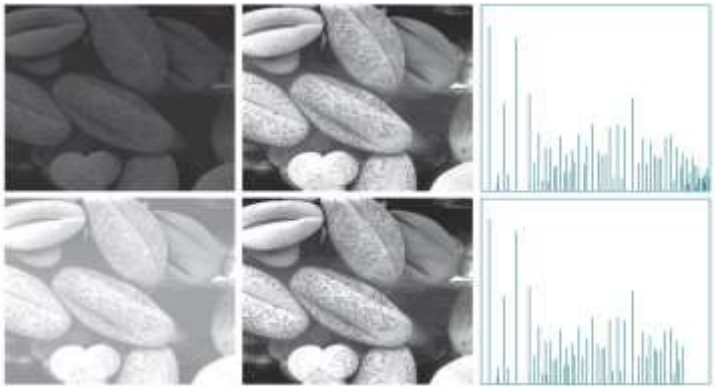


Figure 1. 14: Amélioration de l'image par égalisation d'histogramme

B. Le Dé-bruitage

Le dé-bruitage vise à restaurer une image bruitée en supprimant les perturbations tout en conservant les détails utiles. Cette tâche requiert des algorithmes capables de distinguer le bruit des structures pertinentes. Elle intervient lorsque l'image a été dégradée au moment de la capture, la transmission

ou la compression. Des techniques adaptées à chaque type de bruit ont été développées, soit dans le domaine spatial, soit dans le domaine fréquentiel [14].

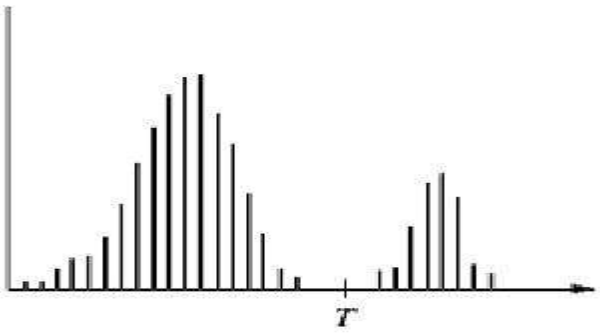
Figure 1. 15: Le débruitage d'image

1.7.4 Segmentation des images

La segmentation d'images est une étape essentielle du traitement d'image, visant à diviser une image en régions significatives ou objets distincts afin d'en faciliter l'analyse et l'interprétation. Parmi les différentes techniques existantes, la segmentation par seuillage est l'une des méthodes les plus simples et les plus utilisées. Elle permet de distinguer les objets du fond en se basant sur les niveaux d'intensité des pixels.

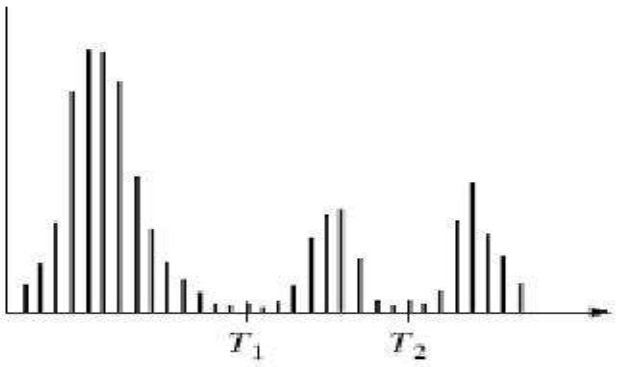
A. Définition du seuillage

Le seuillage (thresholding en Anglais) représente un outil largement utilisé dans la segmentation d'image pour extraire des objets de leurs fonds en fonction d'un seuil. Tout problème de seuillage consiste alors à rechercher la valeur du seuil. La plupart des méthodes de seuillage déterminent le seuil en optimisant une fonction objective.
On distingue le Seuillage de base (simple) (2 classes) où le résultat du seuillage est une image binaire (Figure 1.16), et le multi-seuillage (multi-level thresholding en Anglais) qui est utile quand on a affaire à des images qui contiennent plusieurs objets ayant des luminances différentes. Pour extraire ces objets, plusieurs seuils sont nécessaires. Le résultat du seuillage est une image avec n+1 classes pour n seuils (Figure1.17). [15]



$$g(x,y) = \begin{cases} 0 \; si \; f(x,y) & < T \\ 1 si \; f(x,y) & \geq T \end{cases}$$

Figure 1. 16: Seuillage simple d'un histogramme.



$$G(x,y) = \begin{cases} 2 & si\ f(x,y) \geq T^2 \\ 1 & si\ T'1 \leq f(x,y) < T'^2 \\ 0 & si\ f(x,y) < T'1 \end{cases}$$

Figure 1. 17: Seuillage multiple d'un histogramme.

La segmentation par seuillage d'histogramme constitue un cas particulier de la segmentation par classification. Elle permet de répartir les pixels en classes en fonction de leurs niveaux de gris. Les classes sont alors délimitées par des seuils. Les méthodes de seuillage peuvent être: Seuillage globale et seuillage local, manuelle.

B. Seuillage globale

Le seuillage global est une méthode simple de binarisation d'image consistant à appliquer un seuil unique T à l'ensemble des pixels. Chaque pixel x_i est comparé à ce seuil pour décider s'il devient blanc ou noir. La règle est :

- b_i= 255 si $x_i \geq$T (le pixel est considéré comme appartenant à l'objet),
- b_i= 0 si x_i < T (le pixel est considéré comme appartenant au fond).

C. Seuillage local ou adaptatif

Seuillage local ou adaptatif : un seuil pour une portion de l'image. Les méthodes de seuillage locale prennent en considération la valeur des pixels voisins pour le calcul des seuils. [15]

D. Les Critères d'évaluations

L'évaluation de la qualité d'une segmentation constitue une étape essentielle pour comparer les performances des différents algorithmes. Elle permet de juger dans quelle mesure le résultat d'une segmentation automatique est conforme à une vérité terrain (*ground truth*), généralement construite à partir d'annotations humaines réalisées par des experts. Plusieurs métriques ont été proposées dans la littérature pour quantifier cette conformité, parmi lesquelles les plus utilisées sont : l'Intersection over Union (IoU), **le** Dice Similarity Coefficient (DSC), **l'**indice de similarité structurelle (SSIM). Ces critères sont largement adoptés dans le domaine du traitement d'image, notamment dans des applications sensibles telles que la médecine, la robotique ou l'inspection industrielle.

❖ IoU – Intersection over Union

L'Intersection over Union (IoU), également connu sous le nom d'indice de Jaccard, est l'une des métriques les plus courantes pour la segmentation sémantique. Il est défini comme le rapport entre la surface d'intersection et la surface d'union des régions segmentées et de la vérité terrain :

$$IoU = J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

Où A et B représentent respectivement la vérité terrain et les cartes de segmentation prédites. Cette mesure varie entre 0 (aucune correspondance) et 1 (segmentation parfaite). [16]

❖ **DSC – Dice Similarity Coefficient**

Le coefficient de Dice, parfois appelé indice de Sørensen–Dice, est une autre mesure de similarité entre deux ensembles binaires. [16] :

$$Dice = \frac{2|A \cap B|}{|A| + |B|} \quad (3)$$

Il est particulièrement utilisé dans la segmentation d'images médicales, car il est plus sensible aux petites régions et moins sévère que l'IoU. Cette métrique est très répandue dans l'analyse de segmentation d'organes, de tumeurs, et dans les travaux en radiologie computationnelle.

❖ **SSIM – Structural Similarity Index**

L'indice de similarité structurelle (SSIM) est une mesure perceptuelle introduite par Wang et al. (2004), conçue pour modéliser la manière dont l'œil humain perçoit les différences visuelles entre deux images. Contrairement aux métriques classiques comme le MSE (Mean Squared Error) ou le PSNR (Peak Signal-to-Noise Ratio), le SSIM tient compte de trois composantes fondamentales de l'image : la luminance, le contraste et la structure. [17]

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (4)$$

- μ_x, μ_y: moyennes locales
- σ_x^2, σ_y^2 : variances locales
- σ_{xy} : covariance
- ***C1, C2*** : constantes de stabilisation

Le SSIM retourne une valeur comprise entre 0 et 1, où 1 indique une similarité parfaite, et des valeurs proches de 0 indiquent une forte dissimilarité.

1.7.5 Extraction de contours

L'extraction de contours permet de détecter les bords d'une image, en repérant les variations brutales d'intensité. Ces contours correspondent généralement à des changements de texture, de profondeur ou d'éclairage, et sont essentiels pour l'analyse et la segmentation des objets. On distingue principalement trois types de contours : marche**,** toit et pointe**.**

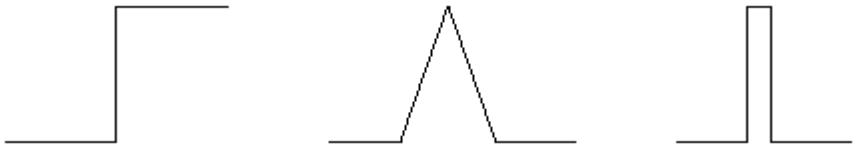


Figure 1. 18: Différents types de contours (marche, toit et pointe)

1.7.6 Classification

A. Apprentissage automatique

L'apprentissage automatique (Machine Learning) est une branche de l'intelligence artificielle qui permet aux machines d'apprendre à partir de données sans être explicitement programmées. Il constitue une technologie clé pour exploiter les grandes quantités de données (Big Data). [18]
Le deep learning, ou apprentissage profond, est une sous-catégorie de cet apprentissage et se divise en trois types : supervisé, où l'on utilise des données étiquetées pour entraîner des modèles tels que les CNN, RNN, LSTM ou GRU [19]. Semi-supervisé, qui combine données étiquetées et non étiquetées, souvent avec des modèles comme les GAN ou les DRL [19]. Non supervisé, où le modèle apprend sans étiquettes, en découvrant des structures cachées à l'aide de techniques comme les auto-encodeurs, les RBM ou les GAN. [20]

B. Réseaux de neurones convolutifs (CNN)

Le nom 'Réseau de neurones à convolution' indique que le réseau emploi une opération mathématique appelée la convolution. Les réseaux de convolution sont un type spécialisé de réseaux neuronaux qui utilisent la convolution à la place de la multiplication matricielle générale dans au moins une de leurs couches. Les CNN sont l'un des meilleurs algorithmes d'apprentissage pour faire l'opération de convolution qui aide à l'extraction de fonctionnalités utiles à partir de points de données corrélés localement. La sortie des noyaux convolutifs est ensuite affectée à l'unité de traitement non linéaire (fonction d'activation), qui non seulement aide à apprendre les abstractions, mais intègre également la non-linéarité dans l'espace des fonctionnalités. Cette non-linéarité génère différents modèles d'activations pour différentes réponses et facilite ainsi l'apprentissage des différences sémantiques dans les images [21].
La topologie de CNN est divisée en plusieurs étapes d'apprentissage composées d'une combinaison des couches convolutives, des unités de traitement non linéaires et des couches de sous-échantillonnage. [21]

C. Règles de décision

Les règles de décision en traitement d'image permettent de classer les pixels ou régions selon leurs caractéristiques. Elles peuvent être simples, comme le seuillage, ou avancées, utilisant des modèles statistiques (ML, MAP) ou l'apprentissage automatique (SVM, réseaux de neurones).Ces

règles sont essentielles pour des tâches telles que la segmentation, la reconnaissance d’objets et l’imagerie médicale.

1.8 Conclusion

Ce chapitre présente les bases du traitement d'image, détaillant les caractéristiques des images numériques, leurs types (binaires, à niveaux de gris, et en couleurs) et les techniques fondamentales de traitement, telles que l'amélioration, la restauration et la segmentation. Nous explorons également les concepts clés comme la résolution, le filtrage, l'égalisation d'histogramme et le débruitage. La segmentation, notamment via le seuillage et le multi-seuillage, constitue un élément crucial du traitement d'image. Ces éléments serviront de base pour les développements dans le chapitre suivant, qui se concentrera sur l'intelligence en essaim, une approche inspirée de la nature pour résoudre des problèmes complexes d'optimisation.

Chapitre 2 : L'intelligence en essaim

1.9 Introduction

En raison des progrès récents de la science informatique, la complexité des problèmes du monde réel à résoudre augmente, et ces problèmes sont souvent caractérisés par la non-linéarité **[22].** Les algorithmes d'optimisation sont généralement conçus pour résoudre ces problèmes, et ils sont classés en (i) algorithmes traditionnels exacts et (ii) méta-heuristiques stochastiques. De nombreux problèmes d'optimisation non déterministes en temps polynomial sont difficiles à résoudre par des algorithmes exacts, quelle que soit la croissance exponentielle de la puissance de calcul. Cependant, comme les algorithmes méta-heuristiques ne nécessitent pas d'informations de gradient, ils sont faciles à mettre en œuvre et peuvent contourner les optima locaux **[23]**. Cette machine autonome adaptée à l'environnement résout les problèmes d'optimisation en ajustant les variables de décision de manière stochastique intelligente et en minimisant ou maximisant la valeur de la fonction objective du problème. **[24] [25]**

En tant que méthode alternative pour atténuer le dilemme des algorithmes exacts, elle est couramment utilisée dans divers domaines en raison de son adaptabilité universelle, sa flexibilité, sa robustesse, son efficacité et de son évitement des optima locaux **[26].**

Au fil du temps, il est devenu plus courant dans la littérature sur les méta-heuristiques d'utiliser des méthodes basées sur la source d'inspiration. Les auteurs de **[27].** Ont classé les algorithmes méta-heuristiques en cinq types, qui sont basés sur l'évolution, les essaims, le comportement humain, la physique et hybrides. Ainsi de nombreuses méthodes de sélection d'attributs utilisent des algorithmes méta-heuristiques pour éviter d'augmenter la complexité de calcul **[28]** Ces algorithmes seront en mesure d'optimiser le problème de sélection d'attributs avec une précision appropriée dans un délai acceptable. Dans ce contexte les algorithmes méta-heuristiques peuvent être classés en deux catégories principales : l'intelligence en essaim (*Swarm Intelligence*) (SI) et les algorithmes évolutionnaires(EA). La figure 08 nous montre une classification de ces algorithmes.

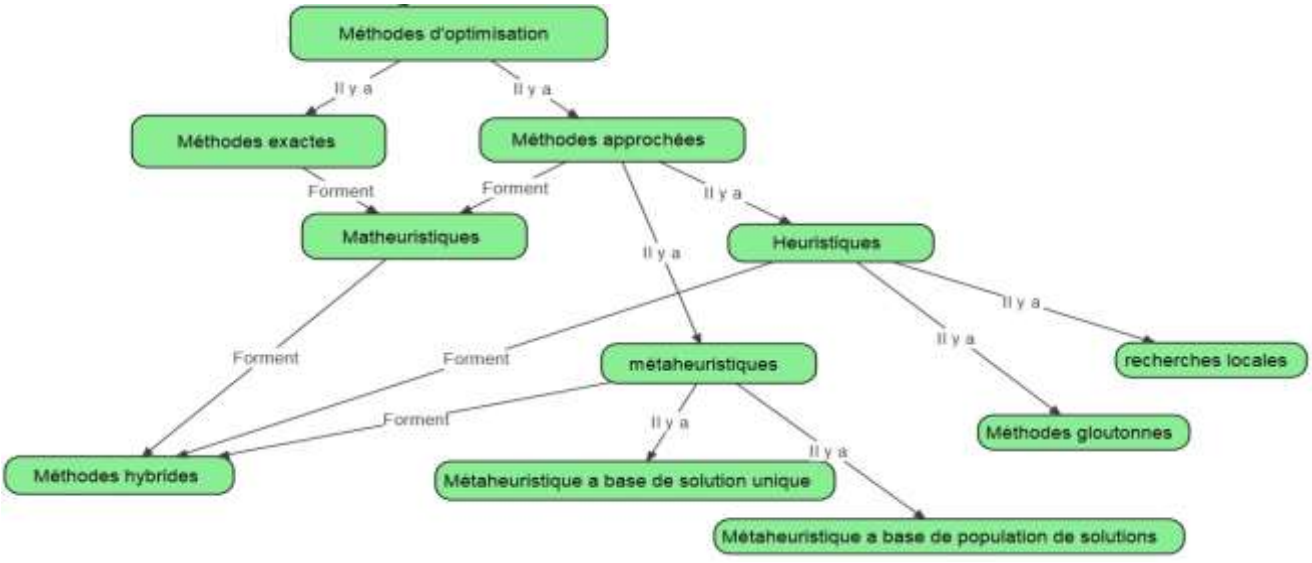


Figure 2. 1: Classification des algorithmes d'optimisation

L'intelligence en essaim est une catégorie relativement nouvelle de calcul évolutif par rapport aux (Evolutionary algorithms) (EA) et à d'autres approches basées sur une solution unique. Les algorithmes SI ont utilisé des techniques approximatives et non déterministes pour explorer et exploiter efficacement l'espace de recherche afin de trouver des solutions quasi optimales. Le groupe méta-heuristique inspiré de la nature le plus populaire est celui des techniques basées sur les essaims. L'intelligence en essaim (SI) est un type d'intelligence artificielle qui repose sur des comportements collectifs dans des systèmes décentralisés et auto-organisés. Ces systèmes sont généralement constitués d'une population d'agents simples qui interagissent localement les uns avec les autres et avec leur environnement.

Il y a longtemps, les gens ont découvert la variété des comportements intéressants d'insectes ou d'animaux dans la nature. Un vol d'oiseaux balaie le ciel. Un groupe de fourmis cherche de la nourriture, etc. Nous appelons ce type de mouvement agrégé "comportement d'essaim". Récemment, des biologistes et des informaticiens ont étudié comment modéliser des essaims biologiques pour comprendre comment ces animaux "sociaux" interagissent, atteignent leurs objectifs et évoluent. De plus, les chercheurs s'intéressent de plus en plus à ce type de comportement d'essaim puisque « l'intelligence d'essaim » résultante peut être appliquée à l'optimisation dans plusieurs domaines. Une vue de haut niveau d'un essaim suggère que les N agents de l'essaim coopèrent pour obtenir un comportement et atteindre un objectif, et ça ce qu'on appelle « intelligence collective ». Les agents utilisent des règles locales simples pour régir leurs actions et, via les interactions de l'ensemble du groupe.

1.10 Définition

Le terme intelligence en essaim a été introduit par Beni et Wang dans le contexte des systèmes robotiques cellulaires. L'intelligence en essaim est l'intelligence collective émergente de groupes d'agents autonomes simples. L'agent autonome ne suit pas les ordres d'un chef, ou un plan global. Par exemple, pour qu'un oiseau participe à un troupeau, il ajuste uniquement ses mouvements pour se coordonner avec les mouvements de ses compagnons de troupeau, généralement ses « voisins » qui sont proches de lui. Un oiseau essaie simplement de rester près de ses voisins, mais évite les collisions avec eux. Chaque oiseau ne reçoit pas des ordres d'un oiseau leader puisqu'il n'y a pas d'oiseau chef. N'importe quel oiseau peut y à l'avant, au centre ou à l'arrière de l'essaim. Le comportement en essaim aide les oiseaux à tirer parti de plusieurs choses, notamment la protection contre les prédateurs et la recherche de la nourriture (essentiellement, chaque oiseau exploite les yeux de tous les autres oiseaux).

Bien que les algorithmes de l'intelligence en essaim utilisent des opérateurs et des stratégies différents, ils doivent satisfaire deux conditions préalables de base lors de la résolution de problèmes d'optimisation : l'exploration et l'exploitation. [29]

L'exploration : est une capacité de recherche globale qui vise à aider l'algorithme à découvrir des régions dans l'espace de recherche prometteuses.

L'exploitation : est une capacité de recherche locale qui permet à l'algorithme de rechercher dans une région prometteuse pour améliorer encore la qualité de la solution. Si les deux capacités sont effectivement équilibrées, la qualité de la meilleure solution sera encore améliorée [30] La recherche complète garantit que les attributs les plus appropriés en termes de qualité sont trouvées.

1.11 Comment fonctionne l'intelligence en essaim

Les systèmes d'intelligence en essaim sont généralement composés d'un grand nombre d'agents simples qui interagissent les uns avec les autres et avec leur environnement afin d'atteindre un objectif commun. Le comportement du système dans son ensemble émerge des interactions entre les agents individuels et est souvent imprévisible et complexe. Ces systèmes naturels sont capables de résoudre des problèmes complexes, comme trouver le chemin le plus court dans un labyrinthe, en les décomposant en sous-problèmes plus petits qui peuvent être résolus par des agents individuels travaillant ensemble. En général, les algorithmes d'intelligence en essaim comportent trois composants principaux :

♣ Population de solutions potentielles (appelées particules),

♣ Fonction objective qui évalue la capacité de chaque particule à résoudre le problème.

♣ Forme de communication ou d'interaction entre les particules afin qu'elles puissent partager des informations sur les solutions qui fonctionnent bien.

Chaque algorithme d'intelligence en essaim doit obéir à certaines phases fondamentales. Ainsi, leurs cadre peut être défini comme suit :

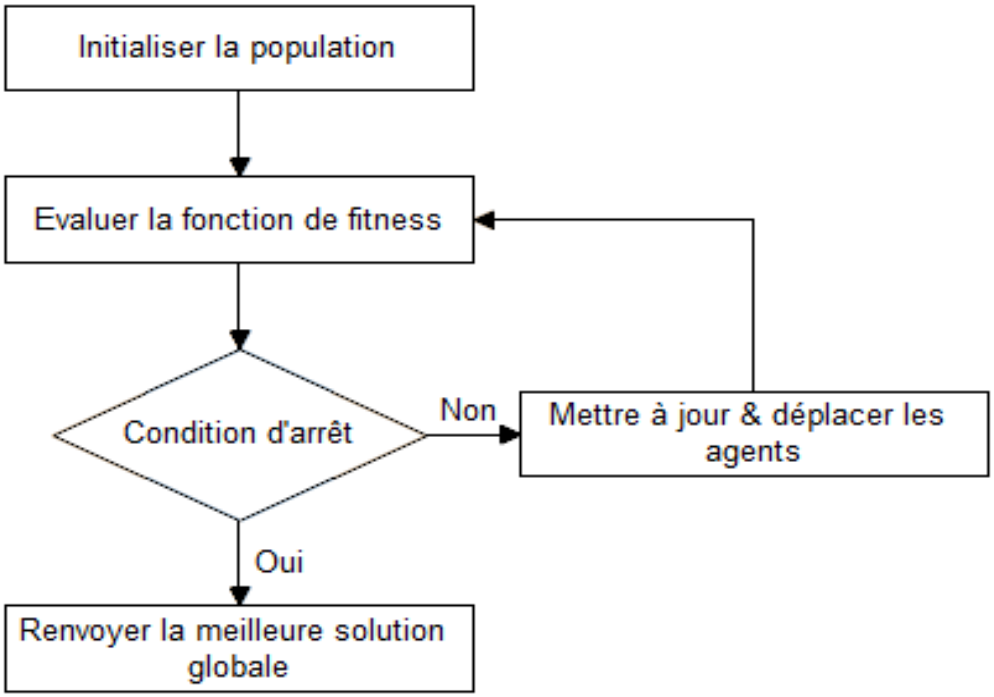


Figure 2. 2: Organigramme de fonctionnement d'intelligence en essaim.

1.11.1 Procédure générale des algorithmes d'intelligence en essaim

Dans les algorithmes SI, une population d'individus, qui indique des solutions potentielles, coopère entre eux et s'améliore statistiquement au fil des itérations, pour finalement trouver des solutions suffisamment bonnes [31]. De manière générale, des différentes opérations tentent d'équilibrer la

convergence et la diversité du processus de recherche, c'est-à-dire l'équilibre entre l'exploration et l'exploitation de l'espace de recherche.

La procédure générale des algorithmes d'intelligence en essaim est résumée dans l'algorithme ci-dessous.

Algorithme 02: Procédure générale de l'algorithme de l'intelligence en essaim

1. **Générer aléatoirement une population de solutions initiales**
2. **Tant que non terminé faire**
   - **Evaluer la qualité de chaque solution**
   - **Sélectionner des solutions avec de meilleures qualité**
   - **Mise à jour de la population**
3. **Selectionner la meilleure solution**

À partir de l'initialisation aléatoire d'une population d'individus dans l'espace de solution, suivie du processus d'évaluation correspondant et du processus de génération de nouvelles solutions, après un certain nombre d'itérations, les algorithmes d'intelligence en essaim pourront éventuellement trouver des solutions acceptables.

En règle générale, la qualité de la solution devrait s'améliorer de manière monotone au fil des itérations, c'est-à-dire la valeur de la fonction objective de la solution au temps t+1, ne devrait pas être pire que la valeur de la fonction objective de la solution au temps t.

1.12 Les applications de l'intelligence en essaim

L'intelligence en essaim s'est révélée utile dans diverses applications, qui peuvent être divisés en deux catégories :

1.12.1 Applications théoriques

Depuis des décennies, l'exploration de données est un sujet académique brûlant dans le domaine des statistiques informatiques. Comme mentionné, l'algorithme SI est principalement utilisé dans les tâches d'exploration de données sous deux formes : le réglage des paramètres ou l'organisation des données. Notamment les principales applications dans la réduction de dimensionnalité, la classification et le regroupement, ainsi que l'apprentissage automatique automatisé **[32].**

♣ **Optimisation :** l'un des cas d'utilisation les plus courants de l'intelligence en essaim concerne les problèmes d'optimisation tels que la minimisation des fonctions, l'optimisation multimodale et le regroupement de données. Les problèmes d'optimisation sont des problèmes dans lesquels trouver une solution à un problème peut se faire en trouvant la meilleure solution possible parmi un nombre colossal de solutions possibles. Ces types de problèmes sont souvent résolus à l'aide de l'intelligence en essaim, qui permet de trouver simultanément de nombreuses solutions différentes, chaque solution étant meilleure que la précédente**. [33]**

♣ **Réduction de la dimensionnalité**

La réduction de la dimensionnalité est le processus de réduction du nombre de variables ou d'attributs aléatoires dans un ensemble de données considéré. Il joue un rôle essentiel dans le prétraitement des données pour l'exploration de données. Il existe généralement deux opérations de réduction de dimensionnalité, la sélection d'attributs et l'extraction d'attributs.

La sélection d'attributs est un processus de sélection d'un sous-ensemble optimal d'attributs pertinents à utiliser dans la construction de modèles. Alors que l'extraction d'attributs est un processus de projection de données originales dans un espace de grande dimension sur un espace plus petit. La précision d'un modèle sera améliorée en utilisant d'attributs judicieusement projetés et/ou sélectionnées plutôt que tous les attributs disponibles dans une grande quantité de données.

Étant donné que FS est un problème d'optimisation combinatoire NP-difficile, les algorithmes SI s'avèrent être une option prometteuse pour résoudre ce type de problèmes. De nombreux travaux connexes ont vu le jour récemment, en voici quelques exemples : Gu et al, ont proposé une méthode de sélection d'attributs pour la classification de haute dimension basée sur une variante très récente de PSO, connue sous le nom de optimiseur d'essaim compétitif (*Competitive Swarm Optimizer*) (CSO) **[34].** Zhang et all **[35].** ont conçu une méthode basée sur (*Firefly Algorithm*) (FA) pour la sélection d'attributs, qui a la capacité d'empêcher une convergence prématurée. Farhad Pourpanah et all **[36]**. ont combiné le modèle Fuzzy ARTMAP (FAM) avec l'algorithme (*Brain Storm Optimization*) (BSO) pour les tâches de sélection d'attributs, etc. Une étude plus détaillée sur la sélection d'attributs alimentée par l'intelligence en essaim peut être trouvée dans **[37].**

♣ Classification et regroupement

La classification et le regroupement sont des aspects essentiels de la science des données. Ils ont été largement étudiés au fil des décennies dans le domaine des statistiques, des réseaux de neurones, de l'apprentissage automatique. En général, la classification consiste à prédire la classe cible en analysant l'ensemble de données d'entraînement, tandis que le regroupement consiste à regrouper le type similaire de cibles en considérant la condition la plus satisfaisante.

Les applications SI dans ces deux aspects sont principalement liées au réglage des paramètres. Pour la classification, on peut trouver dans la littérature des travaux qui combinent des algorithmes SI avec un modèle de régression **[38]**, une machine à vecteurs de support (SVM) **[39] [40] [41],** des classificateurs du k-voisin le plus proche **[42] [43],** des arbres de décision **[44] [45],** ainsi que les réseaux de neurones **[46] [47].** Pour le regroupement, certains travaux récents sont liés à l'utilisation du SI avec des k-moyennes **[48] [49] [50].**

♣ Apprentissage automatique automatisé

La recherche et l'application de l'apprentissage automatique ont connu une croissance explosive, en particulier les réseaux de neurones profonds (*Deep Neural Network*) (DNN) qui ont fait de grands progrès dans de nombreux domaines d'application. Cependant, les performances de nombreuses méthodes d'apprentissage automatique sont très sensibles à un trop grand nombre de décisions de conception. En particulier, la conception de l'architecture des DNN est très complexe et repose fortement sur les connaissances préalables des experts. Pour résoudre ce problème, de nombreuses méthodes basées sur le SI sont proposées pour concevoir automatiquement des DNN **[51].** Wang et all. **[52]** proposent une approche efficace d'optimisation par essaim de particules (EPSOCNN) pour

concevoir automatiquement les architectures de réseaux de neurones convolutifs (CNN). Plus précisément, afin de réduire le coût de calcul, EPSOCNN minimise l'espace hyper-paramétrique des CNN à un seul bloc et évalue les CNN candidats avec le petit sous-ensemble de l'ensemble d'apprentissage. B.Wang et all. [53] ont proposé des CNN évolutifs multi-objectifs (MOCNN) pour rechercher les architectures CNN non dominantes sur le front de Pareto en termes d'objectif de précision de classification et d'objectif de coût de calcul. Il introduit une nouvelle stratégie de codage pour coder les CNN et utilise une optimisation par essaim de particules multi-objectifs (OMOPSO) pour optimiser les architectures de CNN candidates.

1.12.2 Applications dans le monde réel

L'intelligence en essaim a été appliquée pour résoudre divers problèmes dans différents domaines, tels que la conception technique, l'informatique et l'économie. Le SI s'est révélé utile dans diverses applications, notamment :

♣ **Internet des objets:** L'Internet des objets (IoT) est une application du monde réel dans laquelle les algorithmes SI ont été largement utilisés [54]. Par exemple, dans les systèmes basés sur l'IoT, l'algorithme SI a été utilisé pour la planification des tâches [55]. Dans les villes intelligentes basées sur l'IoT, les algorithmes SI ont été utilisés en raison de leur attribut basé sur la population pour rendre le système flexible et évolutif.

♣ **Reconnaissance de formes:** la reconnaissance des formes est la capacité d'identifier des modèles. C'est une tâche difficile pour les humains, car nous devons généralement examiner chaque point de données avant de pouvoir identifier le modèle. Mais une façon de rendre les choses plus faciles consiste à utiliser l'intelligence en essaim. Cela peut être utilisé pour des tâches telles que la classification, le regroupement et la prédiction [56].

♣ **Robotique:** SI peut être utilisé pour contrôler des groupes de robots. Les robots qui utilisent les principes de l'intelligence en essaim ont été utilisés pour des tâches telles que la recherche et le sauvetage, la surveillance et l'exploration. Par exemple, une application robotique en essaim, qui vise à construire des robots simples, en travaillant ensemble en essaims. En SI, des robots simples sont programmés avec des règles très basiques et ils sont lâchés dans un environnement où ils doivent accomplir une tâche. Les robots communiquent entre eux pendant qu'ils travaillent, partageant des informations sur la tâche à accomplir. Au fur et à mesure qu'ils communiquent, les robots apprennent les uns des autres et déterminent la meilleure façon d'accomplir la tâche et construire un bon modèle [57].

♣ **Simulation :** les simulations avec l'intelligence en essaim peuvent être utilisées pour étudier la dynamique de systèmes complexes tels que les réseaux sociaux, les marchés économiques et les écosystèmes écologiques. L'intelligence en essaim est utilisé à des fins de simulation afin de comprendre comment de grands groupes de personnes se comportent et prennent des décisions. Avec l'aide d'intelligence en essaim, les scientifiques peuvent explorer de nombreux résultats différents afin de trouver des solutions à certains problèmes auxquels la société est confrontée aujourd'hui [58].

♣ **Modélisation prédictive:** les techniques de l'intelligence en essaim ont été utilisées à des fins prédictives. La modélisation prédictive est un processus dans lequel les données sont analysées pour prédire des événements futurs. Il peut être utilisé à des fins diverses, depuis la prévision des tendances boursières jusqu'à la prévision de la meilleure façon de procéder pour une procédure médicale.

L'algorithme intelligence en essaim utilise ces données et les combine avec d'autres informations provenant de son environnement afin de prendre des décisions sur ce qui, selon lui, se produira dans le futur. Il s'est avéré utile dans des secteurs tels que la santé, le marketing et la finance, ainsi que dans de nombreux autres domaines où une modélisation prédictive est nécessaire [59].

1.13 Avantages et caractéristiques de l'intelligence en essaim

Les avantages de l'utilisation de l'intelligence en essaim sont nombreux et variés. En exploitant la puissance de l'intelligence collective, les organisations peuvent résoudre les problèmes de manière plus efficace et efficiente.

1.13.1 Avantages

Les principaux avantages de l'intelligence en essaim sont:

♣ **L'autonomie**: il n'existe aucune autorité centrale ni aucun contrôleur dictant la manière dont le système doit fonctionner. Au lieu de cela, chaque agent est autonome et prend ses propres décisions sur la base d'informations locales et de règles simples. Cette décentralisation permet aux systèmes SI d'être hautement évolutifs et efficaces, car ils peuvent facilement ajouter ou supprimer des agents sans affecter les performances globales du système [60]. SI n'ont pas de gestion extérieure, mais chaque agent de l'essaim contrôle son comportement de manière autonome. L'agent dans ce cas représente une solution possible à un problème donné. Sur cette base, nous pouvons déduire le deuxième avantage, qui est l'auto-organisation.

♣ **L'auto-organisation**: L'intelligence ne se concentre pas sur l'agent individuel, mais émerge dans l'essaim lui-même. Par conséquent, les solutions (agents) du problème ne sont pas connues à l'avance, mais se modifient au moment de l'exécution du programme. L'auto-organisation joue un rôle important dans l'adaptabilité. Ce dernier est perceptible dans des environnements changeants où les agents réagissent bien aux changements mentionnés, modifient leur comportement et s'y adaptent de manière autonome. De plus, comme il n'y a pas de coordination centrale, l'essaim est robuste, car il n'y a pas de point de défaillance unique. De plus, dans l'essaim se cachent deux autres avantages. A savoir l'évolutivité et la flexibilité.

♣ **L'évolutivité**: Les systèmes de l'intelligence en essaim peuvent facilement s'adapter à un grand nombre d'agents, Ce qui signifie que l'essaim peut être composé de quelques milliers d'agents et, dans tous les cas, l'architecture de contrôle reste la même.

♣ **Flexibilité :** est pleinement satisfaite car **(i)** Il n'y a pas de chefs chez les agents d'un essaim, chacun travaille pour le bien-être des autres, tous les membres de la population sont considérés comme des leaders. Aucune autorisation n'est requise, et chaque membre travaille selon les informations reçues du plus proche ou collectivement. Le plus souvent, ils ne connaissent pas la situation dans son ensemble. Nous prenons comme exemple chez les abeilles, deux types d'informations sont partagés, les informations sur la nourriture et les menaces.

Lorsque certaines abeilles trouvent une bonne source de nectar, elles exécutent une danse agitée pour signaler aux autres qu'elles sont en sécurité.

Lorsqu'elles recherchent un nouvel endroit pour installer des ruches, les abeilles exécutent une danse frétillante pour signaler aux autres.

♣ **Aucune commande ou autorisation requise:** Les agents de la population fonctionnent sur une structure de partage d'informations sans ordre formel ni autorisation d'entreprendre une tâche. Le système est transparent et il y a une compréhension claire du rôle de chacun. Les commandes doivent être correctement comprises dans leur contexte et plus susceptibles de contenir des erreurs.

1.13.2 Caractéristiques

Un essaim peut être considéré comme un groupe d'agents coopérant pour adopter un comportement déterminé et atteindre un objectif. Cette intelligence collective semble émerger de groupes souvent de grande taille. Selon Milonas, cinq principes de base définissent le paradigme SI **[61].**

A. **Principe de proximité :** l'essaim doit pouvoir réaliser des déplacements spatio-temporels à simples calculs.
B. **Principe de qualité:** l'essaim doit être capable de répondre aux facteurs de qualité de l'environnement.
C. **Principe de la diversité des réponses:** l'essaim ne doit pas engager ses activités excessivement étroites.
D. **Principe de stabilité:** l'essaim ne doit pas changer de comportement à chaque fois au moment où l'environnement change.
E. **Principe d'adaptabilité:** l'essaim doit être capable de changer de comportement lorsque cela en vaut le prix informatique. Noter que les principes quatre (*D*) et cinq (*E*) sont les faces opposées d'une même médaille.

♣ **Caractéristique par rapport aux autres modèles**

De nombreux modèles en informatique ont été inspirés de la nature, mais ne peuvent pas être tous considérés comme une intelligence d'essaim. Les essaims impliquent généralement le mouvement des individus à travers un espace de représentation, et tous les algorithmes inspirés de la nature ne le font pas. Par exemple :

Les réseaux de neurones artificiels et les algorithmes évolutionnaires sont des algorithmes d'inspiration biologique (basés sur les principes des neurosciences et de l'évolution, respectivement), et ont été utilisés pour l'exploration de données, mais ne sont pas considérés comme intelligence en essaims. Les réseaux de neurones sont distribués mais statiques, les algorithmes évolutionnaires sont basés sur la reproduction plutôt que sur le mouvement. De même, les algorithmes génétiques sont basés sur le clonage et la mutation, plutôt que sur le comportement d'essaim.

1.14 Évolution des techniques d'intelligence en essaim

L'évolution des différents algorithmes d'intelligence en essaim et de leurs différentes variantes développées au fil des années, se présenter dans le tableau suivant.

Tableau 2. 1: Algorithmes populaires d'intelligence en essaim

Année	Algorithme proposé	Inspiration
2022	Crocodiles hunting strategy **[62]**	Simule le comportement des crocodiles à la chasse
2021	Flamingo Search Algorithm **[63]**	Il est basé sur le comportement de migration et de recherche de nourriture des flamants roses
2021	Horse herd Optimization Algorithm **[64]**	Il met en œuvre ce que font les chevaux à différents âges en utilisant six caractéristiques importantes : le pâturage, la hiérarchie, la sociabilité, l'imitation, le mécanisme de défense et l'errance.
2020	Black Widow Optimization Algorithm **[65]**	Il s'appuie notamment sur l'intéressant comportement de cannibalisme fraternel proposé par l'araignée Black Widow.
2020	Sparrow Search Algorithm **[66]**	Cet algorithme est basé sur des techniques intelligentes utilisées par les moineaux pour rechercher de la nourriture en fonction de la situation dans laquelle ils se trouvent.
2020	Rat Swarm Optimizer **[67]**	Il s'inspire du comportement de la chasse et d'attaque des rats.
2019	The Sailfish Optimizer **[68]**	Cet algorithme utilise la population de voiliers lors de la chasse et de la recherche de sardines.
2018	Grasshopper Optimization Algorithm **[69]**	Il s'inspire du comportement d'essaim des sauterelles lors de la recherche de nourriture.
2017	Salp Swarm Algorithm **[70]**	Il s'inspire du comportement d'essaim des salpes dans les océans.
2017	Camel Herds Algorithm **[71]**	Cet algorithme est basé sur les chameaux, et comment ils ont un chef pour chaque troupeau et comment ils recherchent de la nourriture et de l'eau en fonction de l'humidité des lieux voisins.
2016	Dragonfly Algorithm **[72]**	Il est basé sur le comportement statique et dynamique des libellules (Dragonfly).
2016	Dolphin Swarm algorithm **[73]**	Il est basé sur les caractéristiques biologiques et les habitudes de vie telles que l'écholocation, les échanges d'informations, la coopération et la division du travail des Dauphins.
2016	Crow Search Algorithm **[74]**	Il est basé sur la façon dont les corbeaux recherchent de la nourriture, cachent leur nourriture aux autres corbeaux et se souviennent de leurs cachettes.
2015	Ant Lion Optimizer **[75]**	Cet algorithme imite la nature de chasse des fourmis-lions dans la nature.
2015	Elephant Herding Optimization **[76]**	Il est basé sur le comportement grégaire des différents groupes d'éléphants.
2015	Moth-flame Optimization algorithm **[77]**	Il est basé sur la méthode de navigation des papillons de nuit dans la nature appelée orientation transversale.
2014	Grey Wolf Optimizer **[78]**	Il imite la hiérarchie et le comportement de chasse des gris loups dans la nature.

2013	Spider Optimization [79]	Il est basé sur les caractéristiques de coopération des araignées.
2012	Bacterial Colony Optimization [80]	Il est basé sur le cycle de vie d'une bactérie nommée, E. Coli.
2012	Zombie Survival Optimization [81]	Il est basé sur le comportement de recherche de nourriture des zombies, lorsqu'ils trouvent un hypothétique antidote aéroporté qui guérit leurs maux.
2010	Bat Algorithm [82]	Il est basé sur l'écholocation de micro-chauves-souris dans la nature, avec des taux d'émission et intensité d'impulsions variables.
2009	Cuckoo Search [83]	Il s'inspire du comportement des coucous pour pondre des œufs dans les nids d'oiseaux d'autres espèces.
2008	Fast Bacterial Swarming Algorithm [84]	Cet algorithme combine le comportement de recherche de nourriture d'E. Coli à partir de l'algorithme de colonie de bactéries et le mécanisme et l'algorithme d'essaim de particules de flocage des oiseaux.
2007	Firefly Algorithm [85]	Il s'inspire des lucioles de nature
2006	Cat Swarm Optimization [86]	Il est basé sur le comportement de chats, et comprend deux sous-processus - le mode de traçage et mode de recherche.
2005	Artificial Bee Colony [87]	Cet algorithme simule comment les colonies d'abeilles mellifères travaillent, y compris les employés, les ouvriers et éclaireurs.
2004	Honey Bee Algorithm [88]	Il est basé sur la façon dont les abeilles mellifères cherchent de la nourriture dans la nature.
1995	Particle swarm optimization (PSO) [89]	PSO est un algorithme de recherche basé sur la population et imite comportement social des oiseaux au sein d'un troupeau.
1992	Ant Colony Optimization [90]	Cet algorithme est basé sur la recherche à base de phéromones fait par les fourmis dans la nature et comment ils gardent une trace de la nourriture trouvé en utilisant leurs phéromones comme traqueurs de localisation.

1.15 Travaux connexes

Diverses techniques ont été développées pour améliorer la qualité, la précision et l'efficacité du processus de segmentation d'images, en combinant des approches classiques avec des méthodes modernes et avancées. Grâce à l'intégration des techniques d'apprentissage profond, des méthodes d'intelligence en essaim (Swarm Intelligence) et d'autres stratégies innovantes, des avancées significatives ont été réalisées. Ces approches récentes ont permis de surmonter les limites des méthodes traditionnelles, telles que la sensibilité au bruit, les variations d'éclairage ou encore la complexité des scènes, ouvrant ainsi la voie à des applications plus complexes, plus intelligentes et plus diversifiées.

1.15.1 Travaux basés sur le Deep Learning

En 2015, Olaf Ronneberger, Philipp Fischer et Thomas Brox [91] ont introduit le modèle U-Net, une architecture en forme de « U » combinant un encodeur, un décodeur et des connexions de saut (skip connections). Ce modèle a été spécialement conçu pour la segmentation biomédicale avec un nombre limité de données annotées, obtenant des résultats remarquables lors du défi ISBI 2015, établissant ainsi une nouvelle référence dans ce domaine.

Suite au grand succès du modèle U-Net introduit par Ronneberger et al. en 2015 [92] , plusieurs travaux ont été proposés pour améliorer sa précision et son efficacité. Parmi ces contributions, Oktay et ses collègues (2018) ont développé le modèle Attention U-Net, qui incorpore des mécanismes d'attention permettant au réseau de focaliser son apprentissage sur les régions importantes de l'image, améliorant ainsi la segmentation de structures complexes comme le pancréas dans les images médicales .Par ailleurs, Zhou et al. (2018) ont proposé le modèle UNet++ [93] , visant à réduire le fossé sémantique entre l'encodeur et le décodeur par des connexions imbriquées et une supervision profonde, ce qui a conduit à des améliorations notables dans des applications telles que la détection de nodules pulmonaires .Chen et son équipe (2018) ont quant à eux présenté DeepLabv3+, combinant convolutions dilatées et architecture encodeur-décodeur, atteignant des performances remarquables avec un mIoU de 89,0 % sur PASCAL VOC 2012 et 82,1 % sur Cityscapes, et ce sans recourir à des étapes de post-traitement [94] .

Dans la continuité de ces avancées, entre 2019 et 2023, d'autres équipes ont proposé des modèles innovants avec des améliorations notables. Wang et al. [95] (2019) ont introduit CE-Net, qui combine un encodeur ResNet avec des modules DAC et RMP, permettant d'améliorer la segmentation de structures fines telles que les vaisseaux sanguins. Ibtehaz et Rahman (2020) [96]ont développé MultiResUNet, intégrant des blocs multi-résolution et des connexions résiduelles, et ont montré une augmentation de performance pouvant atteindre +10,15 % par rapport au U-Net classique. Plus récemment, Hu et ses collaborateurs (2022) ont utilisé un Transformer pyramidal (PVT) dans leur modèle Polyp-PVT pour la segmentation de polypes dans des images d'endoscopie, atteignant une précision de 95 % [97].Enfin, Kirillov et al. (2023) ont présenté le modèle Segment Anything Model (SAM), une approche universelle et interactive capable de segmenter n'importe quel objet dans une image à partir d'indices simples, avec une précision de 97,3 %, marquant ainsi une avancée majeure vers une segmentation généralisée [98] .

1.15.2 Travaux basés sur l'intelligence en essaim (Swarm Intelligence)

L'intelligence en essaim, s'inspirant du comportement collectif d'agents naturels tels que les oiseaux, les abeilles ou les fourmis, s'est révélée particulièrement efficace pour résoudre des problèmes d'optimisation complexes, dont la segmentation d'images.

Mekhmoukh et ses collaborateurs (2018) [99] ont proposé une approche hybride combinant l'optimisation par essaim de particules (PSO), le clustering flou (FCM), la détection de valeurs aberrantes, et la méthode Level Set, afin d'améliorer la précision et la robustesse face au bruit dans les images médicales, obtenant des résultats supérieurs aux méthodes classiques .En 2021, Larabi-Marie-Sainte et al [100] .ont réalisé une revue exhaustive des algorithmes bio-inspirés appliqués à la segmentation d'images, tels que PSO, GWO (Grey Wolf Optimizer) et BFO (Bacterial Foraging Optimization), fournissant une analyse comparative approfondie et soulignant les tendances

émergentes dans ce domaine .La même année, Abualigah et son équipe ont appliqué l'algorithme Dragonfly (DAA) à la segmentation multi-seuils, atteignant une précision de 90 %, avec des performances compétitives par rapport à GWO et PSO [101] .

En 2022, Al-Qaness et al. ont amélioré le Whale Optimization Algorithm (WOA) en y intégrant une perturbation gaussienne, ce qui a permis de réduire la stagnation dans des minima locaux et d'atteindre une précision de 91,8 % sur des images pulmonaires [102] . La même année, Mohamed et ses collègues ont combiné les algorithmes GWO et PSO pour la segmentation d'images satellitaires, obtenant une précision de 90,5 % sur le jeu de données ISPRS, surpassant ainsi les performances du GWO seul [103] . En 2023, Fathy et al. ont fusionné les algorithmes Dragonfly et Firefly pour le traitement d'images thermiques bruitées, atteignant une précision de 87 % et démontrant une robustesse accrue face aux effets du bruit [104] . Enfin, en 2024, Zhang et ses collaborateurs ont utilisé l'algorithme Artificial Bee Colony (ABC) pour segmenter des scènes naturelles complexes issues du jeu de données BSDS500, obtenant une précision de 86,4 %, confirmant ainsi l'efficacité de cette approche dans des environnements visuellement denses [105] .

1.15.3 Autres approches de segmentation

En dehors des méthodes basées sur l'intelligence en essaim, plusieurs approches classiques continuent de jouer un rôle essentiel dans la segmentation d'images. Bien que ces méthodes soient anciennes, elles restent pertinentes, notamment lorsqu'elles sont adaptées ou combinées avec des techniques modernes. De nombreux travaux récents s'appuient sur ces fondations pour proposer des améliorations ciblées.

La méthode d'Otsu (Otsu, 1979) [106] repose sur la maximisation de la variance inter-classes afin de déterminer un seuil optimal pour la segmentation, offrant ainsi une solution simple et efficace pour séparer les régions d'intérêt dans une image .S'appuyant sur ce principe, Vite-Chávez et al [107] . (2023) ont adapté cette méthode en appliquant le seuillage uniquement sur le canal bleu des images RGB agricoles, ce qui a permis d'améliorer la segmentation des objets comme les fruits grâce à une meilleure gestion des variations d'éclairage, confirmée par des évaluations avec les indices de Jaccard et Dice.

D'autres approches combinent des techniques classiques pour optimiser la détection des contours. Par exemple, Salman et Liu (2012) ont proposé une méthode hybride associant K-means et l'algorithme Watershed : d'abord les pixels sont regroupés selon leurs caractéristiques via K-means, puis les contours sont affinés avec Watershed, ce qui donne de bons résultats pour des objets aux contours peu nets [108] .

Parallèlement, les architectures basées sur les Transformers ont montré des performances remarquables en segmentation sémantique. Zheng et al [109] . (2021) ont introduit le modèle SETR, qui applique les Transformers de manière séquentielle pour capturer efficacement les dépendances globales dans les images, atteignant des résultats de pointe sur les bases Cityscapes et ADE20K .Dans la même veine, Cao et al. (2021) ont développé Swin-Unet, un modèle combinant la hiérarchie locale du Swin Transformer avec la structure en U du U-Net, exploitant l'attention glissante pour une reconstruction précise des détails fins, et obtenant d'excellentes performances sur des images médicales [110] .

1.16 Conclusion

L’intelligence en essaim est une approche innovante de l’intelligence artificielle, inspirée du comportement collectif des êtres vivants dans la nature. Grâce à des mécanismes simples comme la collaboration, l’auto-organisation et la communication indirecte, elle permet de résoudre efficacement des problèmes complexes dans différents domaines. Que ce soit pour l’optimisation, la classification, ou encore la robotique et l’Internet des objets, les algorithmes en essaim offrent des solutions performantes, flexibles et adaptatives. Ce domaine reste en constante évolution et ouvre la voie à de nombreuses perspectives pour la recherche et les applications réelles.

Chapitre 3 : Application de l'algorithme Grey Wolf Optimizer à la segmentation d'image

2.1 Introduction

La segmentation d'images constitue l'un des piliers fondamentaux dans le domaine du traitement et de l'analyse d'images. Elle vise à diviser une image en régions ou en classes homogènes. Parmi les méthodes traditionnelles ayant démontré leur efficacité dans ce domaine, la méthode d'Otsu occupe une place importante. Elle repose sur la maximisation de la variance interclasses afin de déterminer le seuil optimal de segmentation. Cette approche est reconnue comme une référence académique incontournable et sert souvent de base pour évaluer les performances d'autres méthodes. Toutefois, la méthode d'Otsu présente certaines limites lorsqu'il s'agit de traiter des images complexes ou multimodales, ou encore dans le cadre d'une segmentation à seuils multiples, où la recherche du seuil optimal devient coûteuse et complexe sur le plan computationnel.
Dans ce contexte, les algorithmes méta-heuristiques ont émergé comme des solutions alternatives prometteuses, grâce à leur capacité à explorer efficacement de vastes espaces de solutions. Parmi ces algorithmes, le Grey Wolf Optimizer (GWO), inspiré du comportement social des loups gris, s'est distingué comme un outil puissant pour résoudre des problèmes d'optimisation complexes [78]. Cet algorithme repose sur les principes de la hiérarchie sociale et de la coopération entre les loups pour converger vers des solutions optimales.

2.2 GWO Originale

L'algorithme Grey Wolf Optimizer (GWO) est une méthode d'optimisation méta heuristique inspirée du comportement social et de la stratégie de chasse des loups gris dans la nature. Il a été proposé en 2014 par Mirjalili et s'appuie principalement sur la hiérarchie sociale des loups ainsi que sur leur manière de traquer les proies. Dans un groupe de loups, on distingue quatre catégories hiérarchiques :

- Alpha (α) : le meilleur loup – représente la meilleure solution**.**
- Bêta (β) : le second meilleur – aide à guider les autres.
- Delta (δ) : le troisième meilleur – suit alpha et beta.
- Oméga (ω) : le reste de la meute – suit les leaders.

Le GWO simule cette hiérarchie en assignant les trois meilleures solutions à ces rôles, où la meilleure solution actuelle est considérée comme le loup alpha. Le comportement collectif des loups est modélisé mathématiquement pour rechercher une solution optimale à un problème donné. Le fonctionnement du GWO peut être décomposé en deux grandes phases fondamentales : l'exploration et l'exploitation. Ces deux phases se succèdent et s'équilibrent au fil des itérations afin d'assurer une recherche efficace et complète de la solution optimale.

♣ Phase d'exploration (Encerclement de la proie)

La phase d'exploration permet à l'algorithme de parcourir l'espace de recherche sans se limiter aux meilleures solutions connues, en évitant ainsi de tomber dans des minima locaux. L'exploration repose sur des vecteurs de coefficient qui contrôlant la convergence :

$$\overrightarrow{A} = 2a.\overrightarrow{r_1} - a \qquad \text{(Ou a décroît de 2 à 0)} \qquad \textbf{(5)}$$

$$\overrightarrow{C} = 2.\overrightarrow{r_2} \qquad \text{(Avec } \quad r_2, r_2[0.1]) \qquad \textbf{(6)}$$

Lorsque $\|\vec{A}\|>1$, cela indique que le loup va s'éloigner de la proie estimée, favorisant ainsi une exploration globale de l'espace de recherche. La position d'un loup est mise à jour selon les formules suivantes :

$$\overrightarrow{D} = |\overrightarrow{C} * \overrightarrow{X_{best}}(t) - \overrightarrow{X}(t)| \qquad \textbf{(7)}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_{best}}(t) - \overrightarrow{A} * \overrightarrow{D} \qquad \textbf{(8)}$$

- $\overrightarrow{X_{best}}(t)$: est la position d'un des meilleurs loups (α, β ou δ),
- $\overrightarrow{X}(t)$: est la position actuelle du loup,
- $\overrightarrow{D}$: est la distance relative à la proie pondérée par$\overrightarrow{C}$.

♣ Phase d'exploitation

À mesure que le paramètre a décroît (vers 0), la norme de $\vec{A}$ devient inférieure à 1, déclenchant la phase d'exploitation. Les loups commencent alors à encercler et converger vers les meilleures solutions connues**.** Lorsque $\|\vec{A}\|<1$, cela signifie que les loups se rapprochent de la proie estimée, renforçant la recherche locale autour des solutions prometteuses. Chaque loup suit alors les positions des leaders α, β et δ, avec les étapes suivantes :

Chasse (mise à jour des positions) :

Les trois meilleurs loups (α, β, δ) guident les autres :

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1} * \overrightarrow{X_a} - \overrightarrow{X}\right| \qquad \textbf{(9)}$$

$$\overrightarrow{D_\beta} = \left|\overrightarrow{C_2} * \overrightarrow{X_\beta} - \overrightarrow{X}\right|$$

$$\overrightarrow{D_\delta} = |\overrightarrow{C_3} * \overrightarrow{X_\delta} - \overrightarrow{X}|$$

Puis on calcule les nouvelles positions :

$$\overrightarrow{X_1} = \overrightarrow{X_a} - \overrightarrow{A_1} * \left(\overrightarrow{D_\alpha}\right) \qquad \textbf{(10)}$$

$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2} * \left(\overrightarrow{D_\beta}\right)$$

$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3} * (\overrightarrow{D_\delta})$

- La position finale est la moyenne des trois : $\overrightarrow{X}(t+1) = \frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}}{3}$ **(11)**

Algorithme GWO

---

1. Initialiser la population de loups gris Xi pour i=1,2,...,N
2. Initialiser a, A et C **(5) (6)**
3. Calculer la valeur de fitness de chaque agent
4. Xα = le meilleur agent
5. Xβ = le deuxième meilleur agent
6. Xδ = le troisième meilleur agent
7. **Tant que** le critère d'arrêt n'est pas atteint **faire**
8. **Pour chaque** loup **faire**
9. Mettre à jour la position en fonction des positions de α, β et δ **(11)**
10. **Fin pour**
11. Mettre à jour a, A et C **(5) (6)**
12. Calculer la valeur de fitness de chaque agent
13. Mettre à jour Xα, Xβ et Xδ
14. t=t+1
15. **Fin tant que**
16. Retourner Xα

L'algorithme GWO consiste à trouver la meilleure solution à un problème donné. Au début, un ensemble de solutions initiales (loups) est créé aléatoirement, avec des paramètres initiaux tels que a, A et C. La valeur de fitness de chaque loup est ensuite calculée pour déterminer les trois meilleurs loups : α (le meilleur), β (le deuxième) et δ (le troisième). Dans chaque cycle de l'algorithme, chaque loup met à jour sa position en fonction des positions des trois premiers loups. Les valeurs de a, A et C sont alors mises à jour, les fitness sont recalculés et les positions de α, β et δ sont mises à jour. Ce processus se répète jusqu'à ce qu'une condition d'arrêt soit remplie, auquel cas le loup α est retenu comme la meilleure solution.

2.3 Approche proposée pour la segmentation d'image

Dans cette partie du travail, nous appliquons deux méthodes distinctes pour déterminer ce seuil. L'algorithme d'Otsu a été utilisé comme méthode analytique visant à trouver le meilleur seuil en maximisant la variance inter-classes, ce qui a permis d'obtenir des résultats précis et optimaux. En parallèle, l'algorithme GWO (Grey Wolf Optimizer) a été appliqué comme approche approximative, produisant des résultats acceptables et proches de l'idéal (OTSU). Ce qui suit décrit ce travail, nous avons commencé par un seuil pour OTSU et utilisé le résultat comme référence pour GWO ce dernier a donné un résultat très proche voici l'explication de cette partie de travail.

2.3.1 OTSU (1 Seuil)

La méthode d'Otsu est l'une des approches les plus reconnues pour le seuillage d'images en niveaux de gris. Elle vise à séparer les pixels en deux classes distinctes : le fond (C1) et l'objet (C2), en déterminant un seuil optimal t. Les pixels ayant une intensité inférieure ou égale à t appartiennent à la classe fond, tandis que ceux ayant une intensité supérieure appartiennent à la classe objet :

$C1 = \{0,1,..., t\}$, $C2 = \{t+1,..., L-1\}$

Le seuil optimal est déterminé en maximisant la variance inter-classes telles que :

$$\sigma^2 b(t) = \omega 1(t)\omega 2(t)\big(\mu 1(t) - \mu 2(t)\big)^2 \quad \textbf{(12)}$$

Soit $\sigma^2 T$ la variance totale de l'image telle que: $\sigma^2 T = \sum_{i=0}^{L-1} p(i) \cdot (i-\mu)^2$. Elle est liée aux deux autres variances comme suit:

$$\sigma^2 T = \sigma^2 b(t) + \sigma^2 w(t) \quad \textbf{(13)}$$

$\mu 1(t)$ Et $\mu 2(t)$ représentent les niveaux de gris moyens des classes : C1 et C2 respectivement:

- Moyenne de la classe 1 (pixels i ≤t) :

$$\mu 1(t) = \frac{1}{\omega 1(t)} \sum_{i=0}^{t} i \cdot P(i) \quad \textbf{(14)}$$

- Moyenne de la classe 2 (i> t) :

$$\mu 1(t) = \frac{1}{\omega 2(t)} \sum_{i=t+1}^{L-1} i \cdot P(i) \quad \textbf{(15)}$$

- Moyenne globale :

$$\mu 1(t) = \sum_{i=0}^{L-1} i \cdot P(i) \quad \textbf{(16)}$$

Soient ω1(t) et ω2(t) les probabilités a priori des classes 1,2 respectivement (définies par le seuil t) :

$$\omega 1(t) = \sum_{i=0}^{t} P(i) \qquad \text{Et} \qquad \omega 1(t) = \sum_{i=t+1}^{t} P(i) \quad \textbf{(17)}$$

Avec :

$$\omega 1(t) + \omega 2(t) = 1$$

Les étapes fondamentales de l'algorithme d'Otsu pour déterminer le seuil optimal

Algorithme Otsu (1 Seuil)

1. Calculer l'histogramme et les probabilités de chaque niveau d'intensité
2. Définir les $\omega i(0)$ et $\mu i(0)$ initiaux
3. Parcourir tous les seuils possibles t=1…….255 intensité max
    1. Mettre à jour ωi et μi
    2. Calculer $\delta^2 b(t)$
4. Le seuil désiré correspond au $\delta^2 b(t)$ maximum.

En raison de sa simplicité et de son efficacité, l'algorithme d'Otsu constitue une méthode de référence en segmentation d'images. Grâce à un principe d'analyse reposant sur la maximisation de la variance inter-classes en fait un excellent critère d'évaluation pour comparer les performances d'algorithmes d'optimisation comme le GWO, en recherchant des solutions optimales proches de celles d'Otsu.

2.3.2 GWO (1 Seuil)

Nous avons utilisé le résultat de la méthode d'Otsu comme référence pour évaluer la qualité des résultats obtenus par l'algorithme GWO. Cette évaluation permet de guider les loups vers le choix des seuils optimaux, assurant ainsi une séparation efficace entre les différentes régions de l'image. Les étapes de base de l'algorithme GWO original ont été conservées sans modification, en maintenant le mécanisme de recherche et la mise à jour des positions des loups tels quels.
Tout d'abord, le résultat de la méthode d'Otsu sert de critère d'évaluation pour guider la recherche du seuil optimal dans un espace délimité par une borne inférieure de 0 et une borne supérieure de 255. De plus, une légère modification a été introduite dans l'étape de calcul de la valeur de fitness des agents (ligne 12 de l'algorithme GWO original)

♣ Fonction objectif (1 Seuil)

Nous avons défini une fonction d'évaluation simple basée sur l'erreur relative normalisée entre les deux seuils. Cette fonction renvoie une valeur de 1 lorsque les deux seuils sont identiques, et diminue proportionnellement à leur écart, la fitness dans cette partie est calculée comme suit :

$$fitness = 1 - \frac{(S_{otsu} - S_{i,gwo})}{255} \qquad (18)$$

Où S_{otsu} est le seuil trouvé par otsu et $S_{i,gwo}$ est le seuil obtenu par GWO dans l'itération $\boldsymbol{i}$. Après cette expérience, les résultats ont été étonnants au niveau d'approximation entre OTSU et GWO, mais travailler sur un seul seuil n'a pas donné une meilleure segmentation, et c'est ce que nous allons voir dans la partie expérience et discussion, ce qui nous a poussé à travailler sur trois seuils.

2.3.3 Segmentation multi-seuils d'images

Le schéma suivant présente les étapes de notre approche

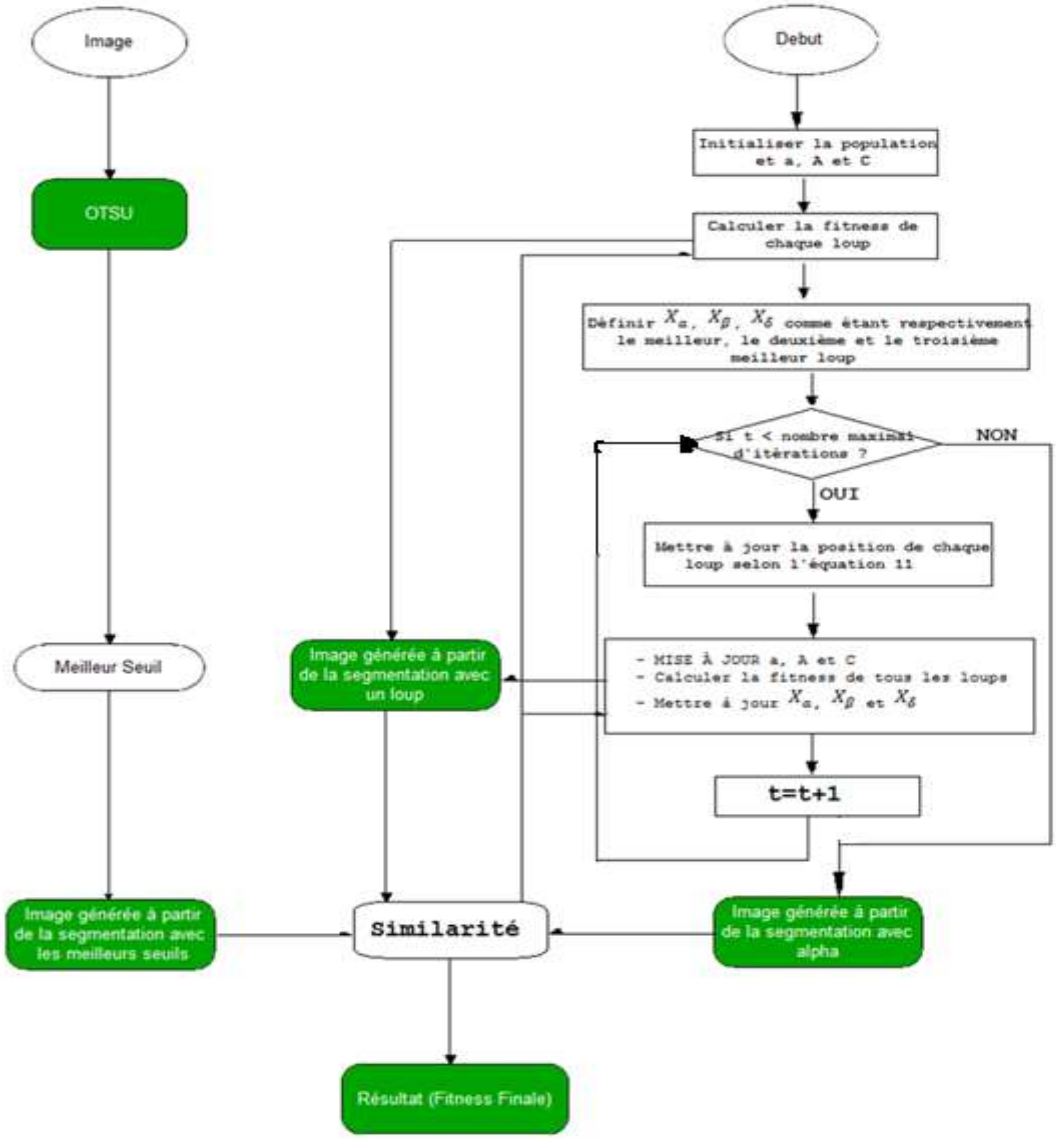


Figure 3. 1: GWO pour la segmentation multi-seuils d'images basée sur la méthode OTSU

2.3.4 Otsu (3 seuil)

La segmentation d'image par la méthode d'Otsu repose sur la recherche du ou des seuil(s) optimaux qui permettent de maximiser la variance inter-classes. Lorsqu'un seul seuil est utilisé, le nombre de combinaisons possibles est relativement faible (255 possibilités pour une image en niveaux de gris

sur 8 bits), ce qui permet un calcul rapide et efficace. En revanche, lorsqu'on applique Otsu avec trois seuils pour segmenter l'image en quatre classes, le nombre de combinaisons possibles explose, atteignant plusieurs millions (plus précisément, le nombre de combinaisons uniques de trois seuils dans l'intervalle [0,255]). Cette augmentation exponentielle du nombre de cas à évaluer engendre un coût computationnel important, ce qui pose un véritable problème en termes de temps de calcul, surtout pour les images de grande taille ou dans des applications en temps réel.

Pour calculer le nombre de combinaisons possibles dans la méthode d'Otsu avec trois seuils**,** on suppose que les seuils sont ordonnés et différents, et qu'ils sont choisis parmi les 256 niveaux de gris **(0 à 255).** La formule mathématique pour le nombre de combinaisons uniques de trois seuils ordonnés est donnée par :

$$\binom{n}{k} = \binom{255}{3} = \frac{255*255*255}{3*2*1} = \mathbf{2\ 731\ 135} \qquad \textbf{(19)}$$

- $\binom{n}{k}$ est le coefficient binomial, qui donne le nombre de combinaisons possibles de k éléments distincts parmi n.
- Ici, n = 256 niveaux de gris, et k = 3 seuils.

Donc, **2 731 135** est le nombre total de combinaisons à tester pour appliquer Otsu à trois seuils. Cela montre clairement l'augmentation exponentielle de la complexité par rapport à la version à un seul seuil (255 cas seulement).

2.3.5 GWO (3 seuil)

Étant donné que la méthode d'Otsu à trois seuils implique une recherche exhaustive parmi plus de 2,7 millions de combinaisons possibles, ce qui entraîne un coût computationnel élevé, GWO permet de réduire considérablement le temps de calcul. Cet algorithme méta heuristique explore intelligemment l'espace de recherche pour approcher les seuils qui maximisent la variance interclasses, critère utilisé par Otsu. L'objectif est d'obtenir une qualité de segmentation comparable à celle d'Otsu multi-seuils, tout en accélérant le processus de calcul grâce à l'optimisation.
Dans l'algorithme GWO, chaque loup est représenté par un vecteur de trois seuils distincts.

♣ Fonction objectif (3 Seuil)

La fonction objectif utilisée dans GWO est basée sur la similarité structurelle (SSIM) entre l'image segmentée par GWO et celle obtenue par Otsu. L'objectif est donc de maximiser la valeur de SSIM**,** qui mesure la ressemblance perceptuelle entre deux images en tenant compte de la luminance, du contraste et de la structure. Ainsi, plus les seuils trouvés par GWO produisent une image similaire à celle de référence, plus la valeur de la fonction objective est élevée.

2.4 Expériences et discutions

Dans cette section, nous présentons les différentes expériences menées pour évaluer l'efficacité de l'algorithme Grey Wolf Optimizer (GWO) appliqué à la segmentation d'images par seuillage

multiple. Dans un premier temps, une étude a été réalisée en considérant un seuillage binaire basé sur un seul seuil, permettant ainsi de valider la pertinence de l'approche GWO en comparaison avec la méthode d'Otsu. Forts des résultats obtenus, nous avons ensuite étendu l'approche à une segmentation par trois seuils, afin de capturer des niveaux de détails plus fins dans les images. La méthode d'Otsu a de nouveau été utilisée pour générer une image de référence, et la qualité des segmentations produites par GWO a été évaluée à l'aide de la mesure de similarité structurelle (SSIM)**.** Les résultats expérimentaux obtenus en utilisant un ordinateur doté d'un processeur 8 ème Génération Intel(R) Core i5 cadencé à 2.20 GHz, de 4 RAM. Ces résultats sont analysés et discutés dans les sous-sections suivantes.

2.4.1 Résultat OTSU, GWO (1 Seuil)

Dans une première étape, nous avons appliqué l'algorithme Grey Wolf Optimizer (GWO) à la segmentation d'image à un seul seuil. Étant donné que l'espace de recherche dans ce cas est limité à 256 valeurs entières possibles (correspondant aux niveaux de gris de 0 à 255), il était raisonnable d'utiliser des paramètres standards sans nécessiter de réglage approfondi. Ainsi, nous avons fixé le nombre de loups à 20 et le nombre d'itérations à 40. Ces paramètres se sont avérés suffisants pour atteindre une convergence stable et rapide.

Nous avons comparé le seuil trouvé par GWO avec celui déterminé par la méthode d'Otsu. Dans cette expérience, Otsu a identifié le seuil optimal à 154, tandis que GWO a convergé vers 156. Pour quantifier cette proximité, La fitness obtenue on appliquant l'équation **(19)** est :

$$\boldsymbol{fitness = 1 - \frac{abs(154 - 156)}{256} = 0.9921\ldots}$$

Ce résultat très proche entre GWO et OTSU indique une excellente concordance entre les deux méthodes, démontrant ainsi la capacité de GWO à localiser efficacement un seuil de segmentation pertinent, même avec des paramètres de base non optimisés.
La figure suivante représente le résultat de la segmentation avec les deux seuils obtenu par Otsu et Gwo.

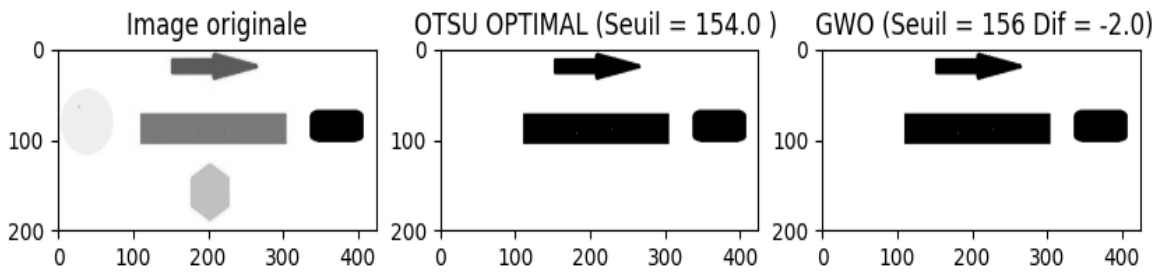


Figure 3. 2: Comparaison des résultats de seuillage binaire entre Otsu et GWO

Il s'avère bien que la segmentation avec OTSU ou avec GWO ne donne pas un bon résultat En particulier, le disque gris clair à gauche de l'image originale n'est pas bien segmenté. Cela met en

évidence les limites des méthodes à seuil unique, notamment dans les cas où l'image contient des objets aux niveaux de gris très proches de l'arrière-plan**.**
L'utilisation d'un seul seuil avec les méthodes Otsu et GWO permet une bonne segmentation des objets à fort contraste. Cependant, cette approche montre ses limites face aux détails fins ou aux zones à faible contraste, ce qui suggère l'intérêt d'explorer des techniques à seuils multiples pour une segmentation plus précise.
Ce premier résultat comparatif entre Otsu et GWO nous a permis de constater que l'approche par optimisation peut aboutir à des seuils légèrement différents mais tout aussi pertinents pour la segmentation. Encouragés par la capacité du GWO à explorer efficacement l'espace des solutions, nous avons étendu notre travail à la segmentation multi-seuils, en particulier à la segmentation à trois seuils. Cette étape vise à mieux distinguer les différentes régions d'intensité dans l'image, ce qui est particulièrement utile pour des images plus complexes.

2.4.2 Résultat OTSU (3 Seuil)

Dans cette étape, l'algorithme vise à calculer toutes les combinaisons possibles des trois seuils ***t1,t2,t3*** dans l'intervalle de 0 à 255, représentant les valeurs des pixels dans les images en niveaux de gris. Le travail consiste à évaluer toutes les combinaisons possibles de seuils (***t1,t2,t3***), avec la contrainte **$0 \leq t1 < t2 < t3 \leq 255$** tq **$t1 < t2 < t3$**, Afin de quantifier le coût de cette approche exhaustive. Pour ce faire, nous avons mis en œuvre un algorithme exploratoire par force brute, basé sur trois boucles imbriquées. Chaque boucle correspond à un seuil :

La première boucle fait varier t1 de 0 à 255,
La deuxième t2 de t1 + 1 à 255,
La troisième t3 de t2+1 à 255.

Un compteur ***cpt*** a été initialisé à zéro et incrémenté à chaque itération, c'est-à-dire à chaque Combinaison évaluée.
L'algorithme exécuté est donc le suivant :

---

1. Initialiser le compteur cpt ← 0
2. Pour t1 allant de 0 à 255 faire :
3. Pour t2 allant de t1 + 1 à 255 faire :
4. Pour t3 allant de t2 + 1 à 255 faire :
5. Incrémenter le compteur : cpt ← cpt + 1
6. Appeler la fonction d'évaluation (fonction (t1, t2, t3))
7. Afficher la valeur de cpt

Après avoir calculé le nombre total de combinaisons via l'équation **(19),** le résultat final de **2 731 135** est vérifié et confirmé par l'exécution du programme. Ce nombre correspond exactement au résultat de la formule mathématique dans l'équation **(19)** dans la section **(3.3.4).**

Ce résultat met en évidence la complexité computationnelle d'une telle méthode exhaustive, ce qui peut poser des défis importants en termes de temps de calcul, surtout pour des traitements en temps réel ou sur des images de grande taille. Cela motive l'exploration de stratégies plus optimisées, à cet égard nous avons, qui permettent de réduire le nombre d'évaluations nécessaires tout en obtenant des résultats proches de l'optimum.
Nous avons appliqué la méthode d'Otsu multi-seuil à trois niveaux sur un ensemble de dix images de tailles variées, allant des plus petites aux plus grandes, afin de couvrir un éventail de cas pratiques en termes de volume de données. Pour chaque image, une recherche exhaustive a été effectuée en évaluant toutes les combinaisons possibles de trois seuils
Cette phase a permis d'obtenir, pour chaque image, les seuils optimaux selon Otsu ainsi que le temps d'exécution nécessaires pour explorer tout l'espace de recherche. Comme la montre Figure 3.5 et le Tableau 3.3

Ces résultats obtenus ont ensuite été utilisés comme référence de qualité (ou solution de base) pour la suite du travail. Dans la deuxième étape, les mêmes dix images ont été soumises à une segmentation par l'algorithme GWO.

2.4.3 Résultat GWO (3 Seuil)

Pour évaluer l'efficacité de l'algorithme GWO appliqué à la segmentation d'images multi-seuils, nous avons mené une série d'expériences sur différentes images. Les résultats obtenus ont été comparés à ceux de la méthode classique d'Otsu à trois seuils. Les images segmentées à l'aide de GWO montrent une qualité visuelle très proche de celle de la méthode d'Otsu, tout en réduisant significativement le temps de calcul, les performances de GWO sont également compétitives.

Cette approche démontre ainsi que GWO constitue une alternative efficace pour la segmentation multi-seuils, surtout dans des contextes où le temps de traitement est un facteur critique.

2.4.3.1 Performance GWO pour différents itérations

Le choix du nombre de particules (ou loups) joue un rôle clé dans la qualité de la solution obtenue, mais aussi dans le coût computationnel de l'algorithme. Ainsi, la démarche adoptée ici vise à identifier un compromis entre qualité de la solution et efficacité computationnelle. En limitant judicieusement le nombre de particules, on maintient de bonnes performances de segmentation tout en réduisant le temps d'exécution, ce qui est particulièrement important dans des contextes de traitement d'image à grande échelle ou en temps réel.

La courbe ci-dessous illustre l'évolution de la valeur de fitness en fonction du nombre d'itérations de l'algorithme GWO utilisé pour la segmentation d'image, avec un effectif de 20 loups (particules).

Tableau 3. 1: Résultat GWO pour différente itérations et avec 20 loups

it_gwo	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80
fit_gwo	0.26	0.38	0.48	0.56	0.65	0.73	0.8	0.86	0.91	0.93	0.95	0.97	0.97	0.97	0.97	0.97

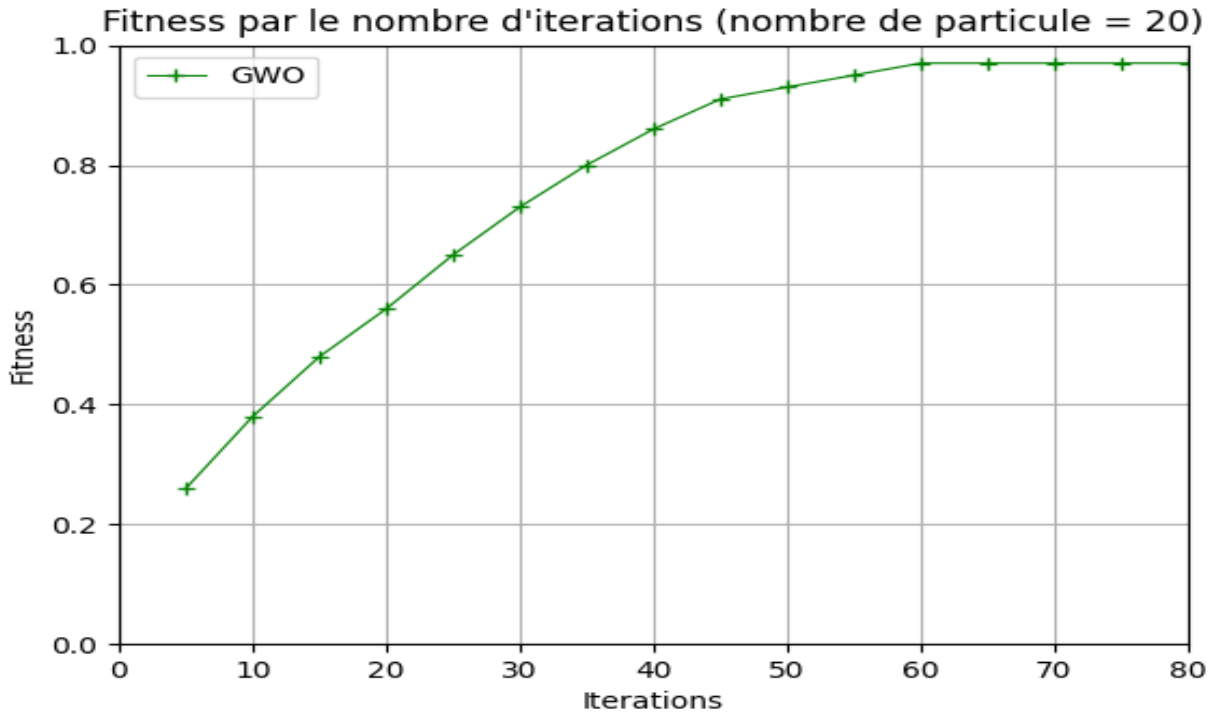


Figure 3. 3: Fitness en fonction du nombre d'itérations (nombre de particules = 20)

On observe une croissance rapide de la fitness durant les premières itérations, notamment entre 0 et 40 itérations. Cette phase indique une exploration efficace de l'espace de recherche, où l'algorithme améliore rapidement la qualité des solutions. À partir d'environ 50 itérations, la courbe commence à se stabiliser, ce qui suggère que l'algorithme entre dans une phase d'exploitation : les améliorations deviennent plus marginales.

Finalement, à partir de 60 itérations, la courbe atteint un plateau autour de la valeur maximale (près de 1), indiquant que l'algorithme converge vers une solution optimale ou quasi-optimale. Ce comportement est typique des algorithmes d'optimisation par essaim, où un compromis s'opère entre exploration et exploitation. On peut conclure que pour ce cas précis (20 particules), un nombre d'itérations compris entre 60 et 70 est suffisant pour atteindre une convergence satisfaisante, et qu'augmenter davantage les itérations au-delà de ce seuil n'apporte pas d'amélioration significative de la performance.

2.4.3.2 Performance GWO pour différents population

La courbe ci-dessous montre l'évolution de la valeur de fitness en fonction du nombre de particules (ou loups) utilisés dans l'algorithme GWO, avec le nombre d'itérations fixé à 60 trouvé dans la phase précédente. Cette analyse permet d'évaluer l'impact de la taille de la population sur la qualité de la solution obtenue pour la tâche de segmentation d'image.

Tableau 3. 2: Tableau Résultat GWO pour différente population et avec 60 itérations

Nbr_loups_gwo	10	13	16	19	22	25	28	31	34	37	40	43	46	49	52
fit_gwo	0.42	0.48	0.56	0.64	0.72	0.75	0.9	0.93	0.96	0.97	0.98	0.98	0.98	0.98	0.98

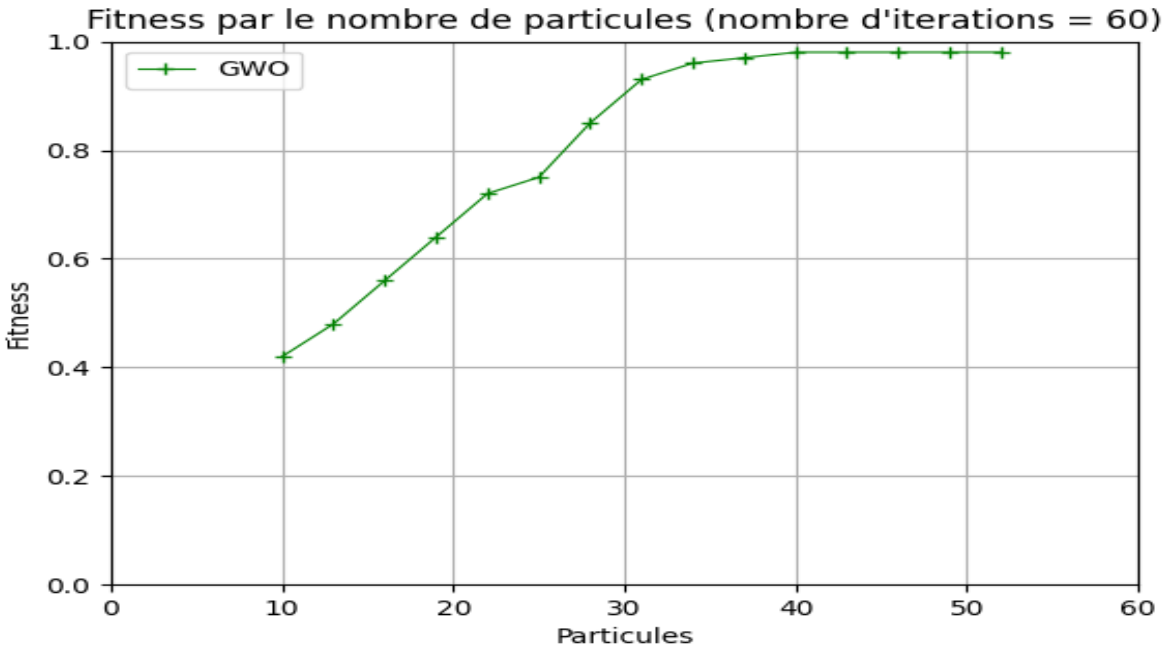


Figure 3. 4: Fitness en fonction du nombre de particules (nombre d'itérations = 60)

On observe une nette amélioration de la fitness lorsque le nombre de particules augmente de 10 à environ 30. Cette phase indique que l'ajout de particules permet à l'algorithme d'explorer plus efficacement l'espace de recherche, ce qui se traduit par de meilleures solutions.
À partir de 30 particules, la courbe montre une saturation progressive : les gains en performance deviennent de plus en plus faibles, et une convergence quasi-complète est atteinte autour de 40 particules. Au-delà de ce seuil, l'augmentation du nombre de particules n'apporte pas d'amélioration significative de la fitness, suggérant que l'algorithme a déjà atteint une solution optimale ou très proche de l'optimum. Ce résultat montre qu'un nombre de particules compris entre 30 et 40 est suffisant pour assurer une convergence efficace sans surcoût computationnel inutile.

2.4.4 Comparaison entre OTSU (3 Seuil) et GWO (3Seuil)

Pour cette expérience, nous avons utilisé un ensemble de dix images de tailles variées, allant de petites dimensions à des images volumineuses. L'objectif était d'analyser l'évolution des performances des deux méthodes, Otsu et GWO, en fonction de la taille de l'image. L'ensemble des images segmentées par les deux méthodes sont présentées dans cette figure.

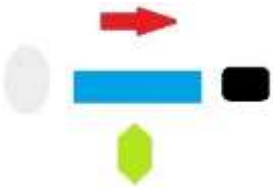
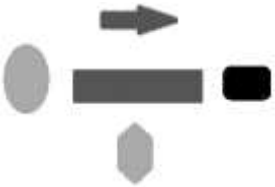
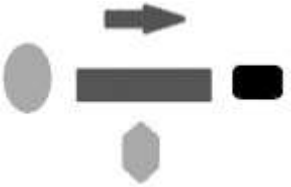
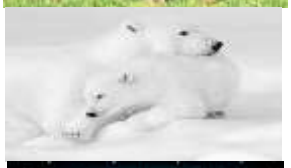
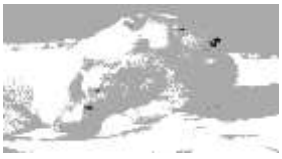
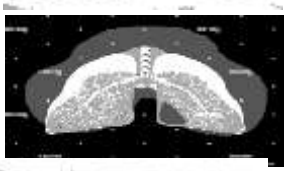
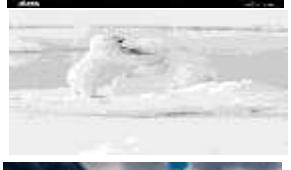
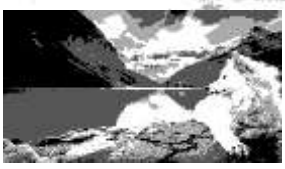
Image	Original	Segmenté(Otsu)	Segmenté(GWO)
image 01			
image 02			
image 03			
image 04			
image 05			
image 06			
image 07			
image 08			
image 09			
image 10			

Figure 3. 5 : Comparaison visuelle des résultats de segmentation par Otsu et GWO sur différentes 10 images.

Les différences entre les images ne sont pas perceptibles visuellement, en raison de la valeur de fitness obtenue par l'algorithme GWO, très proche de 1, indiquant un haut degré de similarité. Ce résultat reflète l'efficacité de GWO en matière de segmentation. En revanche, le tableau1 révèle un écart significatif entre les performances des méthodes OTSU et GWO, confirmant l'avantage de cette dernière.

Tableau 3. 3: Comparaison des performances entre la méthode Otsu et l'optimiseur Grey Wolf (GWO) pour la segmentation en 3 seuils.

Image	Taille (ko)	Otsu 3 Seuils			GWO 3 Seuils			
		Meilleur variance	Meilleur Seuil	Temps d'exécution	Meilleur variance	Meilleur Seuil	Temps d'exécution	Fitness (Similarité)
Image 01	9	56373	(10, 155, 224)	0H:12M	56084	(16. 168. 227)	01M:48S	0.998
Image 02	15	21269	(97, 154, 208)	0H:56M	14439	(97,154,210)	02M:29S	0.9875
Image 03	36	30826	(117, 157,191)	5H:27M	35286	(117,157,192)	06M:55S	0.9882
Image 04	69	18096	(88, 140, 198)	8H:38M	15007	(88,140,198)	08M:37S	0.9858
Image 05	84	14783	(69, 120, 178)	12H:35M	18258	(69,121,178)	10M:42S	0.9769
Image 06	95	29621	(117, 162,204)	13H:32M	24731	(117,163,205)	12M:33S	0.978
Image 07	145	49754	(126,198,226)	01J:3H:24M	44640	(34, 70,226)	16M:43S	0.9469
Image 08	145	7868	(31, 90, 164)	01J:17H:13M	13082	(32, 90,164)	19M:48S	0.9942
Image 09	258	49761	(170, 210, 227)	03J:17H:19M	41562	(75. 211. 228)	34M:24S	0.9779
Image 10	583	18827	(72, 130, 184)	5J:20H:08M	40562	(32,200,235)	49M:24S	0.9879

OTSU atteint plus de 5 jours (5J:20H) pour la dernière image de 583 Ko. GWO, pour la même image, met moins de 50 minutes. Ce comportement montre qu'OTSU devient inadapté dès que le volume augmente, avec une explosion exponentielle du temps. GWO garde un temps maîtrisé et raisonnable même pour les images lourdes.

Les deux méthodes trouvent des seuils proches ou identiques dans plusieurs cas (par ex. image 3, 4, 6). Les variances obtenues par GWO sont parfois meilleures ou comparables à OTSU (ex : image 3, 6, 10). Le score de fitness de GWO reste très élevé (souvent > 0.97), indiquant une très forte similarité

avec OTSU malgré un temps bien plus court. Cela suggère que GWO réussit à approcher, voire dépasser, les performances d'OTSU tout en étant infiniment plus rapide. Cas spécifiques à noter Image 07 et 08 : OTSU explose à plus de 24 heures, tandis que GWO reste en dessous de 20 minutes. Pourtant, la fitness GWO reste autour de 0.95–0.99, ce qui reste très acceptable. Image 09 : malgré un volume modéré (258 Ko), OTSU prend près de 4 jours, contre seulement 34 minutes pour GWO. **Fitness** : 0.9779 → donc GWO reste viable.

Le tableau confirme que GWO est une alternative robuste et efficace à OTSU, surtout lorsqu'on traite des images de grande taille. Les résultats montrent que GWO réduit considérablement le temps de traitement tout en maintenant une qualité de segmentation très satisfaisante. Il devient donc une solution plus pratique, évolutive et réaliste pour les systèmes de traitement d'images modernes, surtout dans des contextes temps réel ou sur des bases de données massives.

La figure suivante est un graphique à deux axes Y met clairement en évidence les différences majeures entre les performances de l'algorithme OTSU et GWO pour la segmentation d'images selon des volumes croissants.

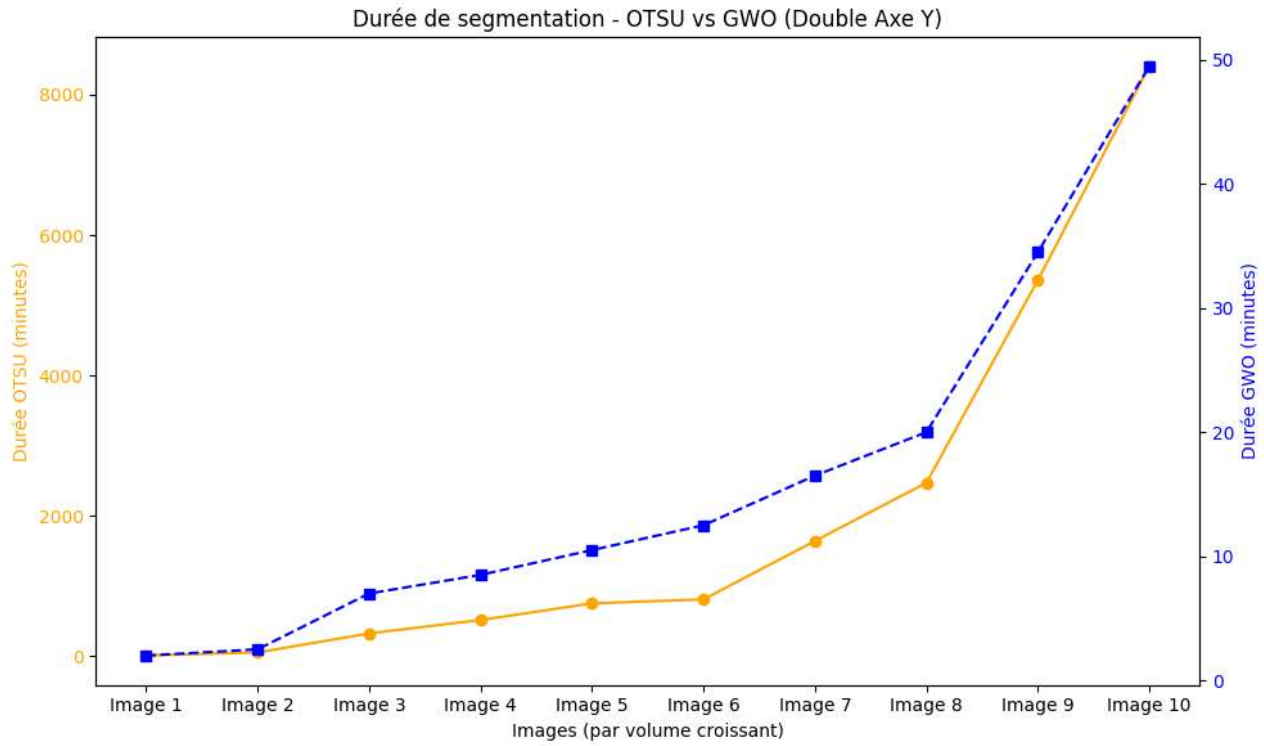


Figure 3. 6: Performance temporelle de segmentation OTSU et GWO

OTSU montre une augmentation exponentielle du temps de traitement à mesure que le volume des images augmente. À partir de la 4e ou 5e image, le temps s'emballe : il passe de centaines à plusieurs milliers de minutes pour les dernières images. Cela indique une mauvaise scalabilité : OTSU devient rapidement inefficace pour des volumes élevés.

GWO, au contraire, présente une croissance modérée et stable**.** Le temps de traitement passe de 2 à 49 minutes sur l'ensemble des images, ce qui reste très raisonnable même pour les images les plus volumineuses. Cela témoigne d'une meilleure robustesse et efficacité de GWO pour des volumes importants.

♣ **Interprétation des performances**

- L'efficacité de GWO est clairement supérieure en termes de temps d'exécution.
- OTSU devient impraticable au-delà d'un certain seuil de volume d'image, sauf à disposer de ressources computationnelles très importantes.
- Ce genre de graphe peut donc justifier l'abandon d'OTSU pour des applications en traitement d'images volumineuses, ou au moins inciter à une optimisation sérieuse.

On observe que les valeurs de fitness sont globalement élevées pour l'ensemble des images, variant entre **0,96** et **0,99**. Cela indique une très bonne performance du GWO en termes de qualité de segmentation, avec une forte stabilité du résultat malgré les variations de contenu d'image. Les trois premières images (`im_01, im_02, im_03`) atteignent la fitness maximale de **0,99**, traduisant une convergence quasi parfaite vers une solution optimale. Les images suivantes présentent une légère baisse progressive des performances, mais la valeur de fitness reste toujours supérieure à **0,96**, ce qui reste très satisfaisant. Cette stabilité des résultats démontre la robustesse de l'algorithme GWO face à des variations de complexité ou de structure dans les images traitées. Elle confirme également que la configuration choisie (40 particules, 60 itérations) est suffisamment généralisable pour maintenir une performance élevée sans ajustement spécifique à chaque image.

2.5 Conclusion

Nous avons étudié l'application de l'algorithme GWO pour la segmentation d'images, en mettant particulièrement l'accent sur l'influence des paramètres clés de l'algorithme à savoir le nombre de particules et le nombre d'itérations sur la qualité de segmentation mesurée par une fonction de fitness. Dans un premier temps, nous avons fixé le nombre de particules à 20 et analysé la convergence de la fitness en fonction du nombre d'itérations. Les résultats ont montré une amélioration rapide de la fitness jusqu'à environ 60 itérations, au-delà desquelles l'amélioration devient négligeable. Cela nous a permis d'identifier 60 itérations comme un compromis optimal entre qualité et temps de calcul.

Ensuite, en fixant le nombre d'itérations à 60, nous avons évalué l'impact du nombre de particules sur la performance de l'algorithme. Les tests ont révélé qu'augmenter le nombre de particules jusqu'à 40 améliore significativement la fitness, mais qu'au-delà de ce seuil, les gains deviennent insignifiants. Cette observation justifie la limitation volontaire du nombre de particules pour réduire les coûts computationnels sans perte de performance.

Enfin, en appliquant ces paramètres optimaux (40 particules et 60 itérations) à dix images de volumes différents, nous avons obtenu des valeurs de fitness comprises entre 0.96 et 0.99, traduisant une excellente stabilité et robustesse de l'algorithme GWO, indépendamment des caractéristiques spécifiques des images.

Ainsi, ce travail confirme que le GWO, avec un bon réglage, est une méthode efficace, stable et peu coûteuse pour la segmentation d'images. Il constitue une alternative sérieuse aux méthodes classiques

comme Otsu, notamment lorsque l'adaptabilité et la précision sont recherchées dans des contextes complexes.

Les résultats montrent que la méthode d'Otsu devient extrêmement lente à mesure que la taille de l'image augmente. Pour les images les plus volumineuses, le calcul des trois seuils par Otsu peut nécessiter plusieurs heures, voire des jours**,** rendant son usage peu pratique dans des contextes à grande échelle ou en temps réel. En revanche, l'approche basée sur GWO permet une réduction significative du temps de traitement, tout en fournissant des résultats visuellement et structurellement comparables à ceux d'Otsu. Le GWO, grâce à sa nature heuristique, converge vers une solution satisfaisante en un nombre limité d'itérations, indépendamment du volume de l'image. Cette efficacité est particulièrement visible sur les images de grande taille, où le gain en temps est considérable.

Conclusion générale

Dans ce travail, nous avons comparé deux approches de segmentation d'images – OTSU et Grey Wolf Optimizer (GWO) en évaluant leur temps d'exécution sur un ensemble de dix images de volumes croissants. Les résultats montrent une différence marquée en termes de performance : alors que la méthode OTSU voit son temps de traitement croître de manière exponentielle avec l'augmentation du volume des images, GWO maintient une croissance modérée et linéaire. Cette observation souligne la meilleure scalabilité et efficacité computationnelle de GWO, en particulier pour les images de grande taille. Par conséquent, GWO apparaît comme une alternative plus robuste et plus adaptée pour les systèmes de traitement d'images modernes, notamment dans des contextes à forte volumétrie de données.

Dans chaque chapitre, nous avons exploré divers aspects importants liés à la segmentation d'images et aux algorithmes d'optimisation. Dans le Chapitre 1, nous avons posé les bases théoriques du traitement d'images, offrant ainsi une compréhension essentielle des concepts et des techniques qui sous-tendent les méthodes de segmentation, en insistant sur les caractéristiques des images et les défis liés à la gestion de la taille et de la qualité des données. Dans le Chapitre 2, nous avons détaillé les principes de l'intelligence en essaim, expliquant le fonctionnement des algorithmes bio-inspirés et leur application à la segmentation d'images, ce qui a permis de mieux situer GWO par rapport à d'autres approches d'optimisation. Enfin, dans le Chapitre 3, nous avons approfondi l'application de GWO à la segmentation d'images en combinant cette méthode avec la technique OTSU, tout en analysant les résultats expérimentaux obtenus pour différents seuils, afin de démontrer la flexibilité et la performance de GWO dans des scénarios plus complexes. Ces chapitres mettent en lumière les fondements théoriques et les applications pratiques qui justifient l'adoption de GWO comme une méthode efficace et adaptable pour des tâches de traitement d'images à grande échelle.

Travaux futurs

Pour enrichir et approfondir ce travail, plusieurs axes de recherche peuvent être envisagés. Tout d'abord, une évaluation qualitative des résultats de segmentation serait essentielle pour compléter l'analyse temporelle, en utilisant des métriques reconnues telles que l'indice de Dice, le coefficient de Jaccard ou encore le taux d'erreur de classification, afin d'évaluer la précision et la pertinence des segmentations obtenues. Ensuite, il serait pertinent d'élargir l'étude à d'autres algorithmes métaheuristiques bio-inspirés, tels que PSO (Particle Swarm Optimization), DE (Differential Evolution) ou WOA (Whale Optimization Algorithm), dans le but de situer GWO dans un cadre comparatif plus large et d'identifier ses avantages et ses limites. Par ailleurs, l'adoption d'une approche multi-objective, combinant la qualité de la segmentation et le temps d'exécution, pourrait permettre d'optimiser les performances globales du système. De plus, afin de réduire les coûts computationnels, en particulier pour les méthodes nécessitant plusieurs seuils, la mise en œuvre d'une version parallèle ou accélérée par GPU constituerait une solution efficace pour améliorer la rapidité d'exécution. Enfin, l'application de ces approches à des données tridimensionnelles, comme les volumes issus de l'imagerie médicale, ou à des données multi-modales telles que l'IRM ou le scanner, permettrait de valider la robustesse et la généricité des méthodes proposées dans des contextes plus complexes et réalisé.

Bibliographies

[1] Gonzalez, R.C., & Woods, R.E. (2008). *Digital Image Processing* (3rd Ed.). Pearson Education.
[2] Houassine, C. (2012). *Segmentation d'images par une approche biomimétique hybride* (Thèse de doctorat).
[3] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
[4] Pratt, W.K. (2007). *Digital Image Processing: PIKS Scientific Inside*. Wiley.
[5] Jain, A.K. (1989). *Fundamentals of Digital Image Processing*. Prentice Hall.
[6] Salomon, D. (2011). *Data Compression: The Complete Reference*. Springer.
[7] Russ, J.C. (2011). *The Image Processing Handbook*. CRC Press.
[8] Castleman, K.R. (1996). *Digital Image Processing*. Prentice Hall.
[9] Jain, R. (1995). *Machine Vision*. McGraw-Hill.
[10] Chikh, M.T. (2011). *Amélioration des images par un modèle de réseau de neurones : comparaison avec les filtres de base*.
[11] Sandeli, M. (2014). *Traitement d'images par des approches bio-inspirées : application à la segmentation d'images*.
[12] Cocquerez, J.-P. (1995). *Analyse d'images : filtrage et segmentation*. Masson.
[13] Bres, S., Jolion, J.-M., & Lebourgeois, F. (2003). *Traitement et analyse des images numériques*. Hermès Science Publications.
[14] Minée, S., Bykovo, Y., Pori li, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2020). Image Segmentation Using Deep Learning: A Survey.
[15] Wang, Z., Bovik, A.C., Sheikh, H.R., & Simoncelli, E.P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612.
[16] Medjaoui, A., & Fares, F. (2012). Segmentation des images par contours actifs : application sur les images satellitaires à haute résolution.
[17] Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., ... & Asari, V.K. (2018). A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8(3), 292.
[18] Arulkumaran, K., Deisenroth, M.P., Brundage, M., & Bharath, A.A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6), 26–38.
[19] Khan, A., Sohail, A., Zamora, U., & Qureshi, A.S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8), 5455–5516.
[20] Jarrett, K., Kavukcuoglu, K., Ranzato, M., & LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision* (pp. 2146–2153). IEEE.
[21] Panigrahi, B.K., Shi, Y., & Lim, M.H. (Eds.). (2011). *Handbook of swarm intelligence: concepts, principles and applications* (Vol. 8). Springer Science & Business Media.
[22] Dhiman, G., & Kumar, V. (2018). Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. Knowledge-Based Systems, 159, 20-50.
[23] Dhiman, G., & Kaur, A. (2019). STOA: a bio-inspired based optimization algorithm for

industrial engineering problems. Engineering Applications of Artificial Intelligence, 82, 148-174.
[24] Morales-Castañeda, B., Zaldivar, D., Cuevas, E., Fausto, F., & Rodríguez, A. (2020). A better balance in metaheuristic algorithms: Does it exist?. Swarm and Evolutionary Computation, 54, 100671.
[25] Tao, X., Li, X., Chen, W., Liang, T., Li, Y., Guo, J., & Qi, L. (2021). Self-Adaptive two roles hybrid learning strategies-based particle swarm optimization. Information Sciences, 578, 457-481.
[26] Dhiman, G., & Kumar, V. (2017). Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Advances in Engineering Software, 114, 48-70.
[27] Ayyarao, T. S., Ramakrishna, N. S. S., Elavarasan, R. M., Polumahanthi, N., Rambabu, M., Saini, G., ... & Alatas, B. (2022). War strategy optimization algorithm: a new effective metaheuristic algorithm for global optimization. IEEE Access, 10, 25073-25105.
[28] Alshamlan, H. M., Badr, G. H., & Alohali, Y. A. (2015). Genetic Bee Colony (GBC) algorithm: A new gene selection method for microarray cancer classification. Computational biology and chemistry, 56, 49-60.
[29] Tilahun, S. L. (2019). Balancing the degree of exploration and exploitation of swarm intelligence using parallel computing. International Journal on Artificial Intelligence Tools, 28(03), 1950014.
[30] Jain, M., Singh, V., & Rani, A. (2019). A novel nature-inspired algorithm for optimization: Squirrel search algorithm. Swarm and evolutionary computation, 44, 148-175.
[31] Panigrahi, B. K., Shi, Y., & Lim, M. H. (Eds.). (2011). Handbook of swarm intelligence: concepts, principles and applications (Vol. 8). Springer Science & Business Media.
[32] Yang, J., Qu, L., Shen, Y., Shi, Y., Cheng, S., Zhao, J., & Shen, X. (2020, July). Swarm intelligence in data science: applications, opportunities and challenges. In International Conference on Swarm Intelligence (pp. 3-14). Cham: Springer International Publishing.
[33] Giovanni Sisinna - How does Swarm Intelligence work and what are its potential applications ? - https://www.linkedin.com/pulse/how-doesswarm-intelligence-work-what-its-potential-giovanni-sisinna/
[34] Gu, S., Cheng, R., & Jin, Y. (2018). Feature selection for high-dimensional classification using a competitive swarm optimizer. Soft Computing, 22, 811-822.
[35] Zhang, Y., Song, X. F., & Gong, D. W. (2017). A return-cost-based binary firefly algorithm for feature selection. Information Sciences, 418, 561-574.
[36] Pourpanah, F., Shi, Y., Lim, C. P., Hao, Q., & Tan, C. J. (2019). Feature selection based on brain storm optimization for data classification. Applied Soft Computing, 80, 761-775.
[37] Nguyen, B. H., Xue, B., & Zhang, M. (2020). A survey on swarm intelligence approaches to feature selection in data mining. Swarm and Evolutionary Computation, 54, 100663.
[38] Soltani, M., Chaari, A., & Ben Hmida, F. (2012). A novel fuzzy c-regression model algorithm using a new error measure and particle swarm optimization.
[39] Zedadra, O., Guerrieri, A., Jouandeau, N., Spezzano, G., Seridi, H., & Fortino, G. (2019). Swarm intelligence and IoT-based smart cities: a review. The internet of things for smart urban ecosystems, 177-200.
[40] Ding, S., An, Y., Zhang, X., Wu, F., & Xue, Y. (2017). Wavelet twin support vector machines based on glowworm swarm optimization. Neurocomputing, 225, 157-163.

[41] Tuba, E., Mrkela, L., & Tuba, M. (2016, April). Support vector machine parameter tuning using firefly algorithm. In 2016 26th International Conference Radioelektronika(RADIOELEKTRONIKA) (pp. 413-418). IEEE.
[42] Tang, H., Xu, Y., Lin, A., Heidari, A. A., Wang, M., Chen, H., ... & Li, C. (2020). Predicting green consumption behaviors of students using efficient firefly grey wolf-assisted K-nearest neighbor classifiers. Ieee Access, 8, 35546-35562.
[43] Wu, Q., Liu, H., & Yan, X. (2016). Multi-label classification algorithm research based on swarm intelligence. Cluster Computing, 19, 2075- 2085.
[44] Bida, I., & Aouat, S. (2019). A new approach based on bat algorithm for inducing optimal decision trees classifiers. In Information Systems and Technologies to Support Learning: Proceedings of EMENA-ISTL 2018 2 (pp. 631-640). Springer International Publishing.
[45] Kozak, J., & Boryczka, U. (2016). Collective data mining in the ant colony decision tree approach. Information Sciences, 372, 126-147.
[46] Karpat, Y., & Özel, T. (2005). Hard turning optimization using neural network modeling and swarm intelligence. Transactions of the North American Manufacturing Research Institute of SME, 33, 179-186.
[47] Vrbančič, G., Fister Jr, I., & Podgorelec, V. (2018, June). Swarm intelligence approaches for parameter setting of deep learning neural network: case study on phishing websites classification. In Proceedings of the 8th international conference on web intelligence, mining and semantics (pp. 1-8).
[48] Kang, Q., Liu, S., Zhou, M., & Li, S. (2016). A weight-incorporated similarity-based clustering ensemble method based on swarm intelligence. Knowledge-Based Systems, 104, 156-164.
[49] Tarkhaneh, O., Isazadeh, A., & Khamnei, H. J. (2018). A new hybrid strategy for data clustering using cuckoo search based on Mantegna levy distribution, PSO and k-means. International Journal of Computer Applications in Technology, 58(2), 137-149.
[50] Tuba, E., Strumberger, I., Bacanin, N., Zivkovic, D., & Tuba, M. (2018, April). Cooperative clustering algorithm based on brain storm optimization and k-means. In 2018 28th international conference radioelektronika (RADIOELEKTRONIKA) (pp. 1-5). IEEE.
[51] Stanley, K. O., Clune, J., Lehman, J., & Miikkulainen, R. (2019). Designing neural networks through neuroevolution. Nature Machine Intelligence, 1(1), 24-35.
[52] Wang, B., Xue, B., & Zhang, M. (2020, July). Particle swarm optimisation for evolving deep neural networks for image classification by evolving and stacking transferable blocks. In 2020 IEEE Congress on Evolutionary Computation (CEC) (pp. 1-8). IEEE.
[53] Wang, B., Sun, Y., Xue, B., & Zhang, M. (2019, July). Evolving deep neural networks by multi-objective particle swarm optimization for image classification. In Proceedings of the genetic and evolutionary computation conference (pp. 490-498).
[54] Chakraborty, T., & Datta, S. K. (2017, November). Application of swarm intelligence in internet of things. In 2017 IEEE international symposium on consumer electronics (ISCE) (pp. 67-68). IEEE.
[55] Boveiri, H. R., Khayami, R., Elhoseny, M., & Gunasekaran, M. (2019). An efficient Swarm-Intelligence approach for task scheduling in cloudbased internet of things applications. Journal of Ambient Intelligence and Humanized Computing, 10, 3469-3479.
[56] Omran, M. G. (2006). Particle swarm optimization methods for pattern recognition and image processing (Doctoral dissertation, University of Pretoria).
[57] Bogue, R. (2008). Swarm intelligence and robotics. Industrial Robot: An International Journal, 35(6), 488-495.

[58] Dutot, A., Olivier, D., & Savin, G. (2010, March). Swarm intelligence to distribute simulations in computational ecosystems. In Swarm Intelligence Algorithms and Applications Symposium within AISB 2010 Convention (p. 25).
[59] Zheng, B., Zhang, J., Yoon, S. W., Lam, S. S., Khasawneh, M., & Poranki, S. (2015). Predictive modeling of hospital readmissions using metaheuristics and data mining. Expert Systems with Applications, 42(20), 7110-7120.
[60] Giovanni Sisinna. (Sep 2022). How does Swarm Intelligence work and what are its potential applications?https://www.linkedin.com/pulse/how-does-swarm-intelligence-work-what-its-potential-giovanni-sisinna
[61] Millonas, M. M. (1993). Swarms, phase transitions, and collective intelligence. arXiv preprint adap-org/9306002.
[62] Balavand, A. (2022). A new feature clustering method based on crocodiles hunting strategy optimization algorithm for classification of MRI images. The visual computer, 38(1), 149-178.
[63] Zhiheng, W., & Jianhua, L. (2021). Flamingo search algorithm: a new swarm intelligence optimization algorithm. IEEE Access, 9, 88564-88582.
[64] MiarNaeimi, F., Azizyan, G., & Rashki, M. (2021). Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems. Knowledge-Based Systems, 213, 106711.
[65] Hayyolalam, V., & Kazem, A. A. P. (2020). Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems. Engineering Applications of Artificial Intelligence, 87, 103249.
[66] Xue, J., & Shen, B. (2020). A novel swarm intelligence optimization approach: sparrow search algorithm. Systems science & control engineering, 8(1), 22-34.
[67] Dhiman, G., Garg, M., Nagar, A., Kumar, V., & Dehghani, M. (2021). A novel algorithm for global optimization: rat swarm optimizer. Journal of Ambient Intelligence and Humanized Computing, 12, 8457-8482.
[68] Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. Engineering Applications of Artificial Intelligence, 80, 20-34.
[69] Mirjalili, S. Z., Mirjalili, S., Saremi, S., Faris, H., & Aljarah, I. (2018). Grasshopper optimization algorithm for multi-objective optimization problems. Applied Intelligence, 48, 805-820.
[70] Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. Advances in engineering software, 114, 163-191.
[71] Al-Obaidi, A. T. S., & Abdullah, H. S. (2017). Camel herds algorithm: A new swarm intelligent algorithm to solve optimization problems. International Journal on Perceptive and Cognitive Computing, 3(1).
[72] Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural computing and applications, 27, 1053-1073.
[73] Wu, T. Q., Yao, M., & Yang, J. H. (2016). Dolphin swarm algorithm. Frontiers of Information Technology & Electronic Engineering, 17(8),717-729.
[74] Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Computers & structures, 169, 1-12.

[75] Mirjalili, S. (2015). The ant lion optimizer. Advances in engineering software, 83, 80-98.
[76] Wang, G. G., Deb, S., & Coelho, L. D. S. (2015, December). Elephant herding optimization. In 2015 3rd international symposium on computational and business intelligence (ISCBI) (pp. 1-5). IEEE.
[77] Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-based systems, 89, 228-249.
[78] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.Bibliographies 90
[79] Cuevas, E., Cienfuegos, M., Zaldívar, D., & Pérez-Cisneros, M. (2013). A swarm optimization algorithm inspired in the behavior of the socialspider. Expert Systems with Applications, 40(16), 6374-6384.
[80] Niu, B., & Wang, H. (2012). Bacterial colony optimization. Discrete Dynamics in Nature and Society, 2012.
[81] Nguyen, H. T., & Bhanu, B. (2012, November). Zombie survival optimization: a swarm intelligence algorithm inspired by zombie foraging. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012) (pp. 987-990). IEEE.
[82] Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010) (pp. 65-74). Berlin, Heidelberg: Springer Berlin Heidelberg.
[83] Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In 2009 World congress on nature & biologically inspired computing (NaBIC) (pp. 210-214). IEEE.
[84] Chu, Y., Mi, H., Liao, H., Ji, Z., & Wu, Q. H. (2008, June). A fast bacterial swarming algorithm for high-dimensional function optimization. In 2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence) (pp. 3135-3140). IEEE.
[85] Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimisation. International journal of bio-inspired computation, 2(2),78-84.
[86] Chu, S. C., Tsai, P. W., & Pan, J. S. (2006). Cat swarm optimization. In PRICAI 2006: Trends in Artificial Intelligence: 9th Pacific Rim International Conference on Artificial Intelligence Guilin, China, August 7-11, 2006 Proceedings 9 (pp. 854-858). Springer Berlin Heidelberg.
[87] Basturk, B. (2006). An artificial bee colony (ABC) algorithm for numeric function optimization. In IEEE Swarm Intelligence Symposium, Indianapolis, IN, USA, 2006 (Vol. 2006, p. 12).
[88] Nakrani, S., & Tovey, C. (2004). On honey bees and dynamic server allocation in internet hosting centers. Adaptive behavior, 12(3-4), 223-240.
[89] Shi, Y. (2004). Particle swarm optimization. IEEE connections, 2(1), 8-13.
[90] Maniezzo, V., Gambardella, L. M., & De Luigi, F. (2004). Ant colony optimization. New optimization techniques in engineering, 1(5).

[91] Ronneberger, O., Fischer, P., & Brox, T. (2015, octobre). U-Net : réseaux convolutionnels pour la segmentation biomédicale d'images. Dans *Conférence internationale sur le calcul médical et l'intervention assistée par ordinateur* (pp. 234-241). Springer, Cham.
[92] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., ... & Rueckert, D. (2018). Attention U-Net : apprendre où regarder pour le pancréas. *arXiv preprint arXiv:1804.03999*.

[93] Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2018). UNet++ : une architecture U-Net imbriquée pour la segmentation d'images médicales. Dans *Apprentissage profond en analyse d'image médicale et apprentissage multimodal pour le support à la décision clinique* (pp. 3-11). Springer, Cham.
[94] Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encodeur-décodeur avec convolution atrous séparable pour la segmentation sémantique d'images. Dans *Actes de la conférence européenne sur la vision par ordinateur (ECCV)* (pp. 801-818).
[95] Wang, Q., Feng, J., Song, Z., & Wang, Y. (2019). CE-Net : réseau encodeur de contexte pour la segmentation d'images médicales 2D. *IEEE transactions on medical imaging*, 38(10), 2281-2292.
[96] Ibtehaz, N., & Rahman, M. S. (2020). MultiResUNet : repenser l'architecture U-Net pour la segmentation biomédicale multimodale. *Neural Networks*, 121, 74-87.
[97] Hu, X., Zheng, Y., Lu, J., & Wang, Y. (2022). Polyp-PVT : segmentation de polypes avec Vision Transformer pyramidal. *Medical Image Analysis*, 75, 102284.
[98] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., ... & Girshick, R. (2023). Segment Anything. *arXiv preprint arXiv:2304.02643*.
[99] Mekhmoukh, A., Bensaid, A., & Benidir, M. (2018). Approche hybride combinant PSO, clustering flou et Level Set pour la segmentation d'images médicales. *Applied Soft Computing*, 68, 741-753.
[100] Larabi-Marie-Sainte, S., Belkasmi, M., & Bensaid, A. (2021). Algorithmes d'optimisation bio-inspirés pour la segmentation d'images : revue comparative. *Journal of Computational Science*, 54, 101439.
[101] Abualigah, L., Abd El Aziz, M., Al-Betar, M. A., & Alomari, O. (2021). Algorithme Dragonfly pour la segmentation d'images multi-seuils. *Expert Systems with Applications*, 178, 115054.
[102] Al-Qaness, M. A., Abd Elaziz, M., Oliva, D., & Fan, H. (2022). Algorithme d'optimisation de la baleine à perturbation gaussienne pour la segmentation d'images pulmonaires. *Biomedical Signal Processing and Control*, 74, 103485.
[103] Mohamed, E. A., Abd Elaziz, M., Oliva, D., & Fan, H. (2022). Algorithme hybride GWO-PSO pour la segmentation d'images satellite. *Remote Sensing*, 14(8), 1812.
[104] Fathy, M., Youssef, M., & Elshazly, S. (2023). Fusion des algorithmes Dragonfly et Firefly pour la segmentation d'images thermiques bruitées. *Infrared Physics & Technology*, 127, 104179.
[105] Zhang, Y., Li, X., & Huang, T. (2024). Optimisation par colonie d'abeilles artificielles pour la segmentation de scènes naturelles sur le dataset BSDS500. *Pattern Recognition Letters*, 153, 148-156.
[106] Otsu, N. (1979). Une méthode de sélection de seuil basée sur les histogrammes niveaux de gris. *IEEE transactions on systems, man, and cybernetics*, 9(1), 62-66.
[107] Vite-Chávez, I., Hernández-Lemus, E., & Hernández-Muñoz, C. (2023). Seuillage par canal bleu pour une meilleure segmentation des fruits agricoles. *Computers and Electronics in Agriculture*, 208, 107843.
[108] Salman, R., & Liu, Y. (2012). Méthode hybride K-means et Watershed pour la segmentation d'objets avec frontières floues. *International Journal of Computer Science and Network Security*, 12(1), 82-87.
[109] Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., & Yang, M. H. (2021). SETR : repenser la segmentation sémantique comme un problème séquence-à-séquence avec transformers.

Dans *Actes de la conférence IEEE/CVF sur la vision par ordinateur et reconnaissance des formes* (pp. 6881-6890).
[110] Cao, H., Wang, Y., Chen, J., Jia, D., Zhou, X., Tong, Y., & Li, L. (2021). Swin-Unet : Unet-like pure transformer pour la segmentation d'images médicales.