

République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El Bachir El Ibrahimi de Bordj Bou Arreridj
Faculté des Mathématiques et d'Informatique
Département d'informatique



MEMOIRE

Présenté en vue de l'obtention du diplôme

Master en informatique

Spécialité : Réseaux et Multimédia

THEME

Application d'algorithmes d'optimisation inspirés de la
nature pour la Template matching

Présenté par :

Benmorsi Mokhtar

Houmour Haithem

Soutenu publiquement le : 22/06/2023

Devant le jury composé de:

Président : BEGHOURA Mohamed Amine

Examineur : SAAD SAOUD Manel

Encadreur : BENMESSAHEL Ilyas

2022/2023

Dédicaces

Je dédie humblement ce travail, à toutes les personnes qui ont cru en mes capacités et m'ont encouragé dans mon cheminement d'études .Merci.

A ma chère maman

Mère, tu as été mon premier soutien tout au long de ma vie, et tu m'as fait confiance et m'a encouragée dans les circonstances les plus difficiles.

A mon cher père

Malgré ton absence, que ton âme soit toujours avec moi, je ne t'oublie jamais et ne t'oublierai jamais, sois fier de moi, que Dieu ait pitié de toi

Mon frère et mes soeurs

Ma belle-famille, je vous aime. Merci à tous. Merci d'avoir été à mes côtés et à mes côtés dans les circonstances les plus difficiles. Vous êtes ma source de fierté.

Chers amis, merci d'être près de moi, je vous aime

A tous ceux qui ont cru en moi, merci beaucoup à vous tous, c'est le début d'autres succès

Benmorsi mokhtar

Dédicace

Nous dédions ce modeste travail, comme preuve de respect, de gratitude, et de reconnaissance à :

Nos chers parents : Qui n'ont jamais cessé de nous encourager et nous conseiller

Nos amis et collègues : Pour leur compagnie et bons moments passés ensemble

Houmour Haithem

REMERCIEMENTS

Nous tenons tout d'abord à exprimer notre profonde gratitude envers **Allah, le Tout-Puissant et Miséricordieux**, pour nous avoir accordé la force, le courage et la patience nécessaires pour mener à bien ce modeste travail.

Nous souhaitons également exprimer notre sincère reconnaissance envers notre Encadrant, ilyes benmessahel, pour ses précieux conseils, son assistance inestimable tout au long de cette période de travail et pour son soutien constant et ses orientations. Nos remerciements vont également à l'ensemble des membres honorables du jury qui ont accepté d'évaluer ce travail avec bienveillance.

Nous tenons à remercier chaleureusement nos parents et nos proches pour leur amour, leur soutien constant et leur encouragement indéfectible tout au long de notre parcours académique. Enfin, nous souhaitons exprimer notre gratitude à nos amis pour leur présence et leur soutien précieux.

Abstract

Template matching is the process of finding a template or sub-image inside an original image (source), this process is necessary in many applications such as shape recognition and image processing, current methods are computationally expensive and inefficient in all cases.

As a solution to this problem, we have proposed the method of optimizing the Locust search swarm as a template matching solution, LS employs unique evolutionary factors in its search strategy, which helps to avoid focusing on known solutions and allows to better explore the image search area. Our experimental results showed that the proposed approach achieves a better balance between estimation accuracy and computational cost compared to similar methods.

Keywords: Template matching, image source, image Template,

Locust swarm.

Résumé

La correspondance de modèle (Template matching) est le processus de recherche d'un modèle ou d'une sous-image à l'intérieur d'une image originale (source), ce processus est nécessaire dans de nombreuses applications telles que la reconnaissance de forme et le traitement d'image, Les méthodes existantes sont coûteuses en calcul et inefficaces dans tous les cas.

Comme solution à ce problème, nous avons proposé la méthode d'optimisation de l'essaim de recherche de criquets comme solution de correspondance de modèles, LS utilise des facteurs évolutifs uniques dans sa stratégie de recherche, ce qui permet d'éviter de se concentrer sur des solutions connues et permet de mieux explorer la zone de recherche d'images. Nos résultats expérimentaux ont montré que l'approche proposée permet d'obtenir un meilleur équilibre entre la précision de l'estimation et le coût de calcul par rapport à des méthodes similaires.

Mots clés : La correspondance de modèle, source d'image,

Modèle d'image, essaim de criquets...

ملخص

مطابقة القوالب هي عملية إيجاد قالب أو صورة فرعية داخل صورة أصلية (المصدر)، تعد هذه العملية ضرورية في العديد من التطبيقات مثل التعرف على الأشكال ومعالجة الصور، الطرق الحالية مكلفة حسابياً.

كحل لهذه المشكلة، قمنا باقتراح طريقة تحسين سرّب البحث عن الجراد كحل لمطابقة القوالب، توظف LS عوامل تطويرية فريدة في استراتيجية البحث الخاصة بها، مما يساعد على تجنب التركيز على الحلول المعروفة ويسمح باستكشاف منطقة بحث الصورة بشكل أفضل. اظهرت النتائج التجريبية التي قمنا بيها أن النهج المقترح يحقق توازناً أفضل بين دقة التقدير والتكلفة الحسابية مقارنة بالطرق المماثلة

الكلمات المفتاحية: مطابقة القالب، مصدر الصورة، قالب الصورة، سرّب الجراد.

Table des matières

Chapitre 01	1
Introduction aux Méthode de Template Matching	1
1 .1.Introduction	2
1.2. Template matching.....	2
1 .3. Origine de Template Matching:	4
1 .3. L'importance de faire Template matching	4
1.4. Techniques traditionnelles d'appariement des moules	5
1.5. Méthodes de correspondance de modèles modernes	6
1.6. Applications de correspondance de modèles.....	7
1.6.1. Détection et reconnaissance d'objets :	7
1.6.2. Imagerie médicale :	7
1.6.3. Contrôle de la qualité et inspection dans l'industrie :	8
1.6.4. Analyse de documents :	8
1.6.5. Suivi vidéo :	9
1.6.6. Biométrie :	9
1.7. Méthodes et mathématiques	10
1.8. Mesurer la similarité	12
1.8.1 Corrélation croisée normalisée (CCN)	12
1.8.2. Somme des différences au carré (SSD)	12
1.8.3. Sum of Absolute Differences (SAD)	12
Conclusion.....	13

Chapitre 02.....	14
Les Systèmes Bio-Inspires	14
2.1. Introduction	15
2.2. Définition de l'intelligence artificielle.....	15
2.2.2. Caractéristiques du comportement intelligent	15
2.2.3. Applications d'intelligence artificielle	15
2.2.4. Branches de l'intelligence artificielle	16
2.2.4.1. L'apprentissage automatique (Machine Learning en anglais)	16
2.2.4.2. Systèmes experts (Expert System)	16
2.2.4.3. Reconnaissance de formes et Traitement d'image (Pattern Recognition and image Processing)	17
2.3. Algorithmes d'optimisation	17
2.3.1. Définition de l'optimisation	17
2.3.2. Classification des algorithmes d'optimisation	18
2.3.2.1. Anciens algorithmes d'optimisation	18
2.3.2.2. Algorithmes d'optimisation modernes	18
2.4. Les System Bio-inspires.....	19
2.4.1. Définition :	19
2.4.2. Algorithmes génétiques AG :	19
2.4.2.1. Les Etape des Algorithmes génétique	21
2.4.2.2. Exemple illustratif montrant le mécanisme d'action d'un Algorithmes génétiques.....	21

2.4.3. Intelligence d'essaim	23
2.4.3.1. L'optimisation par essaim de particules (Particle Swarm Optimization)	23
2.4.3.2. L'optimisation par colonies de fourmis (Ant colony optimization)	27
2.5. Conclusion	29
Chapitre 03	30
Application l'optimisation par essaim criquets pour Template Matching	30
3.1. Introduction	31
3.2. Processus de Template matching	31
3.2.1. La stratégie de recherche	31
3.2.2. Le critère de similitude	31
3.3. Comment ça fonctionne TM-NCC	33
3.4. L'algorithme Essaim de Criquets (LS)	35
3.4.1. La phase solitaire	35
3.4.2 La phase sociale.	37
3.4.3 Choisir entre les phases solitaire et sociale	38
3.5. Adaptation de l'algorithme LS avec TM-NCC	40
3.6. Résumé	43
3.7. Conclusion	44
Chapitre 04	45
Application et Resulta	45
4.1. Introduction	46
4.2. Outils et technologies de programmation	46

4.2.1 Python.....	46
4.2.2 OpenCV (cv2).....	46
4.2.3 PyCharm.....	47
4.3. Structure et organisation du code.....	47
4.4.Resultants D'exécution:	52
4.5. Évaluation.....	53
4.5.1 .Comparaison avec l'algorithme PSO.....	55
4.6.Conclusion.....	57
Conclusion générale.....	58
Les Références	59

Liste des figures

Figure 1 Exemple de Template Matching .[1]	3
Figure 2 Déplacement du modèle sur l'image de l'image source. [1].....	3
Figure 3 Détection des tumeurs cérébrales [4].....	8
Figure 4 Clarifier l'analyse des documents. [1].....	8
Figure 5 . Détection faciale du président Obama [1]	9
Figure 6 Les différentes modalités biométriques. [1].....	10
Figure 7 Branches de l'intelligence artificielle. [6]	16
Figure 8 La différence entre la reconnaissance de formes et le traitement d'image.....	17
Figure 9 Classification des algorithmes d'optimisation.	18
Figure 10 . Les cinq niveaux d'organisation d'Algorithmes génétique. [11]	20
Figure 11 . illustration schématique du codage des variables d'optimisation xi . [11]	20
Figure 12 Déplacement d'une particule. [12]	24
Figure 13 Résultats de l'algorithme sur deux itération	26
Figure 14 Il représente le processus de numérisation des images.....	31
Figure 15 Processus de Template matching.....	33
Figure 16 Représentation numérique d'image	34
Figure 17 Représentation numérique du modèle (Template) d'image.	34
Figure 18 Résultat NCC entre l'image et template.	34
Figure 19 Les différentes distributions de la phase solitaire LS..	38
Figure 20 Choisir entre les phases solitaire et sociale	39
Figure 21 Fonction d'évaluation NCC	40

Figure 22 Matrice d'image et template.	41
Figure 23 Les positions aléatoires (x ,y)	41
Figure 24 Les valeurs de la fonction d'évaluation dans la phase solitaire.	42
Figure 25 Représentation des positions dans la phase solitaire.	42
Figure 26 Les nouveau positions (x ,y)	42
Figure 27 a) Les valeurs de la fonction d'évaluation dans la phase social	43
Figure 28 Un plan est un résumé de ce chapitre	43
Figure 29 Logo python	46
Figure 30 Figure 29. Logo OpenCV.....	47
Figure 31 Importation des bibliothèques	47
Figure 32 Chargement de l'image sur python.....	48
Figure 33 Modèle d'image et image source, utilisés dans cette algorithm.....	48
Figure 34 Code fonction d'évaluation	48
Figure 35 Paramètres de l'algorithme LS	49
Figure 36 Générer des positions aléatoires.	49
Figure 37 Algorithme de La phase solitaire.....	50
Figure 38 Algorithme de La phase social.	51
Figure 39 Le résultat de la reconnaissance du visage du robot.....	52
Figure 40 Le résultat de distinguer le nom de Messi	52
Figure 41 Le résultat de la distinction du nombre 7	52
Figure 42 Le résultat de la reconnaissance de Ka'aba	53
Figure 43 Le résultat de la reconnaissance du visage du joueur	53
Figure 44 La courbe d'évolution de la valeur NCC en terme d'itération, pour Figure 36	55

Figure 45 La courbe d'évolution de la valeur NCC en terme d'itération, pour Figure 3956

Figure 46 La courbe d'évolution de la valeur NCC en terme d'itération, pour Figure 3756

Liste des tableaux

Tableau 1 Les avantages et les inconvénients des algorithmes d'optimisation Anciens.18

Tableau 2 Avantages et inconvénients des nouveaux algorithmes d'optimisation.19

Tableau 3 Le nombre d'opérations NCC.54

Liste des abréviations

<i>TM</i>	Template Matching , La correspondance de modèle
<i>LS</i>	Locust Search , essaim de criquets
<i>PSO</i>	Particle swarms optimization
<i>NCC</i>	Normalized Cross-Correlation
<i>ACO</i>	Ant colony optimization , L'optimisation par colonies de fourmis
<i>GA</i>	Algorithme Genetique
<i>TM-NCC</i>	Template Matching base sur NCC

Introduction générale

Dans le domaine du traitement d'images et de la vision par ordinateur, la localisation et la reconnaissance d'objets ou de motifs dans des images numériques constituent l'une des tâches les plus importantes. Ces techniques sont largement utilisées dans de nombreuses applications telles que l'inspection industrielle, la classification des cibles, la photométrie numérique, la télédétection, etc. .

La correspondance de modèles (Template Matching, TM) est une technique de traitement d'images qui vise à localiser et à reconnaître des objets ou des motifs dans une image en trouvant la ressemblance la plus proche entre un sous-image (généralement appelé modèle) et une région correspondante de l'image source. Cette approche repose sur la mesure de similarité entre le modèle et la région de l'image source, en utilisant des métriques spécifiques telles que la somme des différences absolues (Sum of Absolute Differences, SAD), la somme des différences au carré (Sum of Squared Differences, SSD) et la corrélation croisée normalisée (Normalized Cross-Correlation, NCC) .

Cependant, le calcul de ces mesures de similarité s'accompagne d'un coût informatique élevé, ce qui en fait généralement l'opération la plus longue dans un processus de correspondance de modèles. De plus, la plupart des algorithmes de correspondance de modèles traditionnels effectuent une recherche exhaustive sur chaque pixel de l'image source, ce qui limite l'utilisation de ces techniques de reconnaissance dans la plupart des applications de vision par ordinateur en temps réel.

Pour surmonter ces limitations, les chercheurs se sont tournés vers les systèmes inspirés par la nature, qui sont des approches basées sur les principes et les comportements observés dans la nature. Ces systèmes s'inspirent des processus biologiques, des comportements collectifs d'animaux, des mécanismes évolutifs, et les appliquent à des problèmes d'optimisation et de recherche.

Dans cette étude, nous nous intéressons particulièrement aux algorithmes inspirés par la nature et à leur application dans le domaine de la correspondance de modèles. En utilisant des méthodes d'optimisation basées sur l'intelligence collective, telles que la méthode de recherche

des sauterelles (Locust Search - LS), nous cherchons à améliorer le processus de correspondance de modèles en trouvant un meilleur équilibre entre la précision de l'estimation et le coût informatique.

L'objectif de cette recherche est de proposer une approche alternative plus efficace pour la correspondance de modèles en exploitant les avantages des systèmes inspirés par la nature. Nous évaluerons l'efficacité de notre approche à travers des expérimentations et des comparaisons avec d'autres techniques similaires. Les résultats de cette recherche contribueront à l'avancement des applications de vision par ordinateur en temps réel, ouvrant de nouvelles perspectives dans le domaine de la localisation et de la reconnaissance d'objets.

Chapitre 01

Introduction aux Méthode de Template Matching

1 .1.Introduction

Dans notre ère moderne où la quantité d'images disponibles ne cesse d'augmenter de manière exponentielle, la correspondance de modèles d'images est devenue un outil essentiel pour extraire des informations et comprendre les données visuelles. La correspondance de modèles d'images est un processus important utilisé pour trouver des similitudes entre un modèle de référence et un ensemble d'images à analyser ou à comparer.

La correspondance de modèles d'images est une technique essentielle dans le domaine du traitement d'images et de la vision par ordinateur. Elle est utilisée dans diverses applications telles que la reconnaissance faciale, la détection de formes, l'imagerie médicale, la vérification biométrique, l'imagerie spatiale, et bien d'autres. Elle facilite l'analyse, la classification et la reconnaissance de différents motifs dans les images.

Dans ce chapitre, nous explorerons l'importance de la correspondance de modèles d'images, ses défis et les techniques utilisées. Nous discuterons des différentes mesures de similarité entre les modèles et les images, ainsi que des outils utilisés pour effectuer la correspondance des modèles d'images. Nous présenterons également des exemples d'applications de la correspondance de modèles d'images dans divers domaines.

1.2. Template matching

Le correspondant au modèle est une technique permettant de trouver des zones d'une image qui sont similaires à un patch (modèle). Un patch est une petite image avec certaines caractéristiques. Le but de la correspondance de modèles est de trouver le patch/modèle dans une image. Pour le trouver, l'utilisateur doit donner deux images d'entrée : Source Image (S) - L'image dans laquelle trouver le modèle, et Template Image (T) - L'image qui doit être trouvée dans l'image source

- Il s'agit essentiellement d'une méthode pour rechercher et trouver l'emplacement d'une image modèle dans une image plus grande.
- L'idée ici est de trouver des régions identiques d'une image qui correspondent à un modèle que nous fournissons, en donnant un seuil.
- Le seuil dépend de la précision avec laquelle on veut détecter le modèle dans l'image source.

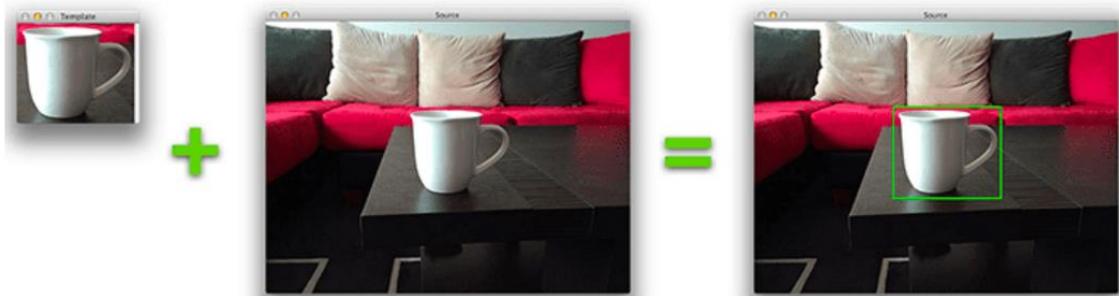


Figure 1 Exemple de Template Matching .[1]

La correspondance de modèles peut être considérée comme une forme très basique de détection d'objets. En utilisant la correspondance de modèle, nous pouvons détecter des objets dans une image d'entrée en utilisant un "modèle" contenant l'objet que nous voulons détecter,

Essentiellement, cela signifie que nous avons besoin de deux images pour appliquer la correspondance de modèle :

1. Image source : C'est l'image dans laquelle nous nous attendons à trouver une correspondance avec notre modèle.
2. Image modèle : Le "patch d'objet" que nous recherchons dans l'image source.

Pour trouver le modèle dans l'image source, nous glissons le modèle de gauche à droite et de haut en bas sur la source :



Figure 2 Déplacement du modèle sur l'image de l'image source. [1]

1.3. Origine de Template Matching:

Le concept de correspondant au modèle trouve son origine dans les premiers jours de la recherche sur la vision par ordinateur, dans les années 1960. David Marr et Tomaso Poggio ont été parmi les premiers chercheurs à proposer un modèle informatique pour le système visuel, qui impliquait l'utilisation de modèles pour l'extraction et la mise en correspondance des caractéristiques. Leur modèle repose sur l'idée que le système visuel extrait des informations d'une image en la comparant à un ensemble de modèles représentant différentes caractéristiques. Les modèles ont été conçus pour capturer les caractéristiques essentielles des éléments, telles que leur taille, leur forme et leur orientation.

Le modèle de Marr et Poggio était limité par la puissance de calcul de l'époque, et ce n'est que dans les années 1970 que la correspondance des modèles a été mise en œuvre pour la première fois dans un programme informatique. En 1974, Sven J. Dickinson et Alex Pentland ont mis au point un algorithme de correspondance de modèles qui utilisait une approche basée sur la corrélation. L'algorithme était capable de détecter et de suivre des objets en mouvement en temps réel à partir d'un flux vidéo.

Au fil des ans, la comparaison de modèles a continué d'évoluer et de nouvelles méthodes et techniques ont été mises au point pour améliorer sa précision et son efficacité. Aujourd'hui, la correspondance de modèles est une technique largement utilisée dans le domaine de la vision par ordinateur, avec des applications dans la reconnaissance d'objets, le suivi et l'analyse de mouvements. Elle est utilisée dans un large éventail d'industries, de la fabrication à la robotique en passant par la sécurité et la surveillance. [1] .

1.3. L'importance de faire Template matching

L'importance de la correspondance des modèles dans l'analyse d'images est cruciale dans le domaine de la vision par ordinateur et du traitement d'images. Cette technique permet d'extraire des informations précieuses à partir des images et de comprendre leur contenu visuel.

Nous pouvons dire, la correspondance des modèles est une technique essentielle dans l'analyse d'images scientifiques. Elle permet d'extraire des informations pertinentes, de détecter des objets, d'analyser des formes et des structures, de suivre le mouvement et de faciliter l'indexation et la recherche d'images. Son utilisation dans la recherche scientifique contribue à

la compréhension des données visuelles et ouvre la voie à de nombreuses applications dans divers domaines de la science et de la technologie. [4] .

1.4. Techniques traditionnelles d'appariement des moules

Les techniques traditionnelles d'appariement des moules font référence aux approches classiques utilisées pour trouver des correspondances entre deux moules ou motifs. Voici quelques exemples de techniques traditionnelles couramment utilisées :

- Méthode de la distance Euclidienne : Cette méthode mesure la distance entre les caractéristiques des moules en utilisant la distance euclidienne. Les caractéristiques peuvent inclure des attributs tels que la couleur, la forme, la texture, etc. Plus la distance euclidienne est petite, plus les moules sont considérés comme similaires.
- Méthode du coefficient de corrélation : Cette méthode mesure le degré de corrélation linéaire entre les caractéristiques des moules. Elle évalue la similarité en utilisant le coefficient de corrélation qui varie de -1 à 1. Une valeur proche de 1 indique une forte corrélation et donc une forte similarité
- Méthode du seuillage : Cette méthode consiste à définir des seuils pour chaque caractéristique des moules. Si les valeurs des caractéristiques des deux moules sont en dessous des seuils, ils sont considérés comme similaires. Sinon, ils sont considérés comme différents.
- Méthode du ratio d'aire : Cette méthode compare les ratios d'aire des moules. Elle calcule le rapport entre l'aire de la partie commune entre les moules et l'aire totale des moules. Plus le ratio d'aire est proche de 1, plus les moules sont considérés comme similaires.

Ces techniques traditionnelles offrent des approches simples et rapides pour l'appariement des moules. Cependant, elles peuvent être limitées en termes de capacité à traiter des variations complexes dans les données des moules. C'est pourquoi de plus en plus de chercheurs se tournent vers des techniques plus avancées, telles que l'apprentissage automatique et l'apprentissage profond, pour améliorer la précision de l'appariement des moules. [1] .

1.5. Méthodes de correspondance de modèles modernes

Les méthodes de correspondance de modèles modernes utilisées dans le domaine du traitement d'images et de la vision par ordinateur ont connu des avancées significatives. Voici quelques-unes des méthodes les plus couramment utilisées :

1. **Descripteurs locaux** : Les descripteurs locaux sont des représentations numériques de régions spécifiques dans une image. Des algorithmes tels que SIFT (Scale-Invariant Feature Transform), SURF (Speeded Up Robust Features) et ORB (Oriented FAST and Rotated BRIEF) sont largement utilisés pour extraire ces descripteurs. Ils sont robustes aux variations d'échelle, aux transformations géométriques et aux changements d'illumination, ce qui les rend efficaces pour la correspondance de modèles.
2. **Descripteurs profonds** : Les descripteurs profonds, tels que les réseaux de neurones convolutifs (CNN), ont révolutionné la correspondance de modèles. Ces descripteurs sont appris automatiquement à partir de grandes bases de données d'images et peuvent capturer des caractéristiques complexes et hiérarchiques des images. Des architectures populaires telles que VGGNet, ResNet et InceptionNet sont souvent utilisées comme descripteurs profonds pour la correspondance de modèles.
3. **Méthodes d'apprentissage par correspondance** : Les méthodes d'apprentissage par correspondance utilisent des algorithmes d'apprentissage automatique pour apprendre à associer des paires d'images correspondantes. Ces méthodes permettent de capturer des relations complexes entre les images et d'améliorer la précision de la correspondance. Des techniques telles que les réseaux de neurones siamois et les réseaux de correspondance de points sont utilisées pour apprendre les correspondances entre les modèles et les images cibles.
4. **Correspondance basée sur la structure** : Les méthodes de correspondance basées sur la structure utilisent des informations géométriques et spatiales pour effectuer la correspondance entre les modèles et les images. Par exemple, la correspondance basée sur le flux optique est utilisée pour estimer le mouvement des objets dans une séquence d'images, tandis que la correspondance basée sur les graphes permet de modéliser les relations entre les points d'intérêt.

Correspondance basée sur l'apparence : Les méthodes de correspondance basées sur l'apparence utilisent les caractéristiques visuelles des objets pour effectuer la correspondance. Cela peut inclure des techniques telles que la corrélation de phase, qui compare les motifs de

phase dans les images, ou l'utilisation de descripteurs d'apparence tels que les histogrammes de couleur ou les histogrammes de gradients orientés (HOG).

Ces méthodes de correspondance de modèles modernes ont considérablement amélioré les performances et la robustesse de la correspondance d'images. Elles sont utilisées dans de nombreux domaines tels que la reconnaissance d'objets, la réalité augmentée, la surveillance vidéo, la robotique et bien d'autres. Les chercheurs continuent de travailler sur de nouvelles techniques et approches pour améliorer encore davantage la précision et l'efficacité de la correspondance de modèles. [4] .

1.6. Applications de correspondance de modèles

La correspondance des modèles a de nombreuses applications dans divers domaines. Voici quelques domaines d'application courants où la correspondance de modèles est largement utilisée [4] :

1.6.1. Détection et reconnaissance d'objets :

La correspondance des modèles est souvent utilisée pour détecter et reconnaître des objets dans des images ou des vidéos. Une image modèle représentant l'objet en question est comparée aux données d'entrée afin d'identifier les instances de l'objet. Cette technique est utile dans des applications telles que la surveillance, la robotique et la conduite autonome.

1.6.2. Imagerie médicale :

L'appariement de modèles trouve des applications dans l'imagerie médicale pour des tâches telles que la détection de tumeurs, la segmentation d'organes et l'analyse de cellules. Des modèles représentant des structures anatomiques spécifiques ou des schémas pathologiques peuvent être comparés à des images médicales pour faciliter le diagnostic et la planification du traitement.

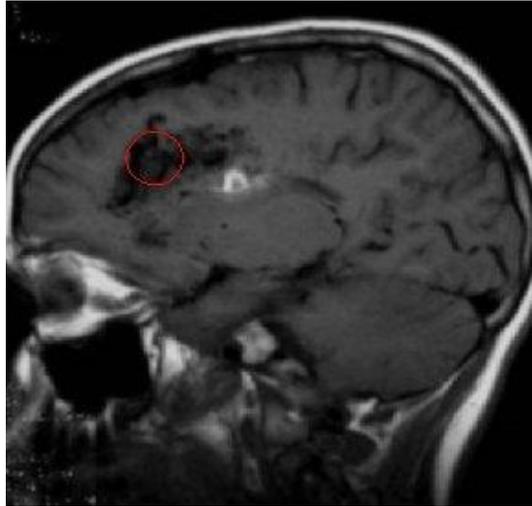


Figure 3 Détection des tumeurs cérébrales [4]

1.6.3. Contrôle de la qualité et inspection dans l'industrie :

La mise en correspondance de modèles est utilisée dans l'industrie pour inspecter et vérifier la qualité des produits fabriqués. Les modèles représentant des composants exempts de défauts ou des motifs spécifiques sont comparés aux images capturées pour identifier les défauts, les anomalies ou les déviations dans le processus de production.

1.6.4. Analyse de documents :

La correspondance de modèles est utilisée dans les tâches d'analyse de documents, telles que la reconnaissance de texte, le traitement de formulaires et la vérification de signatures. Des modèles représentant des caractères, des symboles ou des styles d'écriture spécifiques sont comparés à des images de documents afin d'extraire des informations pertinentes ou d'effectuer des tâches de vérification.



Figure 4 Clarifier l'analyse des documents. [1]

1.6.5. Suivi vidéo :

La correspondance des modèles est utilisée pour suivre des objets ou des régions d'intérêt dans des séquences vidéo. Les modèles représentant l'objet ou la région cible sont comparés à des images consécutives pour estimer son mouvement, ce qui permet des applications telles que le suivi d'objets, la reconnaissance d'actions et la vidéosurveillance.



Figure 5 . Détection faciale du président Obama. [1]

1.6.6. Biométrie :

La correspondance des modèles est utilisée dans les systèmes biométriques pour des tâches telles que la reconnaissance des visages, des empreintes digitales et de l'iris. Les modèles représentant des caractéristiques biométriques uniques sont comparés aux données biométriques acquises pour établir l'identité ou procéder à l'authentification.

Il ne s'agit là que de quelques exemples des diverses applications de la comparaison de gabarits. La polyvalence et la flexibilité de cette technique la rendent applicable à un large éventail de domaines nécessitant une analyse visuelle et une reconnaissance des formes.

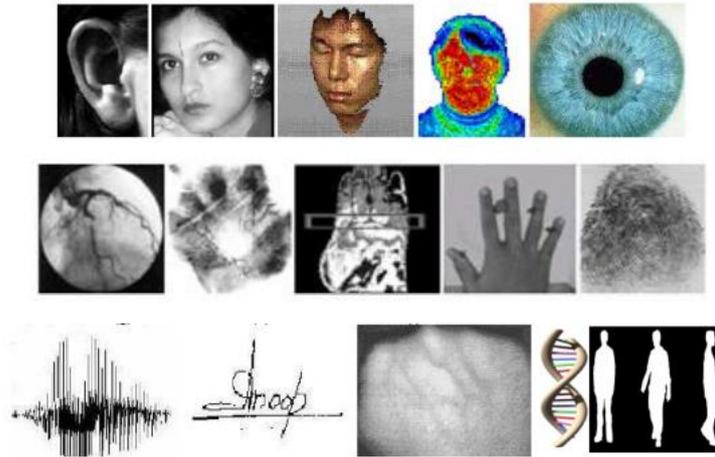


Figure 6 Les différentes modalités. [1]

1.7. Méthodes et mathématiques

Les méthodes de mise en correspondance de modèles font appel à diverses techniques et concepts mathématiques pour comparer une image de modèle à une image plus grande. Voici quelques méthodes et concepts mathématiques couramment utilisés dans la mise en correspondance de modèles [2] :

- **Corrélation** : Les méthodes basées sur la corrélation sont largement utilisées dans la mise en correspondance de modèles. Le coefficient de corrélation mesure la similitude entre le modèle et la région de l'image à comparer. La corrélation croisée et la corrélation croisée normalisée sont des formules couramment utilisées pour calculer le score de corrélation.
- **Convolution** : La convolution est une opération mathématique utilisée dans la mise en correspondance de modèles. Elle consiste à faire glisser le modèle sur l'image et à calculer la somme des multiplications par éléments entre le modèle et la région correspondante de l'image. La convolution est utilisée pour obtenir un score de similarité ou une carte de réponse indiquant la probabilité de présence du modèle à différents endroits de l'image.
- **Extraction de caractéristiques** : Les méthodes basées sur les caractéristiques consistent à extraire les caractéristiques saillantes du modèle et de l'image afin d'établir des correspondances. Les algorithmes d'extraction de caractéristiques, tels que SIFT (Scale-Invariant Feature Transform) ou SURF (Speeded Up Robust Features), sont

utilisés pour détecter et décrire les caractéristiques distinctives qui peuvent être mises en correspondance entre le modèle et l'image.

- **Modèles de transformation** : La mise en correspondance de modèles nécessite souvent la prise en compte de transformations, telles que la rotation, la mise à l'échelle et la translation, afin de tenir compte des différences entre le modèle et l'image. Les modèles de transformation, tels que les transformations affines ou l'homographie, sont utilisés pour estimer les transformations géométriques nécessaires pour aligner le modèle sur l'image.
- **Algorithmes d'optimisation** : Les techniques d'optimisation sont employées pour rechercher la meilleure correspondance entre le modèle et l'image. Des techniques telles que la recherche exhaustive, la descente de gradient ou les méthodes d'optimisation locale sont utilisées pour trouver la position optimale ou les paramètres de transformation qui maximisent la similarité ou minimisent la dissimilarité entre le modèle et l'image.
- **Méthodes statistiques** : Les concepts statistiques sont utilisés dans la mise en correspondance des modèles pour modéliser et comparer les intensités ou les caractéristiques des images. Des mesures telles que la moyenne, l'écart-type, la comparaison d'histogrammes ou les tests d'hypothèses statistiques sont utilisées pour quantifier la similarité entre le modèle et la région de l'image à mettre en correspondance.
- **Apprentissage automatique et apprentissage profond** : Les techniques d'apprentissage automatique et d'apprentissage profond ont révolutionné l'appariement des modèles. Les réseaux neuronaux convolutés (CNN) et d'autres modèles d'apprentissage automatique peuvent apprendre et extraire automatiquement des caractéristiques pertinentes à partir d'images, ce qui les rend très utiles pour les tâches de mise en correspondance de modèles.

Ces méthodes et concepts mathématiques constituent la base de la mise en correspondance de modèles, permettant une comparaison et une identification efficaces des modèles dans des images plus grandes. Le choix spécifique des méthodes et des mathématiques dépend de l'application et des caractéristiques du problème de mise en correspondance des gabarits. [2] .

1.8. Mesurer la similarité

La mesure de la similarité entre le modèle et l'image est une étape cruciale du processus de mise en correspondance des modèles, car elle détermine la précision et la fiabilité des résultats. Nous essaierons de clarifier certains quelques mesures de similarité courantes utilisées dans la mise en correspondance de modèles.

1.8.1 Corrélation croisée normalisée (CCN)

La corrélation croisée normalisée (Normalized Cross-Correlation) est l'une des mesures de similarité les plus populaires utilisées dans la mise en correspondance de modèles. La CCN mesure la similarité entre deux images en calculant le produit de points de leurs valeurs d'intensité normalisées. La CCN est comprise entre -1 et 1, 1 indiquant une correspondance parfaite et -1 une non-corrélation parfaite. [3].

La métrique NCC est calculée à l'aide de l'équation suivante :

$$NCC = \frac{\sum_{i=1}^m \sum_{j=1}^n [I(u+i, v+j) - \bar{I}(u, v)] \cdot [R(i, j) - \bar{R}]}{\sqrt{\left[\sum_{i=1}^m \sum_{j=1}^n [I(u+i, v+j) - \bar{I}(u, v)]^2 \right] \cdot \left[\sum_{i=1}^m \sum_{j=1}^n [R(i, j) - \bar{R}]^2 \right]}} \quad (1)$$

1.8.2. Somme des différences au carré (SSD)

La somme des différences quadratiques (Sum of Squared Differences) est une autre mesure de similarité couramment utilisée dans la mise en correspondance de modèles. La SSD mesure la différence entre les pixels correspondants du modèle et de l'image, élève la différence au carré et additionne les valeurs. La DSS va de 0 à l'infini, 0 indiquant une correspondance parfaite et des valeurs plus élevées indiquant une plus grande différence entre les modèles.

La mesure de la DSS est calculée à l'aide de l'équation suivante :

$$SD = \sqrt{\mathbf{S}^2} = \sqrt{\frac{\mathbf{SS}}{N-1}} = \sqrt{\frac{\sum(\mathbf{x}-\bar{\mathbf{x}})^2}{N-1}} \quad (2)$$

1.8.3. La somme des différences absolues (SAD)

Dans le traitement d'images numériques, la somme des différences absolues (SAD) est une mesure de la similarité entre les blocs d'images. Il est calculé en prenant la différence absolue

entre chaque pixel dans le bloc d'origine et le pixel correspondant dans le bloc utilisé pour la comparaison...

$$\text{SAD}(x, y, u, v) = \sum_{dx=-W}^W \sum_{dy=-W}^W |I_c(x + dx, y + dy) - I_r(x + u + dx, y + v + dy)| \quad (3)$$

1.9. Conclusion

Dans ce chapitre, nous avons expliqué le concept de correspondance de modèle et son origine, en expliquant les utilisations les plus importantes de la correspondance de modèle et la relation des méthodes mathématiques dans la correspondance de modèle.

Chapitre 02

Les Systèmes Bio-Inspires

2.1. Introduction

Dans ce chapitre, nous explorerons les concepts de base de l'intelligence artificielle (IA) et de l'optimisation, ainsi que ses différentes branches. Nous discuterons de la définition de l'intelligence artificielle et de ses caractéristiques de base, telles que l'apprentissage automatique, les systèmes experts et la reconnaissance des formes. De plus, nous examinerons la définition de l'optimisation et la classification des algorithmes d'optimisation, nous nous concentrerons sur les systèmes inspirés de la biologie, y compris les algorithmes génétiques et l'optimisation des essaims de particules, et donnerons des exemples pratiques pour illustrer leur application. Ce chapitre jettera les bases nécessaires à notre étude sur la mise en œuvre de l'algorithme LS à l'aide de la technologie de correspondance de motifs basée sur NCC.

2.2. Définition de l'intelligence artificielle

En termes de langage, l'intelligence est Capacité de saisir une chose par la pensée. [5]

Selon notre foi. L'intelligence concerne les capacités mentales telles que la réflexion, l'analyse, la communication, la déduction, l'apprentissage et la classification.

L'intelligence artificielle est la simulation ou la tentative de simulation de l'intelligence d'organismes vivants (humains, essaims de criquets, etc.) en créant un programme informatique capable de produire un comportement intelligent.

2.2.2. Caractéristiques du comportement intelligent

- conclusion
- Apprentissage
- Comprendre les langues naturelles
- Reconnaître et classer les choses

2.2.3. Applications d'intelligence artificielle

- Systèmes experts.
- Réseaux de neurones.
- Reconnaissance du langage naturel.
- Robotique.
- Vision par ordinateur.

2.2.4. Branches de l'intelligence artificielle

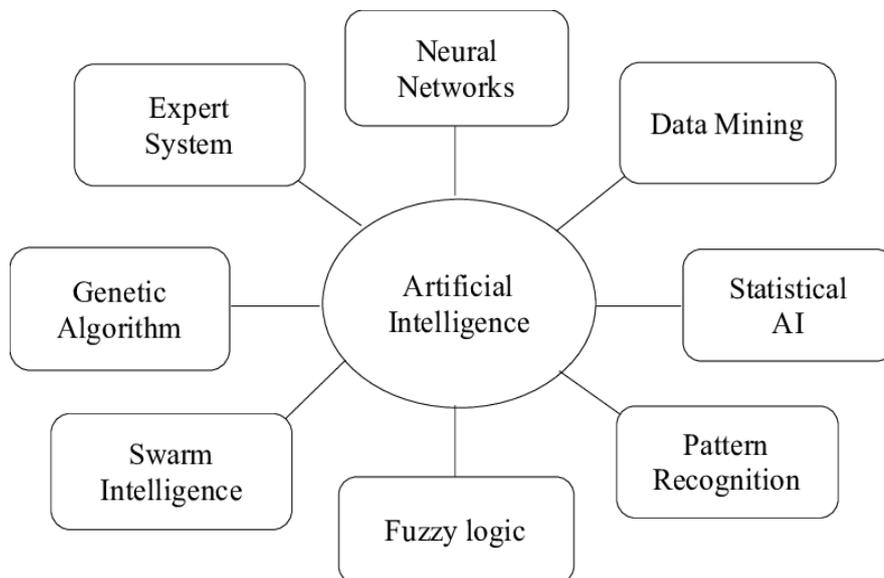


Figure 7 Branches de l'intelligence artificielle. [6]

2.2.4.1. L'apprentissage automatique (Machine Learning en anglais)

C'est une branche de l'intelligence artificielle, et concerne la conception et le développement de logiciels et de technologies qui permettent à une machine d'acquérir une fonction d'apprentissage

L'apprentissage automatique a plusieurs domaines:

- moteurs de recherche
- Reconnaissance du langage naturel.
- Reconnaissance de formes

2.2.4.2. Systèmes experts (Expert System)

‘le système de connaissances ou le système expert est ce programme intelligent qui utilise les règles tirées de l'expérience humaine sous la forme de conditions et de résultats dans un domaine spécifique et l'utilisation de méthodes de dérivation et d'inférence pour extraire et déduire des résultats justifiés par des raisons résultant de la conformité de ces conditions ou résultats avec une condition ou résultat problème spécifique pour lequel une solution doit être trouvée.’ [7]

2.2.4.3. Reconnaissance de formes et Traitement d'image (Pattern Recognition and image Processing)

Une branche de l'intelligence artificielle. Il se soucie de :

- Reconnaître les images saisies dans l'ordinateur.
- Effectuer des opérations sur l'image d'entrée

La différence entre la reconnaissance de formes et le traitement d'image

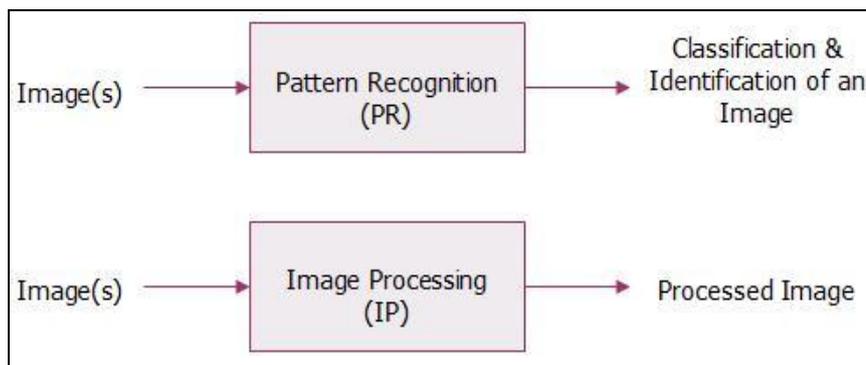


Figure 8 La différence entre la reconnaissance de formes et le traitement d'image

Il existe quatre approches de base utilisées dans le monde des approches ou méthodes de reconnaissance de formes :

- Template-Matching and Corrélation Method / Méthode de correspondance de modèles et de corrélation
- Statically Approche / Approche statique

2.3. Algorithmes d'optimisation

2.3.1 Définition de l'optimisation

L'optimisation est un outil important dans la science de la décision et dans l'analyse des systèmes physiques. Pour utiliser cet outil, nous devons d'abord identifier un objectif, une mesure quantitative de la performance du système à l'étude. Cet objectif peut être le profit, le temps, l'énergie potentielle ou toute quantité ou combinaison de quantités pouvant être représentées par un seul nombre. L'objectif dépend de certaines caractéristiques du système, appelées variables ou inconnues. Notre but est de trouver les valeurs des variables qui optimisent l'objectif. Souvent, les variables sont restreintes ou contraintes d'une manière ou d'une autre. Par exemple, des quantités

telles que la densité électronique dans une molécule et le taux d'intérêt sur un prêt ne peuvent pas être négatives. [8]

2.3.2. Classification des algorithmes d'optimisation

Il existe de nombreuses classifications, mais la plus courante est la classification suivante :

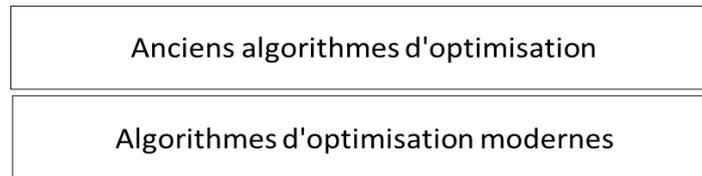


Figure 9 Classification des algorithmes d'optimisation.

2.3.2.1. Anciens algorithmes d'optimisation

Ce sont d'anciens algorithmes qui sont utilisés depuis des décennies et qui ont été développés. Exemples :

Newton-Raphson, Steeples descente, LP & NLP, L'algorithme du simplexe.....

Tableau 1 Les avantages et les inconvénients des algorithmes d'optimisation Anciens.

Avantages	inconvénients
Simplicité	difficile à appliquer
Robustesse	Temps de calcul élevé
Facile à apprendre car il a de nombreuses références	

2.3.2.2. Algorithmes d'optimisation modernes

Les algorithmes d'optimisation modernes sont des approches avancées et adaptatives qui utilisent des techniques heuristiques, stochastiques, d'apprentissage automatique et basées sur la simulation pour résoudre efficacement des problèmes d'optimisation complexes dans une variété de domaines d'application.

Exemples :

- Algorithmes génétiques ou évolutifs . . .
- L'algorithme ACO (Ant Colony Optimization)

Tableau 2 Avantages et inconvénients des nouveaux algorithmes d'optimisation.

Avantages	inconvénients
Très forte pour trouver les meilleures solutions. Résoudre les problèmes les plus difficiles	consommateur de ressources informatiques Manque de références scientifiques Vous devez comprendre certains phénomènes scientifiques

Remarque : Nous ne pouvons pas dire que les nouveaux algorithmes sont plus rapides que les anciens algorithmes ou vice versa, pour les raisons suivantes :

- Variation de l'environnement d'exploitation.
- qualité de l'émission visée.
- possibilité de modifier ou de mixer les algorithmes...(Hybrid Optimization Algorithm) .

Il existe de nombreuses sous-classifications des algorithmes d'optimisation modernes, mais nous ne parlerons que des systèmes inspirés par la nature car c'est l'objet de notre étude.

2.4. Les System Bio-inspires

2.4.1. Définition :

Les System Bio-inspires, ou Les systèmes inspirés de la nature sont des technologies informatiques dont la philosophie est basée sur la prise de phénomènes naturels tels que la biologie, la physique, l'astronomie, la chimie et leur application pour résoudre des problèmes complexes.

Ils simulent simplement l'intelligence de la nature et sont une sous-classe des algorithmes d'optimisation modernes.

2.4.2. Algorithmes génétiques AG :

Aussi appelé calcul évolutif. L'évolution peut-elle être intelligente ? Et les membres actifs agissent-ils envers son environnement de manière intelligente ?

Nous ne parlerons pas de l'origine des théories de survie de Darwin. Nous parlerons des algorithmes génétiques introduits par Holland dans les années 70 du siècle dernier, et son objectif était que l'ordinateur fasse ce que fait la nature. [9]

La description de cet algorithme est la suivante :

- Un chromosome artificiel consiste en une séquence de gènes, chacun représenté par le bit 0 ou 1.
- La nature est capable de s'adapter, d'apprendre et de trouver aveuglément de bons chromosomes (gènes).
- Nous avons une mesure de la modification chromosomique, qui sont deux :
 - L'accouplement : est l'échange de deux parties des deux chromosomes.
 - Mutation : Un changement est effectué à un endroit aléatoire dans le chromosome.

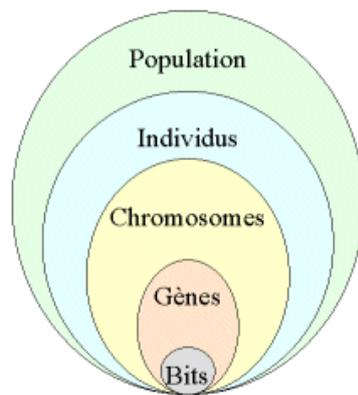


Figure 10 . Les cinq niveaux d'organisation d'Algorithmes génétique. [11]

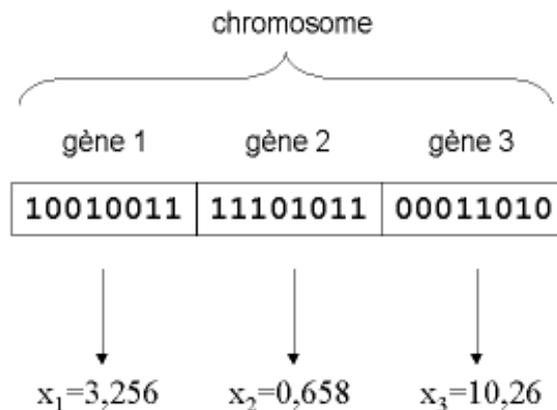


Figure 11 . illustration schématique du codage des variables d'optimisation x_i . [11]

2.4.2.1. Les Etape des Algorithmes génétique

1. Représenter la gamme de variables du problème à résoudre par un chromosome de longueur fixe. supposer un nombre chromosomes N et la probabilité de croisement, et la probabilité de mutation p_m .
2. Définir une fonction de compatibilité qui mesure l'adéquation du chromosome à la solution.
3. Générer des chromosomes au hasard. ..., x_1, x_2, \dots, x_N
4. Calcul de la pertinence de chacun des chromosomes ($f(x), \dots, f(x)$).
5. Sélection d'une paire de chromosomes avec une probabilité liée à leur pertinence.
6. Génération d'une nouvelle paire de chromosomes après l'application de l'accouplement et de la mutation.
7. Mettre les nouveaux chromosomes dans un nouvel ensemble
8. Répétez l'étape (5) jusqu'à ce que le nouveau groupe soit de taille N.
9. Remplacement du groupe parent par le groupe enfants.
10. Revenez à l'étape (4) et répétez jusqu'à ce qu'une condition d'arrêt (y compris la recherche d'une solution) soit remplie.

2.4.2.2. Exemple illustratif montrant le mécanisme d'action d'un Algorithmes génétiques

Le problème : trouver un nombre sur l'intervalle [0 255] et $\text{nombre}(1) = \text{nombre}(0)$

Comme : 11110000. [9]

La solution :

1. Chaque chromosome est un octet et On suppose qu'il y a quatre chromosomes.

$$h = \boxed{0} \boxed{1} \boxed{1} \boxed{1} \boxed{1} \boxed{1} \boxed{1} \boxed{0} = 126$$

2. On connaît la fonction d'appariement suivante (fitness fonction)
 - a. $f(h) = 8 - |n_1 - n_2|$
 - b. La valeur maximal est 8 tel que $n_1 = n_2 = 4$.
 - c. La valeur minimale est 0 tel que $n_1 = 8$ ou $n_2 = 8$.
3. Générer 4 nombres aléatoires (chromosomes) $h_1 h_2 h_3 h_4$

0	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1	0
0	0	1	0	0	1	0	0	0

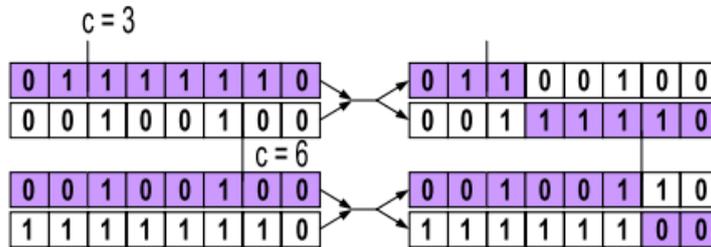
4. Calculs $f(h_i)$.

h_1	0	1	1	1	1	1	1	0	$f(h_1) = 8 - 6 - 2 = 4$
h_2	1	1	1	1	1	1	1	0	$f(h_2) = 8 - 7 - 1 = 2$
h_3	0	0	1	0	0	1	0	0	$f(h_3) = 8 - 2 - 6 = 4$
h_4	0	0	0	0	0	0	0	1	$f(h_4) = 8 - 1 - 7 = 2$

5. Calculs $P(h_i)$ et choisis deux chromosomes

$$P(h_i) = \frac{f(h_i)}{\sum_{j=1}^N f(h_j)} \quad p(h_1) = p(h_3) = 4/12; \quad p(h_2) = p(h_4) = 2/12$$

6. L'accouplement entre (h_1 et h_2), (h_3 , h_2)



L'accouplement a lieu $c=3$ de la première paire et a lieu $c=6$ de la deuxième paire.

7. Mutation : Changer quelques bits

0	1	1	0	0	1	0	0	$f = 8 - 3 - 5 = 6$
0	0	1	1	0	1	1	0	$f = 8 - 4 - 4 = 8$
0	0	1	0	0	1	1	0	$f = 8 - 3 - 7 = 4$
1	0	1	1	1	1	0	0	$f = 8 - 5 - 3 = 6$

Ici, nous n'avons pas besoin de répéter car l'un des chromosomes remplit la valeur 8. Chaque itération de l'algorithme est appelée une génération. Si nous n'obtenons pas de résultats satisfaisants, nous recommençons l'algorithme depuis le début.

2.4.3. Intelligence d'essaim

L'intelligence en essaim est une méthode méta heuristique dans le domaine de l'intelligence artificielle, utilisée pour résoudre des problèmes d'optimisation.

Il est basé sur le comportement collectif d'insectes sociaux, de nuées d'oiseaux ou de bancs de poissons. Ces animaux peuvent résoudre des tâches complexes sans avoir une unité de contrôle centralisée. Les algorithmes bien connus dans le domaine de l'intelligence en essaim qui ont vu le jour ces dernières années comprennent l'optimisation des colonies de fourmis (ACO) , basé sur le comportement de recherche de nourriture des fourmis, particule optimisation en essaim (PSO) , basée sur les comportements de troupeaux d'oiseaux et bancs de poissons, et colonie d'abeilles artificielles (ABC) , basé sur le comportement de butinage des abeilles . [10]

2.4.3.1. L'optimisation par essaim de particules (Particle Swarm Optimization)

2.4.3.1.1. Définition :

Il est indiqué par l'abréviation PSO.

- Technique de recherche utilisée en informatique pour trouver des solutions optimales à des problèmes d'optimisation.
- PSO est une intelligence d'essaim inspirée du comportement social de la nature et des mouvements dynamiques avec des communications d'insectes, d'oiseaux et de poissons.
- Utilise un certain nombre de particules qui constituent un essaim se déplaçant dans l'espace de recherche à la recherche de la meilleure solution.
- Chaque particule dans l'espace de recherche ajuste son "vol" en fonction de sa propre expérience de vol ainsi que de l'expérience de vol des autres particules.

Mots clés de PSO :

- Particules i : Toute solution possible.
- Population N : Groupe de toutes les particules.
- Espace de recherche $[a, b]$ Toutes les solutions possibles au problème.

- Chaque particule est supposée avoir deux caractéristiques :
 - Position x_i et vitesse v_i .
- Chaque particule suit : Personnel Best $P_{best i}$ et Globale Best g_{best}

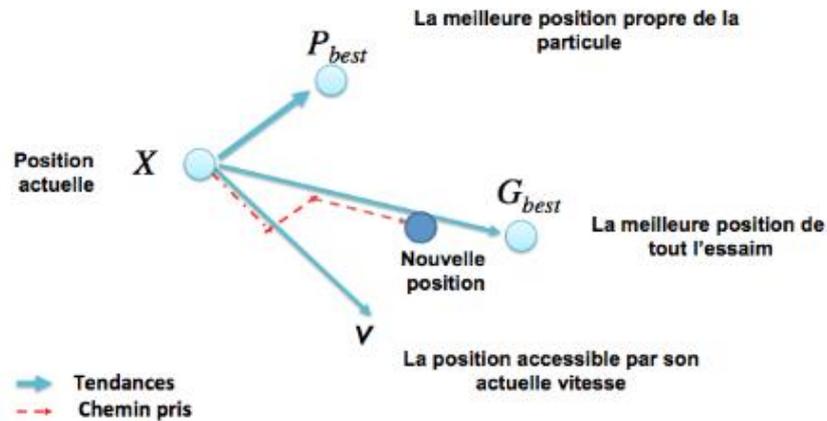


Figure 12 Déplacement d'une particule. [12]

2.4.3.1.2. Les Etapes de PSO :

01. Définir :

- Population initiale (nombre de particules N) .
- Définir la zone de recherche $[x_{lb}, x_{ub}]$
- Position initiale (x) et vitesse initiale (v)
- Affecter $P_{best i}$ et g_{best} (basé sur la fonction objectif)

02. Mettre à jour :

La vitesse et la position de chaque particule sont mises à jour :

La vitesse : $v_i(t) = \theta v_i(t - 1) + c_1 r_1 (p_{best i} - x_i(t - 1)) + c_2 r_2 (g_{best i} - x_i(t - 1))$

$v_i(t)$: La vitesse d'un élément i dans le tour (itération) t .

$v_i(t - 1)$ La vitesse de l'élément x dans l'étape précédente.

θ ' inertie Wight ' est une constante dans l'intervalle $[0.4, 0.9]$.

c_1 et c_2 Cognitive and social paramètres, nombres aléatoires.

r_1 et r_2 aussi nombres aléatoires dans l'intervalle $[0, 1]$.

La position : $x_i(t) = x_i(t - 1) + v_i(t)$

$x_i(t)$ La position d'un élément i dans le tour (itération) t .

$x_i(t - 1)$ La position de l'élément x dans l'étape précédente.

Après avoir calculé les nouvelles positions, Nous vérifions son appartenance à l'espace de recherche

if $x_i(t) > x_{ub}$ $x_i(t) = x_{ub}$

if $x_i(t) < x_{lb}$ $x_i(t) = x_{lb}$

03. Calcul de la fonction objectif (ou fitness value)

- Calcul f_{x_i} .
- Mettre à jour les valeurs $P_{best\ i}$ et g_{best} .

$$\left\{ \begin{array}{l} \text{if } (f_{x_i} > f_{P_{best\ i}}) \ P_{best\ i} = x_i \\ \text{if } (f_{P_{best\ i}} > f_{g_{best}}) \ g_{best} = P_{best\ i} \end{array} \right.$$

2.4.3.1.3. Exemple illustratif montrant le mécanisme d'action d'un Algorithmes PSO :

Le problème : Trouver la plus grande valeur de la fonction dans l'intervalle $[-4, 4]$

$$f(x) = -x^2 + 2x + 11$$

La solution : Application de l'algorithme PSO

- La zone de recherche $[-4, 4]$
- La fonction objectif : $f(x)$
- Supposons qu'il y ait 4 particules :

$$N=4, \ v_1 = v_2 = v_3 = v_4 = 0, \ r_1 = 0.32, \ r_2 = 0.95, \ c_1 = c_2 = \theta = 1$$

01. Quatre positions a hasard :

$$x_1 = -1,5$$

$$x_2 = 0$$

$$x_3 = 0,5$$

$$x_4 = 1,25$$

02. Calcul f_{x_i} :

$$f(x_1) = 5,75$$

$$f(x_2) = 11$$

$$f(x_3) = 11,75$$

$$f(x_4) = 11,93$$

Itération 01

$P_{best 1} = -1,5$
 $P_{best 2} = 0$
 $P_{best 3} = 0,5$
 $P_{best 4} = 1,25$
 $g_{best} = 1,25$
 Parce que la valeur $f(x_4)$ est la plus grande
 vitesses
 $v_1(1) = 2,62$
 $v_2(1) = 1,19$
 $v_3(1) = 0,71$
 $v_4(1) = 0$
 Nouveaux positions
 $x_1 = 1,12$
 $x_2 = 1,19$
 $x_3 = 1,21$
 $x_4 = 1,25$
 $x_i \in [-4, 4]$
 Calcul f_{x_i}
 $f(x_1) = 11,98$
 $f(x_2) = 11,96$
 $f(x_3) = 11,95$
 $f(x_4) = 11,93$

Itération 02

$P_{best 1} = 1,12$
 $P_{best 2} = 1,19$
 $P_{best 3} = 1,21$
 $P_{best 4} = 1,25$
 $g_{best} = 1,12$
 Parce que la valeur $f(x_4)$ est la plus grande
 vitesses
 $v_1(1) = 2,62$
 $v_2(1) = 1,15$
 $v_3(1) = 0,67$
 $v_4(1) = -0,0613$
 Nouveaux positions
 $x_1 = 3,74$
 $x_2 = 2,35$
 $x_3 = 1,88$
 $x_4 = 1,18$
 $x_i \in [-4, 4]$
 Calcul f_{x_i}
 $f(x_1) = 4,44$
 $f(x_2) = 10,17$
 $f(x_3) = 11,21$
 $f(x_4) = 11,96$
 Meilleure solution $g_{best}=1,12$
 $f(g_{best}) = 11,98$

Figure 13 Résultats de l'algorithme sur deux itération .

2.4.3.2. L'optimisation par colonies de fourmis (Ant colony optimization)

2.4.3.2.1. Définition :

- Une technique de recherche utilisée en informatique pour trouver des solutions presque optimales à des problèmes d'optimisation discrets.
- ACO est une intelligence d'essaim inspirée de la façon dont les fourmis communiquent indirectement des directions les unes aux autres.
- L'aspect le plus intéressant du comportement collaboratif des espèces de fourmis est leur capacité à trouver les chemins les plus courts entre le nid des fourmis et les sources de nourriture.

Mots clés de ACO

- Fourmis k : Toute solution possible.
- Population N -Groupe de toutes les fourmis.
- Espace de recherche $[lb, ub]$ - Toutes les solutions possibles au problème.
- L'espace de recherche est divisé par taille d'étape h .
- Piste de phéromones π .
- Paramètre de mise à l'échelle \mathcal{Z} .
- Taux d'évaporation ρ .

2.4.3.2.2 Exemple illustratif montrant le mécanisme d'action d'un Algorithmes colonies de fourmis

Pour donner un exemple mathématique concret de l'optimisation par colonies de fourmis (ACO), nous pouvons prendre le problème classique du voyageur de commerce (Traveling Salesman Problème, TSP) avec un petit ensemble de villes.

Supposons que nous ayons les villes suivantes avec leurs coordonnées (x, y) :

- Ville 1 : (0, 0)
- Ville 2 : (1, 2)
- Ville3 : (3, 1)
- Ville 4 : (4, 3)
- Ville 5 : (2, 4)

L'objet est de trouver le chemin le plus court qui visite chaque ville exactement une fois et revient à la ville de part.

Etapas de l'Aco :

01. Modélisation du problème : Nous représentons les villes et les distances entre elles sous forme de graphe. Chaque ville est un nœud du graphe, et les distances entre les villes sont les arêtes. Par exemple, la distance entre la Ville 1 et la Ville 2 est donnée par la distance euclidienne entre les coordonnées (0, 0) et (1, 2).

02. Initialisation : Nous initialisons une colonie de fourmis dans une ville de départ aléatoire, par exemple, la Ville 1.

03. Construction de solutions : Chaque fourmi se déplace d'une ville à une autre en suivant certaines règles. Par exemple, une fourmi peut choisir la ville suivante en utilisant la règle de sélection proportionnelle à la quantité de phéromone sur l'arête et inversement proportionnelle à la distance entre les villes. Les fourmis explorent le graphe en suivant ces règles et déposent de la phéromone sur les arêtes parcourues.

04. Mise à jour des phéromones : Une fois que toutes les fourmis ont terminé leur parcours, les niveaux de phéromone sur les arêtes sont mis à jour en fonction de la qualité des solutions trouvées. Par exemple, les arêtes appartenant aux chemins les plus courts reçoivent une plus grande quantité de phéromone.

05. Évaporation des phéromones : Pour éviter la stagnation, les niveaux de phéromone sur toutes les arêtes sont réduits progressivement pour simuler l'évaporation naturelle de la phéromone.

06. Répétition des étapes 3 à 5 : Les étapes 3 à 5 sont répétées pour un certain nombre d'itérations afin de permettre aux fourmis d'explorer le graphe et de trouver des solutions de meilleure qualité.

Au fur et à mesure des itérations, les niveaux de phéromone convergeront vers les chemins les plus courts. Le chemin final trouvé par les fourmis représentera la solution optimale ou une approximation de la solution optimale pour le problème du voyageur de commerce.

Veillez noter que dans cet exemple simplifié, les détails tels que les paramètres spécifiques de l'ACO, les formules de mise à jour des phéromones, et les mécanismes de sélection des villes suivantes peuvent varier en fonction de la mise en œuvre spécifique de l'ACO pour résoudre le TSP.

3.5. Conclusion

L'intelligence artificielle ne se limite plus à une simple simulation de l'intelligence humaine. Cela va au-delà de l'imitation de l'intelligence des animaux et de la nature ... Cette nouvelle approche, connue sous le nom de bio-inspiration, vise à développer des algorithmes et des systèmes inspirés de la nature pour résoudre des problèmes complexes. Les exemples incluent les algorithmes génétiques QUI imitent le processus de sélection naturelle, ainsi que l'optimisation de l'essaim de particules PSO.

Chapitre 03

Application l'optimisation par essaim criquets pour Template Matching

3.1. Introduction

Dans le cadre de cette recherche, nous nous intéressons à l'application de l'algorithme LSO en conjonction avec la NCC dans le processus de Template matching. L'objectif principal est d'utiliser les caractéristiques et les comportements des Criquets dans la nature pour optimiser la recherche des correspondances entre le modèle et l'image.

Dans ce chapitre, nous présenterons la méthodologie détaillée de l'application de l'algorithme LSO avec la NCC dans le Template matching. Nous aborderons les différentes étapes du processus, y compris la génération de la population initiale de sauterelles, l'évaluation de la fitness des individus, la mise à jour des positions, ainsi que la convergence vers les meilleures correspondances

3.2. Processus de Template matching

Le TM recherche le motif entré dans l'image dans tous les pixels de l'image. Pour chaque transition, un critère de similarité est calculé (NCC SSD ...) . La technique de TM repose sur deux piliers principaux :

3.2.1. La stratégie de recherche

La stratégie de recherche... est la réponse à la question suivante : Où recherche-t-on l'image ?, La plus courante consiste à rechercher toutes les parties de l'image, mais cette approche est très coûteuse en calcul.

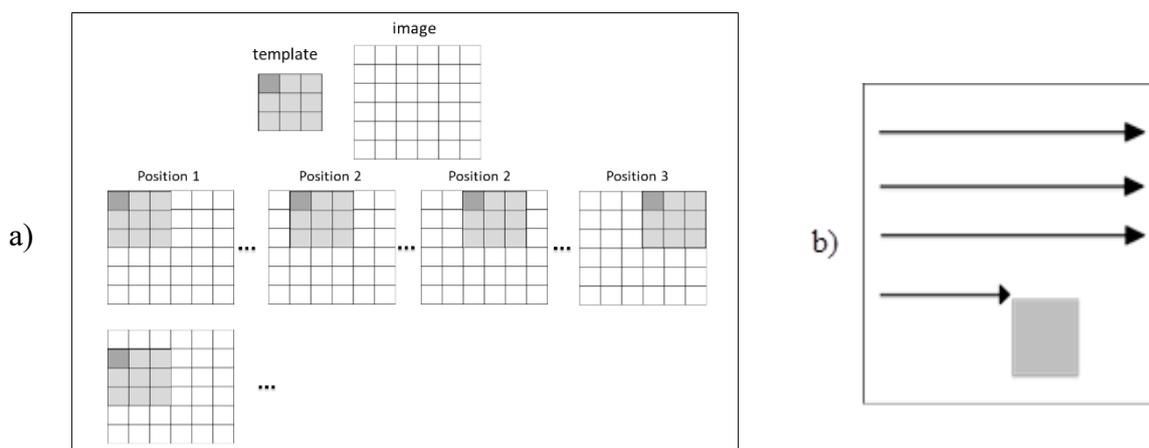


Figure 14 Il représente le processus de numérisation des images

3.2.2. Le critère de similitude

Il existe de nombreuses mesures de similarité entre deux images en TM telles que ssd sad ncc , on sait que le calcul d'une mesure de similarité nécessite un coût de calcul élevé et donc prend plus de temps dans le processus de TM, De plus, malgré cela, Les métriques permettent une mesure appropriée de la similarité entre une paire d'images donnée.

Dans notre application, nous avons utilisé l'échelle NCC (normalized cross correlation) qui est donnée par la relation suivante :

Soit une image source I de taille $N \times M$ et un modèle d'image R de taille $m \times n$

$$NCC = \frac{\sum_{i=1}^m \sum_{j=1}^n [I(u+i, v+j) - \bar{I}(u, v)] \cdot [R(i, j) - \bar{R}]}{\sqrt{\left[\sum_{i=1}^m \sum_{j=1}^n [I(u+i, v+j) - \bar{I}(u, v)]^2 \cdot \left[\sum_{i=1}^m \sum_{j=1}^n [R(i, j) - \bar{R}]^2 \right]}} \quad (1)$$

Voici les composants impliqués dans l'équation :

$I(u, v)$: représente la valeur d'intensité de l'échelle de gris à un emplacement de pixel spécifique (u, v) dans l'image source I.

$R(i, j)$: représente la valeur d'intensité de l'échelle de gris à un emplacement de pixel spécifique (i, j) dans le modèle d'image R.

$\bar{I}(u, v)$: désigne l'intensité moyenne de l'échelle de gris de l'image source I dans la région coïncidente du modèle d'image R. Elle est calculée comme la valeur moyenne des intensités dans cette région.

\bar{R} : Indique l'intensité moyenne de l'échelle de gris du modèle d'image R. Elle est calculée comme la valeur moyenne des intensités dans l'image modèle.

$$\bar{I}(u, v) = \frac{1}{m \cdot n} \sum_{i=1}^m \sum_{j=1}^n I(u + i, v + j) , \quad \bar{R} = \frac{1}{m \cdot n} \sum_{i=1}^m \sum_{j=1}^n R(i, j) \quad (2) \quad (3)$$

$\Sigma [i=1 \text{ à } m] \Sigma [j=1 \text{ à } n]$: Représente la sommation sur les pixels dans les dimensions du modèle d'image

La valeur de la NCC obtenue se situe dans la plage $[-1, 1]$. Une valeur de $NCC = 1$ indique la meilleure similarité possible entre le modèle d'image R et sa sous-image correspondante dans I , tandis qu'une valeur de $NCC = -1$ implique une dis similarité complète entre les images. []

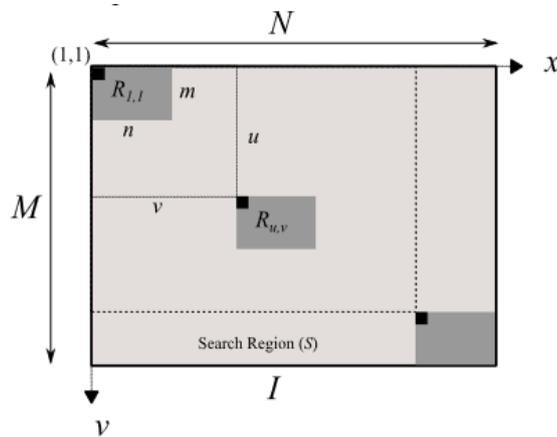


Figure 15 Processus de Template matching.

La valeur de la NCC obtenue se situe dans la plage $[-1, 1]$. Une valeur de $NCC = 1$ indique la meilleure similarité possible entre le modèle d'image R et sa sous-image correspondante dans I , tandis qu'une valeur de $NCC = -1$ implique une dis similarité complète entre les images

3.3. Comment ça fonctionne TM-NCC

Après avoir effectué une recherche en utilisant la technique de mise en correspondance de modèle (Template Matching) avec la mesure de corrélation croisée normalisée (NCC), on calcule la valeur de NCC pour chaque déplacement possible du modèle à l'intérieur de l'image source.

Une fois la recherche terminée, une matrice de résultats, appelée matrice de corrélation croisée, est générée ($[H-h+1, W-w+1]$ où (H, W) est la taille de l'image et (h, w) est la taille du modèle). Cette matrice contient les valeurs de NCC pour chaque déplacement potentiel du modèle à l'intérieur de l'image source, Les coordonnées de la plus grande valeur sont l'emplacement de la correspondance dans l'image.

IMAGE (13, 11)

```

[[255 254 253 253 255 254 254 255 254 254 255]
[255 254 254 254 255 254 254 255 254 254 255]
[254 254 255 255 255 255 255 255 255 255 255]
[254 254 254 254 254 254 254 255 254 255 254]
[254 254 254 254 254 254 254 255 254 255 254]
[254 254 254 254 254 254 254 255 255 255 254]
[255 255 255 255 255 254 254 254 255 255 254]
[255 254 254 255 255 254 254 254 254 255 254]
[254 254 252 253 255 254 254 254 255 255 254]
[253 252 253 253 254 254 250 253 254 255 255]
[245 201 201 200 249 254 255 255 254 254 253]
[246 202 204 202 251 253 255 254 254 255 253]
[251 249 253 251 253 253 253 255 254 254 255]]
    
```

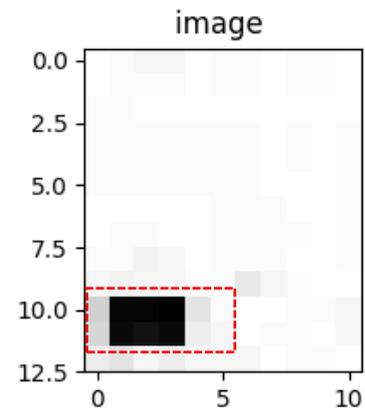


Figure 16 Représentation numérique d'image

TEMPLATE(2, 5)

```

[[242 200 202 201 247]
[243 202 204 203 248]]
    
```

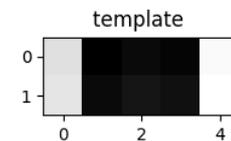


Figure 17 Représentation numérique du modèle (Template) d'image.

NCC(12, 7)

```

[[0.9956202 0.9953775 0.995309 0.9953692 0.9953969 0.9953178 0.9954112 ]
[0.99546075 0.99530727 0.9953516 0.995397 0.9953908 0.9953516 0.9953978 ]
[0.99536186 0.9953273 0.995367 0.9954194 0.9953391 0.9953925 0.9953087]
[0.995377 0.995377 0.995377 0.9954768 0.99531424 0.9954183 0.9952487]
[0.995377 0.995377 0.995377 0.9954768 0.99536544 0.99538934 0.9952216]
[0.99538225 0.99533087 0.9953601 0.995436 0.99548614 0.995384 0.9952216]
[0.99543345 0.99526244 0.995293 0.995393 0.99550825 0.995444 0.9952833]
[0.99556273 0.9953672 0.9952242 0.99530 0.99551004 0.9954476 0.9952868 ]
[0.99559015 0.99541026 0.9950667 0.9953109 0.99559146 0.9955808 0.9951805]
[0.99492586 0.99139535 0.991028 0.9911733 0.99523413 0.9955217 0.9952088]
[0.9999753 0.9918811 0.989111 0.987371 0.99495834 0.99528706 0.9952915]
[0.9960268 0.99208844 0.9920138 0.991414 0.99523485 0.995308 0.9953093]]
    
```

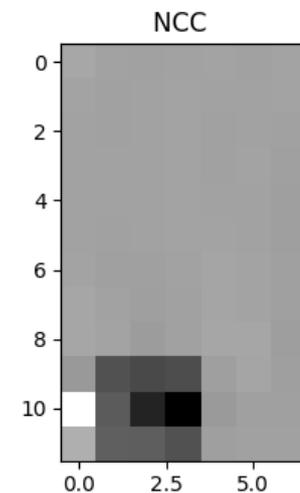


Figure 18 Résultat NCC entre l'image et template.

(10,0) est la valeur la plus élevée dans la matrice NCC, et c'est exactement l'emplacement de la similarité dans l'image. Nous montrons l'emplacement de la similarité en traçant un rectangle autour de la zone de similarité : [10, 0, w, h]. 10 et 0 représentent les coordonnées du coin supérieur gauche du rectangle, et w et h correspondent à la largeur et à la hauteur du rectangle respectivement.

La méthode TM-NCC donne de bons résultats, mais elle est très coûteuse au niveau de la stratégie de recherche (recherche dans toutes les images) et des calculs du NCC à chaque déplacement. Pour traiter ce problème, nous avons appliqué l'algorithme LS pour réduire le nombre des sites de recherche et ainsi réduire le nombre d'opérations de la NCC.

3.4. L'algorithme Essaim de Criquets (LS)

- LS est une approche d'optimisation inspirée du comportement des criquets du désert.
- Les criquets passent par deux phases opposées : solitaire et sociale.
- Dans la phase solitaire, ils évitent le contact avec d'autres criquets et explorent des sources de nourriture prometteuses.
- Dans la phase sociale, ils s'agrègent autour de sources de nourriture abondantes et les dévastent.
- L'agrégation se fait en se basant sur les meilleures sources de nourriture.
- LS combine ces comportements pour maintenir une recherche globale et locale puissante.
- Il est efficace pour résoudre divers problèmes d'optimisation complexes tels que le traitement d'images, l'estimation de paramètres de systèmes chaotiques, et la reconnaissance de motifs.

3.4.1. La phase solitaire

Dans la phase solitaire, la position de chaque criquet est modifiée en fonction de son interaction sociale avec les autres criquets du groupe. L'équation (4) représente la mise à jour de la position i du criquet du pas de temps k à $k+1$

$$x_i^{k+1} = x_i^k + \Delta x_i \quad (4)$$

Pour calculer le changement de position (Δx_i) subi par le i le criquet suite à son interaction sociale, les forces sociales exercées par les autres criquets doivent être prises en compte. Ces forces sont basées sur les principes d'attraction et de répulsion.

L'équation (5) représente le modèle de la force des forces sociales entre deux criquets en fonction de leur distance (r), de la force d'attraction (F) et de l'échelle de longueur attrayante typique (L).

$$s(r_{ij}^k) = F e^{-r_{ij}^k/L} - e^{-r_{ij}^k} \quad , \quad r_{ij}^k = \|x_i^k - x_j^k\| \quad (5)$$

Dans l'équation (6), (r_{ij}) représente la force sociale exercée par le criquet j sur le criquet i . Il est calculé à l'aide de la fonction (r) définie dans l'équation (2). r_{ij} est la distance entre les positions des criquets i et j , et d_{ij} est le vecteur unitaire pointant de i à j .

$$s_{ij}^k = \rho(x_i^k, x_j^k) s(r_{ij}^k) d_{ij} + rand(-1,1) \quad , \quad d_{ij} = (x_j^k - x_i^k) / r_{ij}^k \quad (6)$$

$\rho(x_i^k, x_j^k)$ est la fonction de dominance calculée entre la position de l'élément i et j . Elle est donnée selon la relation suivante :

$$\rho(x_i^k, x_j^k) = \begin{cases} e^{-rank(x_i^k)/N} \\ e^{-rank(x_j^k)/N} \end{cases} \quad (7)$$

L'équation (8) indique que la force sociale totale sur chaque criquet (S_i) peut être modélisée comme la somme des interactions par paires entre ce criquet et tous les autres criquets du groupe. La sommation est effectuée sur tous les criquets sauf le criquet

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^N s_{ij} \quad (8)$$

Enfin on peut calculer la nouvelle position de l'individu

$$\Delta x_i = S_i \quad (9)$$

3.4.2 La phase sociale.

La phase sociale de LS-II comprend les étapes suivantes :

1. Sélection d'un sous-ensemble de solutions : À partir de l'ensemble total de solutions $Lk = \{lk1, lk2, \dots, lkN\}$, on sélectionne un sous-ensemble $Bk = \{bk1, bk2, \dots, bkq\}$. Bk contient les q meilleures solutions de Lk ($q \leq N$).
2. Calcul de la probabilité de sélection : Chaque individu lki dans Lk se voit attribuer une probabilité d'être attiré vers une solution sélectionnée au hasard, bkj . La probabilité, notée (li, bj) , est déterminée par la qualité de la solution bkj et la distance entre lki et bkj . Elle est calculée à l'aide de l'équation (10).

$$P_{li bj}^k = \frac{A(b_j^k) e^{-\|l_i^k - l_j^k\|}}{\sum_{n=1}^q A(b_n^k) e^{-\|l_i^k - l_n^k\|}} \quad (10)$$

3. Calcul de l'attractivité : L'attractivité de chaque solution bkj , notée (bj) , est déterminée en fonction de sa valeur de fitness $(f(bj))$ et de la plage de fitness parmi l'ensemble des meilleures solutions Bk . Elle est calculée à l'aide de l'équation (11).

$$A(b_j^k) = \frac{f(b_j^k) - f_{worst}(B^k)}{f_{best}(B^k) - f_{worst}(B^k) + \varepsilon} \quad (11)$$

4. Mise à jour des positions : Chaque individu lki de la population de l'essaim Lk met à jour sa position en fonction de l'équation (12). La position est mise à jour en se déplaçant dans la direction d'une solution meilleure sélectionnée au hasard, bkr , où r est un entier aléatoire entre 1 et q . La mise à jour implique une somme pondérée de la différence entre la position actuelle lki et la meilleure solution sélectionnée bkr , multipliée par un facteur aléatoire $rand$.

$$I_i^{k+1} = I_i^k + 2(b_r^k - I_i^k)rand \quad (12)$$

Ces étapes définissent collectivement le comportement de la phase sociale de LS-II, où les individus sont attirés vers des solutions sélectionnées en fonction de leur qualité et de leur distance, et leurs positions sont mises à jour en conséquence.

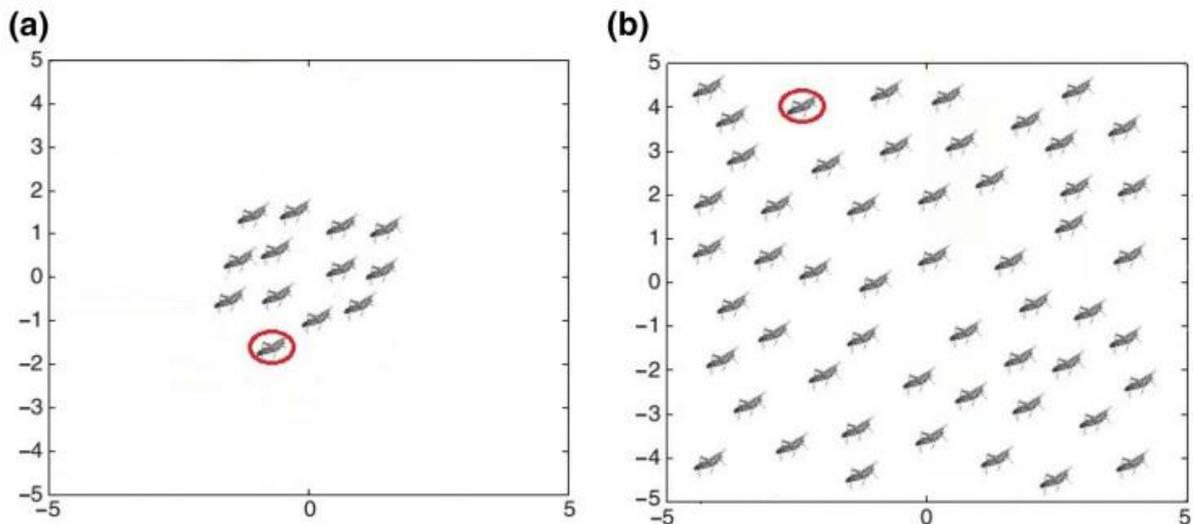


Figure 19 Les différentes distributions de la phase solitaire LS.

3.4.3 Choisir entre les phases solitaire et sociale

Dans LS, une sélection dynamique entre les phases solitaires et sociales est effectuée à chaque itération. Au début du processus de recherche, les individus ont tendance à se comporter de manière solitaire, favorisant ainsi une exploration intensive de l'espace de recherche. Cependant, à mesure que l'évolution progresse, l'exploitation est encouragée par le comportement social. La probabilité qu'un individu se comporte socialement augmente progressivement, tandis que la probabilité de comportement solitaire diminue.

L'équation 13 définit la phase comportementale choisie à chaque itération. Si un nombre aléatoire "rand" est inférieur ou égal à la probabilité de comportement " p^k ", la phase solitaire est sélectionnée. Sinon, la phase sociale est choisie.

$$P(L^k) = \begin{cases} \text{Solitary} & \text{if } rand \leq p^k \\ \text{Social} & \text{if } rand > p^k \end{cases} \quad (13)$$

L'équation 14 calcule la probabilité de comportement p^k à partir de l'itération actuelle "k" et du nombre total d'itérations. Cette probabilité diminue linéairement à mesure que le nombre d'itérations augmente.

$$p^k = 1 - \frac{k}{item} \quad (14)$$

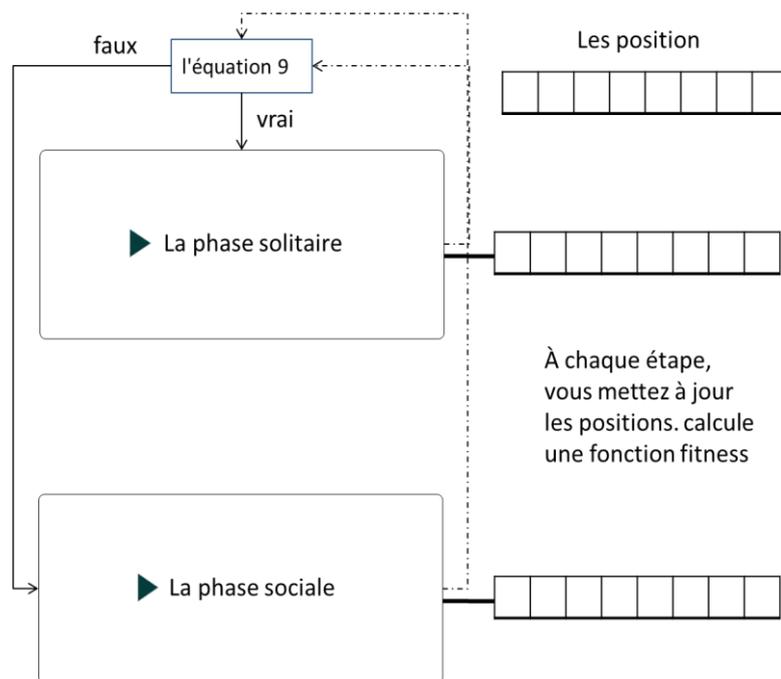


Figure 20 Choisir entre les phases solitaire et sociale

LS utilise une sélection dynamique entre les phases solitaires et sociales, favorisant l'exploration au début et l'exploitation sociale à mesure que l'évolution progresse. Les équations 1 et 2 définissent la règle de sélection et la probabilité de comportement pour guider le choix de la phase à chaque itération.

3.5. Adaptation de l'algorithme LS avec TM-NCC

Afin d'appliquer l'algorithme d'essaim de criquets, le problème doit être défini et décrit avec précision. Cette étape est importante dans tous les algorithmes d'optimisation. Nous allons nous concentrer sur les quatre choses suivantes :

1. **Problème** : Trouver la position (x, y) dans l'image qui a la plus grande valeur NCC entre le modèle et une région dans l'image .
2. **La fonction d'évaluation** est la suivante :
 $(width, height)$ est le size de Template .

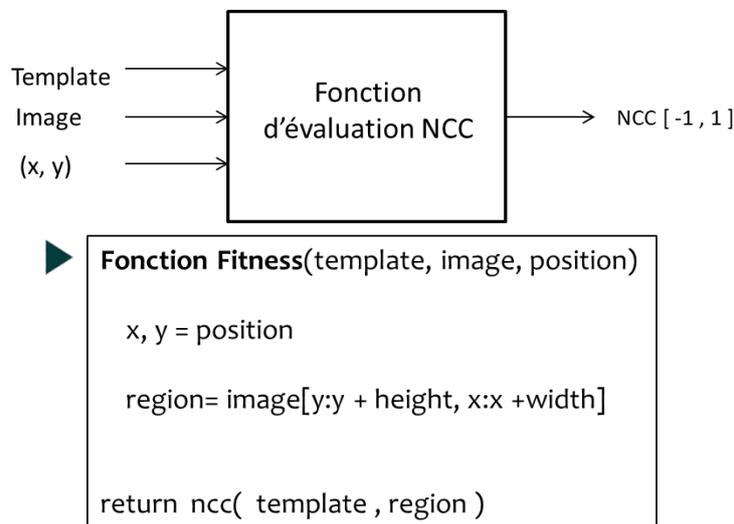


Figure 21 Fonction d'évaluation NCC

3. **Les variables de solution** : les positions (x, y) dans l'image .
4. **Domaine de recherche** : la taille de l'image.

L'explication :

- Nous coupons une partie de l'image à une position aléatoire x, y
- dimensions de cette région sont les mêmes que les dimensions de Template.
- À chaque tour, nous stockons les valeurs et les positions de la NCC entre région et Template et déterminons la meilleure valeur.
- À la fin du nombre de tours, nous obtenons le best qui est la valeur NCC la plus élevée.

- Nous dessinons un rectangle selon le meilleure coordonnées pour indiquer la position de similitude.
- La phase solitaire permet à l'algorithme LS de détecter l'image, (exploration)
- La phase sociale permet de se concentrer sur les meilleures solutions, (exploitation)

De manière générale, nous recherchons une position (x,y) qui a le plus grand NCC .

Après avoir décrit les choses principales, nous préparons l'algorithme LS pour fonctionner.

Exemple : Nous avons l'image et le Template suivants.

1. Convertir les images en niveaux de gris : Les images sont converties en niveaux de gris pour simplifier le traitement.

		Image				
		70	70	75	1	1
		70	70	1	0	1
		1	1	0	1	0
		1	1	10	0	1
		1	1	1	1	1

		Template	
		70	70
		70	70

Figure 22 Matrice d'image et Template.

2. Fixer les paramètres : Nous supposons que nous avons N=3 individus (criquets) et une seule itération.
3. Générer des positions aléatoires pour chaque individu : Pour chaque individu, nous générons des positions aléatoires dans l'image. Ces positions sont représentées par les coordonnées (x, y).

	x	y
l_1	3	2
l_2	0	0
l_3	1	3

Figure 23 Les positions aléatoires (x ,y) .

4. Entrer dans la phase solitaire de l'algorithme LS : Dans cette phase, nous calculons la valeur de la fonction d'évaluation pour chaque individu. La fonction d'évaluation mesure la qualité de la solution représentée par les positions (x, y) de l'individu :

	p_i
l_1	0,1
l_2	1
l_3	-0,5

Figure 24 Les valeurs de la fonction d'évaluation dans la phase solitaire.

5. Sélectionner la meilleure solution : Parmi les N individus, nous sélectionnons la meilleure solution, c'est-à-dire celle qui a la valeur de la fonction d'évaluation la plus élevée. Le meilleur NCC est 1 pour le position $I_2 (0, 0)$ Ce sont les résultats de la phase solitaire.

70	70	75	1	1
70	70	1	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Figure 25 Représentation des positions dans la phase solitaire.

6. Nous entrons dans la phase sociale et ajustons-les Positions (x, y) pour chaque individu

	x	y
l_1	0	0
l_2	0	0
l_3	0	0

Figure 26 Les nouveau positions (x ,y) .

7. Calculer la fonction de fitness

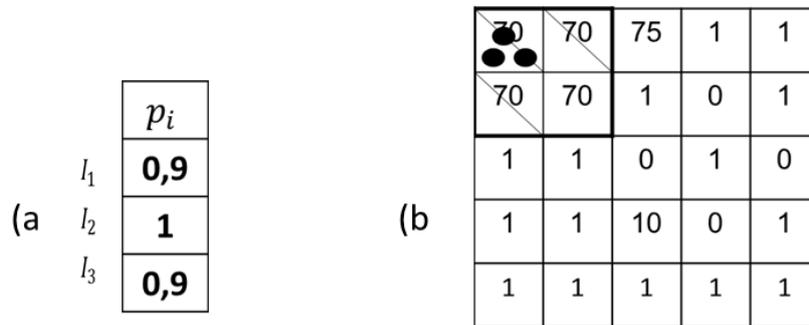


Figure 27 a) Les valeurs de la fonction d'évaluation dans la phase social
 b) Représentation des positions dans la phase sociale

Il convient de noter qu'au stade individuel, la distribution a été faite pour savoir où se trouve la nourriture ou les meilleures zones de solution. , et dans la phase sociale, l'accent était mis sur la meilleure solution.

Ceci est considéré comme un tour et nous devons le répéter si la taille de l'image est grande.

3.6. Résumé

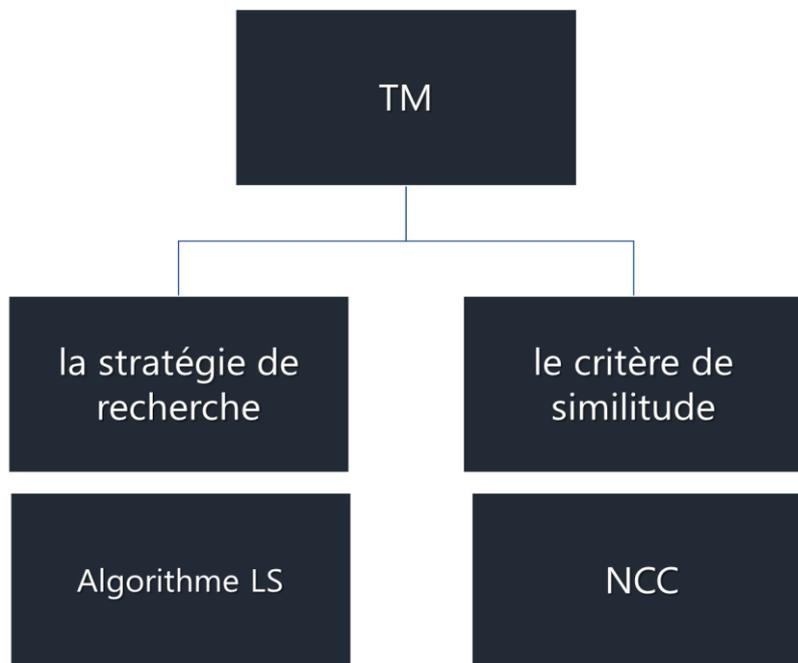


Figure 28 .Un plan est un résumé de ce chapitre

3.7. Conclusion

Dans ce chapitre, nous avons expliqué comment implémenter l'algorithme LS avec la technique d'appariement de modèles, où nous avons revu la TM traditionnelle seule d'une part, et d'autre part nous avons revu l'algorithme LS avec ses étapes. Après cela, nous avons mentionné la méthode de travail, en donnant un exemple simple pour clarifier

Chapitre 04 Application et Résultat

4.1. Introduction

Le chapitre 4 a été consacré à la mise en œuvre et à l'évaluation de TM-NCC (Template Matching with Normalized Cross-Correlation) en utilisant l'algorithme LS , tout en maintenant un contexte de travail approprié. Nous avons examiné les outils logiciels et le code source nécessaires pour cette implémentation, suivi de l'implémentation pratique et de l'évaluation des résultats obtenus.

Enfin, à la fin de ce chapitre, nous avons présenté le résultat global de notre travail, en soulignant les points forts et les limites de l'approche TM-NCC avec l'algorithme LS .

4.2. Outils et technologies de programmation

Dans le cadre de notre implémentation de TM-NCC avec l'algorithme LS, nous avons utilisé le langage de programmation Python et la bibliothèque OpenCV (cv2) dans l'environnement de développement PyCharm.

4.2.1 Python

Python est un langage de programmation polyvalent et populaire, connu pour sa simplicité et sa lisibilité. Il offre une large gamme de bibliothèques et de Framework qui facilitent le développement d'applications, y compris le traitement d'images et la vision par ordinateur. Python est largement utilisé dans le domaine de l'apprentissage automatique (machine Learning) et de l'analyse de données, ce qui en fait un choix naturel pour notre implémentation.



Figure 29 Logo python

4.2.2 OpenCV (cv2)

OpenCV (Open Source Computer Vision Library) est une bibliothèque open source largement utilisée dans le domaine de la vision par ordinateur et du traitement d'images. Elle fournit des fonctions et des outils puissants pour la manipulation, l'analyse et la compréhension

des images et des flux vidéo. La bibliothèque OpenCV est écrite en C++ mais dispose également d'une interface Python (cv2) qui facilite l'utilisation des fonctionnalités d'OpenCV dans des applications Python. Nous avons utilisé cv2 pour effectuer des opérations de traitement d'images, telles que la lecture et l'affichage des images, la conversion des couleurs, et l'application de la corrélation croisée normalisée (NCC) dans notre implémentation de TM-NCC.



Figure 30 Figure 29. Logo OpenCV

4.2.3 PyCharm

PyCharm est un environnement de développement intégré (IDE) spécialement conçu pour le langage de programmation Python. Il offre une multitude de fonctionnalités telles que l'auto complétion, le débogage, la gestion des packages, et la facilité d'intégration avec des outils de contrôle de version. PyCharm facilite le développement, le débogage et la gestion de projets Python, ce qui en fait un choix pratique pour notre travail d'implémentation de TM-NCC avec l'algorithme LS.

4.3. Structure et organisation du code

1. Importation des bibliothèques :

- a) Importer la bibliothèque OpenCV (cv2) : `import cv2 .`
- b) Importer la bibliothèque NumPy : `import numpy as np .`

```
import numpy as np
import cv2
```

Figure 31 Importation des bibliothèques

2. Chargement de l'image et du modèle :

```
template = cv2.imread(r'C:\Users\A\Desktop\ooo.jpg', 0)
# 0 signifie lire l'image en gris
image = cv2.imread(r'C:\Users\A\Desktop\robot_image.jpg', 0)
height,width = template.shape
H,W = image.shape
```

Figure 32 Chargement de l'image sur python.



Figure 33 Modèle d'image et image source, utilisés dans cette algorithm.

3. Définition de la fonction d'évaluation pour une utilisation dans les phases LS :

Couper une partie de l'image à la coordonnée (x, y) avec la même taille que le modèle.

Calculer le NCC entre la partie image et le modèle.

```
def ncc(template, image, position):
    x, y = position
    region = image[y:y + height, x:x + width]
    region = cv2.resize(region, (width, height))
    template_norm = (template - np.mean(template)) / np.std(template)

    image_norm = (region - np.mean(region)) / np.std(region)
    ncc = np.mean(template_norm * image_norm)

    return ncc
```

Figure 34 Code fonction d'évaluation .

4. Paramètres de l'algorithme LS :

```
Esc = 0.1
N = 30
F = 22
L = 9
n=2
```

Figure 35 Paramètres de l'algorithme LS .

F : la force d'attraction. (La phase solitaire eq 2) .

L : l'échelle de longueur attrayante typique. . (La phase solitaire eq 2) .

N : population (nombre de ' locust ' = nombre de position) .

n : demension (x,y) , Chaque particule a des coordonnées x,y .

Esc : est un nombre aléatoire . (La phase sociale eq 8).

5. Générer aléatoirement la position de particules et les placer dans la variable xn :

```
xn = np.random.randint((0, 0), (W- width,H-height), size=(N, 2))
pzn = np.empty(N)
```

Figure 36 Générer des positions aléatoires.

6. Boucle de génération (itérations) et Choix entre les phases (Phase solitaire ou Phase sociale):

```
iteration = 100
for gen in range(iteration):
    pk = 1 - (gen / 100) # Choisir entre les phases eq 14

    if np.random.rand() <= pk: # eq 13
        # La phase solitaire
        .....
    else
        # La phase social
        .....
```

Figure 34. Choix Entre les phases.

7. Calculer les valeurs de la fonction d'évaluation et les stocker dans le tableau pzn .

La phase solitaire

```

for h in range(N):
# calcul la fonction de fitness pour chaque particule (x,y)
    pzn[h] = ncc(template, image,xn[h])
# trier les éléments du tableau pzn par ordre
# croissant et renvoyer les indices correspondants
    index = np.argsort(pzn)[::-1]

```

```

for p in range(N):
for q in range(N):
    if p != q:
        # fonction de dominance eq 7 La phase solitaire
        if index[p] <= index[q]:
            dom = np.exp(-index[p] / N)
        else:
            dom = np.exp(-index[q] / N)
        ip = xn[p] # position x,y de particule p
        iq = xn[q] # position x,y de particule q
        # distance entre ip iq , eq 5
        rpq = np.linalg.norm(np.array(ip) - np.array(iq))
        # la force sociale exercée par ip sur iq , eq 5
        spq = F * np.exp(-rpq / L) - np.exp(-rpq)
        # eq 6
        dpq = (iq - ip) / (rpq + np.finfo(float).eps)
        # eq 3 la force
        va = dpq * dom * spq + Esc * (2 * np.random.rand(2) - 1)
        # Stocker les forces appliquées entre chaque particule
        vs[p, q, :] = va

```

```

for u in range(N):
    for dim in range(n):
# Mettre à jour les positions des particules eq 1 eq 4
        pts[u, dim] = xn[u, dim] + np.sum(vs[u, :, dim])

```

Figure 37 Algorithme de La phase solitaire

La fin de La phase solitaire. Nous renverrons la condition de selection .

```

if np.random.rand() <= pk:
# La phase solitaire
    .....
else
# La phase social

```

La phase sociale

```
for h in range(N):
    pzn[h] = ncc(template, image, xn[h])
index = np.argsort(pzn)[::-1]
```

```
if gen == 0:
    best_value = pzn[index[0]]
    worst_value = pzn[index[-1]]
else:
    if best_value < pzn[index[0]]:
        best_value = pzn[index[0]]
    if worst_value > pzn[index[-1]]:
        worst_value = pzn[index[-1]]
```

#Choisissez la valeur maximale et minimale de la fonction fitness

```
for i in range(N):
    for b in range(N // 2):
        ii = xn[i]
        ib = xn[index[b]]
        # distance
        dis = np.linalg.norm(ii - ib)
        # eq 7 la phase social
        tem = (pzn[index[b]] - worst_value) / (best_value - worst_value +
        np.finfo(float).eps)
        dss = tem * np.exp(-dis) # Calcul la probability
    result = dss // np.sum(dss)
    M = np.max(result)
    I = np.argmax(result)
    # eq 8 de la phase social
    Xn[i, :] = xn[i, :] + 0.8 * (xn[index[I], :] - xn[i, :]) + Ecs * (2 * np.random.random() - 1)

Xn[index[0]] = xn[index[0]] # Conserver la meilleure position
xn = Xn # Mettre à jour les positions des particules
```

Figure 38 Algorithme de La phase social.

La fin de La phase sociale. Nous renverrons la condition de sélection :

```
if np.random.rand() <= pk:
    # La phase solitaire
else:
    # La phase social
```

La condition d'arrêt est: nombre de l'itération.

4.4. Résultats D'exécution:

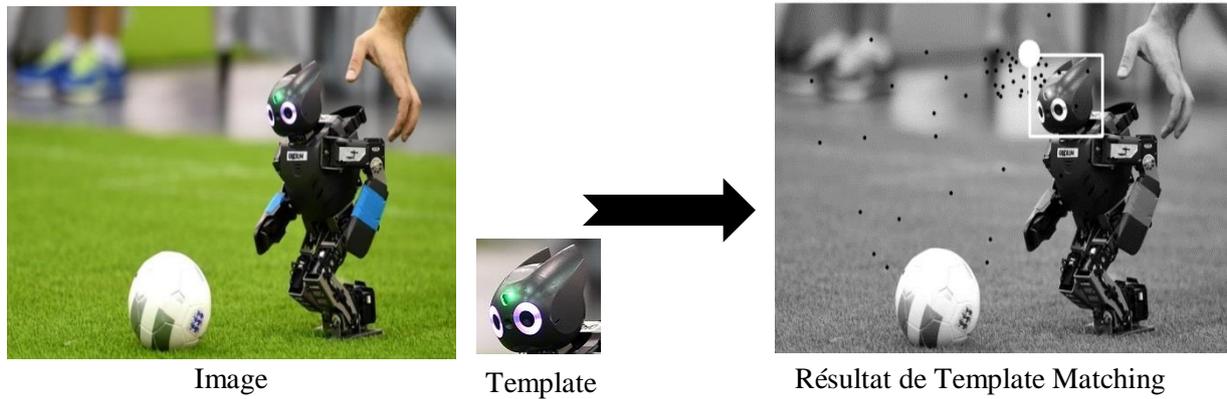


Figure 39 Le résultat de la reconnaissance du visage du robot.

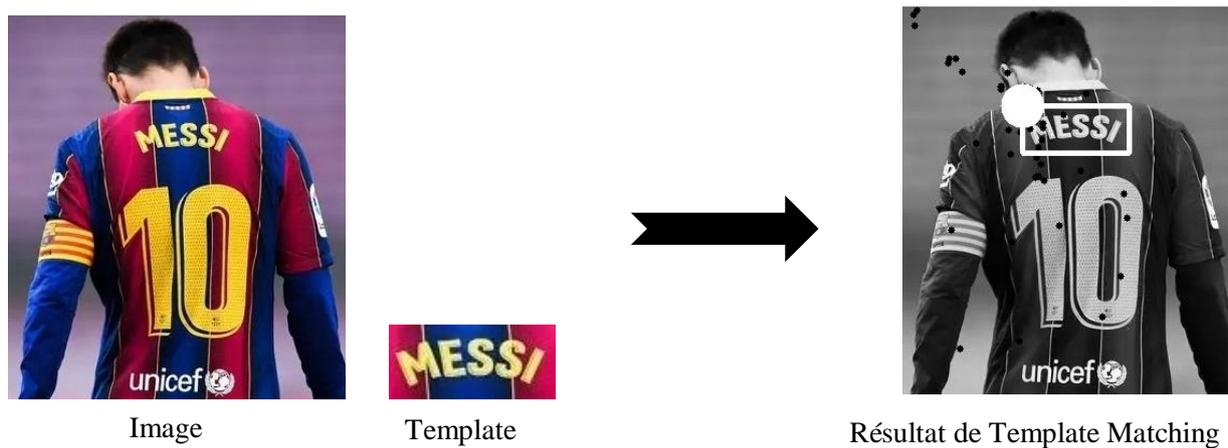


Figure 40 Le résultat de distinguer le nom de Messi

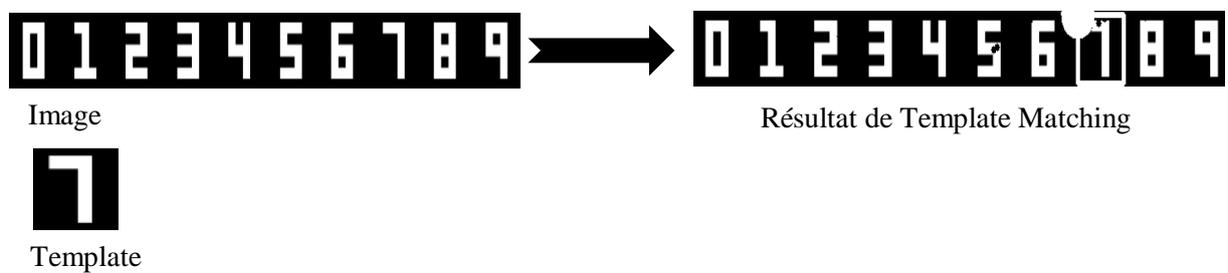


Figure 41 Le résultat de la distinction du nombre 7.

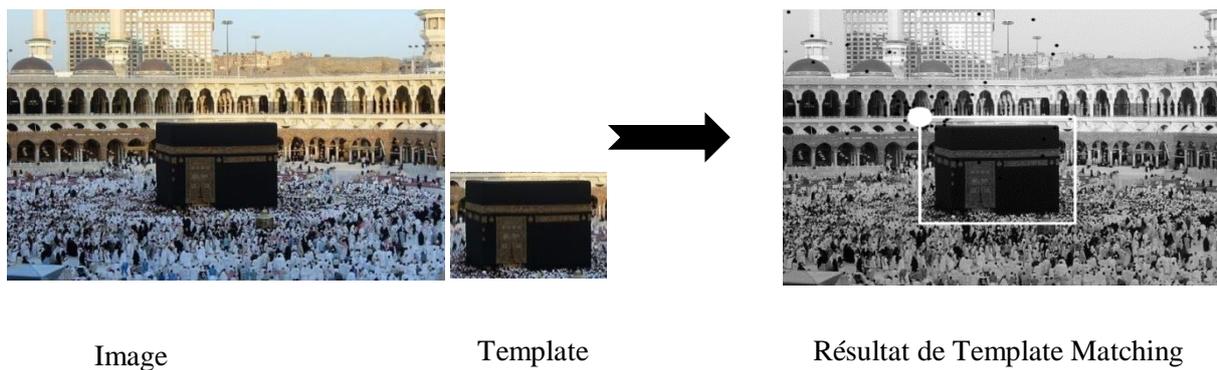


Figure 42 Le résultat de la reconnaissance de Ka'aba



Figure 43 Le résultat de la reconnaissance du visage du joueur

4.5. Évaluation

En général, et sur la base des résultats précédents, l'algorithme LS a montré de bons résultats en discrimination comme le TM traditionnel, mais ce qui est remarquable, c'est le petit nombre d'opérations NCC.

Auparavant, la mise en œuvre de TM-NCC nécessitait le calcul du NCC pour chaque partie de l'image afin de vérifier la présence du modèle. Cela impliquait la génération d'une matrice de taille $(H-h+1, W-w+1)$, où (H, W) représente la taille de l'image et (h, w) représente la taille

du motif. Ensuite, nous devons trouver la valeur maximale pour obtenir le meilleur NCC, ce qui nécessitait $(H-h+1) * (W-w+1)$ opérations de calcul NCC.

Dans notre méthode, nous avons utilisé l'algorithme d'essaim de criquets pour réduire considérablement le nombre d'opérations NCC nécessaires. Nous avons supposé la présence de 30 particules sur 100 itérations, soit en moyenne 3000 opérations NCC. Cela représente une amélioration significative en termes de réduction de la charge de calcul des opérations NCC.

Exemple :

Supposons que nous avons une image de taille $H = 500$ pixels de largeur et $W = 400$ pixels de hauteur. Nous voulons rechercher un modèle de taille $h = 50$ pixels de largeur et $w = 50$ pixels de hauteur dans cette image.

Dans ce cas, la taille de la matrice générée pour le calcul du NCC serait $(H-h+1) * (W-w+1)$, soit $(500-50+1) * (400-50+1) = 451 * 351$.

Cela signifie que nous devrions effectuer $451 * 351 = 158,001$ opérations NCC pour vérifier la présence du modèle dans l'image. Avant l'utilisation de l'algorithme d'essaim de criquets, cette approche nécessiterait le calcul du NCC pour chaque partie de l'image.

Maintenant, en supposant que nous utilisons l'algorithme d'essaim de criquets avec 30 particules sur 100 itérations, cela signifie que nous effectuons en moyenne $30 * 100 = 3,000$ opérations NCC. C'est une réduction significative par rapport à la méthode précédente.

Tableau 3 Le nombre d'opérations NCC.

TM - NCC	TM-NCC avec LS
158,001	3,000

4.5.1 .Comparaison avec l'algorithme PSO

Nous avons évalué notre travail en comparant l'algorithme LS TM-NCC avec l'algorithme PSO TM-NCC , Les tests ont été effectués avec les mêmes paramètres d'itération (100) et le même nombre de particules (30) pour une comparaison équitable.

Une fonction d'évaluation commune a été utilisée pour mesurer la meilleure valeur de similarité à chaque itération.Figure [21]

- Les parameter de pso : $N = 30$, iteration = 100 , $c1 = 2$, $c2 = 2$, $w = 0.7$.
- Les parameter de ls : $N = 30$, $F = 22$, $L = 9$, , iteration = 100 .

Ces expériences ont été réalisées à l'aide de plusieurs images, Toutes les expériences ont été réalisées sur pycharm pour Python, exécuté sur un ordinateur avec Intel Core™ i5 - 2,49 GHz, Window10 (64 bits,4 Go de mémoire) comme système d'exploitation.

Les résultats correspondants représentent Courbes d'évolution de la valeur NCC en terme d'itération, pour les algorithmes PSO-TM-NCC et LS-TM-NCC.

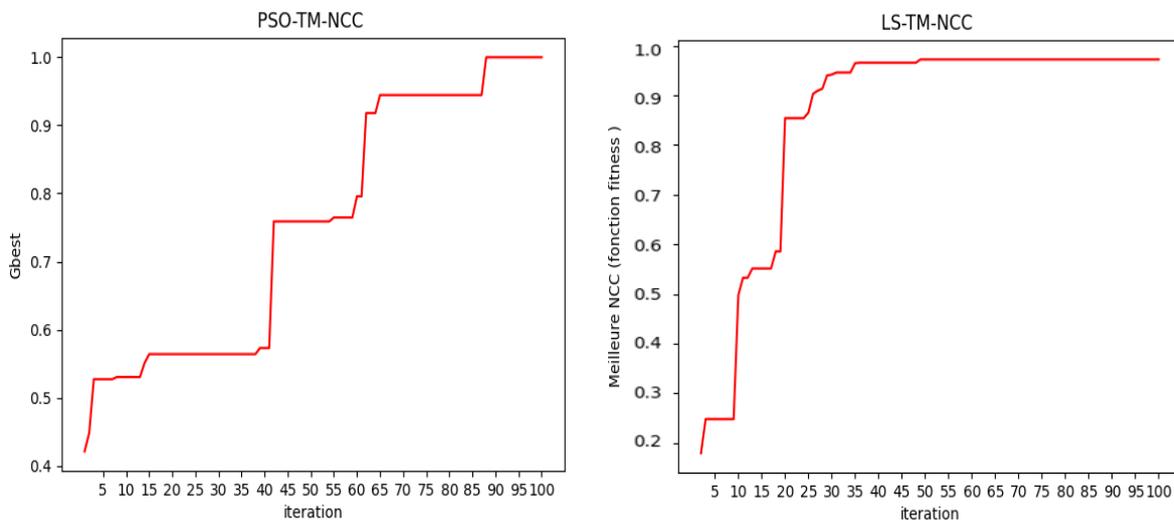


Figure 44 La courbe d'évolution de la valeur NCC en terme d'itération, pour Figure 39

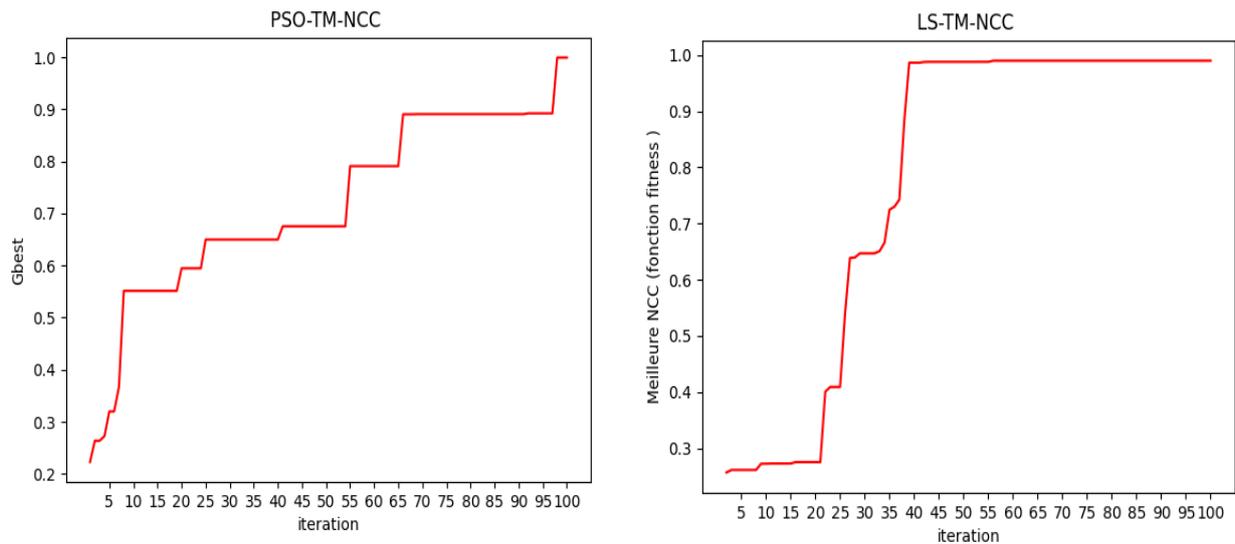


Figure 45 La courbe d'évolution de la valeur NCC en terme d'itération, pour Figure 42

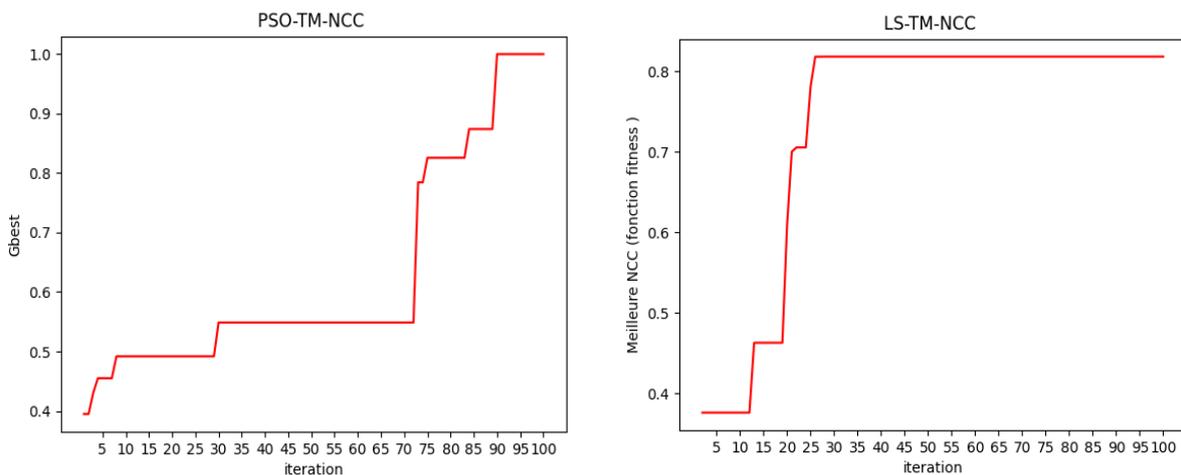


Figure 46 La courbe d'évolution de la valeur NCC en terme d'itération, pour Figure 40

Dans la Figure 44, nous voyons que la méthode LS-TM-NCC atteint la correspondance la plus élevée (NCC = 1) à (itération = 30), tandis que la méthode PSO-TM-NCC atteint la meilleure correspondance à (itération = 90).

Dans les figures 45-46, nous constatons que la méthode PSO-TM-NCC prend plus de 85% de l'itération totale pour atteindre la meilleure similitude, tandis que la méthode LS-TM-NCC prend environ 30% de l'itération totale pour trouver la meilleure correspondance ou similitude. À partir de l'analyse précédente, nous découvrons facilement la rapidité et la supériorité de

l'algorithme LS sur l'algorithme PSO pour détecter la similitude dans le processus de correspondance du modèle TM.

4.6. Conclusion

Dans ce chapitre, nous avons examiné le code source du code source avec les résultats de l'opération et nous avons obtenu de bons résultats dans la recherche de correspondances, et pour évaluer notre travail, nous avons effectué une comparaison informatique du nombre d'opérations NCC de l'algorithme LS-TM-NCC avec la TM traditionnelle, et nous avons comparé le processus de correspondance des modèles en utilisant l'algorithme PSO et LS.

Nous avons obtenu un plus petit nombre d'opérations NCC que la TM traditionnelle et une bonne vitesse de découverte du modèle.

Conclusion générale

Nous avons découvert dans cette thèse l'application d'algorithmes inspirés par la nature pour la correspondance de modèles. Elle propose une analyse approfondie de la technique de correspondance des modèles, en mettant l'accent sur les systèmes inspirés de la Bio-inspires tels que les algorithmes génétiques et l'optimisation par colonies de fourmis.

Dans le premier chapitre, la technique de correspondance des modèles est définie et son importance et ses utilisations sont discutées. Le deuxième chapitre se concentre sur les systèmes bio-inspirés, leur relation avec l'intelligence artificielle, en présentant leurs caractéristiques et en explorant leurs différentes branches, notamment l'apprentissage automatique et les systèmes experts. Les algorithmes d'optimisation, anciens et modernes, sont également présentés.

Dans le chapitre 3, l'algorithme LSO est appliqué en conjonction avec la NCC dans le processus de correspondance de modèles, en utilisant les caractéristiques et les comportements des criquets pour améliorer la recherche des correspondances. Les différentes étapes du processus sont décrites.

Dans le chapitre 4, l'accent est mis sur la mise en œuvre et l'évaluation de la recherche. La structure et l'organisation du code sont présentées. Les résultats de l'exécution sont exposés et comparés à l'algorithme PSO pour évaluer l'efficacité de l'algorithme LS dans la correspondance des modèles. Nous pouvons dire que nous avons obtenu de bons résultats.

En conclusion, cette thèse a approfondi la compréhension et l'application d'algorithmes inspirés par la nature pour la correspondance de modèles. L'algorithme LS avec TM-NCC a montré de bons résultats, ce qui démontre son efficacité dans la résolution de problèmes de correspondance de modèles. Cela ouvre des perspectives intéressantes pour les applications futures dans le domaine de l'analyse d'images et de la reconnaissance de motifs .

Les Références

- [1] <https://pyimagesearch.com/2021/03/22/opencv-template-matching-cv2-matchtemplate/> .
- [2] Nazanin Sadat Hashemi 1,*, Roya Babaei Aghdam 1, Atieh Sadat Bayat Ghiasi 1, Parastoo Fatemi .
- [3] Adrian González 1 · Erik Cuevas1 · Fernando Fausto1 · Arturo Valdivia1 · Raul Rojas
- [4] Segmentation d'images médicales IRM par la méthode d'ensembles de niveaux (Level_Sets) M. Beladgham, F. Derraz, M. Khélif .
- [5] <https://www.larousse.fr/dictionnaires/francais/intelligence/43555> .
- [6] https://www.researchgate.net/publication/312522454_Acoustic_Emission_and_Artificial_Intelligent_Methods_in_Condition_Monitoring_of_Rotating_Machine_-_A_Review .
- [7] Intelligence artificielle et réseaux de neurones. Dr Muhammad Ali Al-Sharqawi
- [8] Livre « Jorge Nocedal Stephen J. Wright Numerical Optimization
- [9] Artificial Intelligence Oumaima Dkkak Publications of the Syrian Virtual University(SVU) Syrian Arab Republic, 2018 .
- [10] Object Detection Based on Template Matching through Use of Best-So-Far ABC Anan Banharnsakun 1 and Supanee Tanathong .