Ministère de l'enseignement Supérieur et de la recherche Scientifique

وزارة التعليم العالي والبحث العلمي

Badji Mokhtar Annaba University
Université Badji Mokhtar – Annaba
Faculté de Technologie

Département Informatique

جامعة باجي مختار – عنابـــة

كـليــة التكنـولـوجيـا

قســـم الإعلام الآلي

Thèse

Présentée pour obtenir le diplôme de

Doctorat Troisième Cycle

Filière : Informatique

Spécialité : Informatiques

Par :

Lemnadjlia Saida

Thème :

Apprentissage profond et super-pixels pour une approche sémantique de compression d'image

Thèse soutenue le 2025 devant le jury composé de :

N°	Nom et prénom	Grade	Etablissement	Qualité
01	Souici-Meslati Labiba	Prof	Université Badji Mokhtar -Annaba	Président
02	Sari Toufik	Prof	Université Badji Mokhtar -Annaba	Rapporteur
03	Bendjena Hakim	Prof	Université Larbi Tébessi- tebessa	Examinateur
04	Yakoubi Mohamed Amine	MCA	Université Badji Mokhtar -Annaba	Examinateur
05	Abainia Kheireddine	MCA	Université 08 Mai 1945-Guelma	Examinateur

Dédicace

Je dédie cette thèse à la mémoire de ma directrice de thèse, Pr Ahlem Melouah, en hommage à son engagement scientifique, à son encadrement rigoureux et à l'empreinte durable qu'elle a laissée dans mon parcours académique.

Je dédie cette thèse à mes chers parents, qui m'ont offert une éducation digne et dont l'amour et les sacrifices ont fait de moi ce que je suis aujourd'hui. C'est avec une profonde gratitude que je leur adresse cette marque de reconnaissance. Que cette thèse soit le plus beau cadeau que je puisse leur offrir.

Je dédie ce travail à mes sœurs Nabila et Djamila qui m'ont accompagné et soutenu tout au long de ce projet.

Je la dédie également à mes neveux et nièces, en les remerciant pour la joie qu'ils ont apportée dans ma vie. Leur présence m'a toujours été une source de bonheur et de motivation.

Enfin, je le dédie à tous les membres de ma famille qui m'ont soutenu et encouragé depuis toujours.

Remerciements

Je tiens tout d'abord à remercier ALLAH tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce projet de recherche.

Je tiens à remercier tout particulièrement ma directrice de thèse, Pr. Ahlem Melouah, pour ses efforts, sa disponibilité, son suivi attentif, ses critiques constructives et ses conseils précieux tout au long de ces années de recherche. Je lui suis profondément reconnaissant(e) pour la confiance qu'elle m'a toujours accordée.

Je voudrais remercier également Pr Souici_Meslati Labiba, Pr Bendjena Hakim, Dr Kheireddine Abainia et Dr Yakoubi Mohamed Amine qui ont manifesté un intérêt à examiner ce travail.

Il m'est impossible d'oublier de remercier mes enseignants, Pr. Hayet Farida Merouani et Pr. Tahar Bouhadada et Pr. Souici Labiba pour leur présence constante, leur soutien, leurs encouragements et leurs conseils avisés chaque fois que j'en avais besoin.

Je tiens à remercier sincèrement Dr. Amel Slim, Dr. Noha Kobsi et Dr. Akila Djebar pour leur aide, le temps qu'elles m'ont consacré, ainsi que pour leurs conseils précieux, leurs explications claires et leurs orientations pertinentes, qui m'ont apporté un soutien considérable dans la réalisation de ce projet.

.

Un grand merci aussi à toutes les personnes qui m'ont aidé de près ou de loin dans la réalisation de mon projet,

Il m'est spécialement agréable, d'exprimer toutes mes reconnaissances envers ma famille qui m'a toujours soutenu et poussé à continuer mes études. Ce présent travail a pu voir le jour grâce à leur soutien.

Résumé :

La compression d'image joue un rôle crucial dans de nombreux domaines liés à l'imagerie numérique, notamment pour le stockage, la transmission rapide sur les réseaux et l'optimisation des ressources dans des contextes contraints (tels que les appareils mobiles, les systèmes embarqués ou le cloud). Face à l'augmentation constante de la résolution et du volume des images générées, il devient indispensable de développer des techniques de compression à la fois efficaces et intelligentes.

Cette thèse propose une approche innovante de compression d'image fondée sur une combinaison de super-pixels et d'apprentissage profond, dans une optique sémantique. Contrairement aux méthodes classiques comme JPEG, qui exploitent principalement des redondances statistiques, cette approche cherche à comprendre le contenu visuel pour adapter dynamiquement le taux de compression.

L'idée repose sur le constat que toutes les régions d'une image n'ont pas la même valeur sémantique. Ainsi, les zones contenant des objets importants (comme les visages ou objets spécifiques) méritent une compression avec perte minimale, tandis que les arrière-plans ou les zones moins informatives peuvent être compressés davantage sans altérer la perception globale.

La méthode commence par une segmentation en super-pixels, permettant de regrouper les pixels similaires tout en respectant les contours des objets. Ensuite, un réseau de neurones profond analyse ces super-pixels pour en estimer l'importance sémantique. Cette estimation sert de guide pour ajuster le taux de compression localement.

Les résultats expérimentaux démontrent que cette stratégie permet de réduire significativement le débit binaire tout en maintenant une qualité visuelle élevée, notamment sur les régions sémantiquement importantes. Des mesures objectives (PSNR, MSE, SS) et des évaluations subjectives confirment les performances de la méthode.

En intégrant l'intelligence artificielle à la compression, cette approche ouvre la voie à des systèmes adaptatifs et conscients du contenu, adaptés à des usages variés comme la télémédecine, la vidéosurveillance, ou les véhicules autonomes, où la qualité des informations transmises est aussi cruciale que leur volume.

Mots clés : Compression, JPEG, Facteur de Qualité, Compression Sémantique, Apprentissage Profond, CNN, Super Pixel, ROI.

Abstract:

Image compression plays a crucial role in many areas of digital imaging, including storage, fast transmission over networks and resource optimization in constrained contexts (such as mobile devices, embedded systems or the cloud). In view of the ever-increasing resolution and volume of images generated, it is becoming essential to develop compression techniques that are both efficient and intelligent.

This thesis proposes an innovative approach to image compression based on a combination of super-pixels and deep learning, from a semantic perspective. Unlike conventional methods such as JPEG, which mainly exploit statistical redundancies, this approach seeks to understand the visual content in order to dynamically adapt the compression ratio.

The idea is based on the observation that not all regions of an image have the same semantic value. Thus, areas containing important objects (such as faces or specific objects) deserve minimal lossy compression, while backgrounds or less informative areas can be compressed further without altering overall perception.

The method begins with super-pixel segmentation, enabling similar pixels to be grouped together while respecting object contours. Next, a deep neural network analyzes these super-pixels to estimate their semantic importance. This estimate serves as a guide for adjusting the compression ratio locally.

Experimental results show that this strategy significantly reduces the bit rate while maintaining high visual quality, particularly in semantically important regions. Objective measurements (PSNR, MSE, SS) and subjective evaluations confirm the method's performance.

By integrating artificial intelligence with compression, this approach paves the way for adaptive, content-aware systems, suitable for a variety of uses such as telemedicine, video surveillance or autonomous vehicles, where the quality of the information transmitted is as crucial as its volume.

Key words : Compression, JPEG, Quality Factor, Semantic Compression, Deep Learning, CNN, Super Pixel, ROI.

الملخص:

يلعب ضغط الصور دورًا حاسمًا في العديد من المجالات المتعلقة بالتصوير الرقمي، لا سيما للتخزين والنقل السريع عبر الشبكات وتحسين الموارد في السياقات المقيدة (مثل الأجهزة المحمولة أو الأنظمة المدمجة أو السحابة). ونظراً للزيادة المستمرة في دقة وحجم الصور التي يتم إنشاؤها، فقد أصبح من الضروري تطوير تقنيات ضغط تتسم بالكفاءة والذكاء في آن واحد.

تقترح هذه الأطروحة نهجًا مبتكرًا لضغط الصور استنادًا إلى مزيج من وحدات البكسل الفائقة والتعلم العميق، من منظور دلالي. على عكس الطرق التقليدية مثلJPEG ، التي تستغل بشكل أساسي التكرار الإحصائي، يسعى هذا النهج إلى فهم المحتوى المرئي من أجل تكييف نسبة الضغط بشكل ديناميكي.

تعتمد الفكرة على ملاحظة أنه ليست كل أجزاء الصورة لها نفس الأهمية و القيمة الدلالية. وبالتالي، تستحق المناطق التي تحتوي على معلومات مهمة (مثل الوجوه أو كائنات محددة) الحد الأدنى من الضغط المفقود، بينما يمكن ضغط الخلفيات أو المناطق الأقل إفادة بشكل أكبر دون تغيير الإدراك العام.

تبدأ الطريقة بتجزئة وحدات البكسل الفائقة، والتي تجمع وحدات البكسل المتشابهة معًا مع احترام خطوط الأجسام. ثم تقوم شبكة عصبية عميقة بتحليل هذه البكسلات الفائقة لتقدير أهميتها الدلالية. يُستخدم هذا التقدير كدليل لضبط معدل الضغط محليًا.

تُظهر النتائج التجريبية أن هذه الاستراتيجية تقلل بشكل كبير من معدل البت مع الحفاظ على جودة بصرية عالية، خاصة في المناطق المهمة دلالياً. تؤكد القياسات الموضوعية (PSNR, MSE, SS) والتقييمات الذاتية أداء الطريقة.

من خلال دمج الذكاء الاصطناعي مع الضغط، يمهد هذا النهج الطريق لأنظمة تكييفية مدركة للمحتوى ومناسبة لمجموعة متنوعة من الاستخدامات مثل التطبيب عن بُعد والمراقبة بالفيديو والمركبات ذاتية القيادة، حيث تكون جودة المعلومات المنقولة مهمة مثل حجمها.

كلمات مفتاحية: الضغط، JPEG، عامل الجودة، الضغط الدلالي، التعلم العميق، CNN، البكسل الفائق، ROI.

Table des matières

Résumé

Abstract

الملخص

Table des matières

Liste des Figures

Liste des tableaux

Introduction générale

Partie1 : Notions préliminaires

Chapitre 1 : Deep Learning

1.1 Introduction 6

1.2 Apprentissage profond (Deep Learning) 6

1.2.1 Réseaux de Neurones artificiels (ANN) 8

1.2.2 Réseaux de Neurones Récurrents (RNN) 8

1.2.3 Auto-Encodeurs (AE) 10

1.2.4 Réseau de neurone convolutionel (CNN) 12

1.2.5 Boltzmann Machines et Deep Belief Networks (DBN) 16

1.2.6 Réseaux de Neurones Spiking (SNN) 17

1.2.7 Réseaux Antagonistes Génératifs (GAN) 17

1.3 Conclusion 18

Chapitre 2 : Compression JPEG et Compression sémantique

2.1 Introduction 20

2.2 Joint Photographic Experts Group (JPEG) 21

2.2.1 Les différentes étapes de JPEG 22

2.2.1.1 DCT 22

2.2.1.2 Quantification 25

2.2.1.3 Linéarisation des blocs 26

2.2.1.4 Codage entropique dans le format JPEG 26

2.2.2 Principe de la Compression JPEG 27

2.2.3 Algorithme de Compression JPEG 28
2.2.4 Exemple de Compresseur JPEG sur une image 29
2.2.5 Les Limites de JPEG 33
2.3 Compression Sémantique 34
2.4 Conclusion 39

Chapitre 3 : Super Pixel

3.1 Introduction 42
3.2 Super Pixel 42
3.2.1 Pourquoi utiliser les super pixels ? 42
3.2.2 Propriétés essentielles des super pixels 42
3.2.3 Comment Construire un super pixel ? 44
3.3 Algorithmes de construction des Super Pixels 44
3.3.1 Super pixel basé sur la forêt minimale couvrante 44
3.3.2 Approche basée sur les k-means pour les super pixels 46
3.3.2.1 SLIC Super Pixel 47
3.3.2.2 SPASM segmentation 50
3.4 Conclusion 51

Partie2 les contributions

Chapitre 4 : Approche de compression des images basée sur le découpage de l'image

4.1 Introduction 55
4.2 Problématique 55
4.3 Motivation 56
4.4 Approche proposée 57
4.5 Tests et Résultats 61
4.6 Discussion 65
4.7 Conclusion 66

Chapitre 5 : Approche hybrides de compression des images basée super pixel, découpage et CNN

5.1 Introduction 68
5.2 Motivation 68
5.3 Approche Proposée 69

5.3.1 Super pixel 73
5.3.2 Partitionnement 74
5.3.3 Classification 75
5.3.4 Calcul du FQ 76
5.3.5 Compression 76
5.3.6 Groupement 77
5.3.7 Décompression 78
5.4 Conclusion 79

Chapitre 6 : Résultats et Discussion

6.1 Introduction 81
6.2 Résultats 81
6.2.1 Résultats de la Segmentation et de Découpage 81
6.2.2 Résultats de Classification 82
6.2.3 Résultats de Compression 84
6.3 Discussion 87
6.3.1 Comparaison de l'approche proposé et JPEG 87
6.3.1.1 Implémentation de l'approche proposée sur des images contenant des zones d'intérêt 87
6.3.1.2 L'implémentation de l'approche proposée sur des images naturelles 100
6.3.1.3 L'implémentation de l'approche proposée sur des images texturées 102
6.3.1.4 L'implémentation de l'approche proposée sur des images satellitaires 105
6.3.2 Comparaison de l'Approche proposée avec des méthodes similaires 107
6.4 Conclusion 110

Conclusion Générale

Mes Contributions Scientifiques

Bibliographie

Liste des Figures

Chapitre 1 : Deep Learning

Figure 1.1 Evolution de l'IA ……. 07
Figure 1.2 Conception de réseaux de neurones …… 07
Figure 1.3 Structure d'un RNN …… 09
Figure 1.4 Schéma d'un AE avec 6 neurones dans les couches d'entrée /sortie et trois neurones dans la couche cachée ……. 11
Figure 1.5 Illustration d'un exemple d'un CNN comprenant plusieurs couches, conçu pour analyser une image et prédire correctement sa classe ……. 12
Figure 1.6 Représentation visuel en 3D de l'opération de convolution …… 13
Figure 1.7 Illustration de l'opération de la convolution ……... 13
Figure 1.8 Illustration du processus de convolution appliqué à une image ……. 14
Figure 1.9 Illustration du processus de convolution appliqué à une image ……. 15
Figure 1.10 La principale architecture de CNN ……. 15
Figure 1.11 Architecture de GAN …….. 17

Chapitre 2 : Compression JPEG et Compression sémantique

Figure 2.1 Les types de compression des images ……. 20
Figure 2.2 Algorithme JPEG ……. 22
Figure 2.3 Balayage en Séquence Zig-Zag ……... 26
Figure 2.4 Les étapes de la Compression JPEG ……... 27
Figure 2.5 Les étapes de la décompression JPEG ……. 27
Figure 2.6 Conversion d'une image RGB en YCBCR ……. 28
Figure 2.7 Exemple d'image de type BMP ……... 29
Figure 2.8 Image découpée en bloc 8*8 ……... 30
Figure 2.9 Exemple de balyage Zig Zag et AC Coefficients …… 31
Figure 2.10 Calcul des itérations des coefficients …….. 32
Figure 2.11 Exemple d'image restituée …….. 33
Figure 2.12 Perte des informations ……. 34
Figure 2.13 Concept de la Zone d'intérêt ROI ……... 35
Figure 2.14 Comparaison des différentes tâches de reconnaissance d'images…… 36
Figure 2.15 Classification d'image ……. 36
Figure 2.16 Segmentation par instance …….. 36
Figure 2.17 Segmentation sémantique …… 37
Figure 2.18 Analyse sémantique d'images …… 38
Figure 2.19 Segmentation sémantique d'instance …….. 39

Chapitre 3 : Super Pixel

Figure 3.1 Différents résultats de Super Pixel …….. 43
Figure 3.2 Schéma de la méthode SLIC pour R = 2 ……. 49
Figure 3.3 Exemples d'application de l'algorithme SPASM sur des images issues du jeu de données Middlebury …… 51

Chapitre 4 : Approche de compression des images basée sur le découpage de l'image

Figure 4.1 Exemples des images homogènes et hétérogènes ……... 57
Figure 4.2 Architecture de la compression de l'approche proposée ……. 58
Figure 4.3 Partitionnement de l'image de Lina en 4 parties ……. 59
Figure 4.4 Architecture de la décompression de l'approche proposée ……. 61
Figure 4.5 Interface de compresseur proposée avec découpage en 4 ……... 62
Figure 4.6 Interface de compresseur proposée avec découpage en 8 ……... 62
Figure 4.7 Interface de compresseur proposée avec découpage en 16 ……. 63

Chapitre 5 : Approche hybrides de compression des images basée super pixel, découpage et CNN

Figure 5.1 Exemple des partitions de l'image de Lina ……. 70
Figure 5.2 Qualité de l'image compressée par rapport au Facteur de Qualité (FQ) ……. 71
Figure 5.3 Différentes étapes de compression JPEG …… 71
Figure 5.4 Matrice de Quantification (FQ=10) ……. 72
Figure 5.5 Flux de données de l'Approche Proposée ……... 73
Figure 5.6 Architecture D'AlexNet …….. 75
Figure 5.7 Illustration de la compression de partition et du codage FQ ……... 78

Chapitre 6 : Résultats et Discussion

Figure 6.1 Exemples de segmentation et de découpage ……... 81
Figure 6.2 Exemple de dix images de peau testées …….. 82
Figure 6.3 Niveaux de perte d'informations ……. 85
Figure 6.4 Comparaison des PSNRs des images médicales dermatologiques …….. 90
Figure 6.5 Comparaison des CRs des images médicales dermatologique …… 92
Figure 6.6 Comparaison de PSNRs des images d'oiseau ……. 95
Figure 6.7 Comparaison des CRs des images des oiseaux ……... 96
Figure 6.8 Comparaison des PSNRs des voitures ……. 99
Figure 6.9 Comparaison des CRs de voiture …… 100
Figure 6.10 Comparaison des espaces de stockage des images naturelles ……. 102
Figure 6.11 Comparaison des espaces de stockage des images texturées …….. 104
Figure 6.12 Comparaison des espaces de stockage des images satellitaires ……... 106
Figure 6.14 Comparaison des CR, PSNR, MSE et SS de l'approche proposée et les autres méthodes …….. 110

Liste des Tableaux

Chapitre 2 : Compression JPEG et Compression sémantique

Tableau 2.1 Exemple de Bloc 8*8 …… 23
Tableau 2.2 Matrice DCT d'un bloc ……. 24
Tableau 2.3 Matrice DCT arrondie ……... 24
Tableau 2.4 Matrice de Quantification (FQ=10) …….. 25
Tableau 2.5 Exemple de Matrice d'un bloc …….. 30
Tableau 2.6 Exemple de Matrice DCT d'un bloc ……. 30
Tableau 2.7 Exemple de Matrice quantification FQ= 5 ……... 31
Tableau 2.8 Exemple de Matrice DCT Quantifiée ……... 31
Tableau 2.9 Exemple de Matrice DCT Déquantifiée ……... 32
Tableau 2.10 Exemple de Matrice DCT Quantifiée ……... 32
Tableau 2.11 Exemple de Matrice DCT inverse ……. 33

Chapitre 3 : Super Pixel

Tableau 3.1 Exemples de segmentation par SLIC super pixel ……. 50

Chapitre 4 : Approche de compression des images basée sur le découpage de l'image

Tableau 4.1 Exemple de compression JPEG avec influence du FQ ……. 56
Tableau 4.2 Exemple des valeurs différentes de degrés d'homogénéité …….. 59
Tableau 4.3 Équivalence entre la valeur du facteur de qualité et le degré d'homogénéité ... 60
Tableau 4.4 Tableau illustratif des résultats ……. 64
Tableau 4.5 Taille de stockage ……. 65
Tableau 4.6 Valeurs des FQ ……. 65

Chapitre 5 : Approche hybrides de compression des images basée super pixel, découpage et CNN

Tableau 5.1 Architecture D'AlexNet …… 75

Chapitre 6 : Résultats et Discussion

Tableau 6.1 Performances d'AlexNet ……... 82
Tableau 6.2 Exemple de classification des partitions des dix images …….. 83
Tableau 6.3 Valeurs FQ …… 84
Tableau 6.4 Autres mesures de l'approche proposée …… 86
Tableau 6.5 Comparaison de la qualité des images médicales dermatologique restaurée ... 89
Tableau 6.6 Taille de stockage et CR de Jpeg avec différentes valeurs et l'approche proposée …… 91
Tableau 6.7 Comparaison de la qualité des images oiseau restaurée ……... 93
Tableau 6.8 Comparaison de la qualité des images de voiture restaurée ……. 97

Tableau 6.9 Comparaison de la qualité et d'espace de stockage des images de nature restaurée …… 101
Tableau 6.10 Comparaison de la qualité et d'espace de stockage des images texturées restaurée …… 104
Tableau 6.11 Comparaison de la qualité et d'espace de stockage des images satellitaires restituées ……... 105
Tableau 6.12 Tableau comparatif des résultats du différent type d'image ……. 107
Tableau 6.13 Comparaison de CR, PSNR, MSE et SS de l'approche proposée avec d'autres méthodes ……... 109

Introduction Générale

1. Contexte général

Dans un monde où les volumes de données visuelles ne cessent de croître, que ce soit à travers les réseaux sociaux, les systèmes de vidéosurveillance, la médecine ou encore les objets connectés, la compression d'image s'impose comme une problématique cruciale. Elle vise à réduire la taille des fichiers tout en préservant un maximum d'information utile, afin de faciliter le stockage, la transmission et le traitement des contenus visuels. En imagerie, la compression représente un défi majeur, en particulier lorsqu'il s'agit de concilier efficacité de stockage ou de transmission avec la conservation de la qualité des informations essentielles.

2. Problématique et état de l'art synthétique

Depuis plusieurs décennies, de nombreuses techniques de compression d'image ont été développées, allant des méthodes sans perte aux approches avec perte, chacune répondant à des objectifs spécifiques. Les méthodes avec perte, souvent privilégiées pour des taux de compression élevés, soulèvent des enjeux liés à la perte d'informations visuelles critiques.
Parmi les directions explorées, la notion de régions d'intérêt (ROI) est devenue centrale, notamment dans les applications médicales, où il est crucial de préserver l'intégrité des zones diagnostiques. De plus, les méthodes traditionnelles, comme JPEG, bien qu'efficaces, présentent des limites lorsqu'il s'agit de prendre en compte la sémantique des images.
L'émergence de l'apprentissage profond a ouvert de nouvelles perspectives en matière de compression intelligente. Les réseaux de neurones, capables d'apprendre des représentations riches et significatives, ont permis d'adapter les taux de compression aux contenus et de préserver la qualité dans les zones critiques. L'usage des super pixels, en tant que technique de segmentation préalable, permet de structurer l'image en régions homogènes, favorisant une compression contextuelle et hiérarchisée.

3. Objectifs et hypothèses

Cette thèse se propose d'explorer une approche novatrice de la compression d'image, fondée sur l'intégration de trois éléments clés :

- la robustesse du compresseur JPEG,
- la puissance d'abstraction de l'apprentissage profond,
- et la structuration visuelle offerte par les super pixels.

L'objectif principal est de développer une méthode de compression sémantiquement guidée, capable d'adapter le taux de compression selon le contenu, et en particulier de préserver la qualité visuelle des régions importantes (ROI).
Nous faisons l'hypothèse que la combinaison intelligente de ces trois composantes permettra d'améliorer significativement la qualité des images compressées, tout en réduisant leur taille, en comparaison avec les méthodes classiques.

4. Méthodologie

La méthodologie suivie dans cette thèse repose sur plusieurs étapes :

- ✓ Une revue des approches existantes, qu'elles soient classiques (JPEG, DCT), hybrides, ou basées sur l'IA ;
- ✓ La mise en œuvre de deux contributions principales : une compression par découpage spatial préalable, et une compression combinant JPEG, super pixels et CNN ;
- ✓ Une évaluation expérimentale rigoureuse a été conduite à l'aide de plusieurs jeux de données représentatifs, incluant des images naturelles, médicales et à contenu structuré, afin de valider la robustesse et la généralité des approches proposées. Cette évaluation s'appuie sur des comparaisons quantitatives, à travers des indicateurs tels que le PSNR, le MSE, le SS et le taux de compression, ainsi que sur une analyse qualitative de la qualité visuelle des images compressées.

5. Originalité et contribution

L'originalité de ce travail réside dans l'articulation inédite entre :

- la segmentation par super pixels, pour une identification précise des zones homogènes ;
- L'apprentissage profond, capable de capter les informations sémantiques des images ;
- et l'usage stratégique du JPEG, afin de tirer parti d'un format largement répandu tout en en étendant les capacités.

Ce triptyque permet de construire un schéma de compression adaptatif, où le taux et la stratégie de compression sont conditionnés par la nature du contenu image, avec un souci constant de préserver la qualité visuelle perçue.

6. Plan de la thèse

Cette thèse est structurée en six chapitres, organisés comme suit :

- **Chapitre 1** : *Apprentissage profond* : introduction aux fondements de l'apprentissage profond, aux principaux modèles utilisés (CNN, RNN, auto-encodeurs), et à leurs domaines d'application.
- **Chapitre 2** : *Compression JPEG et Compression sémantique* : présentation du compresseur JPEG, de ses mécanismes (DCT, quantification, codage), et de ses limitations, notamment dans la gestion de la qualité perceptuelle. . Enfin, il introduit la notion de compression sémantique comme une réponse aux limites du JPEG, en soulignant l'importance de l'exploitation du contenu significatif de l'image pour une compression plus intelligente et adaptée aux besoins actuels.
- **Chapitre 3** : *Super pixels* : exploration du concept de super pixels, des algorithmes de génération (SLIC, SEEDS, etc.), et de leur utilité dans la structuration des images.
- **Chapitre 4** : *Compression par découpage* : première contribution, avec une méthode de segmentation préalable de l'image pour une compression ciblée et différenciée.

- **Chapitre 5** : *Compression combinée JPEG–CNN–super pixels* : deuxième contribution, avec une approche intégrée combinant JPEG, CNN et super pixels, pour une compression sémantiquement guidée.
- **Chapitre 6** : *Évaluation expérimentale* : protocole d'expérimentation, jeux de données utilisés (généraux et médicaux), analyse des performances comparées, et discussion des résultats.

Partie1 :

Notions préliminaires

Chapitre1 :

L’apprentissage profond

1.1 Introduction

Avec l'explosion des volumes de données visuelles générées chaque jour, la compression des images est devenue une nécessité pour le stockage, la transmission et le traitement efficace des informations. Les méthodes traditionnelles de compression, comme JPEG ou JPEG2000, bien qu'efficaces, atteignent leurs limites face aux exigences croissantes de qualité, de rapidité et d'adaptabilité aux nouvelles applications. C'est dans ce contexte que l'apprentissage profond a émergé comme une alternative prometteuse, transformant profondément les approches de compression.

Les réseaux neuronaux, et en particulier les architectures avancées comme les auto_encodeurs convolutionnels et les réseaux génératifs adverses (GAN), offrent des capacités inédites pour apprendre des représentations compactes des images. Contrairement aux approches classiques fondées sur des transformations fixes (DCT, ondelettes), ces modèles apprennent directement à partir des données à optimiser les compromis entre compression et qualité visuelle.

Ce chapitre propose une exploration approfondie du concept d'apprentissage profond (deep Learning). Il présente les principaux types de modèles utilisés, tels que les réseaux neuronaux convolutifs (CNN), les réseaux récurrents (RNN), ou encore les auto_encodeurs. Chacun de ces modèles est examiné à travers ses mécanismes de fonctionnement, ses domaines d'application (comme la vision par ordinateur, la reconnaissance vocale, ou le traitement du langage naturel), ainsi que son rôle dans l'évolution des technologies intelligentes. L'objectif est de mettre en lumière l'importance stratégique de l'apprentissage profond dans le développement de solutions innovantes et performantes en intelligence artificielle.

1.2 Apprentissage profond (Deep Learning)

Les techniques conventionnelles du Machine Learning ont été limitées dans leur capacité à traiter les données naturelles sous leur forme brute. Pendant des décennies, La mise en place d'un système de reconnaissance ou de modèle de machine Learning requis Une ingénierie soignée et une expertise de domaine considérable pour concevoir un extracteur qui transforme les données brutes (telles que les valeurs de pixel d'une image) en une représentation interne en un vecteur de caractéristique approprié à partir du quel le sous-système d'apprentissage, souvent un classificateur, pourrait détecter ou classer les motifs en entrée [Goodfellow et al., 2016].

Le deep Learning, ou apprentissage structuré en profondeur, est une technique de machine Learning apparue autour de 2006. Elle repose sur l'utilisation de réseaux de neurones à couches multiples pour effectuer une série de traitements hiérarchisés de manière autonome, sans intervention humaine. Depuis son émergence, cette approche, également appelée apprentissage hiérarchique, s'est imposée comme un domaine novateur du machine learning [Deng et Yu,

2013]. Elle a permis de réaliser des avancées significatives dans la résolution de problèmes complexes qui avaient longtemps résisté aux efforts de la communauté scientifique. Le deep learning s'est notamment révélé particulièrement efficace pour explorer et analyser des structures complexes au sein de données de grande dimension.

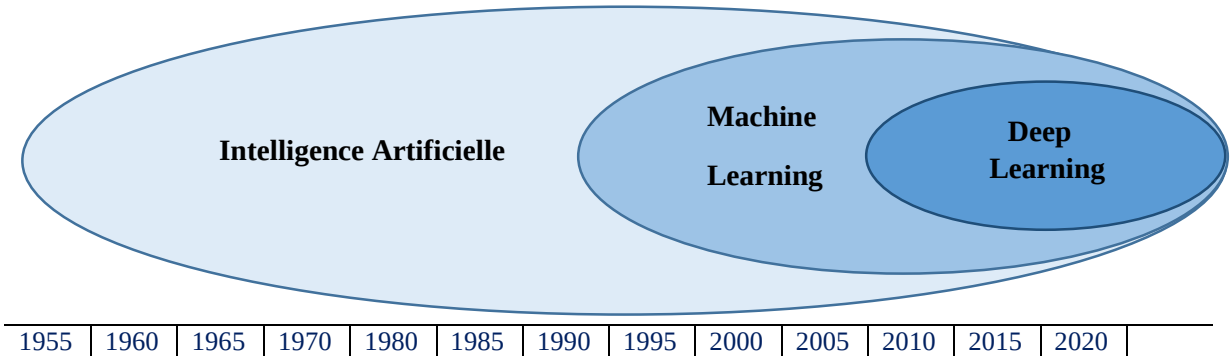


Figure 1.1 : Evolution de l'IA.

Les réseaux de neurones, constitués de structures artificielles inspirées des cellules biologiques, offrent une perspective innovante pour explorer des domaines tels que la perception, la mémoire, l'apprentissage et le raisonnement. Ils représentent également une alternative prometteuse pour surmonter certaines limites des méthodes d'apprentissage traditionnelles. En exploitant le traitement parallèle de l'information et des mécanismes semblables à ceux des neurones biologiques, ces réseaux permettent de faire émerger des propriétés adaptées à la résolution de problèmes autrefois considérés comme complexes.

Après plusieurs cycles d'enthousiasme et de désillusion quant à leur utilité réelle, les réseaux de neurones sont désormais au cœur des progrès du "Deep Learning" ou "apprentissage profond". Ce domaine émergent a été développé pour rapprocher la machine learning (ML) de l'un de ses objectifs initiaux : atteindre une véritable intelligence artificielle. La figure 1.1 retrace les grandes étapes de l'évolution de l'apprentissage profond, mettant en évidence les avancées majeures et les jalons technologiques qui ont marqué son développement au fil du temps.

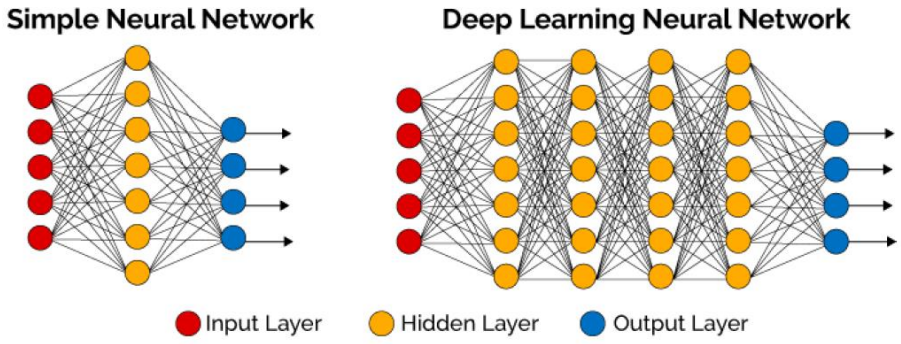


Figure 1.2 : Conception de réseaux de neurones.

Le deep learning regroupe diverses architectures de réseaux neuronaux adaptées à des types de données et à des applications spécifiques. Voici les principaux types de deep learning :

1.2.1 Réseaux de Neurones artificiels (ANN)

> *Un réseau de neurones artificiels : est un circuit composé d'un nombre très important d'unités de calcul simples basées sur des neurones formels.*
> *'Albert Negrin, 1993, Cambridge'*

Les réseaux de neurones artificiels se composent de réseaux densément interconnectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur produit une sortie unique à partir des informations qu'il reçoit. Toute structure hiérarchique de réseaux peut naturellement être considérée comme un réseau [Touzet, 1992]. La figure 1.2 tirée de [Dassi, 2023] illustre le fonctionnement des réseaux de neurones.

Cette approche a été largement adoptée dans le domaine de l'intelligence artificielle. Les réseaux de neurones, fortement inspirés du système nerveux biologique, visent à simuler l'intelligence humaine en modélisant mathématiquement le cerveau afin d'en reproduire certaines capacités : la mémoire, l'apprentissage et le traitement d'informations incomplètes. Un réseau de neurones peut être entraîné pour accomplir une tâche spécifique, comme la reconnaissance de caractères, en ajustant les valeurs des connexions (ou poids) entre ses neurones.

Au cours des dix dernières années, les réseaux de neurones artificiels (ANN) ont trouvé des applications dans de nombreuses disciplines, notamment l'économie, l'écologie, l'environnement, la biologie et la médecine. Ils sont particulièrement utilisés pour résoudre des problématiques telles que la classification, la prédiction, la catégorisation, l'optimisation, la reconnaissance de formes et la mémoire associative. Dans le traitement des données, les ANN se révèlent être un outil puissant pour approximer des systèmes complexes, notamment lorsque ces derniers ne peuvent pas être modélisés efficacement par des méthodes statistiques classiques. Ils sont également adaptés aux situations où une relation non linéaire existe entre une variable explicative et une variable à prédire. Grâce à leur conception et à leur mode de fonctionnement, les ANN sont capables de détecter et de gérer des interactions multiples non linéaires entre plusieurs variables d'entrée, ce qui leur permet de modéliser des relations complexes entre variables indépendantes et dépendantes [Schmitt et al., 2001].

1.2.2 Réseaux de Neurones Récurrents (RNN)

Les réseaux de neurones récurrents (RNN) sont conçus pour traiter des flux de données grâce à leurs unités cachées capables d'analyser des séquences. Cela les rend particulièrement utiles pour des applications où la sortie dépend des calculs précédents, comme l'analyse de séquences de texte, de parole ou d'ADN. Les RNN sont généralement entraînés avec des données d'apprentissage qui présentent de fortes interdépendances et contiennent des informations significatives à chaque étape. Le résultat produit par le réseau à l'instant t-1 influence les

décisions prises à l'instant t. Ainsi, les RNN combinent deux sources d'information : le présent et le passé récent, pour générer une sortie adaptée aux nouvelles données. Grâce à cette capacité à conserver et utiliser des informations précédentes, on dit souvent que les RNN possèdent une forme de "mémoire"[Williams et Zipser, 1989].

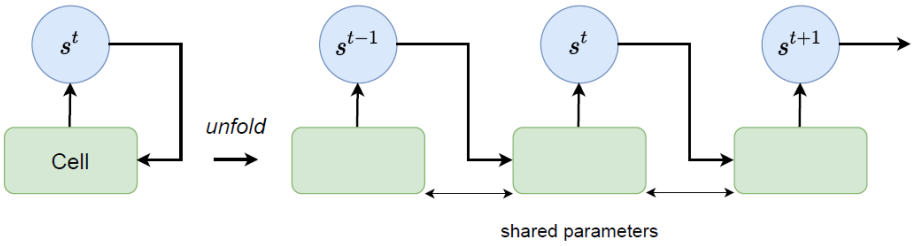


Figure 1.3 : Structure d'un RNN.

Les réseaux de neurones récurrents (RNN) traitent une séquence d'entrée élément par élément, en maintenant dans leurs unités cachées un "vecteur d'état" qui contient implicitement des informations sur l'historique des éléments précédents de la séquence. Lorsque l'on considère les sorties des unités cachées à différents instants discrets comme les sorties de neurones dans un réseau multicouche profond (voir figure 1.3, à droite), il devient évident qu'une rétropropagation dans le temps peut être utilisée pour entraîner les RNN.

Les réseaux de neurones récurrents (RNN) ont démontré leur efficacité pour prédire le caractère suivant dans un texte [Parizeau, 2006] ou le mot suivant dans une séquence [Mikolov et al., 2013], mais leur utilité s'étend également à des tâches plus complexes. Par exemple, après avoir analysé une phrase en anglais mot par mot, un réseau "encodeur" peut être formé de manière à ce que le vecteur d'état final de ses unités cachées représente adéquatement l'idée exprimée par la phrase. Ce vecteur d'idée peut ensuite être utilisé comme état caché initial (ou comme entrée supplémentaire) d'un réseau de "décodeur" français, formé simultanément, qui génère une distribution de probabilité pour le premier mot de la traduction en français. Si un mot particulier est sélectionné à partir de cette distribution et fourni en entrée au décodeur, il produira une distribution de probabilité pour le deuxième mot, et ainsi de suite, jusqu'à ce qu'un mot de fin soit généré [Sutskever et al., 2014]-[Cho et al., 2014]-[Bahdanau et al., 2015].

Ce processus génère ainsi des séquences de mots en français en fonction de la distribution de probabilité liée à la phrase en anglais. Cette méthode relativement simple de traduction automatique est rapidement devenue compétitive avec les meilleures approches de l'époque. De plus, au lieu de traduire une phrase d'une langue à une autre, il est désormais possible d'apprendre à "traduire" la signification d'une image en une phrase en anglais. Dans ce cas, l'encodeur est un réseau de neurones convolutif profond (DCNN), qui transforme les pixels de l'image en un vecteur d'activité dans sa dernière couche cachée. Le décodeur, quant à lui, est un RNN similaire à ceux utilisés pour la traduction automatique et la modélisation du langage neural. Ces systèmes ont récemment suscité un regain d'intérêt.

Les réseaux de neurones récurrents (RNN) sont aujourd'hui utilisés dans de nombreux domaines scientifiques, commerciaux et gouvernementaux. En plus de surpasser d'autres méthodes dans

des domaines tels que la reconnaissance d'image [Krizhevsky et al., 2012], et la reconnaissance vocale [Hin et al., 2012]. ils ont également démontré leur supériorité dans des applications telles que la prédiction de l'activité des molécules de médicaments potentiels [Ma et al., 2015]. l'analyse des données provenant des accélérateurs de particules [Leung et al., 2014],[Xiong et al., 2015], la reconstruction des circuits cérébraux [Helmstaedter et al., 2013] et la prédiction des effets des mutations dans l'ADN non codant sur l'expression des gènes et les maladies [Leung et al., 2014]-[Xiong et al., 2015]. De manière particulièrement impressionnante, l'apprentissage profond a montré des résultats extrêmement prometteurs dans diverses tâches de compréhension du langage naturel [Collobert et al., 2011], notamment la classification des sujets, l'analyse du sentiment, la réponse aux questions [Bordes et al., 2014] et la traduction [Jean et al., 2015]-[Sutskever et al., 2014].

1.2.3 Auto-Encodeurs (AE)

L'auto-encodeur profond (AE) est un type particulier de réseau de neurones profonds (DNN) où les données en sortie sont identiques à celles en entrée, d'où son autre nom, "auto-associateur". Cet algorithme d'apprentissage non supervisé est conçu pour coder efficacement des ensembles de données, principalement dans le but de réduire leur dimensionnalité [Ravi et al., 2016]. Plus précisément, il s'agit d'une méthode d'extraction de caractéristiques non linéaire, ne nécessitant aucune étiquette de classe.

Les auto-encodeurs (AEs) ont joué un rôle central dans les recherches sur les réseaux de neurones artificiels (ANN). En 1988, Bourlard et Kamp [Schmidhuber, 2015] ont démontré qu'un perceptron multicouche, lorsqu'il fonctionne en mode auto-associatif, pouvait effectuer la compression des données et réduire leur dimensionnalité, notamment dans des applications liées au traitement de l'information. En règle générale, un auto-encodeur repose sur un réseau artificiel (AN) comportant trois couches ou plus [Deng, 2012].

- ✓ Une couche d'entrée, destinée à recevoir les données à coder efficacement, comme des pixels d'une image ou un spectre sonore.
- ✓ Une ou plusieurs couches cachées, généralement de taille réduite, avec un nombre d'unités inférieur à celui des couches d'entrée et de sortie. Ces couches permettent d'encoder les données dans un espace de dimension inférieure.
- ✓ Une couche de sortie, où chaque neurone correspond directement à un neurone de la couche d'entrée.

Lorsque l'auto-encodeur comporte plus d'une couche cachée, il est qualifié de "profond".

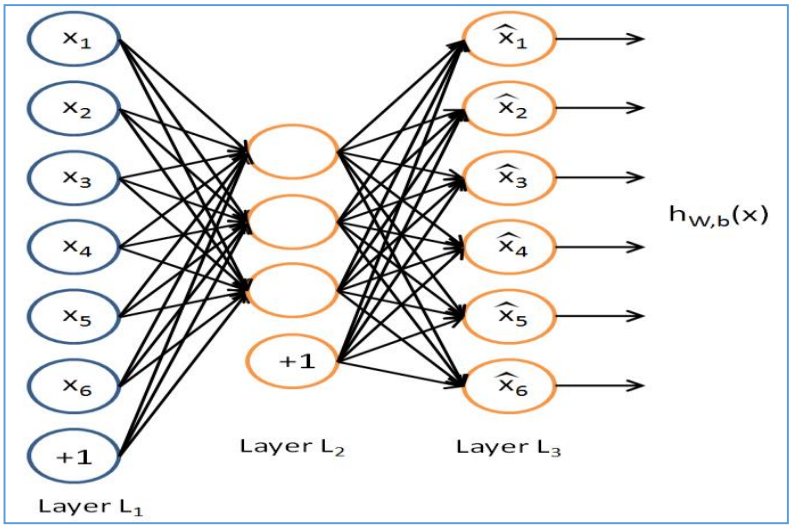


Figure 1.4 : Schéma d'un AE avec 6 neurones dans les couches d'entrée /sortie et trois neurones dans la couche cachée [Ji et al., 2023].

Les valeurs de la couche de sortie dans un AE sont définies pour être égales aux valeurs de la couche d'entrée (y = x). La fonction la plus simple de cette "transformation" est la fonction d'identité, f (x) = x. Par exemple, si nous avons une couche d'entrée avec 100 neurones, mais sur la couche intermédiaire, nous avons seulement 50 neurones, alors le RN devra compresser les données d'entrée, car il ne peut pas mémoriser 100 valeurs non compressées avec seulement 50 neurones. Si nos données d'entrée sont complètement aléatoires, cela n'est pas possible et il y aura des erreurs lors de la reproduction des données d'entrée. Si nos données ont une certaine structure, par exemple, elles représentent les pixels d'une image 10x10, nous pouvons obtenir une représentation plus compressée du pixel. Si un pixel dans une image est vert, les 8 pixels environnants sont également verts. S'il y a des pixels de la même couleur sur un cercle, au lieu de se rappeler la position de chaque pixel (pour un cercle de rayon 4, cela signifie environ 25 valeurs), il suffit de savoir qu'en position x, y on a un cercle de rayon r, qui peut être stocké avec seulement 3 valeurs. Bien sûr, un tel niveau de compression ne sera pas atteint en utilisant une seule couche.

Le processus d'entraînement pour un AE peut être divisé en deux étapes : la première étape consiste à apprendre les « *features* » en mode d'apprentissage non supervisé et le second consiste à affiner le réseau (*fine-tuning*) en mode d'apprentissage supervisé. Pour être précis, dans la première étape, La propagation feed-forward est d'abord effectuée pour chaque entrée pour obtenir la valeur de sortie x~. Ensuite, les erreurs au carré sont utilisées pour mesurer l'écart de x~ à partir de la valeur d'entrée. Enfin, L'erreur sera rétrogradée via le réseau pour mettre à jour les poids. En phase de réglage, avec le réseau possédant des caractéristiques appropriées dans chaque couche, nous pouvons adopter la méthode d'apprentissage supervisée standard et l'algorithme de descente en gradient pour ajuster les paramètres à chaque couche.

1.2.4 Réseau de neurone convolutionel (CNN)

Les réseaux neuronaux convolutifs CNN (Convolutional Neural Networks) (ou ConvNets) constituent une classe particulière de réseaux de neurones artificiels (ANN) largement utilisés pour l'analyse des images. Ils ont émergé en réponse au besoin de traiter des données organisées selon une topologie en grille, comme les pixels d'une image, et de capturer des caractéristiques locales pertinentes présentes dans ces données [de Oliveira, 2022].

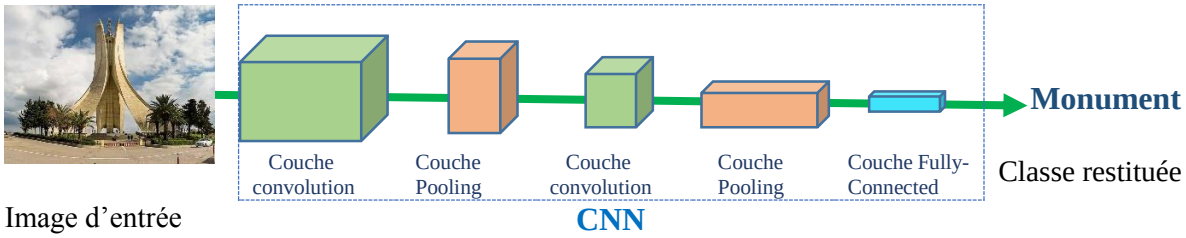


Figure 1.5 : Illustration d'un exemple d'un CNN comprenant plusieurs couches, conçu pour analyser une image et prédire correctement sa classe.

Les architectures profondes, notamment les CNN, sont généralement organisées en trois étapes principales [Berthet, 2022] :

1. **L'entrée** qui consiste généralement en une image brute, c'est-à-dire non traitée.

2. **L'apprentissage des caractéristiques** qui consiste à réduire la taille des images afin de faciliter le traitement, tout en préservant les informations essentielles nécessaires pour faire des prédictions précises. Cette étape inclut des couches convolutives et des couches de sous-échantillonnage, qui servent respectivement à extraire des caractéristiques de haut niveau (comme les bords, les textures, etc.) et à réduire la taille des caractéristiques extraites. Le nombre de couches dans une architecture CNN dépend essentiellement du compromis entre le temps de calcul et la précision. Les couches convolutives sont définies par des paramètres supplémentaires, qui doivent être spécifiés pour chaque couche *i*, tels que le noyau K*i*, le padding P*i* et le stride S*i* (voir Eq.1), où N*i* représente la taille de l'entrée de la couche), ainsi que les fonctions d'activation et de normalisation.

L'équation pour la taille de la couche suivante est donnée par [Berthet, 2022] :

$$N_{I+1} = 1 + \frac{(N_i + 2P_i - k_i)}{S_i} \qquad (1)$$

Filtre :

Le noyau représente le filtre (par exemple, de taille 3x3, Comme illustré dans les Figures 1.6 et 1.7 tirées de [Simard, 2021] qui se déplace sur l'ensemble de l'image d'entrée pendant l'opération de convolution. Dans un réseau de neurones convolutif (CNN), les filtres sont considérés comme des matrices de poids utilisées pour extraire des motifs spatiaux des images, tels que les arêtes et les lignes, en identifiant les variations des valeurs d'intensité dans l'image d'entrée. En général, les dimensions spatiales d'un filtre sont plus petites que celles de l'image d'entrée. Les filtres sont mis à jour en effectuant des opérations de convolution complète sur les cartes de caractéristiques entre une couche convolutionnelle et la couche qui la précède [Saideni, 2022].

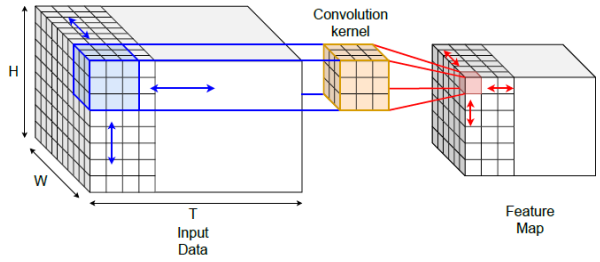


Figure 1.6 : Représentation visuel en 3D de l'opération de convolution.

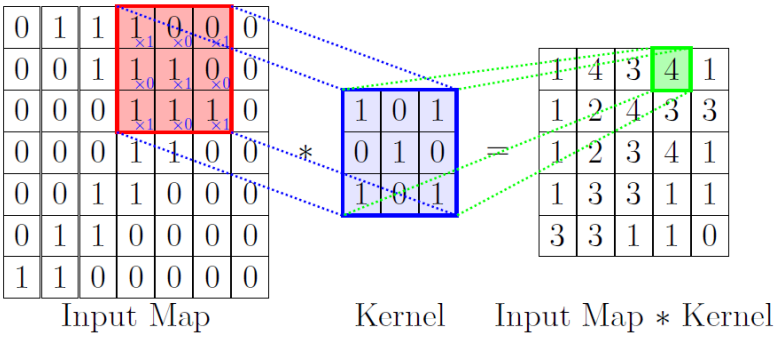


Figure 1.7 : Illustration de l'opération de la convolution

Padding :

Dans une image, le nombre de pixels ajoutés lors de l'extraction de caractéristiques par un filtre d'un modèle CNN est appelé « remplissage »(ou padding). Techniquement, cela consiste à ajouter des couches de zéros autour du cadre de l'image pour permettre une extraction efficace des caractéristiques des pixels situés aux coins et sur les bords, car ces pixels sont généralement moins exploités que ceux situés au centre (comme montré en Figure 1.8 tirée de [Berthet, 2022]). Ainsi, le remplissage permet d'éviter la réduction de la taille de l'image d'entrée résultant des opérations de convolution [Saideni, 2022].

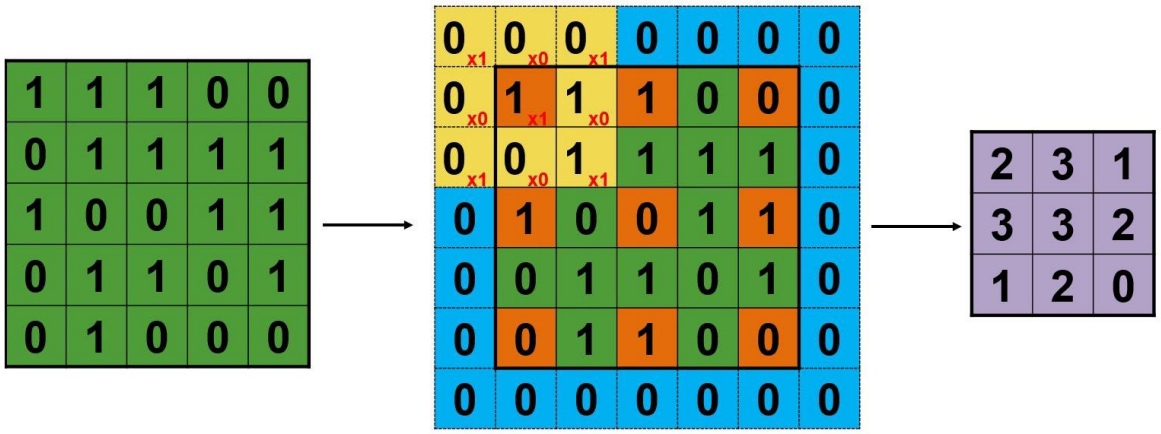


Figure 1.8 : Illustration du processus de convolution appliqué à une image.
À gauche: Image d'entrée de dimensions 5x5.
Au centre: Fenêtre du noyau (en jaune) se déplaçant sur l'image d'entrée,
avec le padding ajouté (en bleu) et le pas de déplacement défini par le stride (en orange).
À droite: Image de sortie obtenue après l'application de la convolution.

L'activation (par exemple, Rectified Linear Unit (ReLU), Tangente Hyperbolique (TanH), etc.) et les fonctions de normalisation (par lot ou réponse locale) fonctionnent en complément : la première doit produire une sortie non linéaire, tandis que la seconde contrôle les valeurs non bornées. Pour les couches de pooling, on retrouve également des paramètres tels que le noyau et le stride, mais le plus important est la fonction utilisée. En effet, ce type de couche réduit la taille des caractéristiques convolutives à l'aide d'un filtre, similaire aux convolutions. Cependant, dans ce cas, toute la fenêtre est remplacée par une valeur déterminée par la fonction de pooling (maximale, minimale ou moyenne).

Le Pooling :

Le « pooling » est une opération réalisée pour réduire le nombre de paramètres dans le réseau (sous-échantillonnage ou downsampling) et éviter le surapprentissage. Cette opération n'affecte pas la dimension de profondeur. Elle est utilisée pour généraliser les caractéristiques extraites par les filtres et permettre au modèle d'identifier des caractéristiques indépendamment de leur position dans l'image. Le type de pooling le plus couramment utilisé est une couche de filtres 2 × 2 appliquant la fonction max (max pooling), comme illustré dans la Figure 1.9, car il permet une convergence plus rapide et améliore les capacités de généralisation du réseau. Par ailleurs, d'autres fonctions de pooling peuvent également être utilisées, telles que le pooling moyen (average pooling) et le pooling minimum (min pooling) [Saideni, 2022].

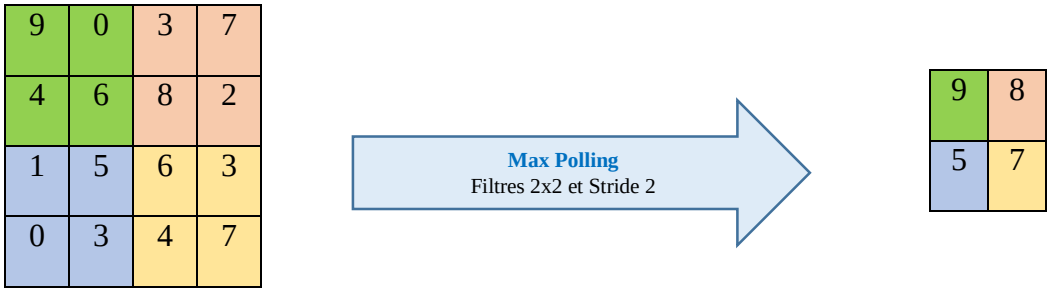


Image Originale Image Après le Max Pooling

Figure 1.9 : Illustration du processus de convolution appliqué à une image.

3. **La classification** consiste à attribuer une étiquette à l'image d'entrée, correspondant à la classe à laquelle elle appartient (par exemple, un chat pour une classification animale ou masculin pour une classification de genre). La sortie finale est générée par une couche entièrement connectée (Fully-Connected, FC), qui relie tous les neurones entre eux. La couche entièrement connectée (Fully Connected Layer) prend en entrée un volume (qui correspond à la sortie de la couche de convolution, ReLU ou pooling qui la précède) et produit un vecteur de dimension N, où N représente le nombre de classes que le système doit distinguer [Bakir, 2020].

Dans un premier temps, les cartes de caractéristiques sont aplaties (les données en 3D issues de l'étape d'apprentissage des caractéristiques sont transformées en un vecteur 1D). Ensuite, ce vecteur est réduit à une taille N, où N représente le nombre de classes, grâce à des couches FC successives. Cette séquence est souvent associée à une fonction de régularisation appelée dropout, qui supprime aléatoirement et temporairement certains neurones (leurs valeurs sont mises à zéro) afin de limiter le surapprentissage pendant l'entraînement.

Enfin, une fonction d'activation appelée Softmax est appliquée pour normaliser les valeurs : elle attribue 1 à la meilleure classe et 0 aux autres [Berthet, 2022].

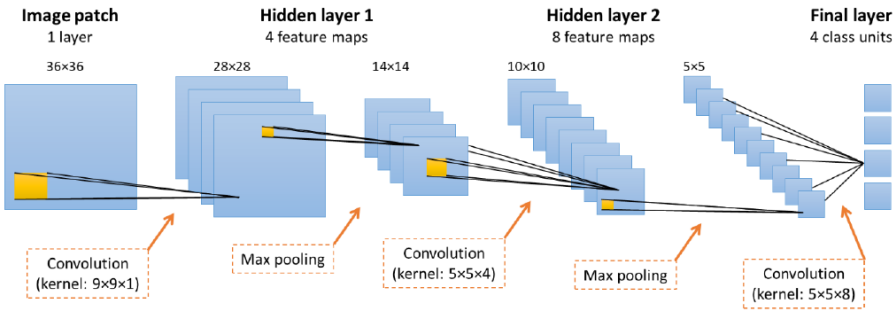


Figure 1.10 : la principale architecture de CNN.

L'architecture des réseaux de neurones convolutifs (CNN) comme il est présenté dans la figure 1.10 [Saideni, 2022], est principalement utilisée en traitement d'images, mais elle trouve également des applications dans le traitement du langage naturel [Zhang et Wallace 2016] et l'analyse des séries temporelles [Fawaz et al., 2019]. En raison de la grande ressemblance entre les images et les données spatiales mobiles (comme les instantanés du trafic mobile ou les schémas de mobilité des utilisateurs), les modèles basés sur les CNN offrent un potentiel considérable pour l'analyse des données mobiles à l'échelle des réseaux [Meslet-Millet, 2023].

1.2.5 Boltzmann Machines et Deep Belief Networks (DBN)

Le (RBM) Restricted Boltzmann Machine proposé comme concept par (Smolensky) est devenu prometteur grâce à la publication de (Hinton) en 2006 [Hinton, 2006]. Les RBM constitue une variante des machines Boltzmann (BM). Les BM peuvent être interprétés comme des Neurones Networks avec des unités de traitement stochastiques connectées bi-directionnellement. Les RBM ont été utilisés pour générer des modèles stochastiques des ANNs qui peuvent apprendre la distribution de probabilité à partir des données en entrée. Comme il est difficile d'apprendre des aspects d'une distribution de probabilité inconnue, les RBM ont été proposés pour simplifier la topologie du réseau et améliorer l'efficacité du modèle [Liu et al., 2017].

En 2006, (Hinton) a construit les DBN en empilant plusieurs RBM. Plus précisément, les DBN sont composés de plusieurs couches de variables stochastiques et latentes et peuvent être considérées comme une forme spéciale du modèle génératif probabiliste Bayésien.

Par rapport aux ANNs, les DBN sont plus efficaces, surtout lorsque ils sont appliqué dans un contexte non supervisé. [Liu et al., 2017] On peut voir la structure comme suit [Tau92]:

- Chaque deux couche adjacente forment un RBM. La couche visible de chaque RBM est connectée à la Couche cachée de RBM précédant et les deux couches en haut ne sont pas directionnelles.
- La connexion est dirigée entre la couche supérieure et la couche inférieure de haut en bas.
- Les différentes couches des RBM sont formées séquentiellement.
- Les RBM les plus bas sont formés d'abord, puis les plus élevés. Une fois les caractéristiques extraites par le RBM supérieur, elles seront propagées en arrière aux couches inférieures.
- Par rapport à un seul RBM, le modèle empilé augmentera la limite supérieure de la probabilité logarithmique, ce qui implique une meilleure capacité d'apprentissage.

Le processus d'entraînement d'un DBN comprend un apprentissage non supervisé réalisé dans la phase « *pre-training* » avec l'algorithme *« Greedy wise layer by layer* » et un apprentissage supervisé dans la phase de « *Fine Tuning* ». Le « *pre-training* » peut être considéré comme un apprentissage des *features* qui nous permettent d'avoir une meilleure initialisation des poids et l'algorithme supervisé de *Up-Down* utilisé en *Fine tuning* permet d'ajuster tout le réseau. Il est

digne de mentionner qu'avec les DBN, les données non étiquetées sont traitées efficacement [Liu et al., 2017].

1.2.6 Réseaux de Neurones Spiking (SNN)

Les réseaux neuronaux siamois (SNN) sont également divisés en deux parties : l'extraction des caractéristiques et la comparaison de la similarité. La partie extraction consiste en des sous-réseaux, qui sont des modèles identiques avec des poids partagés, permettant d'extraire des caractéristiques similaires à partir des données d'entrée. Ensuite, les résultats de ces sous-réseaux sont comparés pour effectuer une classification binaire. Par exemple, [Mazumdar, 2018] utilisent une couche de distance pour comparer les caractéristiques et classer les images en tant que paires identiques ou différentes. [Mandelli et al., 2020] ont proposé une autre méthode de réseau de similarité pour la classification de deux images. [Rao, 2020] adoptent une approche particulière en utilisant un SNN comme descripteur local, qui est utilisé parallèlement avec un CNN classique pour l'extraction des caractéristiques. Cette méthode est similaire au deuxième type de TSN, à savoir le réseau à deux domaines [Berthet, 2022].

1.2.7 Réseaux Antagonistes Génératifs (GAN)

Un GAN (Generative Adversarial Network) est une technique d'intelligence artificielle qui permet de générer des imitations réalistes d'images ou d'autres types de données. Ce modèle est composé de deux réseaux (générateur et discriminateur) qui s'opposent pour créer des données réalistes. La figure 1.11 [Bakir, 2020] montre les composants de GAN.

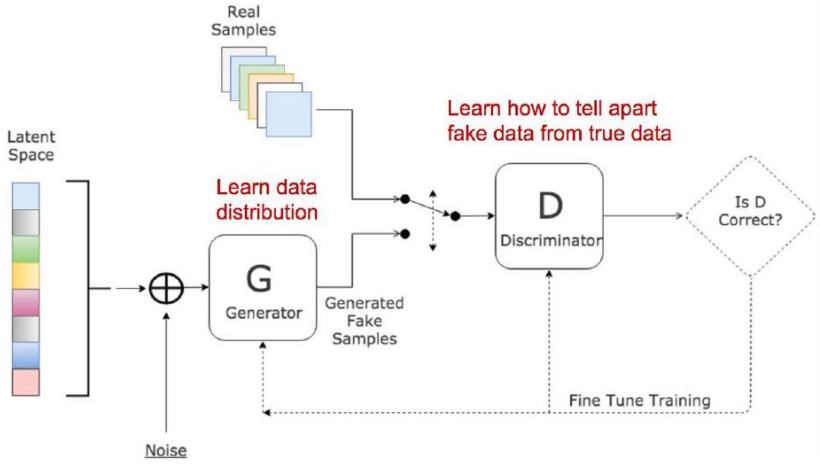


Figure 1.11 : Architecture de GAN.

Ces deux réseaux, appelés « générateur » et « discriminateur », jouent des rôles distincts mais complémentaires. Le générateur, basé sur un réseau neuronal convolutionnel, a pour mission de créer de nouvelles instances d'un objet. Le discriminateur, quant à lui, est un réseau neuronal spécialisé dans la détection, chargé d'évaluer si un objet généré est authentique ou s'il appartient à l'ensemble de données d'origine. Durant le processus d'apprentissage, ces deux réseaux s'affrontent dans un mécanisme compétitif, ce qui leur permet de perfectionner leurs

performances respectives. Ce processus repose sur la technique de la rétropropagation [Bakir, 2020].

1.3 Conclusion

Le deep learning s'est imposé comme une révolution dans le domaine de l'intelligence artificielle, redéfinissant les approches traditionnelles en matière de traitement des données. Grâce à ses architectures inspirées des réseaux neuronaux profonds, il permet d'extraire automatiquement des caractéristiques complexes à partir de données brutes, ouvrant ainsi la voie à des performances inédites dans divers domaines.

Ce chapitre a mis en évidence les fondements du deep learning, ses principales architectures, ainsi que ses applications variées. Parmi ces applications, on trouve la reconnaissance vocale, la vision par ordinateur, la traduction automatique, et bien d'autres, démontrant la polyvalence et l'efficacité de ces modèles. En exploitant de grandes quantités de données et des capacités de calcul avancées, le deep learning a transformé des secteurs comme la santé, les transports, et la finance, en apportant des solutions innovantes à des problèmes complexes.

En conclusion, le deep learning représente un pilier majeur de l'intelligence artificielle moderne, avec un potentiel immense pour transformer notre manière de résoudre les problèmes et d'interagir avec les données. Son développement continu laisse entrevoir un avenir encore plus riche en innovations et en opportunités.

Ce chapitre a mis en lumière l'importance croissante du deep learning dans le domaine de l'imagerie et de la compression des images. Grâce à des architectures avancées telles que les réseaux de neurones convolutifs (CNN) et les autoencodeurs, le deep learning a révolutionné la manière dont les images sont analysées, manipulées et compressées. En imagerie, le deep learning permet d'extraire automatiquement des caractéristiques complexes et pertinentes des données visuelles, rendant possibles des avancées majeures dans des applications telles que la reconnaissance d'objets, la détection de motifs, et le diagnostic médical assisté par intelligence artificielle. Ces technologies contribuent à des performances nettement supérieures par rapport aux approches traditionnelles, avec une précision accrue et une robustesse face aux variations dans les données. Dans le domaine de la compression d'images, le deep learning a introduit des solutions innovantes qui réduisent considérablement la taille des fichiers tout en préservant leur qualité visuelle. En exploitant des modèles comme les réseaux adverses génératifs (GAN) ou les autoencodeurs variationnels (VAE), il est désormais possible de compresser des images de manière plus efficace tout en minimisant les pertes d'information. Ces avancées sont particulièrement cruciales pour le stockage et la transmission de données dans des environnements où les ressources sont limitées. Ainsi, le deep learning s'impose comme un outil incontournable, transformant en profondeur le domaine de l'imagerie et ouvrant de nouvelles perspectives pour des applications toujours plus performantes et innovantes.

Chapitre 2 :

Compression JPEG et Compression Sémantique des images

2.1 Introduction

Ces dernières années, la compression d'images s'est imposée comme un sujet majeur de recherche, attirant l'attention des scientifiques et des ingénieurs dans divers domaines. Avec l'explosion des volumes de données visuelles générées quotidiennement par des applications telles que les réseaux sociaux, le streaming vidéo, les systèmes de surveillance et l'imagerie médicale, la nécessité de réduire efficacement la taille des fichiers tout en maintenant une qualité visuelle acceptable est devenue cruciale. Ce besoin a conduit à une prolifération de travaux de recherche, à tel point que le nombre de publications dans ce domaine ne cesse de croître.

Cette dynamique a favorisé l'émergence de nouvelles méthodes et algorithmes, dont plusieurs ont été intégrés dans des standards de compression largement utilisés, tels que JPEG, HEVC, PNG, ou encore des formats récents comme JPEG XL et AV1. Ces technologies combinent des approches traditionnelles, basées sur des transformées mathématiques, avec des innovations issues de l'intelligence artificielle et de l'apprentissage profond, pour repousser les limites des performances de compression.

Les méthodes de compression des images sont généralement classées en deux catégories principales, la figure 2.1 montre les deux types de compression des images :

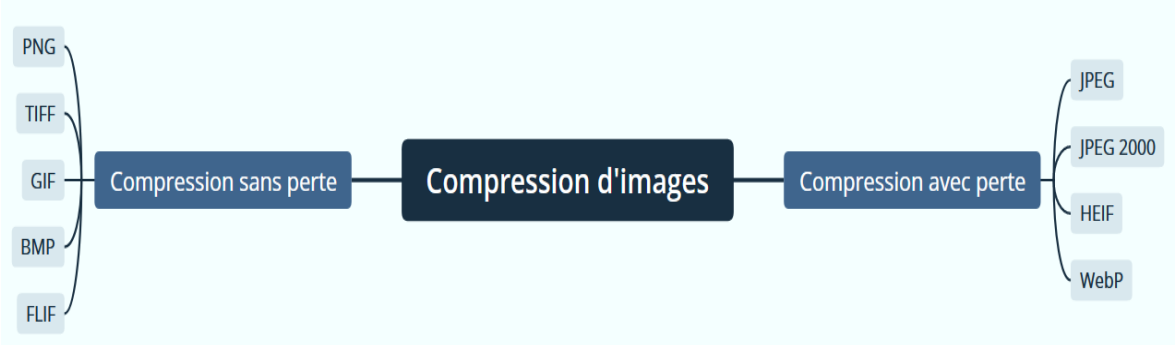


Figure2.1 : les types de compression des images

1. **Compression sans perte (lossless)** : Dans le cas de la compression réversible, les niveaux de gris de l'image compressée restent strictement identiques à ceux de l'image originale, sans subir aucune altération. Ce mode de compression est particulièrement adapté aux applications sensibles, comme la médecine, où toute erreur, même minime, est inacceptable. L'avantage principal de cette approche est de garantir une reconstruction parfaitement fidèle de l'image originale. Cependant, son principal inconvénient réside dans le faible taux de compression qu'elle permet d'atteindre [Voinson, 2003]. Ces méthodes préservent l'intégrité des données de l'image d'origine, permettant une reconstruction exacte après décompression. Cependant, elles offrent des taux de compression généralement plus faibles. Des exemples incluent des formats comme PNG, TIF et GIF.

2. **Compression avec perte (lossy)** : La compression avec perte, ou non réversible, introduit des modifications aux valeurs de l'image lors de la phase de quantification. Cette approche offre l'avantage de permettre des taux de compression élevés, mais cela se fait au prix d'une dégradation de la qualité de l'image reconstruite. Malgré cette limitation, la plupart des applications grand public privilégient ce type de compression, notamment pour les appareils photo numériques (traitement d'images naturelles), la transmission d'images sur divers réseaux ou encore le stockage sur CD-Rom [Voinson, 2003]. Ces méthodes permettent une réduction plus importante de la taille des fichiers en sacrifiant une partie de l'information visuelle, souvent imperceptible pour l'œil humain. Le JPEG est un exemple typique de compression avec perte. Il est largement utilisé pour son équilibre entre réduction de taille et qualité visuelle acceptable [Hamano et al., 2023].

Ce chapitre est consacré à la présentation des concepts fondamentaux de la compression avec perte des images. Il offre une vue d'ensemble des principes théoriques qui sous-tendent la réduction de redondance et la préservation de l'information utile, tout en explorant les techniques modernes et les standards qui définissent l'état de l'art. En outre, une attention particulière sera portée aux défis actuels, comme le compromis entre qualité perçue et taux de compression. Précisément dans ce chapitre, nous présenterons en détail l'algorithme de base de la norme JPEG. Nous examinerons les étapes fondamentales du processus, notamment la segmentation en blocs, l'application de la DCT, la quantification des coefficients, et les techniques de codage utilisées pour compresser les données. Nous discuterons également des avantages et des limites de cette norme, ainsi que de son impact sur l'évolution des technologies de compression. Cette exploration fournira une compréhension approfondie du rôle de la norme JPEG dans les solutions modernes de traitement et de transmission d'images

2.2 Joint Photographic Experts Group (JPEG)

La norme JPEG (Joint Photographic Experts Group), introduite en 1992, a été la première norme dédiée à la compression d'images. La norme JPEG représente le premier standard international conçu pour la compression des images fixes en tons continus, applicable aussi bien aux images monochromes qu'aux images en couleur [Liu, 2015]. Ces dernières années, le groupe Joint Photographic Experts Group (JPEG) a joué un rôle majeur dans le développement d'un standard international pour la compression des images numériques. Cet effort visait à répondre à une demande croissante de solutions universelles pour réduire la taille des fichiers d'image tout en conservant une qualité suffisante pour divers usages. L'objectif principal était de concevoir une norme capable de satisfaire les besoins variés d'applications telles que l'astronomie, l'imagerie médicale, le journalisme, ou encore les usages militaires, en couvrant un large éventail d'exigences liées aux images couleur et en niveaux de gris.

Le standard devait également faciliter la transmission et l'échange des données visuelles entre différentes plateformes et organisations. Pour répondre à ces exigences, plusieurs propositions de méthodes de compression ont été rigoureusement étudiées par le groupe. Parmi celles-ci, la transformée en cosinus discrète (DCT) appliquée sur des blocs 8×8 a émergé comme la solution la plus efficace. Elle a été adoptée pour sa capacité à fournir un bon compromis entre taux de compression, qualité visuelle et complexité de calcul. Le choix de la DCT 8×8 repose sur plusieurs facteurs. Cette méthode exploite les propriétés des images naturelles, qui présentent souvent des corrélations spatiales fortes, pour concentrer l'énergie de l'image dans un petit

nombre de coefficients. En transformant les blocs d'image en domaine fréquentiel, elle permet de réduire les informations redondantes et d'éliminer celles qui sont moins perceptibles par l'œil humain, grâce à des processus comme la quantification et le codage entropique. La figure 2.2 [Ma et al., 2019] exprime en générale les principales étapes de compression JPEG.

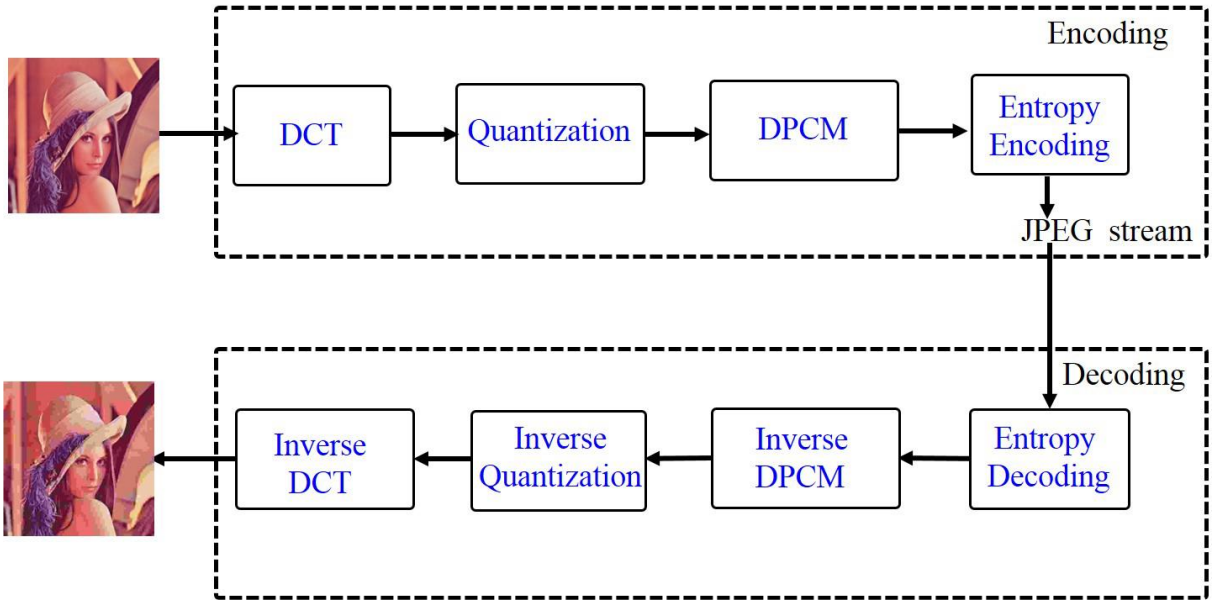


Figure 2.2 : Algorithme JPEG.

Comme nous le verrons plus en détail dans les sections suivantes, ses principales étapes incluent le sous-échantillonnage chromatique, la transformation discrète en cosinus (DCT) par blocs, suivie de la quantification des coefficients...etc.

2.2.1 Les différentes étapes de JPEG

2.2.1.1 DCT

La transformée DCT (Discrete Cosine Transform, en français transformée en cosinus discrète), est l'une des transformées les plus répandues en compression d'images car elle est exploitée dans ce domaine depuis les années 80. Elle est surtout employée en compression avec pertes, et notamment par le standard JPEG adopté en 1992. Cette dernière a su montrer ses faiblesses au fil du temps. La DCT est une transformation linéaire permettant de disproportionner certains coefficients transformés de telle sorte que leur abandon n'entraîne pas de distorsion significative après reconstruction. De plus, la DCT génère des coefficients réels, les plus petits de ceux-ci étant localisés dans une zone fréquentielle où l'œil a une acuité faible. Le calcul de la DCT ne peut pas se faire sur une image entière d'une part parce que cela générerait trop de calculs et d'autre part parce que le signal de l'image doit absolument être représenté par une matrice carrée. Dès lors, le groupe JPEG impose que L'image initiale est découpée en blocs de 8 × 8 pixels Le tableau 2.1 présente les valeurs de pixels d'un bloc de *8 x 8* pixels à *256* niveaux de gris.

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	156	156	156	156
159	161	162	160	159	159	159	159
159	160	161	162	155	155	155	155
161	161	161	161	157	157	157	157
162	162	161	163	157	157	157	157
162	162	161	161	158	158	158	158

Tableau 2.1 : Exemple de Bloc 8*8

Ce découpage permet de réduire le temps de calcul et facilite les étapes de traitement qui suivent. La taille des blocs est toujours la même, telle que définie dans la norme.

La méthode de compression sera donc appliquée indépendamment sur chacun des blocs. Les plus petits blocs en bordure devront être traités par une autre méthode (par exemple accomplir le reste des pixels par des zéros).

La DCT est donc effectuée sur chaque bloc (matrice 8x8) de valeurs de pixels, et elle donne une matrice 8x8 de coefficients de fréquence, l'élément (0,0) représente la valeur moyenne du bloc, les autres indiquent la puissance spectrale pour chaque fréquence spatiale. La DCT est Conservative si l'on ne tient pas compte des erreurs d'arrondis qu'elle introduit. Lorsqu'on travaille avec le signal Img(x,y), les axes X et Y représentent les dimensions horizontales et verticales de l'image. Lorsqu'on travaille avec la transformée de cosinus discrète du signal DCT(i,j), les axes représentent les fréquences du signal en deux dimensions [Sahir, 2018].

La transformée DCT s'exprime mathématiquement par la formule (1) [Sahir, 2018] :

$$F(u,v) = \left(\frac{1}{4}\right).c(u).c(v)\sum_{i=0}^{7}\sum_{j=0}^{7} f(i,j).\cos\left((2.i+1).u.\frac{\pi}{16}\right).\cos\left((2.j+1).v.\frac{\pi}{16}\right) \quad (1)$$

Et la transformée DCT inverse s'exprime par :

$$f(i,j) = \left(\frac{1}{4}\right)\sum_{i=0}^{7}\sum_{j=0}^{7} c(u).c(v)F(u,v).\cos\left((2.i+1).u.\frac{\pi}{16}\right).\cos\left((2.j+1).v.\frac{\pi}{16}\right) \quad (2)$$

Avec : $\begin{cases} c(u) = c(v) = \frac{1}{\sqrt{2}} & pour\ u = v = 0 \\ c(u) = c(v) = 1 & pour\ u \neq 0, v \neq 0 \end{cases}$

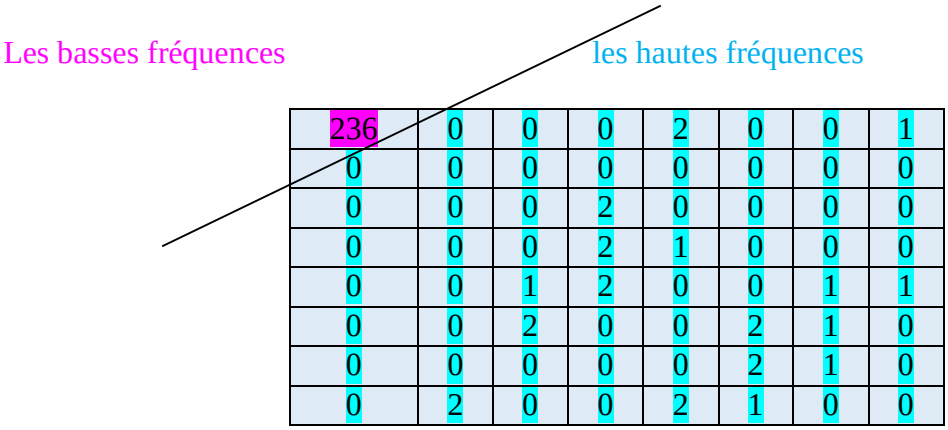
- $f(i,j)$: représente la valeur du pixel de coordonnées (i, j) dans le bloc de l'image originale de 8× 8 pixels.
- $F(u,v)$: représente la valeur de la DCT au point de coordonnées (u, v) dans le bloc résultat de 8 × 8 pixels.

Ensuite la Transformée en Cosinus Discrète (DCT) est appliqué sur les pixels de chaque bloc.
Le tableau 2.2 présente le calcul DCT du bloc illustré dans le tableau 2.1.

236	-1	-12	-5	2	-2	-3	1
-23	-18	-6	-3	-3	0	0	-1
-11	-9	-2	2	0	-1	-1	0
-7	-2	0	2	1	0	0	0
-1	-1	1	2	0	-1	1	1
2	0	2	0	-1	2	1	-1
-1	0	0	-2	0	2	1	-1
-3	2	-4	-2	2	1	-1	0

Tableau 2.2 : Matrice DCT d'un bloc.

Les valeurs de la matrice DCT ont été arrondies à l'entier le plus proche, le tableau 2.3 propose les valeurs arrondies.



236	0	0	0	2	0	0	1
0	0	0	0	0	0	0	0
0	0	0	2	0	0	0	0
0	0	0	2	1	0	0	0
0	0	1	2	0	0	1	1
0	0	2	0	0	2	1	0
0	0	0	0	0	2	1	0
0	2	0	0	2	1	0	0

Tableau 2.3 : Matrice DCT arrondie.

Le coefficient de l'élément$F(0,0) = 236$, appelé coefficient continu (DC), possède la valeur la plus élevée qui est proportionnelle à l'intensité moyenne du bloc de 64 pixels. Les autres valeurs de la DCT représentent des ``écarts" par rapport à cette moyenne. Les valeurs de la matrice sont appelés aussi les coefficients alternatifs (AC), correspondent aux variations d'un pixel à l'autres, ils représentent donc la contribution des différentes fréquences non nulles dans le Spectre ainsi obtenu. Typiquement, les valeurs des pixels varient lentement de point au point adjacent à travers une image, en particulier pour des images qui ne présentent pas de discontinuités brutales

Les coefficients d'indices $(0, j)$ (respectivement$(i, 0)$) sont les composantes continues le long de l'axe Y (resp.X) pour la fréquence j (resp. i) le long de l'axe X (resp. Y). On remarque une tendance générale des valeurs de la matrice à s'approcher de 0 lorsqu'on s'éloigne du coin supérieur gauche, c'est-à-dire lorsqu'on monte dans les plus hautes fréquences. Cela traduit le fait que l'information effective de l'image est concentrée dans les basses fréquences. C'est le cas de la majorité des images. On d'autre terme les coefficients des basses fréquences, correspondant à des plages spatiales uniformes, ont des valeurs plus grandes que les coefficients des hautes fréquences. L'énergie du signal se trouve donc concentrée sur les fréquences spatiales inférieures, alors que la majorité des autres fréquences sont nulles ou presque.

2.2.1.2 Quantification

La matrice obtenue après transformation nécessite autant, voire plus, d'espace de stockage que la matrice de pixels d'entrée. Il devient donc nécessaire d'introduire une nouvelle opération visant à réduire les valeurs de la matrice transformée. Cette opération, connue sous le nom de quantification, permet d'y parvenir [Makkaoui, 2012]. Cette opération a pour objectif de supprimer une partie des hautes fréquences dans les blocs DCT, en exploitant la distinction entre basses et hautes fréquences. La quantification s'effectue à l'aide de tables de quantification, pouvant aller jusqu'à deux, définies dans les en-têtes des fichiers JPEG conformément à la norme. Une table de quantification, sous la forme d'une matrice de dimensions (8 *8), est utilisée pour réaliser l'opération de quantification de la formule (3) [Der22].

$$Q(i,j) = 1 + (1 + i + j)FQ \qquad (3)$$

Où i l'indice de ligne, j l'indice de colonne et FQ Représente le facteur de qualité de la matrice de quantification (choisi entre 1 et 100). Plus le facteur de qualité FQ est grand, plus la compression sera forte mais moins bonne sera la qualité de l'image reconstituée : FQ détermine le compromis à faire entre le gain en compression et la qualité de l'image reconstituée par rapport à l'image d'origine.
L'objectif de cette étape est d'exploiter les caractéristiques du système visuel humain afin d'optimiser le taux de compression en supprimant les informations auxquelles l'œil humain est moins sensible. En effet, la vision humaine peut être assimilée à un filtre passe-bas, étant moins réceptive aux variations de haute fréquence. Pour accomplir cette tâche, les coefficients DCT sont quantifiés en fonction de tables de quantification adaptées à la sensibilité psycho visuelle [Haddad, 2020].

Exemple :

Supposons que FQ=10
On va calculer la matrice de quantification
Pour $i = 1$ et $j = 1$, $Q(i,j) = 1 + (1 + i + j)FQ \;\; = \;\; 1 + (1 + 1 + 1) * 10 = \;\; 31$
Pour $i = 1$ et $j = 2$, $Q(1\,,2) = \;\; 1 + (1 + 1 + 2) * 10 = \;\; 41$
…etc. jusqu' à $Q(8{,}8)$. Le tableau 2.4 démontre la matrice de quantification pour un facteur de qualité égale à 10.

31	41	51	61	71	81	91	101
41	51	61	71	81	91	101	111
51	61	71	81	91	101	111	121
61	71	81	91	101	111	121	131
71	81	91	101	111	121	131	141
81	91	101	111	121	131	141	151
91	101	111	121	131	141	151	161
101	111	121	131	141	151	161	171

Tableau 2.4 : Matrice de Quantification (FQ=10).

2.2.1.3 Linéarisation des blocs

Après la quantification et avant le codage entropique (RLE ou HUFFMAN), la lecture des blocs ce fait d'une manière en Zig-Zag (balayage depuis les basses fréquences jusqu' aux hautes fréquences) afin d'obtenir un maximum de zéros à la fin de la séquence. Les coefficients AC sont réorganisés dans un vecteur 1D selon un schéma de balayage en zigzag. Cela transforme la structure bidimensionnelle en une séquence linéaire, tout en regroupant les coefficients proches en termes de fréquence spatiale.

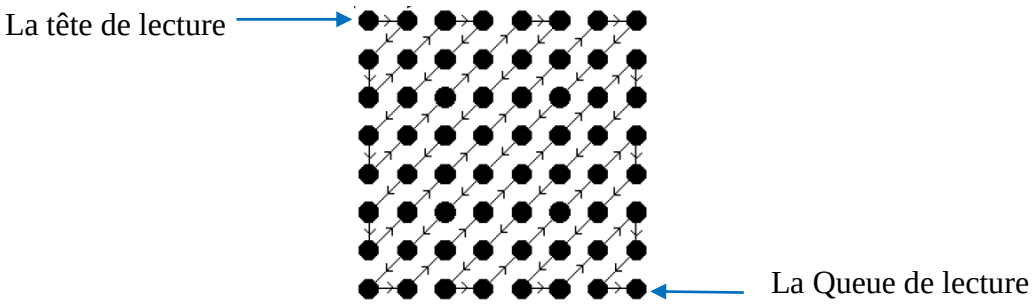


Figure 2.3 : Balayage en Séquence Zig-Zag.

2.2.1.4 Codage entropique dans le format JPEG

Après l'étape de quantification, les coefficients DC (composante continue) et AC (composante alternative) sont traités différemment pour optimiser la compression des données.

- **Codage RLE (Run-Length Encoding)** : Cette séquence contient souvent de longues séries de zéros consécutifs, conséquence de la quantification et du balayage en zigzag. Le codage RLE représente cette séquence sous forme de paires `(run, value)`, où *run* correspond au nombre de zéros précédant un coefficient non nul, et *value* est la valeur de ce coefficient.

- **Codage de Huffman** : Les données transformées (DC après DPCM et AC après RLE) sont ensuite compressées à l'aide d'un codage de Huffman. Ce codage associe à chaque symbole une séquence de bits de longueur variable, proportionnelle à la fréquence d'apparition du symbole. Les symboles les plus fréquents reçoivent les codes les plus courts, optimisant ainsi l'efficacité de la compression.
- **Tables fixes** : Pour simplifier le calcul et accélérer le processus, la norme JPEG utilise des tables de Huffman prédéfinies (une pour la luminance et une pour la chrominance). Ces tables, qui représentent des approximations statistiques raisonnables pour la majorité des images, sont incluses dans l'en-tête du fichier JPEG [Haddad, 2020].

2.2.2 Principe de la Compression JPEG

La figure 2.4 illustre les différentes étapes du processus de compression JPEG, depuis la transformation de l'image d'origine jusqu'à l'obtention du fichier compressé. Elle met en évidence les opérations successives telles que la transformation en blocs, l'application de la transformation discrète en cosinus (DCT), la quantification, et enfin l'encodage entropique.

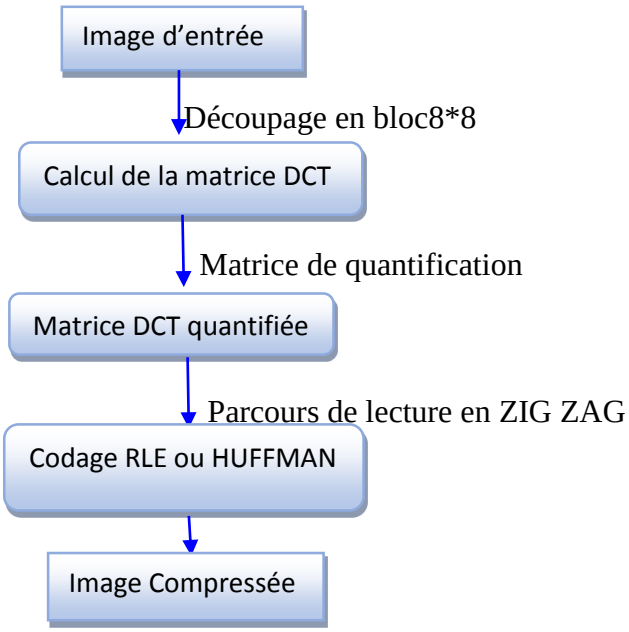


Figure 2.4 : Les étapes de la Compression JPEG

Pour reconstituer l'image décompressée, il faut suivre le processus inverse de la compression. Toutefois, l'image obtenue ne sera pas strictement identique à l'originale, car elle aura subi des distorsions durant l'étape de compression. Voir figure 2.5.

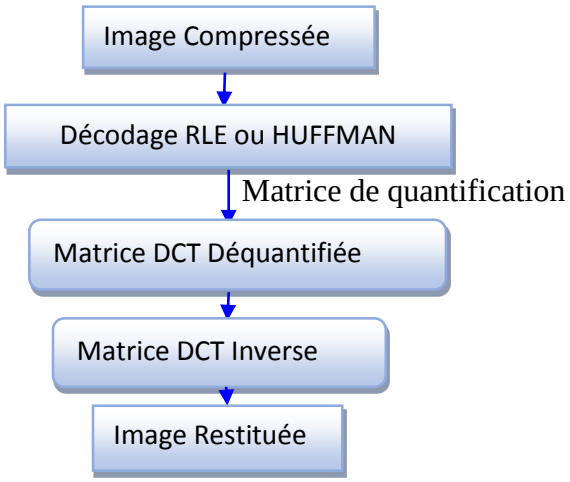


Figure 2.5 : Les étapes de la décompression JPEG

2.2.3 Algorithme de Compression JPEG

Avant d'aborder le fonctionnement général de la compression JPEG, il est essentiel de comprendre la première étape du processus : la conversion de l'image de l'espace colorimétrique RVB (Rouge, Vert, Bleu) vers l'espace YCbCr. Cette transformation permet de séparer les informations de luminance (Y) des informations de chrominance (Cb et Cr), ce qui est fondamental pour le JPEG, car elle facilite la réduction des données tout en conservant une bonne qualité visuelle.

Les images numériques sont généralement représentées sous une forme RGB, mais les composants de cette représentation manifestent une forte corrélation entre eux. Certains codeurs d'images (et de vidéo aussi) exige une transformée sur les couleurs RGB avant le codage réel pour représenter l'image dans un domaine qui diminue la corrélation entre les pixels. Cette transformée est appelée « transformée en YCbCr ». La transformée en YCbCr décompose les plans RGB en trois plans orthogonaux, notamment en Luminance (Y) et deux plans de chrominance (Cb & Cr). La Figure 2.6 [Zeybek, 2011] illustre l'image Lena, décomposée en YCbCr.

Figure 2.6 : Conversion d'une image RGB en YCBCR.

Le format JPEG, commence par traité séparément chaque composante de couleur. Il est possible d'utiliser directement le modèle RVB, mais JPEG est plus efficace sur le modèle YCbCr (après Conversion éventuelle en luminance / chrominance). Comme l'œil humain est moins sensible aux changements de teinte (chrominance) que changements de luminosité (luminance).

L'algorithme JPEG est basé sur cette différence de perception. D'ailleurs, il est même possible de sous – échantillonner ces composantes de chrominance, afin d'aboutir à un modèle prédéfini. Ensuite le format JPEG, commence par découper chaque composante YCbCr en blocs carrés de 64 (8 × 8) pixels (L'algorithme de compression est calculé pour chaque bloc séparé, ce qui explique pourquoi ces blocs ou groupes de blocs deviennent visibles lorsque la compression est trop appliquée). L'étape suivante dans le processus de compression consiste à appliquer une transformée en cosinus discrète (DCT) à chaque bloc 8x8 pixels, Cette méthode permet de décrire chaque bloc en une carte de fréquences et en amplitudes plutôt qu'en pixels et couleurs, C'est l'étape qui coûte le plus de temps et de ressources dans la compression et la décompression ,

mais la plus importante car elle nous a permis de séparer les basses fréquences et les hautes fréquences présentes dans l'image. L'étape DCT lui-même est sans perte. La quantification est l'étape dans laquelle on perd réellement des informations (et donc de la qualité visuelle), mais c'est celle qui fait gagner beaucoup de place. Le but est ici est de garder les basses fréquences, c'est-à-dire celles auxquelles l'œil humain est très sensible et d'atténuer et de vouloir annuler les hautes fréquences, c'est à dire celles auxquelles l'œil humain est très peu sensible. JPEG utilise des tables de quantification, pour la luminance et pour la chrominance. Une fois que ces tables ont été construites, les constantes des tables sont utilisées pour quantifier les coefficients DCT. Chaque coefficient DCT est divisé par sa constante correspondante dans la table de quantification et arrondie au plus proche entier. Le résultat de la quantification des coefficients DCT sont parcouru en zig zag avant d'appliquer le codeur entropique qui est ici, Le codeur de Huffman qui compresse sans perte. Pour pouvoir reconstruire l'image originale à partir des données compressées, il faut appliquer l'étape de décompression donc de décodage, cette étape s'obtient en appliquant les procèdes inverses de l'étape du codage [Sahir, 2018].

2.2.4 Exemple de Compresseur JPEG sur une image

La Figure 2.7 présente une image originale est taille 261 KB

Figure 2.7 : Exemple d'image de type PNG

La figure 2.8 montre le découpage de la figure 2.6 en bloc de 8*8

Figure 2.8 : Image découpée en bloc 8*8

Le bloc sélectionné en vert dans la figure 2.8 est présenté par les valeurs du tableau 2.5, et le tableau 2.7 montre la matrice de quantification avec FQ =5.

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	156	156	156	156
159	161	162	160	159	159	159	159
159	160	161	162	155	155	155	155
161	161	161	161	157	157	157	157
162	162	161	163	157	157	157	157
162	162	161	161	158	158	158	158

Tableau 2.5 : Exemple de Matrice d'un bloc.

Les tableaux 2.6 et 2.8 présentent respectivement le calcul de la matrice DCT et le calcul de la matrice DCT quantifiée de la matrice de tableau 2.5.

236	-1	-12	-5	2	-2	-3	1
-23	-18	-6	-3	-3	0	0	-1
-11	-9	-2	2	0	-1	-1	0
-7	-2	0	2	1	0	0	0
-1	-1	1	2	0	-1	1	1
2	0	2	0	-1	2	1	-1
-1	0	0	-2	0	2	1	-1
-3	2	-4	-2	2	1	-1	0

Tableau 2.6 : Exemple de Matrice DCT d'un bloc

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	3	56	68	109	103	77
24	35	55	64	51	14	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Tableau 2.7 : Exemple de Matrice quantification FQ= 5

15	0	-1	0	0	0	0	0
-1	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Tableau 2.8 : Exemple de Matrice DCT Quantifiée.

La figure 2.9 décrit le schéma de La lecture des valeurs en parcours ZIG ZAG

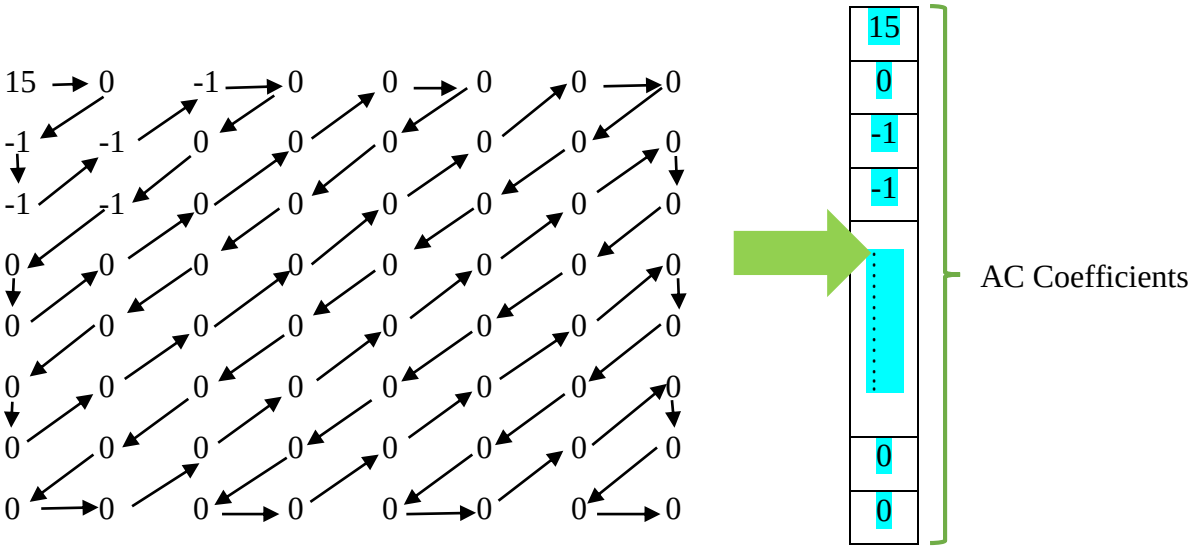


Figure 2.9 : Exemple de balyage Zig Zag et AC Coefficients.

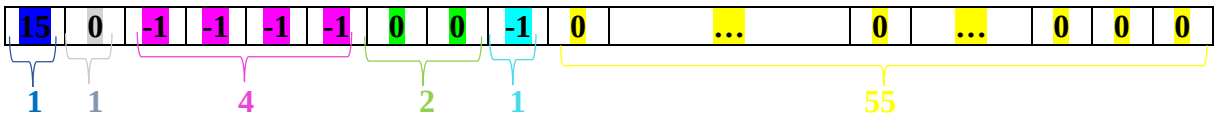


Figure 2.10 : Calcul des itérations des coefficients

La Figure 2.10 illustre la construction de La séquence RLE des AC coefficients de la Figure 2.8. La séquence RLE obtenue est la suivante **:** (1,15), (1,0), (4,-1), (2,0), (1,-1), (55,0).

Lors de la phase de décompression, les étapes inversement appliquées aux procédés utilisés durant la compression permettent de retrouver l'image originale (ou approximative). Cela implique la reconstruction des données en suivant la séquence inverse des opérations : déquantification des coefficients, application de la transformée DCT inverse et recomposition de l'image à partir des blocs traités. Cette inversion vise à restaurer au mieux l'image tout en minimisant les pertes de qualité causées par la compression.

Le tableau 2.9 représente la DCT Déquantifiée restituée après le Balayage ZIG ZAG inverse de la séquence de la figure 2.9.

15	0	-1	0	0	0	0	0
-1	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Tableau 2.9 : Exemple de Matrice DCT Déquantifiée

Le tableau 2.10 résume les valeurs de la multiplication de la matrice de tableau 2.7 et la matrice de tableau 2.9 pour l'obtention de la matrice DCT quantifiée.

240	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Tableau 2.10 : Exemple de Matrice DCT Quantifiée

Le tableau 2.11 illustre les résultats de calcul de la DCT inverse de la matrice de tableau 2.10.

144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	159	157	155
163	163	164	163	162	160	158	157
163	164	164	164	162	160	159	158
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158

Tableau 2.11 : Exemple de Matrice DCT inverse

La figure 2.11 présente l'image restituée après compression JPEG avec une taille égala à 9 KB

Figure 2.11 : Exemple d'image restituée.

2.2.5 Les Limites de JPEG

Le compresseur JPEG, bien qu'efficace pour la compression d'images, présente plusieurs limites. Tout d'abord, il s'agit d'une compression avec distorsion, ce qui entraîne une dégradation de la qualité, particulièrement visible dans les zones à fort contraste et les détails fins. La Figure 2.12 montre la perte des informations De plus, l'effet de blocage dû à la transformation en blocs de 8x8 pixels peut provoquer des artefacts visibles, surtout à des taux de compression élevés.

(A) Image originale 22.8 MO (B) JPEG (159 Ko, PSNR= 22)

Figure 2.12 : Perte des informations.

Un autre inconvénient est la sensibilité aux recompressions successives, qui entraînent une perte cumulative d'informations. JPEG ne prend pas non plus en charge la transparence et n'est pas optimal pour les images avec du texte ou des contours nets, où des formats comme PNG ou WebP sont préférables. Ces limites soulignent l'importance de la **compression sémantique**, qui vise à préserver les éléments les plus significatifs d'une image selon son contexte. Par exemple, dans une photographie de visage, une compression intelligente donnerait la priorité aux détails des yeux et de la bouche plutôt qu'à un fond uniforme. Contrairement à la compression traditionnelle qui traite toutes les zones de manière uniforme, une approche sémantique permettrait de minimiser la perte de qualité perçue tout en réduisant la taille du fichier. Cette évolution pourrait améliorer considérablement le stockage et la transmission des images, notamment dans des applications comme la vision par ordinateur et la transmission vidéo.

2.3 Compression Sémantique

La compression sémantique représente un changement de paradigme prometteur dans le domaine de la compression d'images à faible débit. Contrairement aux méthodes traditionnelles qui cherchent à minimiser la perte d'information au niveau des pixels en utilisant des métriques comme le PSNR ou la MSE, la compression sémantique repose sur la préservation du contenu sémantique d'une image, c'est-à-dire le sens ou les informations perceptuellement importantes qu'elle véhicule [Bachard, 2024].

Dans ce nouveau cadre, l'objectif n'est plus de reconstituer une image identique pixel par pixel, mais de produire une version qui reste fidèle sur le plan interprétatif, même si elle diffère visuellement de l'original. Autrement dit, deux images peuvent apparaître très différentes en termes de texture ou de couleurs, mais tant qu'elles transmettent le même message ou les mêmes objets reconnaissables, la compression est considérée comme réussie.

Ce basculement de perspective peut sembler à première vue contre-intuitif, surtout dans un domaine historiquement centré sur la fidélité visuelle. Toutefois, il ouvre des perspectives nouvelles et fascinantes : il permet de repenser les critères de qualité d'une image compressée, en les rapprochant davantage de la perception humaine et des usages finaux de l'image (reconnaissance d'objet, compréhension de scène, etc.). De plus, il soulève des questions fondamentales sur la manière dont nous mesurons la qualité d'une image et sur la pertinence des métriques actuelles face à des applications plus intelligentes ou orientées machine [Bachard, 2024].

La compression sémantique est une technique de réduction de données qui vise à optimiser le stockage ou la transmission en préservant principalement le sens ou la signification (sémantique) de l'information, plutôt que sa forme exacte. Contrairement à la compression traditionnelle, qui se concentre sur la réduction des redondances statistiques ou structurelles dans les données, la compression sémantique utilise des techniques intelligentes pour identifier et représenter uniquement les éléments essentiels ou significatifs pour une tâche spécifique.

(A) Image Originale (B) ROI de l'image (C) Non-ROI de l'image

Figure 2.13 : Concept de la Zone d'intérêt ROI.

La compression sémantique**,** basée sur le concept de **Region of Interest (ROI)**, la figure 2.13 présente la zone d'intérêt (ROI) d'une image d'un véhicule et la zone Non-ROI. Ce type de compression offre une approche plus intelligente et adaptative. Contrairement à la compression standard qui traite toutes les parties de l'image de manière uniforme, la compression basée sur les Régions d'Intérêt identifie et préserve les zones contenant des informations essentielles, tout en appliquant une compression plus forte aux régions moins importantes. Par exemple, dans une photographie de visage, cette approche donnerait la priorité aux détails des yeux, de la bouche et des contours faciaux, tout en appliquant une compression plus élevée sur le fond ou les zones uniformes. Cette méthode est particulièrement utile dans des domaines comme la vision par ordinateur, la télémédecine, la surveillance vidéo et la transmission d'images en temps réel, où la préservation des détails critiques est essentielle. En combinant des techniques d'intelligence artificielle et d'analyse d'image.

L'extraction de la zone d'intérêt (ROI) repose principalement sur le processus de segmentation, qui consiste à délimiter les régions pertinentes d'une image en fonction de critères spécifiques (forme, couleur, texture, etc.). La figure 2.14 illustre un axe représentant les différentes tâches de reconnaissance d'image, classées de la moins précise à la plus précise. Cet axe met en évidence l'évolution des techniques d'analyse d'image, allant des méthodes globales et approximatives vers des approches de plus en plus fines, comme la segmentation sémantique, qui permettent une localisation et une identification précises des régions d'intérêt.

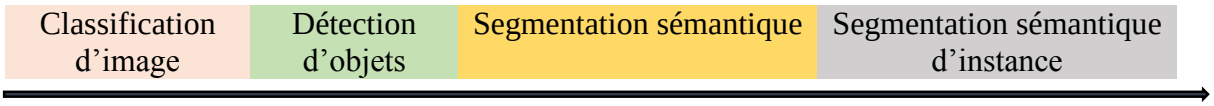


Figure 2.14 : Comparaison des différentes tâches de reconnaissance d'images.

La figure 2.15 présente le processus de classification d'image, qui consiste à attribuer une étiquette ou une catégorie à une image dans son ensemble. Quant à la figure 2.16, elle illustre la détection d'objets, une tâche plus avancée qui vise à localiser et identifier plusieurs objets présents dans une image à l'aide de cadres englobants (bounding boxes).

Figure 2.15 : Classification d'image.

Figure 2.16 : Segmentation par Instance (Détection d'objets).

La segmentation sémantique d'images est une tâche qui consiste à attribuer une classe sémantique (par exemple personne, arbre, route etc.) à chacun des pixels d'une image [Fourure, 2017]. C'est-à-dire son rôle est d'étiqueter chaque pixel d'une image en lui attribuant une classe spécifique. Elle permet ainsi de regrouper des zones amorphes présentant des textures ou des matériaux similaires comme l'herbe, le ciel ou la chaussée en leur assignant une étiquette correspondant à leur nature. En revanche, la segmentation par instance (détection des objets) s'intéresse aux objets dénombrables, tels que les personnes, les véhicules ou les animaux. Elle vise à identifier et distinguer chaque instance de ces objets dans l'image, en les délimitant à l'aide de boîtes englobantes ou de masques de segmentation individuels **[**Diaz-Zapata, 2019**].** La figure 2.14 montre le principe de la segmentation sémantique et la figure 2.17 illustre la segmentation sémantique d'instance.

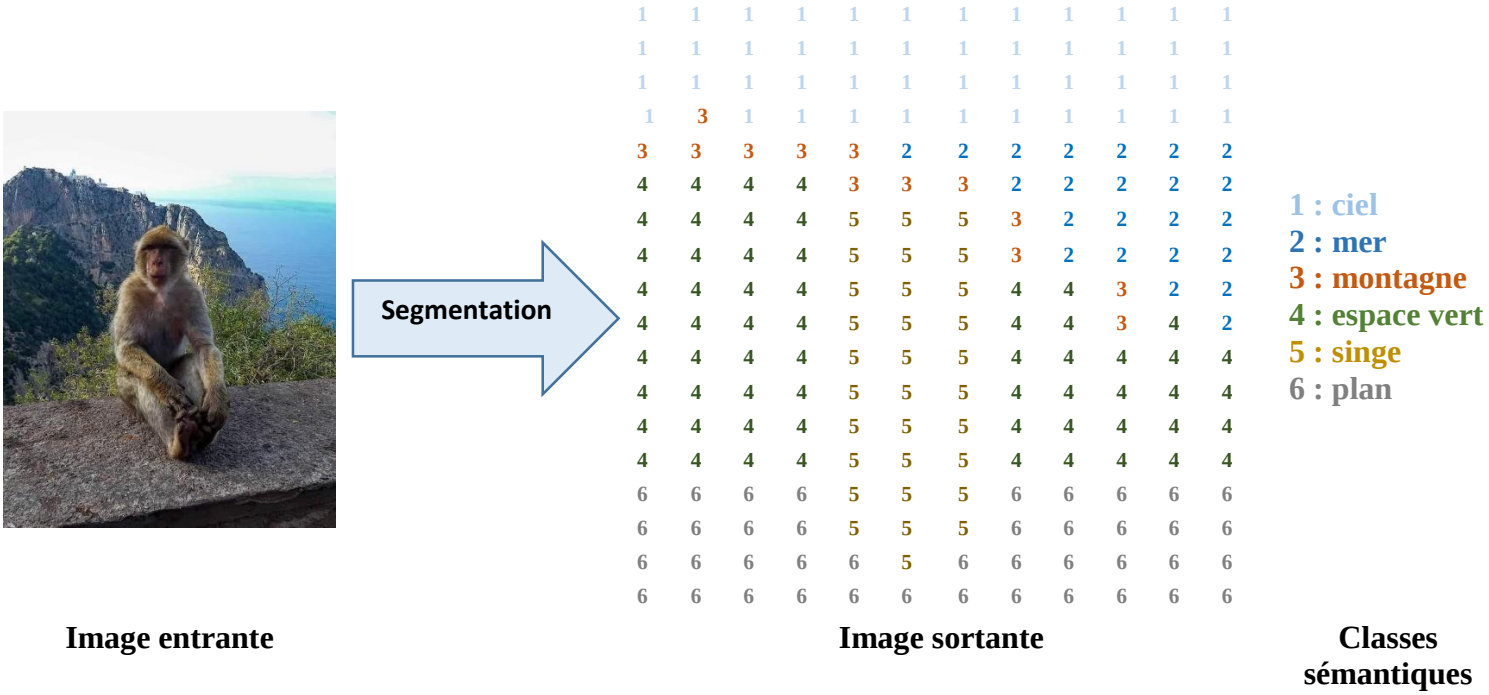


Figure 2.17 : Segmentation sémantique

Une illustration de la tâche de segmentation sémantique. L'image entière est divisée en segments en fonction de la signification sémantique. Chaque pixel de l'image est classé dans l'une des six classes sémantiques prédéfinies, et une région connectée portant la même étiquette représente un « objet ».

La **segmentation sémantique** constitue une tâche essentielle en vision par ordinateur. Divers jeux de données ont été développés pour permettre l'évaluation comparative des méthodes de segmentation, que ce soit dans le cadre des images multimédia [Lin et al., 2014], [Everingham et al., 2014], de la conduite autonome [Cordts et al., 2016], [Brostow et al., 2009], de la télédétection [Maggiori et al., 2017], [Rottensteiner et al., 2012] ou encore des applications médicales [Ulman et al., 2017]. De nombreuses approches en compréhension de scène reposent sur une classification dense à l'échelle du pixel, notamment pour des tâches telles que la détection

et la segmentation d'objets [He et al., 2017], [Arnab et Torr, 2017], [Audebert et al., 2017], [Sommer et al., 2017], [Ronneberger et al., 2015], [Che et al., 2016].

L'introduction des réseaux entièrement convolutifs (FCN) [Long et al., 2015] a marqué une avancée importante en permettant d'étendre les architectures de réseaux convolutifs classiques à la segmentation sémantique. Cette approche a permis de passer d'une classification globale de l'image à une classification dense, pixel par pixel. Par la suite, de nombreux modèles ont vu le jour afin d'exploiter davantage le contexte multi-échelle [Che et al., 2018], [Zhao et al., 2017] ou d'adopter des architectures symétriques de type encodeur-décodeur [Badrinarayanan et al., 2017], [Ronneberger et al., 2015]. [Audebert et al., 2018]

La figure 2.18 présente des exemples d'analyse d'images réalisés à l'aide de l'architecture encodeur-décodeur décrite précédemment dans le chapitre 1.

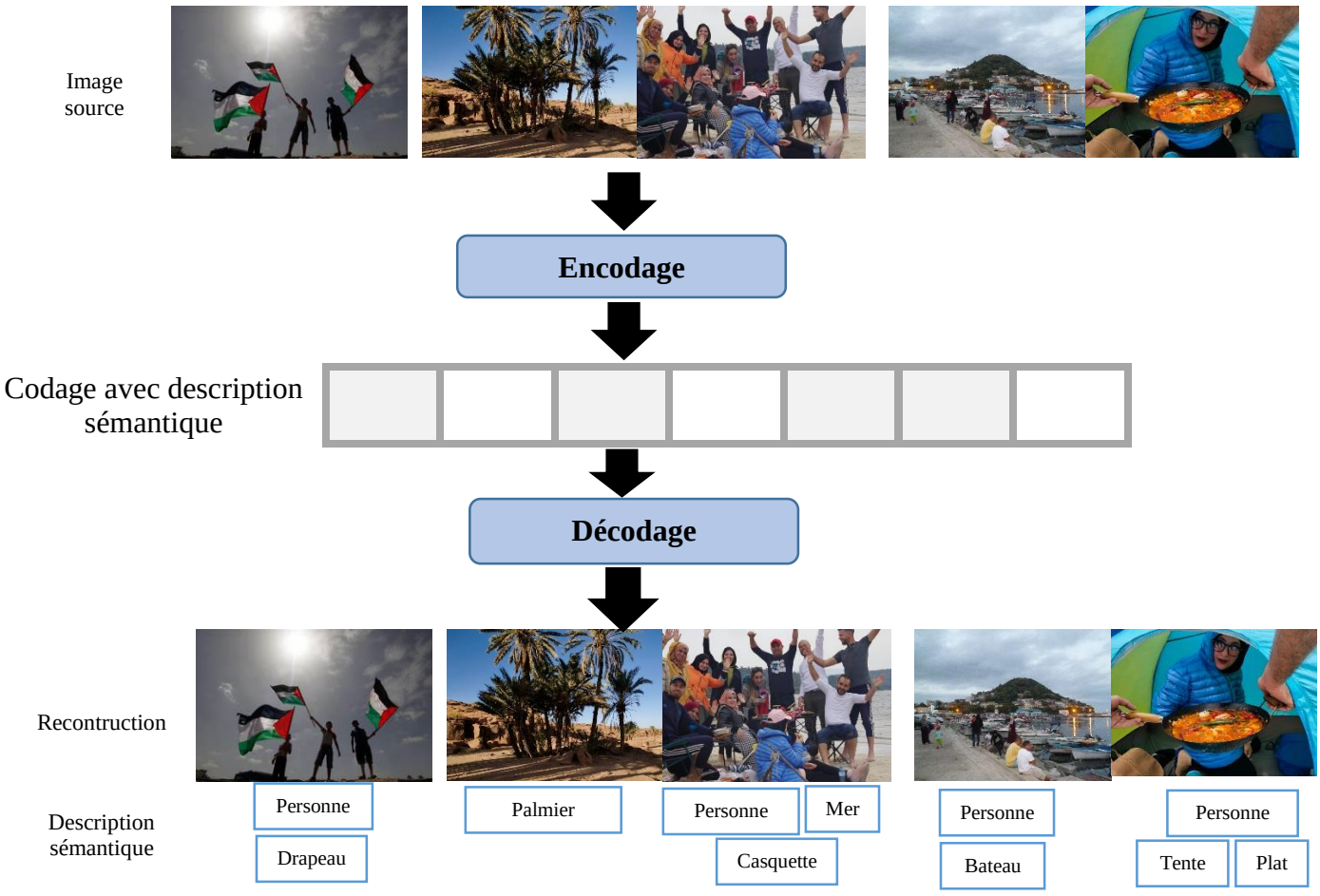


Figure 2.18 : Analyse sémantique d'images

Figure 2.19 : Segmentation sémantique d'instance.

La figure 2.19 illustre le type le plus avancé de reconnaissance d'image : la segmentation sémantique d'instances. Cette méthode ne se limite pas à prédire la classe sémantique de chaque pixel, mais elle distingue également les différentes instances d'une même classe. Autrement dit, deux objets distincts appartenant à la même catégorie (par exemple, deux voitures) seront identifiés séparément et associés à des étiquettes différentes. [Fourure, 2017]

2.4 Conclusion

À ce jour, le format JPEG demeure une référence incontournable parmi les techniques de compression d'images, grâce à son équilibre entre simplicité théorique et efficacité pratique. Basé sur des calculs mathématiques fondamentaux, il atteint des taux de compression élevés tout en maintenant une qualité visuelle satisfaisante pour l'œil humain. Cependant, la véritable complexité réside dans les démonstrations détaillées de son algorithme et dans l'optimisation des paramètres tels que les matrices de quantification.
L'utilisation généralisée de ce type de compression s'étend à divers domaines tels que l'informatique, la téléphonie et la vidéo. L'adaptabilité des implémentations, permettant de moduler la qualité en fonction des besoins, en fait un outil flexible et largement adopté. La popularité d'un algorithme de compression, comme le JPEG, repose sur plusieurs critères essentiels : le rapport taille/qualité, la rapidité des processus de compression et de décompression, ainsi que la robustesse de ses résultats.
Comparer les différentes méthodes disponibles reste une étape cruciale pour choisir l'approche la plus adaptée à des besoins spécifiques. Ces analyses permettent non seulement d'optimiser les performances selon les applications envisagées, mais aussi de guider les choix technologiques dans un paysage en constante évolution. Cette démarche souligne l'importance des recherches

continues dans le domaine de la compression d’images pour répondre aux défis croissants des usages modernes.

Chapitre 3 :

Super Pixel

3.1 Introduction

En vision par ordinateur, L'analyse d'images numériques repose traditionnellement sur le traitement de pixels individuels, qui sont les plus petites unités d'information d'une image. Cependant, cette approche présente des limites lorsqu'il s'agit d'extraire des structures visuelles cohérentes ou d'effectuer des traitements avancés, notamment en segmentation et en reconnaissance d'objets. En effet, le traitement pixel par pixel est coûteux en calcul et ne prend pas en compte la cohérence régionale des objets présents dans l'image.

Pour répondre à ces défis, la notion de **superpixel** a été introduite. Un superpixel est un ensemble de pixels adjacents partageant des caractéristiques visuelles similaires, telles que la couleur, l'intensité ou la texture. Contrairement aux pixels isolés, les superpixels offrent une représentation plus riche et plus structurée d'une image. Les superpixels permettent une représentation plus compacte et plus significative des images en réduisant le nombre d'éléments à traiter, ce qui est particulièrement utile pour des tâches telles que la segmentation, la reconnaissance d'objets et le suivi d'éléments dans une scène. Plusieurs algorithmes ont été développés pour générer des superpixels, notamment SLIC (Simple Linear Iterative Clustering), SEEDS, Felzenszwalb et QuickShift. Il existe une grande variété d'algorithmes de génération de superpixels, et plusieurs études ont comparé leurs performances [Achanta et al., 2012] , [Machairas et al., 2015] , [Wang et al., 2017] et [Stutz et al., 2018] [Ayres et al., 2024].

Ce chapitre explore les différentes approches de génération de superpixels, leurs avantages et leurs applications dans des domaines variés de la vision par ordinateur, notamment en segmentation d'image, en reconnaissance d'objets et en traitement vidéo.

3.2 Super Pixel

3.2.1 Pourquoi utiliser les super pixels ?

Plutôt que de traiter chaque pixel individuellement (ce qui est inefficace et peu pertinent sur le plan perceptuel), les super pixels regroupent des ensembles de pixels similaires, La Figure 3.1 présente différents résultats des super pixels. Les pixels similaires forment des régions cohérentes Cela permet [Machairas et al., 2015] :

- ✓ De réduire la complexité des calculs,
- ✓ D'améliorer la précision de la segmentation,
- ✓ De faciliter l'analyse des objets dans l'image.

3.2.2 Propriétés essentielles des super pixels

Pour être efficaces, les super pixels doivent respecter plusieurs critères fondamentaux [Machairas et al., 2015] :

1. Connexion spatiale : Chaque super pixel doit être une région connectée, sans parties séparées.

2. Homogénéité : Les pixels au sein d'un super pixel doivent avoir des couleurs ou des niveaux de gris similaires.

3. Respect des contours des objets : Les frontières des super pixels doivent épouser celles des objets afin d'éviter de mélanger plusieurs objets dans une même région.

4. Régularité : Les super pixels doivent être répartis de manière équilibrée dans l'image pour faciliter la comparaison des caractéristiques.

5. Facilité de calcul : La génération des super pixels doit être rapide et offrir un contrôle simple via des paramètres compréhensibles.

Le principal défi dans la génération des super pixels réside dans le compromis entre régularité et adhérence aux contours. Des super pixels bien réguliers permettent de faciliter l'apprentissage et la comparaison des caractéristiques visuelles, tandis qu'une meilleure adhérence aux contours garantit une segmentation plus précise mais peut entraîner des formes irrégulières. Un bon algorithme doit donc trouver un équilibre entre ces deux aspects pour optimiser la qualité de la segmentation. Les super pixels sont généralement utilisés comme étape préliminaire dans diverses applications de vision par ordinateur, telles que la segmentation d'images, la reconnaissance d'objets ou l'analyse d'images médicales. Leur génération doit être rapide et efficace, car ils s'intègrent souvent dans un traitement plus complexe et potentiellement gourmand en ressources [Machairas et al., 2015].

Figure 3.1 : différents résultats de Super Pixel.

3.2.3 Comment Construire un super pixel ?

La partition en super pixels est construite à partir des graines initiales. Parmi les méthodes de complexité linéaire, on distingue deux stratégies principales [Machairas et al., 2015] :

1. Méthodes par chemin le plus court (Type 1) : Ces approches, basées sur la croissance de régions, débutent avec un ensemble de graines (points ou régions) qui sont progressivement étendues en incorporant les pixels voisins selon une fonction de coût, généralement dépendante de l'image. Ce processus continue jusqu'à ce que chaque pixel de l'image soit assigné à un unique super pixel. Il peut être itératif ou non.

2. Méthodes par distance minimale (Type 2) : Ces procédures itératives sont inspirées des techniques d'apprentissage non supervisé. À chaque itération, les graines (comme des centroïdes) sont recalculées à partir de la partition précédente, et les pixels sont réassignés au centre le plus proche, à l'image de l'algorithme des **k-means**.

Bien que les méthodes inspirées du **clustering** (Type 2) puissent sembler avantageuses, notamment lorsqu'elles optimisent globalement une fonction de coût, elles ne garantissent pas toujours la connectivité des super pixels pour certains choix de distances entre pixels et graines. Par exemple, la distance définie par [Achanta et al., 2012], combinant la distance euclidienne et la différence de couleur, peut aboutir à des super pixels non connectés, ce qui est indésirable.
Pour pallier ce problème, un post-traitement est nécessaire, impliquant soit un re-étiquetage pour garantir la connectivité des régions [Wang et Wang, 2012], au détriment de la régularité des tailles et formes des super pixels, soit une réattribution des régions isolées au super pixel voisin le plus grand [Achanta et al., 2012], ce qui peut compromettre l'optimisation de la solution. Dans les deux cas, le nombre final de super pixels devient imprévisible. De plus, ce post-traitement augmente le coût computationnel, et peut devenir particulièrement lourd si l'image contient de nombreux petits objets par rapport à la taille moyenne des super pixels.
En revanche, les méthodes par croissance de régions (Type 1) appliquent naturellement une distance de type chemin, où la distance entre deux pixels dépend non seulement de leur position et de leur valeur, mais aussi des caractéristiques des pixels le long du chemin qui les relie. Ces méthodes garantissent des super pixels connectés, dont le nombre est exactement égal au nombre de graines initiales, offrant ainsi un contrôle plus précis sur la segmentation.

3.3 Algorithmes de construction des Super Pixels

3.3.1 Super pixel basé sur la forêt minimale couvrante

Les approches basées sur les graphes considèrent chaque pixel comme un nœud dans un graphe, où les liens entre pixels voisins sont pondérés selon leur similarité [Mandelli et al., 2020]. La segmentation en super pixels est alors obtenue en minimisant une fonction de coût définie sur le graphe [Wang et al., 2017].

Parmi les méthodes représentatives de cette catégorie, on retrouve [Wang et al., 2017] :

- **Normalized Cuts (N-cut)** [Shi et Malik, 2000], qui segmente l'image en minimisant un critère de coupure normalisée du graphe.
- **Felzenszwalb et Huttenlocher (FH)** [Felzenszwalb et Huttenlocher, 2004], une approche efficace qui fusionne des pixels en segments selon un critère d'homogénéité.
- **Homogeneous Superpixels (HS)** [Perbet et Maki, 2011], qui vise à produire des superpixels homogènes en termes de couleur et de texture.
- **Superpixels via Pseudo-Boolean Optimization (PB)** [Zhang et al., 2011], qui applique une optimisation pseudo-booléenne pour la segmentation.
- **Topology Preserved Regular Superpixel (TPS)** [Tang et al., 2012], qui génère des superpixels tout en préservant la structure topologique de l'image.
- **Lazy Random Walks Superpixel (LRW)** [Shen et al., 2014], qui utilise des marches aléatoires pour propager les superpixels de manière progressive.
- **Entropy Rate Superpixels (ERS**) [Liu et al., 2011], qui exploite le taux d'entropie pour optimiser la segmentation en superpixels.

Les arbres couvrants minimaux (MST - Minimum Spanning Tree) et les forêts couvrantes sont utilisés depuis longtemps pour le clustering et la segmentation d'images. [Felzenszwalb et Huttenlocher, 2004] ont proposé une approche de segmentation basée sur l'algorithme de Kruskal appliqué aux MSTs. Cette méthode consiste à [Mandelli et al., 2020] :

1. Trier les arêtes du graphe selon leur poids.

2. Fusionner progressivement les pixels aux extrémités des arêtes pour former des super pixels ou segments, à condition qu'ils soient similaires et qu'ils ne créent pas de cycle.

3. Définir un seuil par segment pour décider si un pixel doit être intégré au segment ou non.

Cet algorithme, proche de la complexité linéaire, génère une forêt couvrante minimale (MSF - Minimum Spanning Forest) où chaque sous-arbre correspond à un segment. Afin d'optimiser le traitement parallèle sans diviser les objets aux frontières, [Wassenberg et al.,2009] ont proposé une amélioration permettant de fusionner plusieurs MSF générés en parallèle. De plus, pour réduire le problème de sur-segmentation, ils ont introduit une fonction objective d'étiquetage, qui calcule une distance au seuil afin d'améliorer la segmentation.

Les méthodes de segmentation par bassins versants sont également largement utilisées. [Cousty et al., 2009] ont démontré que cette approche est équivalente à une MSF construite sur les minima d'une fonction de coût. L'algorithme 3.1 [Mandelli et al., 2020] montre les étapes de MSF.

Algorithm **MSF superpixel algorithm**

Input: Image *f*, parameters *depth, cut*
Output: Root map representing clusters *gRoot*
//Generate the gradient image
1: *g ← gradientMorpho(f);*
//Minima (seeds) detection as an indicator grid
2: *gMin ← minimaDetection(g,depth);*
//create groups of seeds at distance *cut* by euclidean MST
3: gRoot ← run EMST(gMin,cut);
// connect seed groups by euclidiean MST
4: gRoot2 ←runEMST(gMin,gRoot,∞);
//create forest by MST on image graph with gradient edge cost
5: gRoot ← runGMST(f,gRoot,gRoot2);
*6:*return *gRoot;*

Algorithme 3.1 : Algorithme MSF Super pixel.

- [Roe Roerdink et Meijster, 200000] ont proposé une étude approfondie sur différentes implémentations et adaptations de la segmentation par bassins versants et MST.

- Une implémentation open-source de cette technique est disponible dans la bibliothèque OpenCV [OpenCV, 2019], qui propose une segmentation par bassins versants marqués.

- Dans cette variante, les marqueurs sont fournis en entrée et influencent fortement le résultat de la segmentation. Par exemple, [Meyer, 1992] utilise les minima locaux du gradient comme seeds. Ces marqueurs s'étendent itérativement pixel par pixel jusqu'à atteindre une frontière définie par un autre segment, formant ainsi les lignes de partage des eaux.

3.3.2 Approche basée sur les k-means pour les super pixels

Les méthodes de segmentation basées sur le regroupement (ou clustering) regroupent les pixels en clusters, appelés super pixels, et les affinent itérativement jusqu'à atteindre un critère de convergence [Wang et al., 2017].

Parmi les méthodes populaires de cette catégorie, on retrouve [Wang et al., 2017] :

- **Simple Linear Iterative Clustering (SLIC)** [Liu et al, 2016], une approche efficace inspirée du clustering k-means.
- **SPASM (Superpixel Adaptive Segmentation Map),** proposé par [Wang et al., 2017],
- **Watershed** [Vincent et Soille, 1991], qui segmente l'image en super pixels en simulant un processus d'inondation.
- **MeanShift** [Comaniciu et Meer, 2002], qui regroupe les pixels en fonction de leur densité locale dans l'espace des couleurs.
- **QuickShift** [Vedaldi et Soatto, 2008], une méthode basée sur les distances de similarité et le parcours de graphes.

- **TurboPixel** [Levinshtein et al., 2009], qui génère des super pixels réguliers en simulant un flux géodésique.
- **Manifold SLIC** [Liu et al, 2016], une variante de SLIC améliorant la régularité des super pixels.
- **Contour Relaxed Superpixels (CRS)** [Conrad et al., 2013], qui ajuste la segmentation en respectant mieux les contours.
- **Superpixel Segmentation using Linear Spectral Clustering (LSC)** [Li et Chen, 2015], qui applique le clustering spectral pour générer des super pixels.
- **Depth-Adaptive Superpixels (DAS)** [Wang et Wang, 2012], qui ajuste la segmentation en fonction de la profondeur dans une scène 3D.
- **Compact Superpixels (CS)** [Veksler et al., 2010], qui privilégie des super pixels de forme régulière et compacte.
- **Constant Intensity Superpixels (CIS)** [Veksler et al., 2010], qui segmente en super pixels de luminance homogène.
- **SEEDS** [Van den Bergh et al., 2012], qui affine progressivement la segmentation en optimisant les frontières.
- **VCells** [Wang et Wang, 2012] et **VCCS** [Papon et al., 2013], qui exploitent des cellules volumétriques pour une segmentation en 3D.
- **Superpixel Lattices** [Moore et al., 2008] et **Lattices Cut** [Moore et al., 2010], qui génèrent des super pixels en formant une grille structurée.
- **Scene Shape Superpixel (SSP)** [Moore et al., 2009], qui adapte la segmentation en fonction de la géométrie de la scène.
- **Co-superpixel** [Xie et al., 2014], qui améliore la segmentation en intégrant des informations contextuelles.
- **Saliency-based Superpixel (SBS)** [Xu et al., 2014], qui optimise la segmentation en mettant en avant les régions saillantes.
- **Structure Sensitive Superpixels (SSS)** [Wang et al., 2013], qui privilégie la cohérence structurelle des super pixels.

3.3.2.1 SLIC Super Pixel

L'algorithme **SLIC** (Simple Linear Iterative Clustering), proposé par [Achanta et al., 2012], est une méthode de segmentation qui génère des super pixels en faisant croître des régions à partir de centres initiaux, en prenant en compte à la fois la similitude des couleurs et la proximité spatiale. Il repose sur un clustering de type k-means local, où les centres de clusters sont initialisés sur une grille régulière et leur zone d'influence est restreinte à un voisinage spatial autour du centre. L'algorithme utilise une métrique combinant la distance de couleur dans l'espace LAB et la distance spatiale dans l'image. Un paramètre de compacité permet d'ajuster l'importance de la composante spatiale, influençant ainsi la forme et la taille des super pixels obtenus [Mandelli et al., 2020].

La seule variable d'entrée est k, qui correspond au nombre de super pixels souhaité. Le processus de segmentation suit ces étapes [Mandelli et al., 2020]:

1. **Initialisation** : les k centres des super pixels sont placés sur une grille régulière afin d'obtenir des super pixels de tailles similaires.

2. **Assignation des pixels** : chaque pixel est affecté au centre de cluster le plus proche dont la zone d'influence le recouvre. La proximité est évaluée à l'aide d'une distance euclidienne dans un espace 5D combinant coordonnées spatiales et couleur.

3. **Mise à jour des centres** : après assignation, chaque centre est recalculé comme la moyenne des pixels qui lui sont attribués.

4. **Itérations** : les étapes précédentes sont répétées jusqu'à la convergence de l'erreur.

5. **Post-traitement** : une correction finale est appliquée pour assurer la connectivité des super pixels, en réattribuant les pixels isolés aux super pixels les plus proches.

L'algorithme 3.2 [Achanta et al., 2012] résume tous ces étapes.

Algorithm SLIC Superpixel Segmentation

*/*Initialization*/*
Intialize cluster centers $C_K = [L_K, a_K, b_K, x_K, y_K]^T$ by
Sampling pixels at reglar grid step *S*.
More cluster centers to the lowest gradient position
in a
3x3 neighborhood.
Set label *l(i)*= -1 for each pixel *i.*
Set distance *d(i)*= ∞ for each pixel *i.*
Repeat
/*Assignment*/
For each cluster center C_K **do**
For each pixel *i* in a *2S* x *2S* region around C_K **do**
Compute tehe distance *D* between C_K and *i*.
If *D*< d(*i*) **then**
Set *d(i)= D*
Set *l(i)= k*
End if
End for
End for
*/*Update*/*
Compute new cluster centers.
Compte residual error *E.*
Until *E*≤ threshold

Algorithme 3.2: SLIC segumentation par superpixel.

Il existe plusieurs implémentations de **SLIC**, notamment :

- L'implémentation originale proposée par [Achanta et al., 2012],
- Une version intégrée à la bibliothèque VLFeat [Vedaldi et Soatto, 2008],
- Une implémentation parallèle sur GPU développée par [Ren et Reid, 2011].

La **Figure 3.2** [Ayres et al., 2024] illustre le fonctionnement de **SLIC**, en mettant en évidence le processus de génération des superpixels. Il est important de noter que le terme **"oversegmentation"** est une autre désignation pour les **super pixels**, soulignant ainsi le fait que l'image est segmentée en régions plus petites et homogènes avant une analyse plus approfondie.

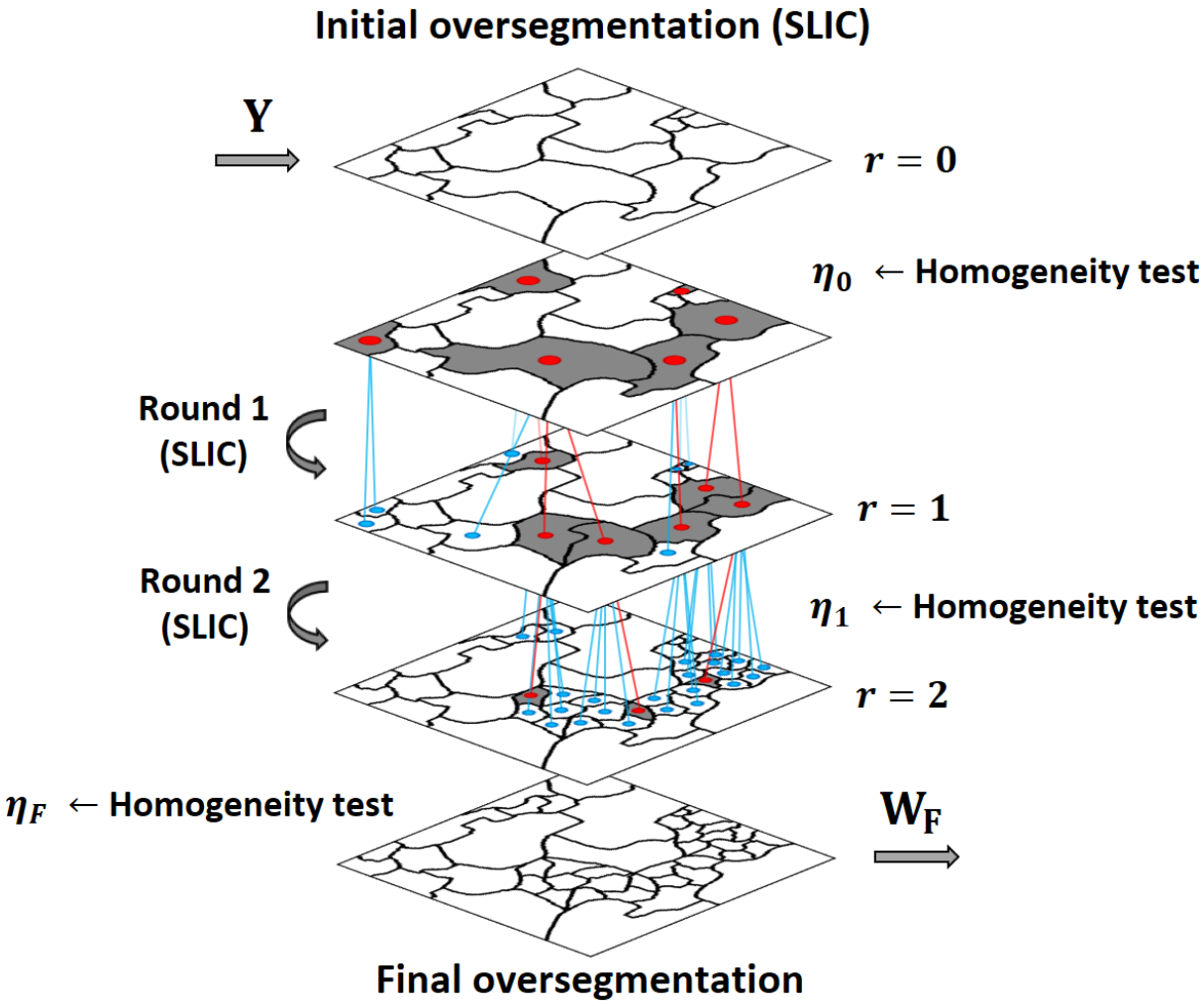


Figure 3.2 : Schéma de la méthode SLIC pour R = 2.
Les super pixels identifiés comme non homogènes (homogènes) sont représentés par les régions grises avec des points rouges (régions blanches avec des points bleus). À l'échelle (r), les superpixels non homogènes sont subdivisés à l'échelle (r+1) en superpixels homogènes (non homogènes), indiqués respectivement par des lignes bleues (rouges).

Enfin, une étude comparative menée par [Wang et al., 2017] a évalué **gSLIC** face à **SLIC** et **SPASM**. Les résultats ont montré que gSLIC est plus lent que SLIC et SPASM en termes de

temps de calcul. Le tableau 3.1 illustre des exemples obtenus avec l'implémentation originale de SLIC.

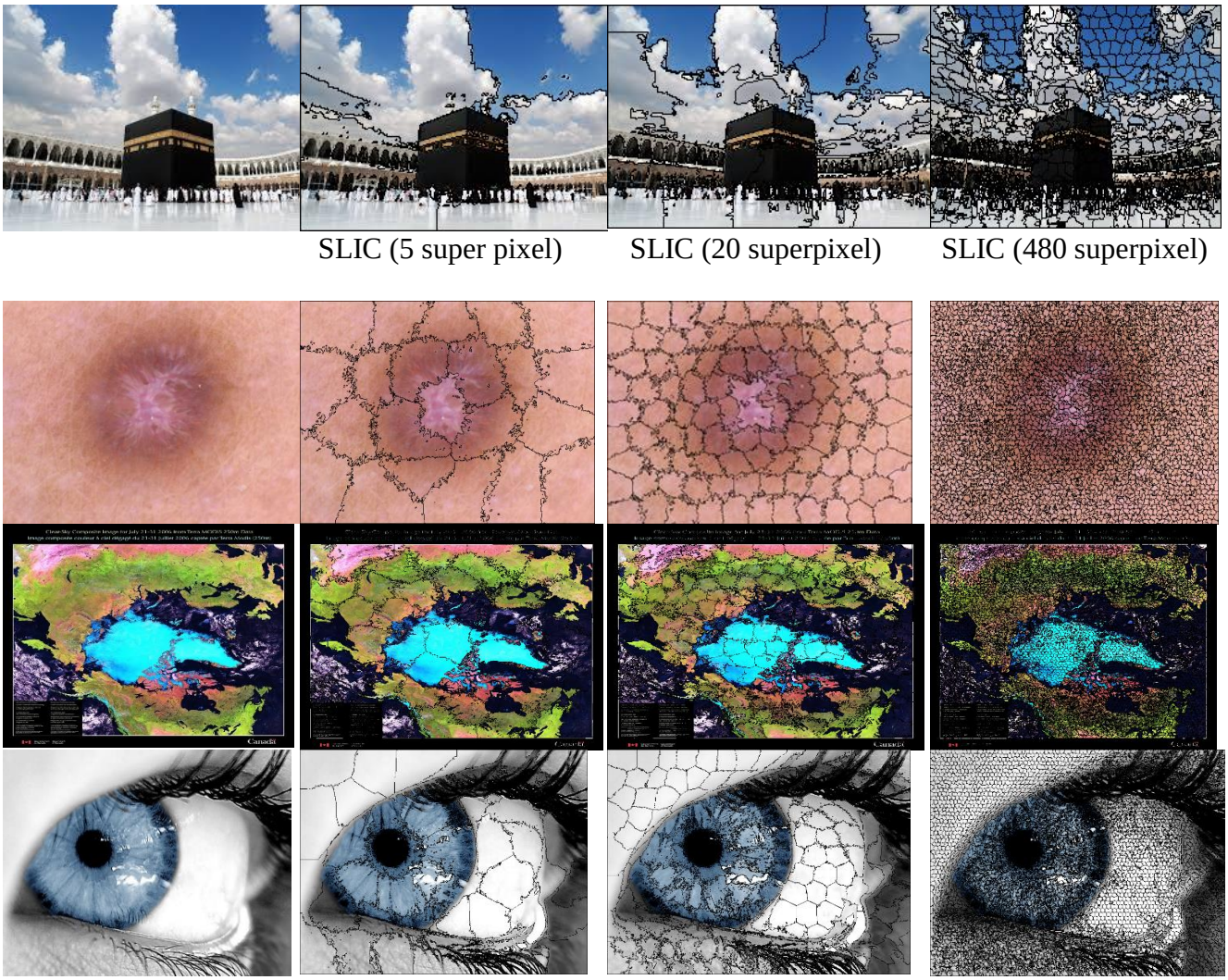


Tableau 3.1 : Exemples de segmentation par SLIC super pixel

3.3.2.2 SPASM segmentation

L'algorithme SPASM (Superpixel Adaptive Segmentation Map), est une méthode de segmentation des super pixels optimisée pour une exécution sur GPU. Le terme "adaptatif" signifie que la distribution des super pixels s'ajuste dynamiquement en fonction de certaines caractéristiques de l'image, telles que les contours, les textures et la profondeur. Dans un premier

temps, SPASM initialise une grille régulière bidimensionnelle de nœuds, uniformément répartis sur le plan Euclidien défini par l'image d'entrée. Chaque nœud représente le centre d'un super pixel [Mandelli et al., 2020].

L'algorithme s'appuie ensuite sur une adaptation dynamique des nœuds en deux étapes [Mandelli et al., 2020] :

1. Une première phase utilise une version en ligne de l'algorithme SOM(Self-Organizing Map) de [Kohonen, 2001] pour ajuster la répartition des nœuds en fonction des caractéristiques de l'image (contours, textures, profondeur, etc.).

2. Une seconde phase applique une version batch de SOM avec une nouvelle métrique de distance dans l'espace couleur, permettant d'affiner la segmentation finale.

Toutes ces opérations sont massivement parallélisées sur GPU, garantissant une exécution rapide et efficace.

La Figure 3.3 [Mandelli et al., 2020] illustre des exemples de segmentation en superpixels obtenus avec l'implémentation parallèle de SPASM sur GPU.

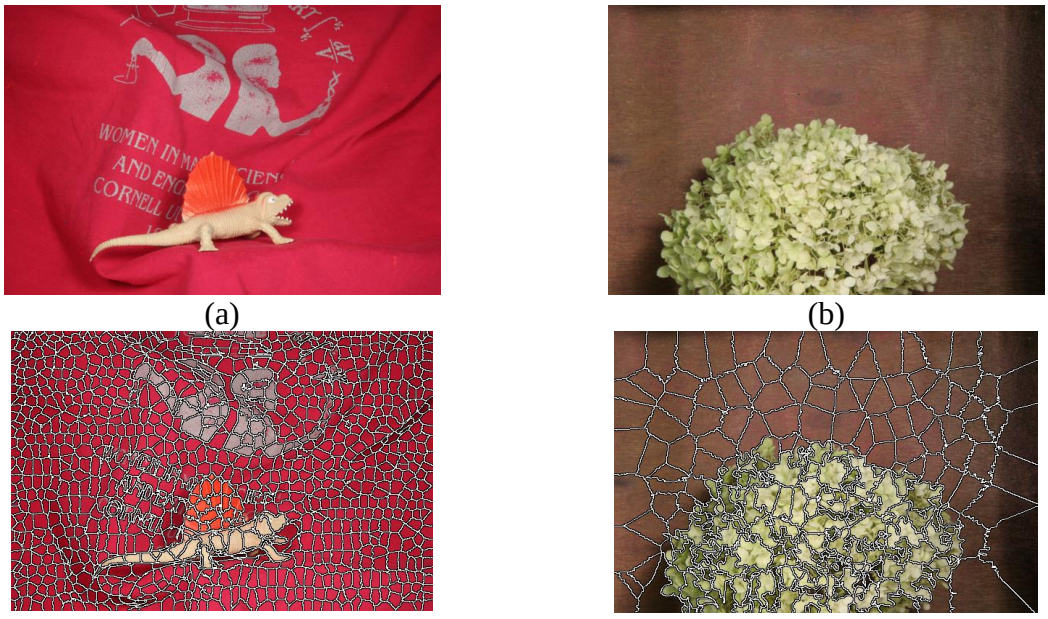


Figure 3.3 : Exemples d'application de l'algorithme SPASM sur des images issues du jeu de données Middlebury : (a) Image d'entrée Dimetrodon ; (b) Image d'entrée Hydrangea; (c) Carte de segmentation SPASM avec k = 1015 ; (d) Carte de segmentation SPASM avec k = 260.

3.4 Conclusion

Dans ce chapitre, nous avons exploré les différentes approches de segmentation en super pixels, en mettant en évidence leurs principes, avantages et limites. Nous avons tout d'abord présenté le concept de super pixels et leur importance dans le prétraitement des images pour diverses

applications de vision par ordinateur. Ensuite, nous avons étudié plusieurs méthodes de segmentation, en les classant en deux grandes catégories : les méthodes basées sur les graphes et les approches inspirées du clustering. Parmi ces approches, nous avons cité des algorithmes populaires tels que SLIC, Watershed, MeanShift, TurboPixel, ainsi que des méthodes plus récentes comme SPASM et les méthodes à base de graphes minimaux. Nous avons également discuté des enjeux liés à la segmentation en super pixels, notamment le compromis entre régularité et adhérence aux contours, la connectivité des super pixels, ainsi que les défis liés à l'optimisation des performances.

L'analyse de ces différentes techniques met en évidence les critères essentiels pour choisir un algorithme de segmentation en super pixels en fonction des contraintes applicatives. Parmi ces approches, l'algorithme SLIC se distingue particulièrement par sa simplicité, son efficacité et sa capacité à générer des super pixels réguliers et bien adhérents aux contours des objets. Contrairement à d'autres méthodes plus complexes, comme celles basées sur les graphes ou l'optimisation adaptative, SLIC offre un excellent compromis entre précision, rapidité de traitement et facilité de paramétrage, le rendant ainsi idéal pour une large gamme d'applications en vision par ordinateur.

Partie 2 :

Les Contributions

Chapitre 4 :

Approche de compression des images basée sur le découpage de l'image

4.1 Introduction

Dans ce chapitre, nous introduisons une méthode novatrice de compression d'images fondée sur le compresseur JPEG, avec une amélioration apportée par un choix adaptatif du facteur de qualité en fonction de l'homogénéité de l'image. L'approche développée dans ce travail se distingue par l'ajustement dynamique du facteur de qualité, permettant de moduler le niveau de compression en tenant compte des variations de contenu au sein de l'image. Cette stratégie permet d'obtenir un taux de compression optimisé, tout en préservant une qualité d'image supérieure comparativement aux méthodes classiques basées sur JPEG. Les résultats montrent que notre méthode parvient à concilier efficacement réduction de taille de fichier et fidélité visuelle, rendant ainsi cette technique particulièrement intéressante pour des applications nécessitant à la fois efficacité de stockage et qualité de rendu.

4.2 Problématique

La norme de compression JPEG, proposée par le *Joint Photographic Experts Group* et adoptée depuis 1993, s'est imposée comme une référence incontournable dans le domaine de la compression d'images. Elle repose sur une méthode de compression dite avec pertes, permettant une réduction significative de la taille des images numériques. Grâce à sa simplicité computationnelle, sa flexibilité et son efficacité, le JPEG permet d'adapter le taux de compression selon les besoins, en préservant une qualité visuelle acceptable, voire quasi intacte à l'œil nu dans certaines configurations [Cogranne, 2018].

Cependant, malgré ses avantages, le JPEG présente plusieurs limitations. Tout d'abord, la compression entraîne inévitablement une perte d'information, ce qui peut dégrader la qualité visuelle, notamment lorsque des taux de compression élevés sont appliqués. Ce compromis qualité-taille est contrôlé principalement par un facteur de qualité (FQ), dont la valeur influence directement le degré de dégradation : un FQ élevé augmente le taux de compression mais altère davantage l'image, tandis qu'un FQ faible réduit la compression mais préserve mieux les détails [Kadhim, 2016].

(a) (b) (c)

Tableau 4.1 : Exemple de compression JPEG avec influence du FQ (a) taille de l'image originale=225Kb, (b) image JPEG avec FQ=30 : qualité acceptable et taille acceptable =18Kb, (c) image JPEG avec FQ=70 : faible qualité et bonne taille =5Kb.

Le tableau 4.1 est un exemple illustratif présente les effets de la compression JPEG sur une image initiale de 225 kilo-octets (KO) selon deux valeurs distinctes de facteur de qualité (FQ). En

appliquant la compression JPEG avec un FQ de 30, l'image compressée atteint une taille réduite de 18 KO, tout en maintenant une qualité visuelle acceptable. En revanche, lorsqu'un FQ de 70 est utilisé, l'image compressée voit sa taille diminuer davantage, atteignant 5 KO, mais au prix d'une perte de détails plus marquée. Cet exemple démontre clairement que la valeur du facteur de qualité impacte directement la qualité visuelle de l'image restituée après compression, illustrant le compromis nécessaire entre la réduction de l'espace de stockage et la préservation des détails.

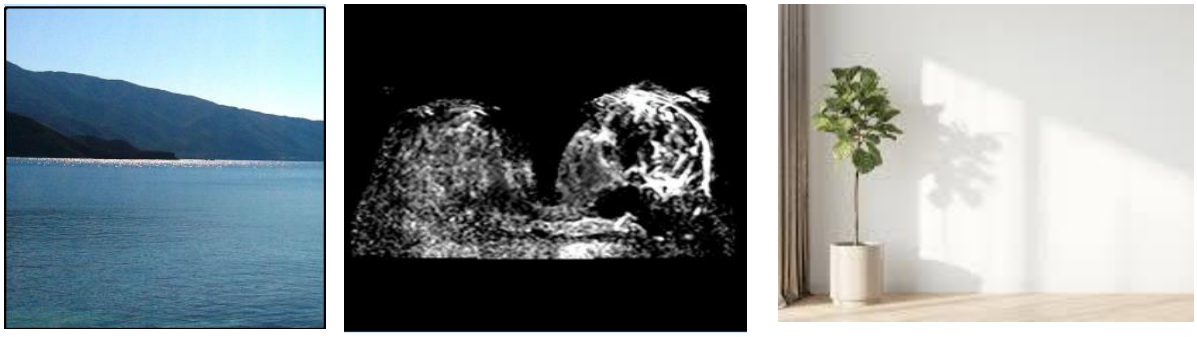
Par ailleurs, le JPEG applique généralement une stratégie uniforme sur toute l'image, ignorant les variations de contenu local. Ainsi, les zones riches en détails subissent la même compression que les régions homogènes, ce qui peut engendrer une perte excessive d'informations visuelles critiques dans certaines parties de l'image. De plus, les besoins en qualité diffèrent selon les domaines d'application : imagerie médicale, surveillance, photographie, télédétection, etc. Une méthode unique, appliquée de manière globale, peut alors s'avérer sous-optimale [Jin et al., 2017].

4.3 Motivation

L'objectif fondamental de la compression d'images est de réduire la quantité de données nécessaires pour représenter une image tout en conservant une qualité visuelle satisfaisante [Zhang et Zhang, 2014]. Le JPEG, en tant que norme dominante, a démontré son efficacité en atteignant un bon compromis entre qualité visuelle et taux de compression. Cependant, l'approche uniforme du JPEG ne prend pas en compte la structure spatiale de l'image. Notre proposition consiste à améliorer cette méthode en introduisant une compression adaptative basée sur l'homogénéité de l'image. Autrement dit, nous cherchons à déterminer localement la valeur optimale de FQ en fonction des caractéristiques visuelles de chaque région :

- **Pour les zones homogènes** (surfaces lisses, couleurs uniformes), un FQ élevé est approprié, car une forte compression n'entraîne pas de dégradation perceptible.
- **Pour les zones hétérogènes** (textures complexes, contours nets), un FQ plus faible est préféré, afin de préserver les détails visuels importants.

Images hétérogène



Images homogène

Figure 4.1 : Exemples des images homogènes et hétérogènes.

La figure 4.1 illustre cette distinction entre images homogènes et hétérogènes en présentant des exemples concrets. Elle montre comment des images homogènes, comme des surfaces lisses ou des zones de couleur uniforme, peuvent tolérer une compression plus forte, tandis que les images hétérogènes, comme des scènes détaillées ou des textures complexes, nécessitent une approche plus prudente pour conserver les éléments visuels critiques.

4.4 Approche proposée

Dans ce travail, nous proposons une amélioration de la méthode JPEG en adaptant le facteur de qualité (FQ) de manière intelligente selon le degré d'homogénéité de différentes zones de l'image. Contrairement à l'approche standard de JPEG, où un FQ unique est appliqué à l'ensemble de l'image, notre méthode segmente l'image en régions et ajuste le FQ de chaque région en fonction de son homogénéité.

Ce processus est illustré par la figure 4.2, où l'image est divisée en parties avec une évaluation de l'homogénéité pour chacune, suivie de l'attribution d'un FQ approprié. Ce schéma met en évidence les étapes de découpage, d'évaluation de l'homogénéité, et de sélection adaptative du FQ pour chaque partie de l'image. Grâce à cette approche, notre méthode permet de conserver une haute qualité visuelle dans les zones détaillées tout en optimisant la compression des zones plus uniformes, ce qui se traduit par une amélioration du taux de compression global sans compromettre la qualité de l'image dans ses parties critiques.

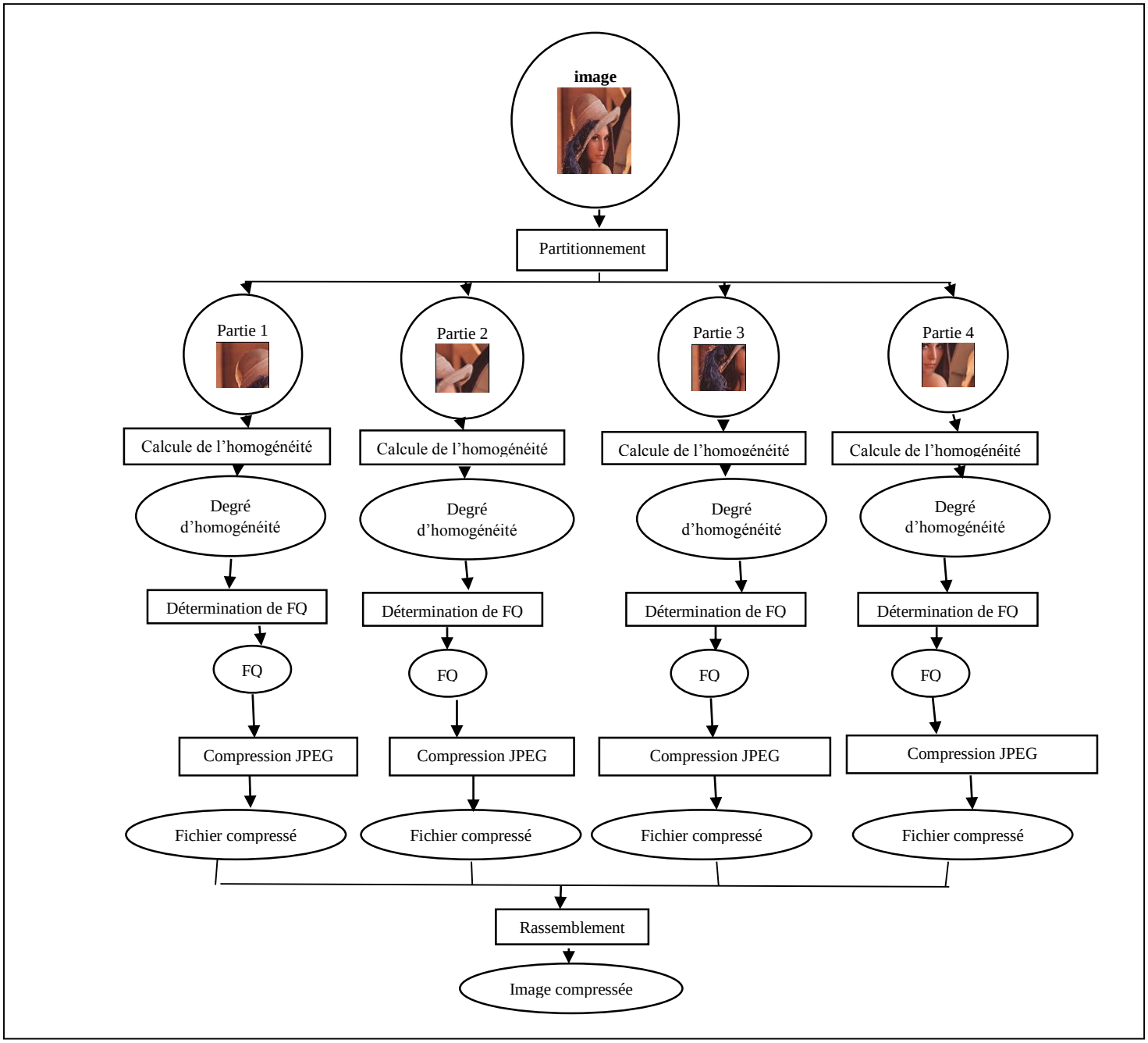


Figure 4.2 : Architecture de la compression de l'approche proposée

La première étape de l'approche proposée est de diviser l'image en quatre parties égales.

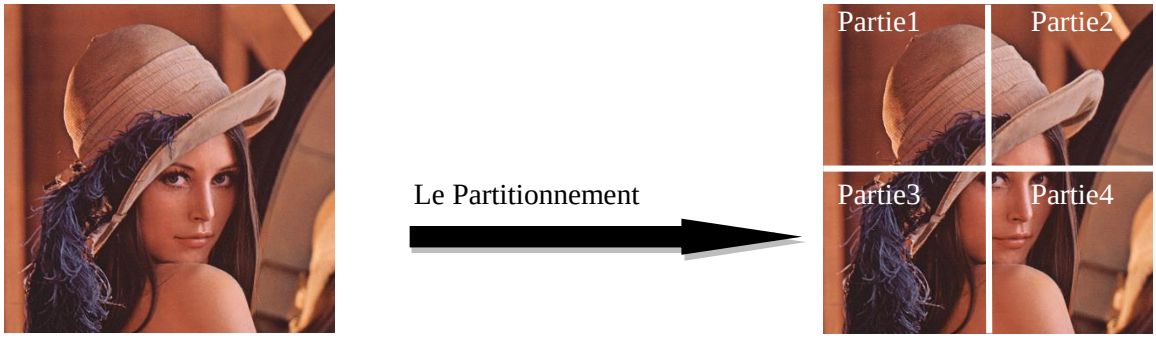


Image de Lina Image Découpée

Figure 4.3 : partitionnement de l'image de Lina en 4 parties.

Après la phase de découpage, l'objectif devient de calculer le degré homogénéité D^{pi} de chaque partie de l'image p^i. Pour évaluer cette homogénéité, nous avons adopté une méthode basée sur l'espace de couleur RGB, en appliquant la formule suivante :

$$D^{pi} = \frac{1}{N^i}\sum_{j=1}^{N^i}(c_j - \mu)^2 \quad (1)$$

Ou:

- μ: la valeur moyenne de l'intensité de la partie d'image p^i,
- N^i: la taille de p^i
- c_j : la valeur moyenne de la couleur d'un pixel calculée par la formule (2)

$$c_j = \frac{r_j + b_j + g_j}{3} \quad (2)$$

Si le degré d'homogénéité D^{pi} est faible, cela signifie que l'image est hétérogène, avec de nombreux détails, objets et variations de couleurs. En revanche, si le degré d'homogénéité est élevé, l'image est considérée comme homogène, où les valeurs d'intensité des pixels sont proches les unes des autres. Le tableau 4.2 présente un exemple d'images avec leurs degrés d'homogénéité respectifs.

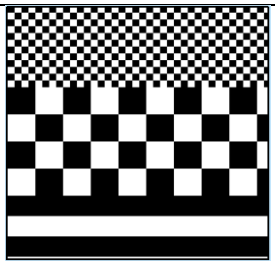
image				
Degré d'homogénéité	39	126	74	6

Tableau 4.2 : Exemple des valeurs différentes de degrés d'homogénéité.

Après avoir effectué ce calcul, le système attribue un degré d'homogénéité à chaque partition de l'image. Ensuite, il passe à l'étape de la sélection de valeur de FQ de chaque partie d'image. En fonction de ces valeurs de degré d'homogénéité, un facteur de qualité est déterminé pour chaque partition. Autrement dit, le facteur de qualité de chaque partition dépend directement de son degré d'homogénéité. Le choix de la valeur de FQ est définit dans le tableau 4.3 :

D^{pi}	<50	[50,100[	[100,150[	>150
FQ	10	30	50	70

Tableau 4.3 : Équivalence entre la valeur du facteur de qualité et le degré d'homogénéité.

Une fois que les facteurs de qualité associés à chaque partition de l'image sont obtenus, le compresseur proposé applique la compression JPEG à chaque partition individuellement, en traitant chaque partition comme une image distincte.
Le processus de décompression, illustré par le schéma de la figure 4.4, suit les étapes inverses de celles utilisées lors de la compression.

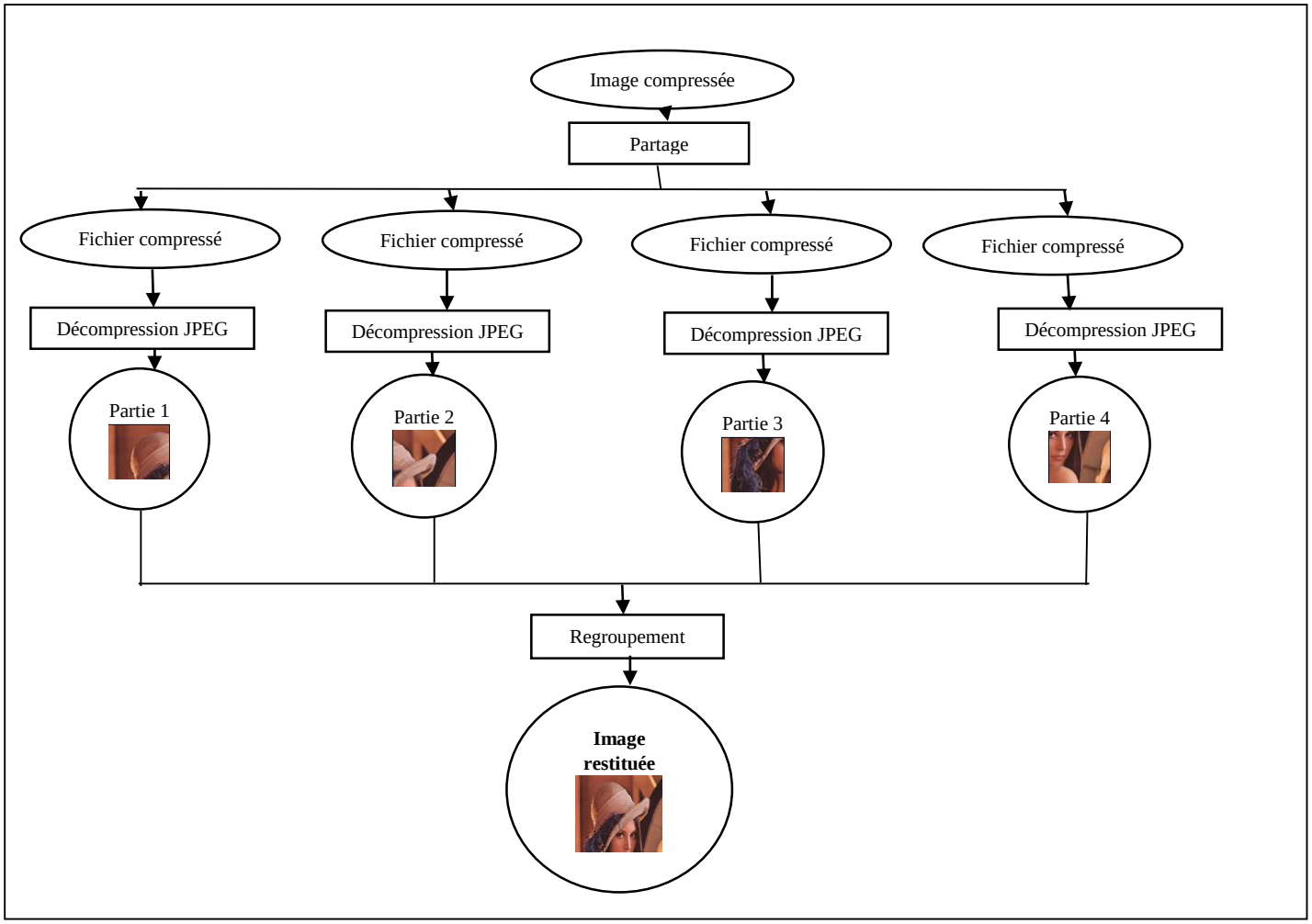
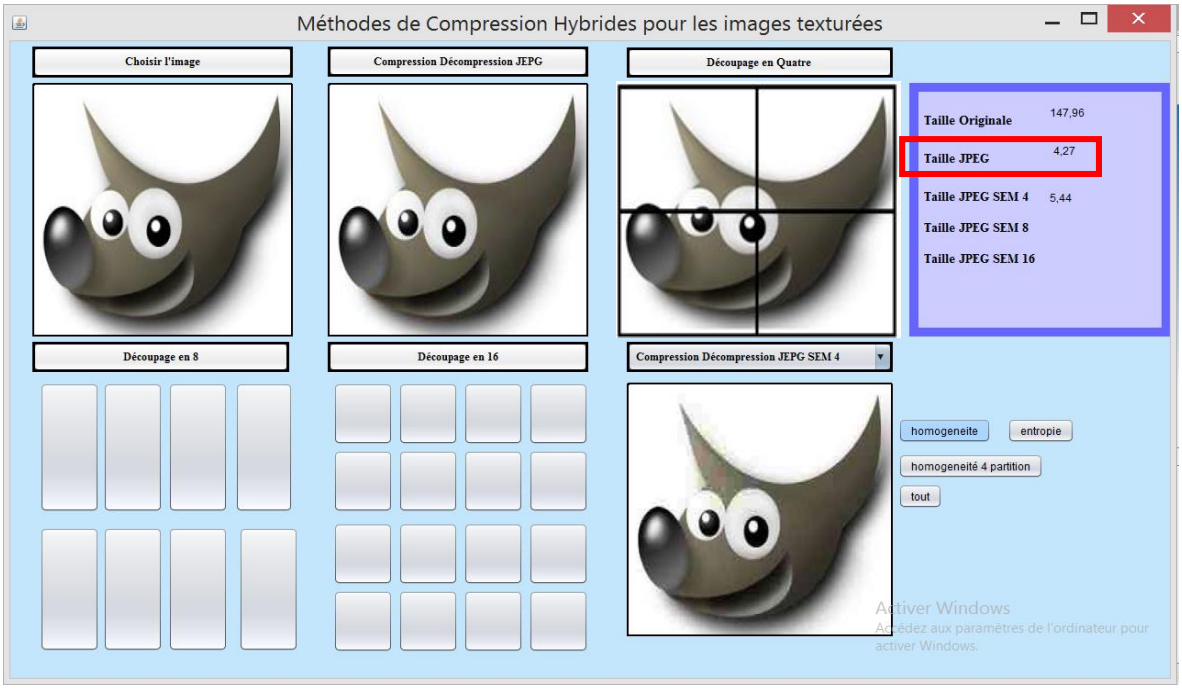


Figure 4.4 : Architecture de la décompression de l'approche proposée.

Le compresseur proposé prend l'image compressée (le fichier global) et la divise en quatre fichiers distincts. Ensuite, chaque fichier est décompressé en JPEG en respectant le facteur de qualité (FQ) attribué à chaque partition. Enfin, le système assemble les différentes parties pour restituer l'image initiale.

4.5 Tests et Résultats

Le compresseur proposé a été implémenté en Java, utilisant l'environnement de développement Eclipse. L'application se compose de trois sections principales : la partie dédiée à la compression JPEG, celle réservée au découpage de l'image en 4, 8 et 16 parties, et une section pour l'affichage des résultats. L'interface utilisateur comporte six panneaux : un pour l'image originale, un pour l'image JPEG compressée, ainsi que des panneaux pour les images découpées en 4, 8 et 16 parties, et un panneau pour afficher les résultats de l'approche proposée. De plus, trois boutons



sont disponibles pour afficher les valeurs d'entropie et les degrés d'homogénéité. Les figures 4.5, 4.6 et 4.7 illustrent les résultats du test du système proposé, avec un découpage en 4, 8 et 16 parties respectivement.

Figure 4.5 : Interface de compresseur proposée avec découpage en 4.

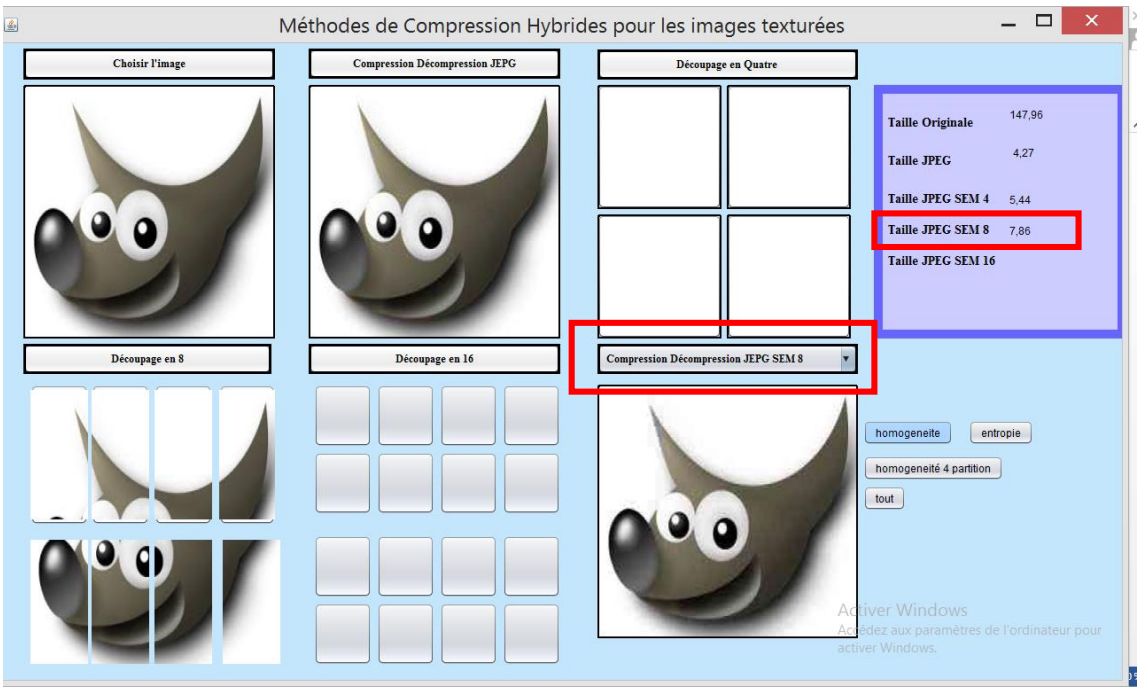


Figure 4.6 : Interface de compresseur proposée avec découpage en 8.

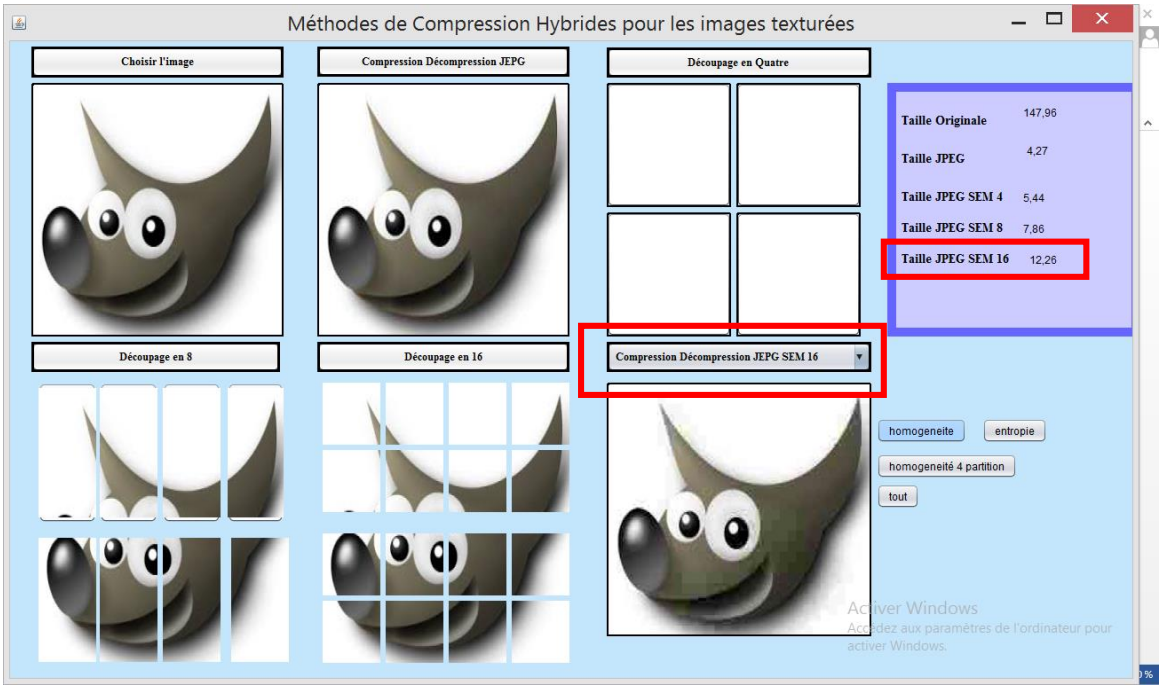


Figure 4.7 : Interface de compresseur proposée avec découpage en 16.

Le compresseur proposé a démontré son efficacité en comparaison avec la méthode JPEG classique. Cette performance se manifeste à la fois par la qualité d'image obtenue après compression et par la réduction significative de l'espace de stockage nécessaire. En outre, nous avons effectué une comparaison des résultats obtenus avec différentes stratégies de découpage, à savoir en 4, 8 et 16 partitions. Les résultats ont révélé que le découpage en 4 partitions fournit des performances optimales, en termes de qualité d'image et d'efficacité de compression, par rapport aux découpages en 8 et 16 partitions. En effet, bien que le découpage en 8 et 16 partitions puisse offrir une certaine précision supplémentaire, il entraîne une complexité accrue et n'améliore pas de manière significative la qualité ou la réduction de l'espace de stockage par rapport à la méthode à 4 partitions. Par conséquent, le découpage en 4 partitions apparaît comme un compromis idéal, équilibrant efficacement la qualité d'image et l'efficacité de compression. Le tableau 4.4 ci-dessous illustre les résultats obtenus, mettant en évidence le succès du compresseur proposé.

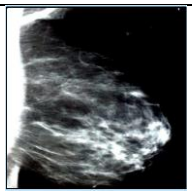
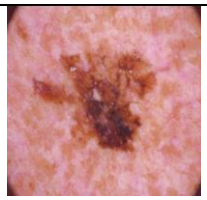
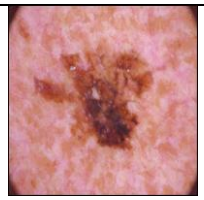
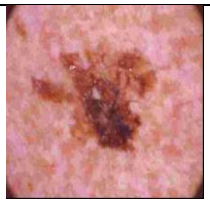
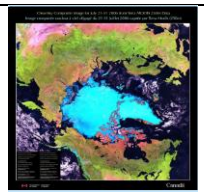
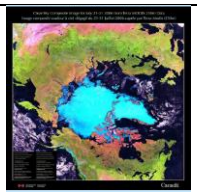
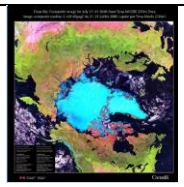
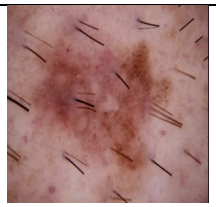
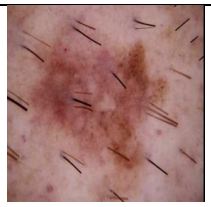
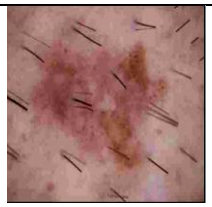
Image	Image Originale	JPEG(FQ=50)	Approche Proposée
1			
2			
3			
4			
5			
6			

Tableau 4.4 : Tableau illustratif des résultats.

Image	Image Originale	JPEG	Approche Proposée
1	1304b	43 Kb	30Kb
2	639 Kb	21Kb	16Kb
3	790 Kb	23.20Kb	9.52Kb
4	2037Kb	105Kb	71Kb
5	148Kb	6.12Kb	6.02Kb
6	790Kb	17.57Kb	9.09Kb

Tableau 4.5 : Taille de stockage

Image	Approche Proposée			
	FQ Part1	FQ Part2	FQ Part3	FQ Part4
1	30	30	10	30
2	30	10	30	30
3	10	10	10	10
4	30	10	30	30
5	10	30	10	30
6	10	10	10	10

Tableau 4.6 : valeurs des FQ.

4.6 Discussion

Le tableau 4.4 présente quelques exemples de résultats de compression obtenus avec la méthode JPEG classique et l'approche proposée. Les deux méthodes offrent une qualité d'image satisfaisante. Cependant, le tableau 4.5 met en évidence la supériorité de l'approche proposée, qui permet d'obtenir une réduction plus importante de la taille de stockage.Pour mieux comprendre l'impact du facteur de qualité (FQ), le tableau 4.6 présente les valeurs du FQ attribuées à chaque partition des quatre images analysées.

Cette approche a montré des résultats prometteurs, surpassant même le célèbre compresseur JPEG en termes de qualité d'image et de taux de compression des images restituées. En ajustant le facteur de qualité (FQ) en fonction de l'homogénéité de l'image, nous avons pu obtenir un meilleur compromis entre la compression et la qualité, surpassant les méthodes traditionnelles en offrant des images compressées de qualité supérieure tout en réduisant efficacement l'espace de stockage nécessaire.

D'après le tableau 4.6, les valeurs de FQ des images dermatologiques se révèlent stables dans les quatre parties de l'image, reflétant une homogénéité quasi similaire entre ces zones. Cette stabilité résulte de la centralisation de la zone d'intérêt dans les images médicales dermatologiques. Lors du découpage en quatre, chaque partie intègre approximativement un quart de la zone centralisée et trois quarts du fond de l'image, illustrant ainsi les limites du concept de découpage en quatre dans le cadre de la compression JPEG pour ce type d'images.

4.7 Conclusion

Les performances obtenues, tant au niveau de la qualité que du taux de compression, sont encourageantes et offrent un potentiel significatif pour des applications nécessitant à la fois des images de haute qualité et des capacités de compression élevées, telles que dans les domaines du stockage d'images, des transmissions en temps réel et des systèmes de vision par ordinateur. Ces résultats nous motivent à poursuivre et approfondir nos recherches dans ce domaine, afin de rendre cette approche encore plus robuste et polyvalente.

Les limites de la méthode proposée mettent en évidence la nécessité d'explorer des solutions alternatives et d'orienter les travaux vers d'autres approches capables de mieux répondre aux spécificités des images médicales dermatologiques et en global les images qui ont des zone d'intérêt. Le chapitre suivant abordera ces pistes en détail, en proposant des méthodes adaptées pour surmonter les limites identifiées.

Pour les perspectives futures, nous envisageons d'intégrer des techniques avancées telles que l'apprentissage profond (Deep Learning) pour affiner encore le processus de compression. En effet, les réseaux neuronaux et autres modèles d'apprentissage automatique peuvent aider à identifier des structures complexes dans les images et à optimiser davantage les paramètres de compression, tout en minimisant la perte de qualité. Ainsi, l'intégration de ces techniques pourrait ouvrir de nouvelles avenues pour améliorer non seulement le taux de compression, mais aussi la qualité des images dans des applications de plus en plus exigeantes, répondant aux besoins croissants en termes de traitement d'images de haute qualité dans des environnements à faible bande passante et avec des ressources limitées.

Chapitre 5 :

Approche hybrides de compression des images basée super pixel, découpage et CNN

5.1 Introduction

Dans ce chapitre, nous présenterons en détail notre contribution et démontrerons les avancées significatives qu'apporte cette recherche dans le domaine de la compression des images. L'objectif de ce travail est de proposer une solution au problème de stockage des images en utilisant une approche qui combine l'apprentissage profond (DL) et les super-pixels pour contrôler de manière sélective la dégradation introduite lors de la compression de l'image. Cette solution vise à maintenir une qualité d'image élevée dans les régions contenant des informations pertinentes, tout en autorisant une compression et une dégradation accrues dans les zones moins significatives.

5.2 Motivation

La compression d'image est un domaine crucial dans les applications modernes de traitement de l'information, notamment à l'ère de l'explosion des données visuelles. Avec la prolifération des smartphones, des réseaux sociaux et des plateformes de streaming, le besoin de transmettre et de stocker efficacement des images tout en préservant leur qualité est devenu plus important que jamais. Cependant, les méthodes de compression traditionnelles, bien qu'efficaces dans de nombreux cas, présentent des limites lorsqu'il s'agit de répondre à des exigences croissantes en termes de qualité visuelle, de résolution, et de compréhension contextuelle des données visuelles.

Dans ce contexte, l'utilisation de l'apprentissage profond associé à des représentations sémantiques telles que les super-pixels ouvre de nouvelles perspectives. Les approches traditionnelles de compression d'image, telles que JPEG ou PNG, s'appuient sur des modèles mathématiques génériques pour réduire la taille des fichiers, sans considérer la signification contextuelle des différents éléments visuels. Cela peut entraîner une dégradation perceptible de la qualité, en particulier dans des applications critiques comme la médecine ou les systèmes de surveillance.

Les super-pixels permettent de regrouper des régions homogènes au sein d'une image, offrant une représentation plus compacte et sémantiquement cohérente. Ces regroupements peuvent être exploités par des modèles d'apprentissage profond pour mieux comprendre les structures importantes d'une image, réduisant ainsi les pertes de qualité dans les zones critiques tout en optimisant la compression dans les parties moins significatives. Cette combinaison s'inscrit parfaitement dans une approche sémantique de la compression, où le contenu et la signification des données visuelles jouent un rôle central.

En outre, l'évolution des réseaux neuronaux convolutifs (CNN) et des architectures avancées comme les auto-encodeurs et les GANs (Generative Adversarial Networks) a permis de repousser les limites de la compression. Ces modèles permettent de capter des détails subtils et de les reconstruire avec une précision exceptionnelle, tout en respectant les contraintes de stockage et de transmission.

L'objectif de ce projet est donc de développer une approche novatrice de compression d'image en intégrant les techniques d'apprentissage profond et les super-pixels pour améliorer la qualité perçue et l'efficacité. Cette méthodologie vise non seulement à optimiser les ressources mais aussi à répondre aux besoins des applications modernes, qui exigent des performances élevées tout en respectant des contraintes strictes en termes de bande passante et de stockage.

Pour atteindre cet objectif, les étapes suivantes sont mises en œuvre :

- **Méthode des super-pixels** : d'abord, une méthode de segmentation en super-pixels est appliquée pour regrouper les pixels homogènes de l'image. Les super-pixels sont de petites régions qui partagent des caractéristiques similaires, permettant ainsi une analyse plus ciblée du contenu de l'image.
- **Partitionnement et classification** : ensuite, l'image est divisée en différentes parties, chacune étant classée selon sa pertinence à l'aide d'un modèle d'apprentissage profond (DL). Ce modèle DL est entraîné pour identifier et différencier les régions pertinentes des moins pertinentes, en se basant sur des caractéristiques apprises.
- **Attribution d'un facteur de compression** : en fonction des résultats de la classification, un facteur de compression est assigné à chaque partie de l'image. Ce facteur détermine le niveau de compression et de dégradation appliqué à chaque région. Les parties les plus pertinentes reçoivent un facteur de compression réduit, minimisant la perte de qualité de l'image, tandis que les régions moins importantes se voient attribuer des facteurs de compression plus élevés, permettant une compression et une dégradation accrues.
- **Évaluation sur diffèrent types des images** : enfin, l'approche proposée est testée sur plusieurs types des images. Les résultats obtenus sont comparés aux techniques les plus récentes, afin de démontrer que la méthode proposée offre une meilleure préservation des informations pertinentes tout en optimisant la compression.

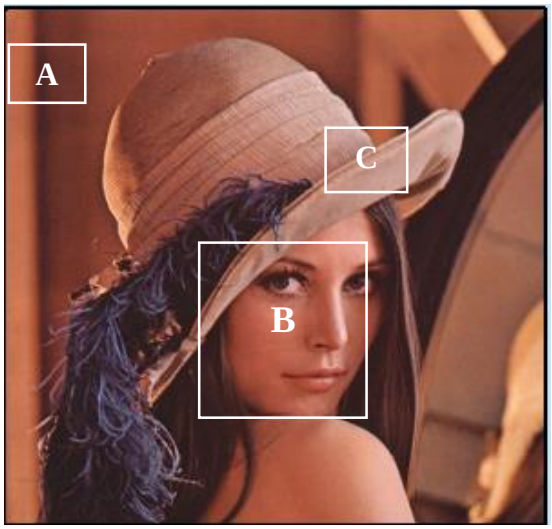
L'objectif global de ce travail est de proposer une solution de compression des images qui équilibre le besoin d'un stockage optimisé avec l'exigence de préserver la qualité et les informations essentielles de l'image. En combinant l'apprentissage profond et les super-pixels, l'approche permet de contrôler de manière sélective la compression et la dégradation en fonction du contenu de l'image. Cela permet d'augmenter le taux de compression global tout en maintenant la qualité des informations pertinentes.

5.3 Approche Proposée

Dans cette méthode, chaque image a été subdivisée en 16 partitions. La base de données utilisée pour la phase de classification est donc constituée de l'ensemble des partitions de toutes les images. Chaque partition a été classée manuellement en trois catégories :

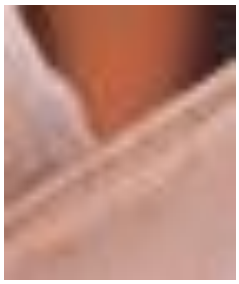
- **ROI** : partition essentielle, qui ne doit subir aucune perte.
- **Non-ROI** : partition de l'arrière-plan où les pertes sont tolérées.
- **Ambiguë** : partition moins précise où une perte minimale est acceptable.

La figure 5.1 illustre un exemple de partitionnement et de classification d'une image de Lina.



A : Exemple de Non-ROI

B : Exemple de ROI



C : Exemple d'Ambiguë

Figure 5.1 : Exemple des partitions de l'image de Lina.

Compte tenu de la complexité des images, qui contiennent souvent une grande quantité de détails, il est essentiel de proposer une approche adaptative de la compression. Dans ce contexte, l'adaptabilité désigne la capacité à ajuster le niveau de perte d'information en fonction du contenu de l'image compressée. Certaines régions de l'image, telles que celles contenant des informations critiques, nécessitent une préservation intégrale, tandis que d'autres zones, comme l'arrière-plan, sont moins cruciales. L'approche que nous proposons vise ainsi à compresser les images en préservant les détails vitaux dans les zones d'intérêt sans perte d'information, tout en permettant une compression avec perte pour les zones moins importantes. Ce principe d'adaptabilité s'intègre bien à la technique de compression JPEG.

Le format JPEG est une technique de compression d'images très répandue, reconnue pour sa compression avec perte basée sur des mesures perceptuelles. Ce procédé permet de réduire la taille des fichiers tout en maintenant la qualité visuelle de l'image à un niveau indiscernable pour l'œil humain. JPEG offre aux utilisateurs un paramètre de « facteur de compression » pour ajuster le niveau de compression selon leurs besoins. Plus le facteur de compression est élevé, plus la compression est forte, ce qui entraîne une diminution de la qualité de l'image. Cette relation inverse entre le facteur de compression et la qualité d'image est illustrée dans la figure 5.2, où le facteur de compression, souvent appelé facteur de qualité (FQ), montre son impact direct sur l'apparence visuelle de l'image finale.

Figure 5.2 : Qualité de l'image compressée par rapport au Facteur de Qualité (FQ).

L'un des principaux avantages de la compression JPEG réside dans la flexibilité qu'elle offre pour contrôler la qualité de l'image compressée, en ajustant le niveau de compression selon les besoins. Ce contrôle se fait par le biais d'un paramètre appelé facteur de qualité (FQ), qui influence directement la balance entre la taille du fichier et la qualité visuelle de l'image résultante.

La figure 5.3 illustre les différentes étapes du processus de compression JPEG, qui sont détaillées en profondeur dans le chapitre 2. Ce processus commence par le découpage de l'image d'origine en blocs de 8x8 pixels, une taille qui facilite le traitement. Ensuite, chaque bloc est soumis à une transformation en cosinus discrète (DCT), qui convertit les valeurs de pixels en une représentation de fréquences. Cette étape permet de concentrer la majeure partie de l'information visuelle dans quelques coefficients, en séparant les détails fins des informations globales de l'image. Après la transformation DCT, chaque bloc passe par une étape de quantification. C'est à ce stade que le FQ intervient de manière décisive : en ajustant le facteur de qualité, on peut modifier l'intensité de la compression. Un FQ plus élevé entraîne une compression plus forte, avec une perte plus importante des détails fins, tandis qu'un FQ plus bas préserve davantage d'informations au détriment de la taille de fichier.

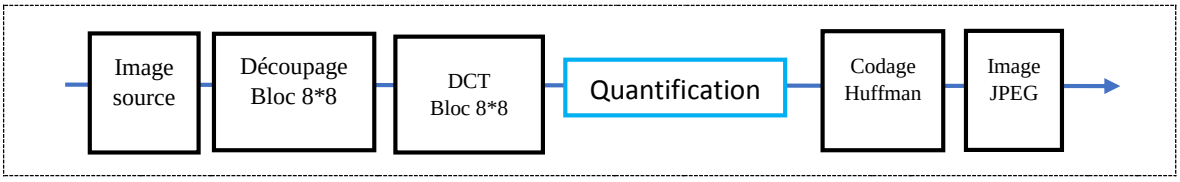


Figure 5.3 : Différentes étapes de compression JPEG.

La quantification dans le processus de compression JPEG vise à réduire le nombre de bits requis pour représenter chaque valeur entière dans la matrice obtenue après l'application de la transformation en cosinus discrète (DCT). En simplifiant ces valeurs, la quantification permet de compresser davantage l'image en minimisant l'importance des détails peu perceptibles à l'œil humain, ce qui contribue à une réduction significative de la taille du fichier.

La matrice de quantification, définie par la formule (1), est au cœur de cette étape. Elle agit comme un filtre en ajustant chaque coefficient de la matrice DCT selon une valeur prédéfinie, ce qui aide à conserver uniquement les informations visuelles les plus essentielles. Ainsi, le niveau de compression peut être modulé par cette matrice de quantification en fonction du facteur de qualité (FQ), permettant aux utilisateurs de faire un compromis entre la fidélité de l'image et le taux de compression souhaité.

$$Q(i,j) = 1 + (1 + i + j)FQ \qquad (1)$$

Où :

- ✓ i: indice de ligne.
- ✓ j : indice de colonne.
- ✓ FQ : facteur de qualité de la matrice de quantification.

La figure 5.4 montre un exemple de matrice de quantification où FQ=10.

31	41	51	61	71	81	91	101
41	51	61	71	81	91	101	111
51	61	71	81	91	101	111	121
61	71	81	91	101	111	121	131
71	81	91	101	111	121	131	141
81	91	101	111	121	131	141	151
91	101	111	121	131	141	151	161
101	111	121	131	141	151	161	171

Figure 5.4 : Matrice de Quantification (FQ=10).

Pour obtenir la version quantifiée de la DCT notée $\hat{}DCT$, les coefficients de la matrice DCT sont divisés par les éléments correspondants d'une matrice de quantification prédéfinie, $Q(i,j)$, puis arrondis à l'entier le plus proche. Ce processus produit ainsi la représentation quantifiée de la DCT. Cette étape joue un rôle crucial dans l'efficacité de la compression.

$$\hat{}DCT(i,j) = [DCT(i,j)/Q(i,j)]. \qquad (2)$$

Le format JPEG traditionnel ne prend pas en compte le contenu de l'image et applique un facteur de qualité (FQ) uniforme à toutes les partitions. Dans l'approche que nous proposons, chaque partition dispose de son propre facteur de qualité, déterminé en fonction de son importance dans le processus de diagnostic. Ce choix repose sur l'impact de chaque partition sur la précision diagnostique globale. En ajustant le FQ pour chaque partition, notre méthode adaptative de compression des images médicales permet de préserver les détails critiques pour le diagnostic, tout en autorisant une compression plus élevée dans les zones moins importantes. Cette personnalisation du processus de compression améliore l'efficacité

globale de la compression des images, en optimisant les besoins de stockage et de transmission. La figure 5.5 illustre les différentes étapes de l'approche proposée.

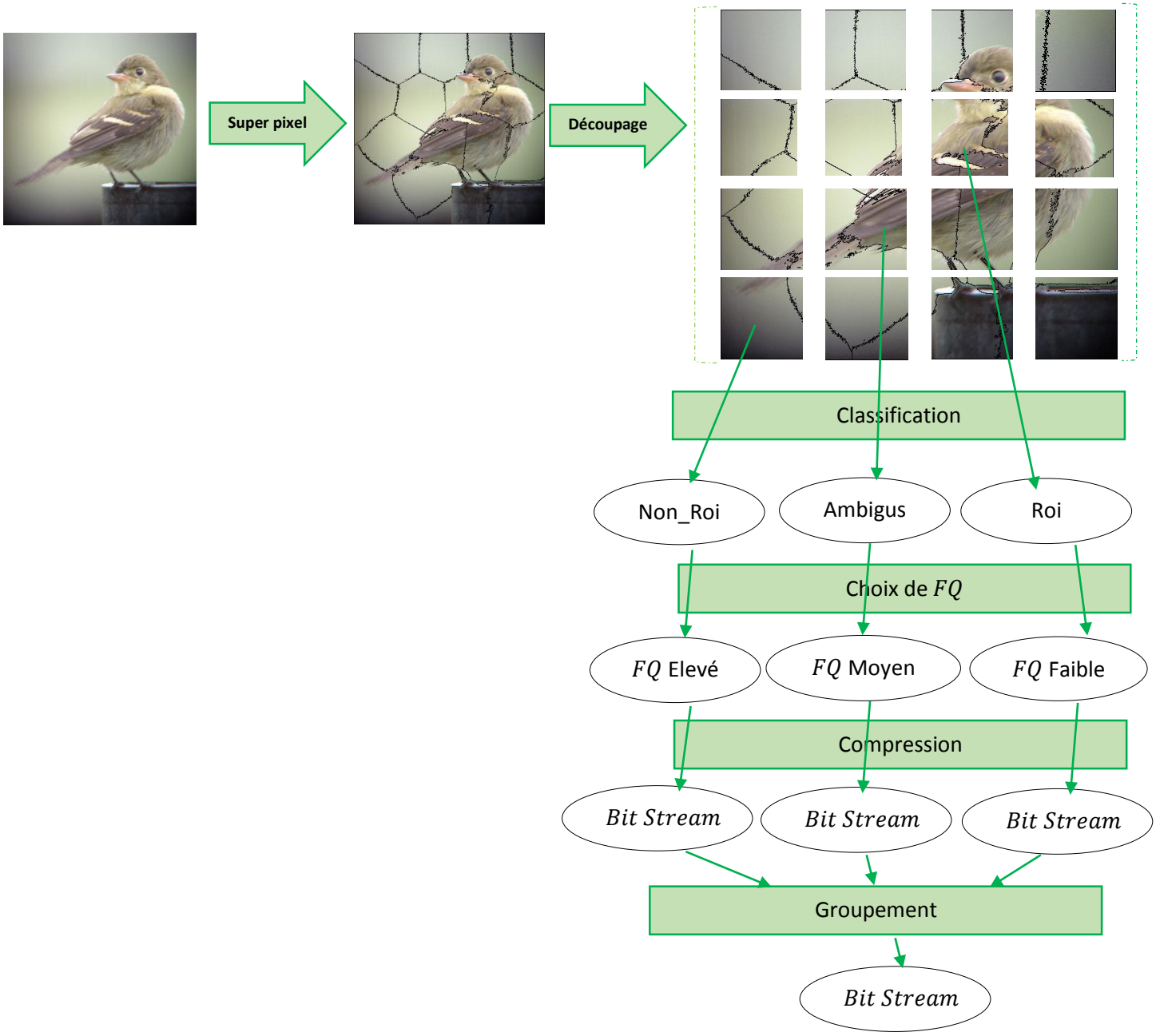


Figure 5.5 : Flux de données de l'Approche Proposée.

5.3.1 Super pixel

Pour optimiser la couverture de la région d'intérêt, nous avons appliqué la technique de segmentation en superpixels appelée SLIC le concept de super pixel est expliqué en détail dans le chapitre3. Cette

méthode regroupe les pixels similaires, permettant ainsi une représentation plus complète et précise de la zone d'intérêt. L'algorithme SLIC [Achanta et al., 2012] est une méthode populaire de segmentation en superpixels. Il permet de générer des superpixels de haute qualité, compacts et de taille uniforme. SLIC est une approche efficace pour regrouper les pixels ayant des couleurs similaires et proches spatialement, en tenant compte des informations de couleur et en exploitant pleinement ces données pour une segmentation précise. L'algorithme 5.1[Achanta et al., 2012] démontre les différentes étapes de l'algorithme SLIC.

Algorithm SLIC Superpixel Segmentation

*/*Initialization*/*
Intialize cluster centers $C_K = [L_K, a_K, b_K, x_K, y_K]^T$ by
Sampling pixels at reglar grid step *S*.
More cluster centers to the lowest gradient position
in a
3x3 neighborhood.
Set label *l(i)*= -1 for each pixel *i*.
Set distance *d(i)*= ∞ for each pixel *i*.
Repeat
/*Assignment*/
For each cluster center C_K **do**
  For each pixel *i* in a *2S* x *2S* region around C_K **do**
    Compute tehe distance *D* between C_K and *i*.
    If *D*< d(*i*) **then**
      Set *d(i)= D*
      Set *l(i)= k*
    End if
  End for
End for
*/*Update*/*
Compute new cluster centers.
Compte residual error *E*.
Until *E*≤ threshold

Algorithme 5.1: SLIC segumentation par superpixel.

5.3.2 Partitionnement

Après la segmentation, l'image est divisée en *N* parties égales. Dans ce travail, nous avons fixé *N = 16*. Chaque partition est alors traitée comme une unité indépendante, permettant un traitement spécifique et adapté.

5.3.3 Classification

L'évaluation de la pertinence de chaque partition de l'image est effectuée à l'aide de l'apprentissage profond (DL). Nous avons choisi le modèle AlexNet pour classifier chaque partition. Entraîné avec la base de données décrite en Introduction Générale, AlexNet attribue à chaque partition l'une des trois catégories suivantes : ROI (région d'intérêt), ambiguë, ou Non_ROI (région sans importance diagnostique). L'architecture AlexNet [Chavan et al., 2020] est le modèle de réseau de neurones convolutifs (CNN) employé dans ce travail. Le tableau 5.1 et la Figure 5.6 présentent l'architecture détaillée de ce modèle.

Layer	#Filters/	Filter size	Stride	Padding	Activation Function
Input	1	-	-	-	-
Convolution	96	11x11	4x4	3x3	relu
Max pooling	-	3x3			relu
convolution	256	5x5	1x1	2x2	relu
Max pooling	-	3x3	-	-	relu
Convolution	384	3x3	-	-	relu
Convolution	384	3x3	-	-	relu
Convolution	256	3x3	-	-	relu
Max pooling	-	3x3	-	-	relu
Fully connected	4096	-	-	-	relu
Fully connected	4096	-	-	-	relu
Output	1	-	-	-	softmax

Tableau 5.1 : Architecture D'AlexNet.

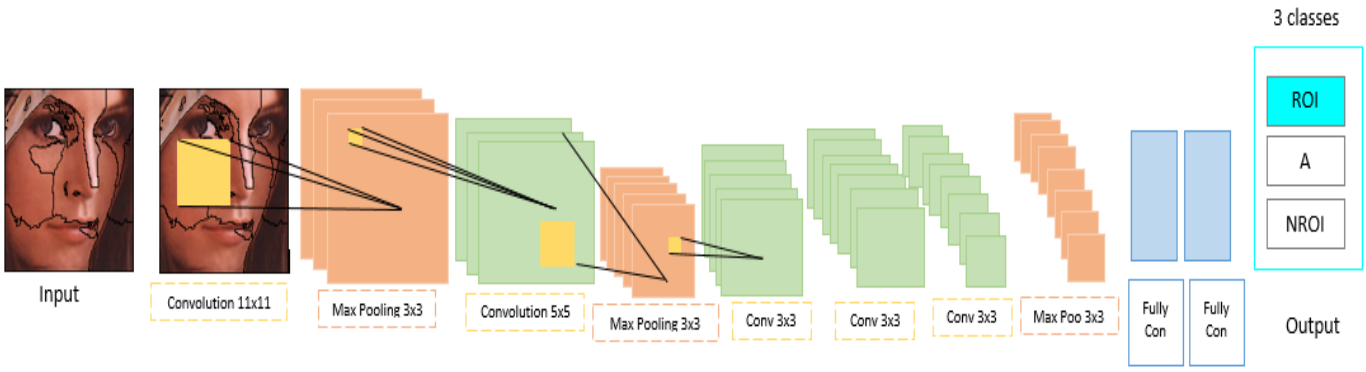


Figure 5.6 : Architecture D'AlexNet.

5.3.4 Calcul du FQ

La valeur du facteur de qualité (FQ) est déterminée en fonction de la classification de chaque partition. Ainsi, la valeur FQ n'est pas uniforme pour l'ensemble de l'image I. Différentes valeurs de FQ, notées FQ_{Pi}, sont attribuées à chaque partition Ip_i de l'image. Selon le résultat de la classification par apprentissage profond (DL), une valeur FQ_{Pi} spécifique est assignée à chaque partition Ip_i. En pratique, les valeurs de FQ sont fixées soit en fonction des objectifs, soit par expérimentation. Par exemple, pour une partition classée comme ROI (région d'intérêt), l'objectif est de la compresser tout en évitant toute perte d'information. Dans ce cas, aucune perte n'est autorisée, et la valeur de FQ doit être choisie en conséquence. En revanche, pour une partition classée comme ambiguë ou Non_ROI, la valeur de FQ est déterminée empiriquement. Sur la base de la décision du modèle DL, la valeur FQ_{Pi} appropriée pour chaque partition de l'image est sélectionnée à l'aide de la formule (3).

$$FQ_{Pi} = \begin{cases} Si\ ROI\ Alors\ FQ = FQ^{ROI} \\ Si\ Non-ROI\ Alors\ FQ = FQ^{NROI} \\ Sinon \quad FQ = FQ^{Amb} \end{cases} \qquad (3)$$

$$FQ_I = [FQ_{P1}, FQ_{P2}, \dots, FQ_{Pn}]$$

Où :

- FQ_{Pi}: FQ de la partition pi.
- FQ_I : Vecteur de tous les FQ_{Pi}.
- $FQ = FQ^{NROI}$ Pour l'image classée comme Non-ROI, étant donné que cette zone représente l'arrière-plan et ne contient pas d'informations pertinentes, le facteur de qualité devient moins important. Elle peut ainsi être compressée à un taux élevé, permettant ainsi d'économiser de l'espace de stockage.
- $FQ = FQ^{Amb}$ Pour la zone ambiguë de l'image, qui contient à la fois des régions ROI et Non-ROI, le choix optimal est de privilégier une compression moyenne, afin de trouver un compromis équilibré entre la qualité de l'image et l'économie d'espace de stockage.
- $FQ = FQ^{ROI}$ Pour le ROI de l'image, il est essentiel de préserver la qualité de la région d'intérêt autant que possible. Ainsi, un taux de compression faible est choisi afin de maintenir les informations importantes de l'image.

5.3.5 Compression

Une fois la valeur du facteur de qualité (FQ) déterminée pour une partition donnée P_i, celle-ci sera compressée à un taux faible, élevé ou moyen, en fonction de son classification. Le processus de compression est réalisé de manière à ce qu'il n'y ait aucune différence perceptible dans la qualité de l'image. Les étapes de la compression JPEG sont alors appliquées à chaque partition Ip_i de la manière suivante :

1. Ip_i est segmenté en blocs de pixels de 8 × 8 à partir du coin supérieur gauche.
2. Appliquer la transformée en cosinus discrète (DCT) à chaque bloc segmenté, en commençant par le bloc supérieur gauche et en progressant jusqu'au bloc inférieur droit.
3. Compresser chaque bloc de Ip_i en utilisant la quantification (en utilisant FQ_{Pi}).
4. Les blocs compressés sont conservés sous forme de tableau C_{p_i} qui contient les données compressées de la partition. Ip_i.

$$\forall Ip_i \in I,\ C_{p_i} = JPEG(Ip_i, FQ_{Pi}) \qquad (4)$$

5.3.6 Groupement

Après l'étape de compression, nous obtenons N blocs de flux de bits ($C_{p_i}\ i = 1 \dots N$), qui sont ensuite concaténés pour former les données compressées de l'image *I*. Le résultat de la compression de l'image entière *I* est donné par la formule (5).

$$C_I = C_{p1} \gg C_{p2} \gg \cdots C_{pi} \gg \cdots C_{pn} \qquad (5)$$

$$O\text{ù} \gg est\ \text{é}quivalent\ \text{à}\ l'op\text{é}ration\ de\ concat\text{é}nation$$

Le processus de décompression présente un défi : nous ne disposons pas des valeurs de séquence des FQs utilisés lors de la compression des différentes partitions, et aucune méthode automatique ne peut déterminer cette séquence. Il est donc nécessaire de l'ajouter au code C_I lors de la compression. Une solution simple consiste à insérer directement cette séquence dans le code C_I, mais cela réduit le taux de compression. Une approche plus efficace consiste à compresser cette séquence de valeurs FQ avant de l'ajouter au code C_I. Le codage arithmétique [Howard et Vitter, 1994] s'avère être la meilleure méthode pour ce faire, car nous avons une séquence fixe de 16 valeurs, réparties sur trois symboles. Ainsi, une valeur de codage arithmétique de taille fixe est concaténée au début du code C_I. La figure 5.7 illustre ces deux processus de compression. Le JPEG est appliqué à chaque partition de manière individuelle en utilisant le FQ approprié, tandis que le codage arithmétique est utilisé pour compresser la séquence des FQ. Le code de compression final généré est alors donné par :

$$Comp_I = C_{QF} \gg C_I$$

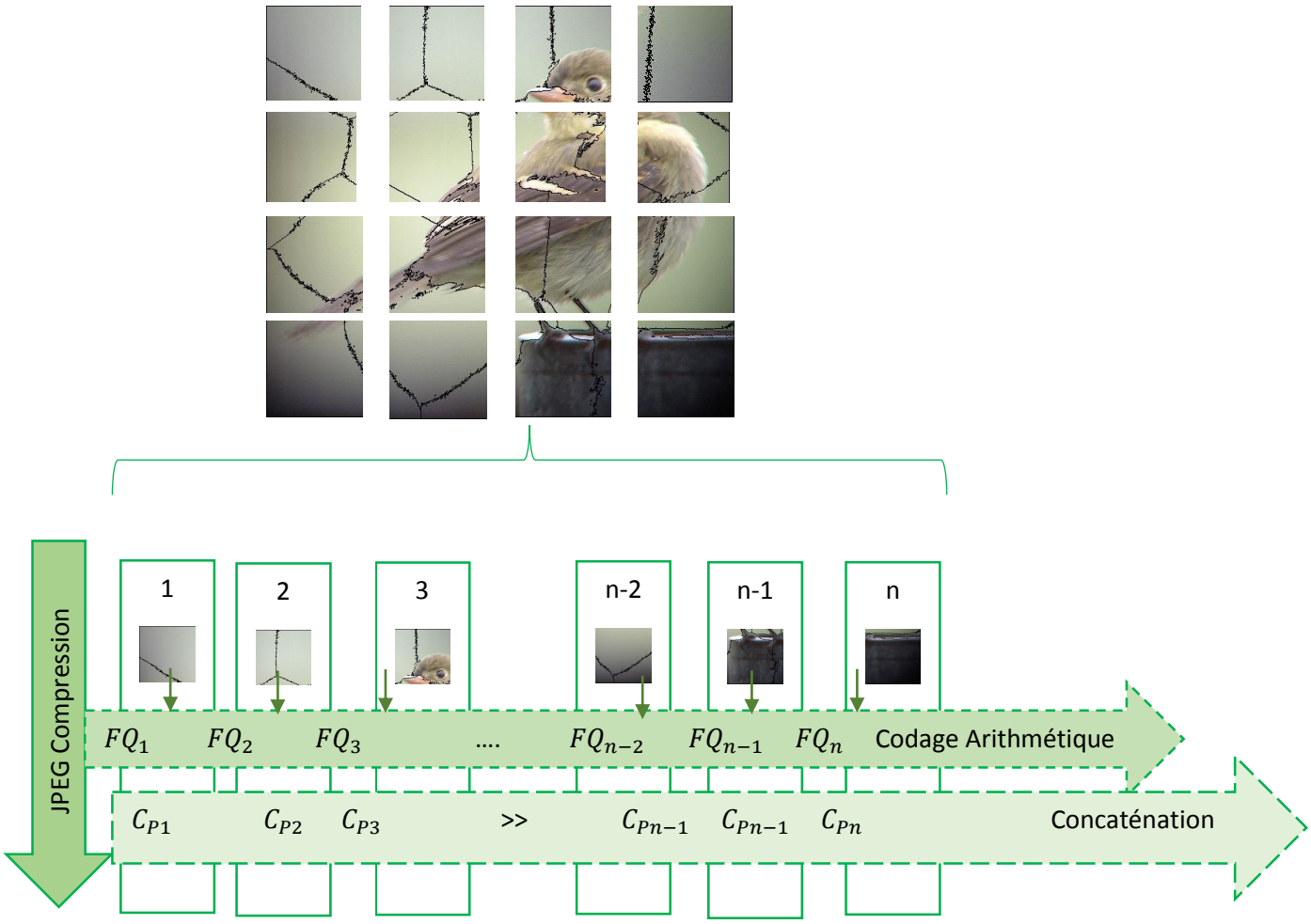


Figure 5.7 : Illustration de la compression de partition et du codage FQ.

5.3.7 Décompression

La phase de décompression s'effectue en suivant un chemin inverse à celui de la compression. Elle peut être décrite en plusieurs étapes successives :

1. **Dégroupement**
   Le processus débute par la lecture du fichier compressé global. Celui-ci contient les différentes parties de l'image, qui ont été préalablement compressées séparément. Ce fichier est alors « dégroupé » pour extraire les 16 sous-fichiers JPEG correspondants aux segments de l'image d'origine.

2. **Décompression individuelle des segments**
   Chaque sous-fichier est ensuite traité indépendamment à l'aide de l'algorithme de décompression JPEG standard. Cette opération permet de restaurer les petites images compressées correspondant à différentes zones de l'image initiale.

3. **Assemblage final (concaténation)**
   Une fois toutes les petites images décompressées, elles sont réassemblées selon leur position d'origine dans la grille de partitionnement (par exemple, une grille 4x4 si 16 segments sont utilisés). Ce processus de concaténation spatiale permet de reconstituer l'image complète dans sa version décompressée.

5.4 Conclusion

Dans ce chapitre, nous avons présenté une méthode innovante de compression d'images médicales qui combine l'apprentissage profond (DL) et la segmentation en superpixels pour améliorer la gestion des informations pertinentes et optimiser l'utilisation de l'espace de stockage. Notre approche a permis de répondre efficacement au défi de la compression d'images en préservant la qualité des zones intéressantes significatives tout en permettant une compression plus agressive des zones non pertinentes.

Les résultats expérimentaux du chapitre suivant montrent que l'utilisation de partitions adaptatives et la personnalisation du facteur de qualité (FQ) pour chaque région de l'image ont permis d'atteindre un compromis optimal entre la réduction de la taille des fichiers et la préservation des détails essentiels pour le diagnostic. Cette méthode a prouvé sa capacité à réduire les besoins en stockage et en transmission tout en garantissant la précision du diagnostic.

Chapitre 6 :

Résultats et Discussion

6.1 Introduction

Dans ce chapitre, nous présentons et analysons les résultats obtenus à partir de l'implémentation de l'approche de compression d'images que nous avons proposée. Nous discutons de l'efficacité de la méthode, en comparant les performances de notre solution avec celles des techniques existantes. Nous évaluerons la qualité des images compressées en termes de préservation des détails et d'efficacité de la compression, tout en tenant compte des exigences en matière de stockage et de transmission des données.

Nous commencerons par présenter les résultats des expérimentations menées sur différentes type des images, en détaillant les métriques utilisées pour évaluer la qualité d'image et la performance de compression. Ensuite, nous discuterons de l'impact de la segmentation, de la classification des partitions et de l'adaptation du facteur de qualité sur les performances globales du système. Enfin, nous analyserons les avantages et les limites de l'approche proposée et suggérerons des pistes d'amélioration pour de futures recherches.

Les résultats obtenus permettent de tirer des conclusions importantes sur la viabilité de notre méthode pour le traitement d'images médicales, en particulier dans le cadre de la compression des données pour un stockage et une transmission optimisés.

6.2 Résultats

6.2.1 Résultats de la Segmentation et de Découpage

L'algorithme de superpixels SLIC est utilisé pour effectuer la segmentation de l'image en regroupant les pixels similaires. Cela permet d'obtenir une représentation plus précise et complète des régions d'intérêt (ROI). Ensuite, l'image segmentée est divisée en 16 partitions, ce qui facilite l'attribution d'un facteur de qualité (FQ) adapté à chaque partition. Ce choix du FQ permet de contrôler la perte d'information en fonction du contenu spécifique de chaque partition. La figure 6.1 illustre les résultats de la segmentation et du découpage appliqués à quelques types d'images.

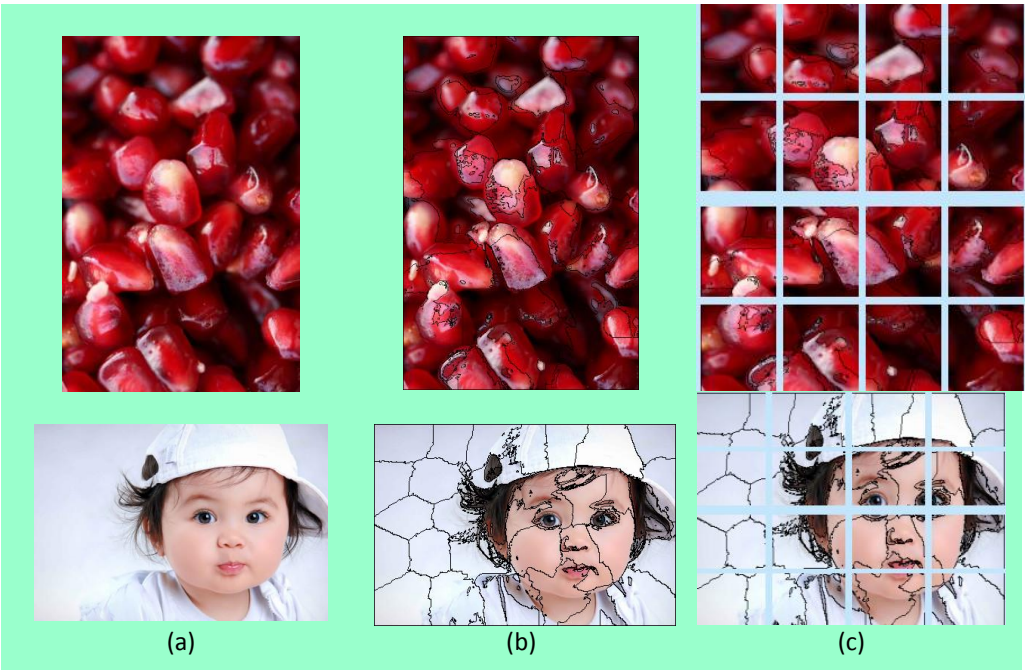


Figure 6.1 : Exemples de segmentation et de découpage, (a) image originale, (b) résultat SLIC, (c) résultat du découpage.

6.2.2 Résultats de Classification

Nous avons entraîné le modèle AlexNet sur 50 époques avec un lot de 20 par itération. Ce modèle classe chaque partition en l'une des trois catégories : classe Ambiguë, classe ROI, et classe Non-ROI. Le tableau 6.1 présente les performances du modèle, évaluées en termes de précision, rappel, score F1 et exactitude.

Accuracy	*Precision*	*Recall*	*F1 score*
66.67	72.22	66.67	65.56

Tableau 6.1 : Performances d'AlexNet

Le tableau 6.2 présente le résultat de la décision DL pour chaque partie des dix images de la figure 6.2.

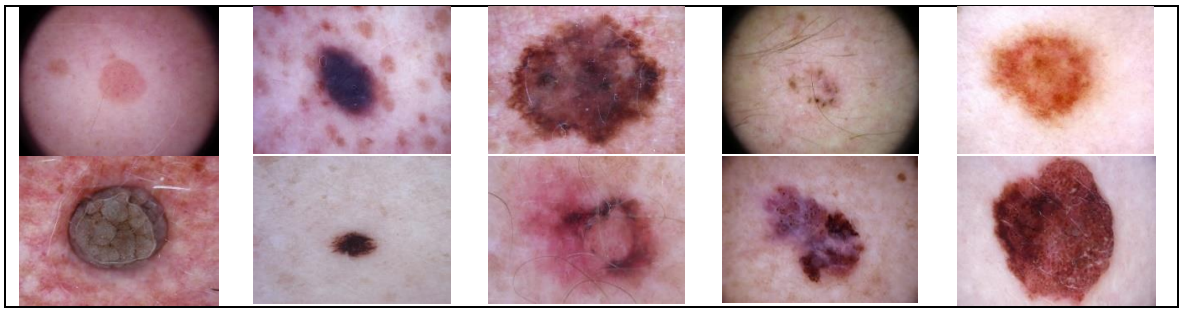


Figure 6.2 : Exemple de dix images de peau testées : première ligne de gauche à droite, respectivement les images 1, 2, 3, 4 et 5 ; deuxième ligne de gauche à droite, respectivement les images 6, 7, 8, 9 et 10.

Où :

- Pi : numéro de partition i de l'image *I*
- A : classe ambiguë
- R : classe ROI
- N : classe Non- ROI.

Résultat du DL pour l'approche proposée																
Pi	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**	**P13**	**P14**	**P15**	**P16**
1	*R*	*N*	*N*	*A*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*A*	*N*	*N*	*A*
2	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*
3	*N*	*A*	*A*	*N*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*N*	*R*	*A*	*N*
4	*A*	*N*	*N*	*A*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*A*	*N*	*N*	*A*
5	*N*	*N*	*N*	*N*	*N*	*A*	*A*	*N*	*N*	*A*	*A*	*N*	*N*	*N*	*N*	*N*
6	*N*	*A*	*N*	*A*	*N*	*A*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*
7	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*
8	*N*	*N*	*N*	*N*	*N*	*A*	*N*	*N*	*N*	*N*	*N*	*A*	*N*	*N*	*N*	*N*
9	*N*	*N*	*N*	*N*	*N*	*N*	*R*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*	*N*
10	N	A	A	N	N	A	A	N	N	A	A	N	N	A	A	N

Tableau 6.2 : Exemple de classification des partitions des dix images (R : ROI, N : Non-ROI et A : Ambiguë).

En fonction de la décision d'AlexNet, une valeur FQ appropriée est affectée à chaque partition. Nous adoptons :

- $^{FQROI} = 20$
- $^{FQAmb} = 50$
- $^{FQNROI} = 100$

Le tableau 6.3 présente les valeurs de FQ pour chaque partie des dix partitions énumérées dans la figure 6.2.

Valeurs FQ de l'approche proposée																
FQ i	**FQ 1**	**FQ 2**	**FQ 3**	**FQ 4**	**FQ 5**	**FQ 6**	**FQ 7**	**FQ 8**	**FQ 9**	**FQ10**	**FQ11**	**FQ12**	**FQ13**	**FQ14**	**FQ15**	**FQ16**
1	*20*	*100*	*100*	*50*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*50*	*100*	*100*	*50*
2	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*
3	*100*	*50*	*50*	*100*	*50*	*50*	*50*	*50*	*50*	*50*	*50*	*50*	*100*	*20*	*50*	*100*
4	*50*	*100*	*100*	*50*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*50*	*100*	*100*	*50*
5	*100*	*100*	*100*	*100*	*100*	*50*	*50*	*100*	*100*	*50*	*50*	*100*	*100*	*100*	*100*	*100*
6	*100*	*50*	*100*	*50*	*100*	*50*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*
7	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*
8	*100*	*100*	*100*	*100*	*100*	*50*	*100*	*100*	*100*	*100*	*100*	*50*	*100*	*100*	*100*	*100*
9	*100*	*100*	*100*	*100*	*100*	*100*	*20*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*
10	100	50	50	100	100	50	50	100	100	50	50	100	100	50	50	100

Tableau 6.3 : Valeurs FQ

6.2.3 Résultats de Compression

La figure 6.3 illustre comment la perte d'information varie en fonction de la nature des différentes zones de l'image. Les partitions identifiées comme régions d'intérêt (ROI) sont compressées avec un facteur de qualité faible, minimisant ainsi la perte d'information afin de préserver au maximum les détails critiques. En revanche, les partitions de moindre importance, situées en dehors des zones ROI, sont compressées avec un facteur de qualité plus élevé, ce qui entraîne une perte d'information plus importante mais permet une réduction substantielle de la taille des données. Les régions ambiguës, qui contiennent potentiellement à la fois des éléments d'intérêt et des zones secondaires, sont compressées de manière intermédiaire, avec une perte modérée, équilibrant ainsi la qualité et le gain d'espace de stockage.

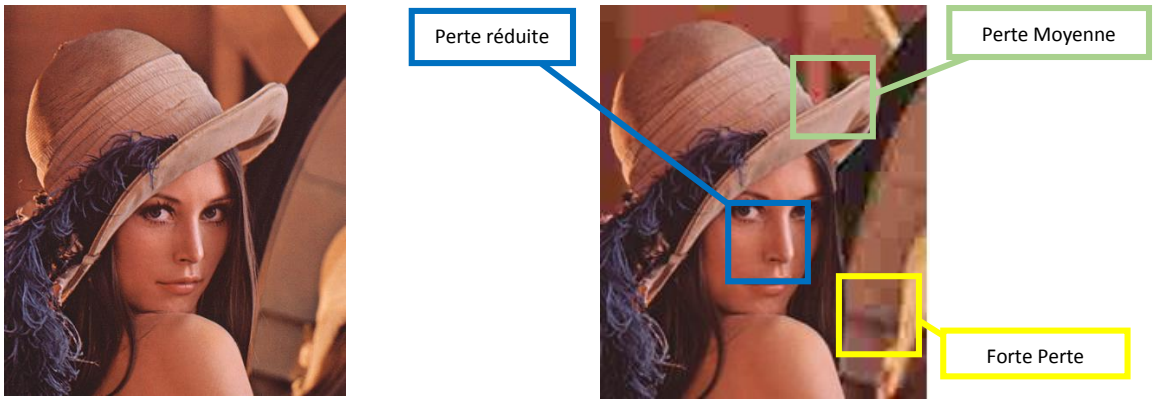


Image Originale Image Compressée par l'approche Proposée

Figure 6.3 : Niveaux de perte d'informations.

Outre la qualité d'image, plusieurs autres mesures de performance ont été utilisées, notamment :

- ✓ Le *taux de compression (Compression ratio)* Le CR est défini comme le nombre de bits représentant la taille de l'image originale par rapport au nombre de bits représentant l'image compressée. Le CR peut être déterminé à l'aide de la formule (1) [Muthalaguraja et Kalavathi, 2000]..

$$CR = \frac{N*M*3}{T} \qquad (1)$$

Ou :

N : longueur de l'image
M : largeur de l'image
T : taille du flux de bits compressés.

- ✓ Le gain d'espace (Space saving) Le pourcentage de gain d'espace est une mesure de la quantité d'espace de stockage économisée grâce à la méthode de compression proposée. Cette mesure permet d'évaluer l'efficacité de la compression en indiquant la proportion de réduction de taille par rapport à l'image originale. Elle est particulièrement utile pour les applications où l'optimisation de l'espace de stockage est cruciale, comme dans le cas des images médicales à haute résolution. Le calcul du pourcentage de gain d'espace est exprimé par la formule suivante (2) [Sabbavarapu et al., 2021], qui prend en compte la taille initiale de l'image et la taille de l'image compressée. En général, cette mesure de gain d'espace est essentielle pour évaluer la balance entre la qualité visuelle préservée et l'espace économisé, indiquant ainsi la performance de la méthode de compression dans des conditions pratiques [Sabbavarapu et al., 2021].

$$SS\% = 1 - \frac{taille\ de\ l'image\ compressée}{taille\ de\ l'image\ originale} * 100 \qquad (2)$$

- ✓ L'erreur quadratique moyenne (MSE) et le rapport signal-bruit maximal (PSNR) sont des

indicateurs clés pour évaluer la différence entre l'image décompressée et l'image originale. Le PSNR et le MSE mesurent le niveau de distorsion introduit par la compression : une valeur PSNR élevée indique une meilleure qualité d'image, tandis qu'une valeur MSE faible correspond à un degré élevé de similitude entre les deux images. Les formules permettant de calculer les valeurs de PSNR et de MSE sont les suivantes [Li et al., 2022] :

$$MSE = \frac{1}{\mathrm{M} \times \mathrm{N}} \sum_{i=1}^{M} \sum_{j=1}^{N} (\mathrm{p(i,j)} - \mathrm{c(i,j)})^2 \qquad (3)$$

$$PSNR = 10lg\left(\frac{I_{max}^2}{MSE}\right) \qquad (4)$$

où M×N est la taille de l'image, P(i,j) est la valeur de gris du pixel dans l'image originale, C(i,j) est la valeur de gris du pixel dans l'image compressée, et Imax est la valeur maximale du pixel de l'image originale [Li et al., 2022].

Le tableau 6.4 présente les valeurs des mesures de performance précédentes, calculées sur les dix images illustrées dans la figure 6.2.

Image	*CR (%)*	*PSNR*	*MSE*	*SS (%)*
1	*43.94*	24.99	140.61	97,72
2	*31.76*	31.20	49.28	96,85
3	*24.57*	30.51	57.82	95,93
4	*30.07*	26.85	134.27	96,67
5	*29.63*	35.58	18.01	96,63
6	*25.77*	29.06	80.77	96,12
7	*33.95*	32.43	37.13	97,05
8	*28.97*	31.39	47.19	96,55
9	*30.31*	39.61	7.12	96,70
10	*25.93*	32.39	37.52	96,14
Moyenne	***30.49***	**31.40**	**60.97**	**96,72**

Tableau 6.4 : Autres mesures de l'approche proposée.

En appliquant une compression minimale (20 % de facteur de qualité), comme illustré dans la figure 6.3, la qualité de la zone d'intérêt (ROI) est efficacement préservée. Ce niveau de compression, sélectionné avec soin, minimise la perte d'information et garantit la conservation des détails essentiels du ROI. L'algorithme réussit ainsi à maintenir une haute qualité pour la zone d'intérêt.

L'évaluation quantitative de l'approche de compression proposée repose sur quatre indicateurs : le taux de compression (CR), l'erreur quadratique moyenne (MSE), le rapport signal-bruit maximal (PSNR) et le pourcentage de gain d'espace (SS).

Le **taux de compression (CR)** permet de quantifier la réduction de taille des images, un critère crucial dans la gestion des données visuelles massives, notamment dans les environnements nécessitant des ressources de stockage limitées. L'**erreur quadratique moyenne (MSE)**, quant à elle, mesure la différence entre l'image originale et l'image compressée, ce qui permet de détecter toute perte d'information perceptible et de garantir que l'intégrité visuelle des régions essentielles de l'image est préservée. En complément, le **rapport signal-bruit maximal (PSNR)**

indique la qualité de l'image décompressée en relation avec l'originale : plus le PSNR est élevé, plus la qualité de l'image restituée se rapproche de celle de l'image d'origine. Enfin, le **gain d'espace (SS)** exprime le pourcentage d'espace économisé grâce à la méthode de compression, illustrant l'efficacité globale de la technique en termes de stockage.

Le tableau 6.4 présente les valeurs de ces mesures pour les dix images de la figure 6.2. En moyenne, les résultats obtenus sont les suivants : un taux de compression (CR) de 30.49, une MSE de 60.97, un PSNR de 31.40 et un gain d'espace (SS) de 96.72 %. Ces valeurs attestent de l'efficacité et de la fiabilité de l'approche de compression proposée.

6.3 Discussion

6.3.1 Comparaison de l'approche proposé et JPEG

Ce projet introduit une méthode de compression JPEG adaptative basée sur un facteur de qualité (FQ) ajusté selon la pertinence des différentes régions d'une image. Contrairement à la compression JPEG classique, où une seule valeur de FQ est appliquée de manière uniforme sur toute l'image, cette méthode adaptative ajuste le FQ pour chaque partition en fonction de sa classification comme ROI (zone d'intérêt), ambiguë, ou Non-ROI. Ce principe permet de préserver les informations cruciales dans les zones d'intérêt tout en appliquant une compression plus élevée aux régions moins importantes, optimisant ainsi la balance entre qualité de l'image et économies d'espace.

6.3.1.1 Implémentation de l'approche proposée sur des images contenant des zones d'intérêt

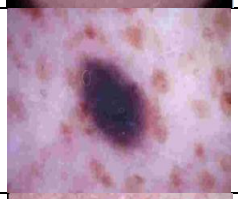
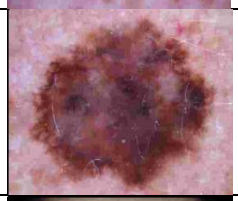
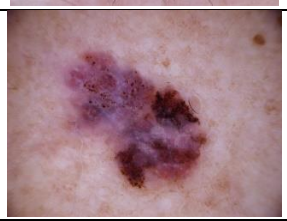
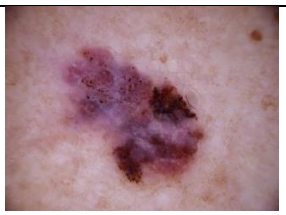
6.3.1.1.1 Implémentation de l'approche proposée sur des images médicales dermatologiques

Les résultats expérimentaux de l'implémentation de l'approche proposée sur la base de données HAM10000 (*Human Against Machine with 10,000 training images*), développée par Philipp Tschandl [Tschandl et al., 2018], démontrent une amélioration significative par rapport à la compression JPEG traditionnelle. Les tableaux 6.5 et 6.6 mettent en évidence cette supériorité : non seulement la qualité de restitution de l'image est préservée dans les zones critiques, mais les gains d'espace sont également substantiels, permettant une utilisation plus efficiente du stockage. Par exemple, la méthode proposée a permis de maintenir une haute valeur de PSNR dans les zones d'intérêt tout en atteignant des taux de compression très avantageux, ce qui se traduit par une importante réduction des besoins de stockage et des coûts de transmission, tout en conservant les détails diagnostiques essentiels. [Lemnadjlia et al., 2024]

Ces résultats montrent que la méthode FQ adaptative est particulièrement bien adaptée aux applications de compression d'images médicales, où la préservation de l'information diagnostique est primordiale. En ajustant dynamiquement le FQ pour chaque partie de l'image,

cette méthode combine haute qualité de restitution et économie d’espace, illustrant ainsi une avancée dans le domaine de la compression adaptative.

Le tableau 6.5 présente les résultats de la compression appliquée aux dix images de référence, comparant le JPEG standard (avec un FQ de 50 %) à l'approche proposée. En analysant ces résultats, on observe clairement la supériorité de l'approche proposée en matière de préservation des informations dans la région d'intérêt (ROI).

Image Originale	JPEG	Approche Proposée	PSNR
			PSNR JPEG = 38.24 PSNR Approche =24.99 PSNR ROI = 40.83
			PSNR JPEG = 37 PSNR Approche = 31.20 PSNR ROI =40
			PSNR JPEG = 34 PSNR Approche = 30.51 PSNR ROI = 37.16
			PSNR JPEG = 36.38 PSNR Approche = 26.85 PSNR ROI = 40.43
			PSNR JPEG = 35.92 PSNR Approche = 35.85 PSNR ROI = 38.63
			PSNR JPEG = 34.37 PSNR Approche = 29.06 PSNR ROI = 37.16
			PSNR JPEG = 37.63 PSNR Approche = 32.43 PSNR ROI = 40.68
			PSNR JPEG = 35.54 PSNR Approche = 31.39 PSNR ROI = 38.47
			PSNR JPEG = 36 PSNR Approche = 39.61 PSNR ROI = 39.05

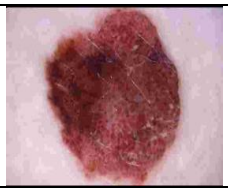
			PSNR JPEG = 34.49 PSNR Approche = 32.39 PSNR ROI = 37.87

Tableau 6.5 : Comparaison de la qualité des images médicales dermatologique restaurée.

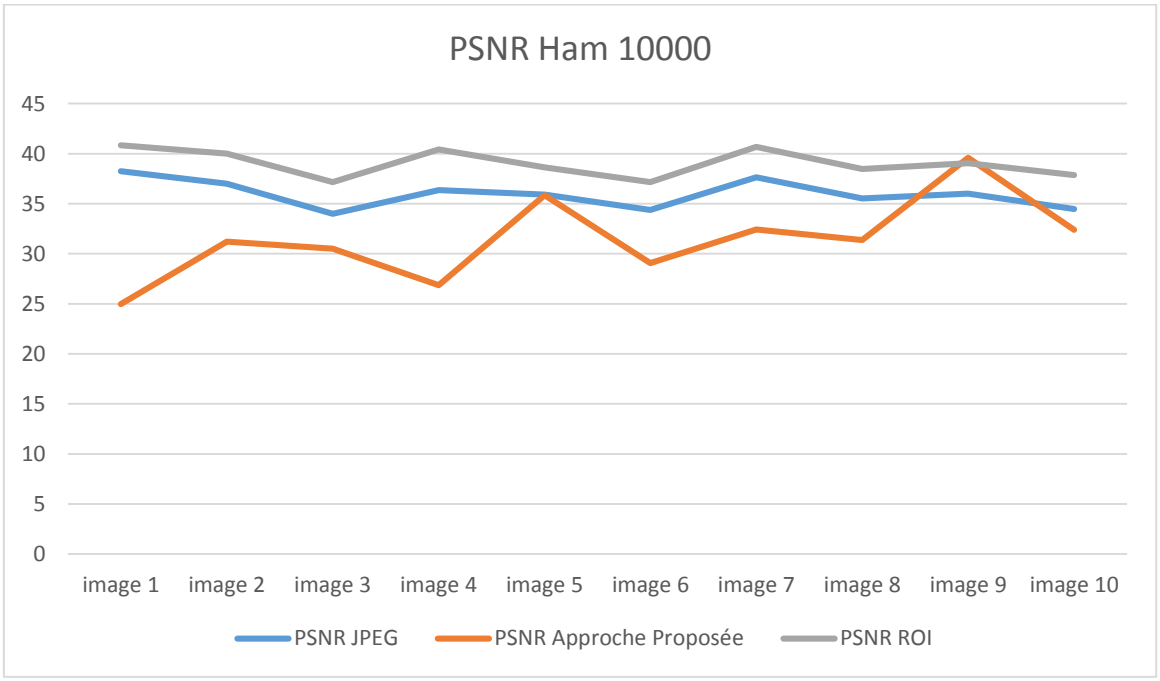


Figure 6.4 : Comparaison des PSNRs des images médicales dermatologiques.

Le tableau 6.6 compare le taux de compression obtenu avec l'approche proposée à celui du JPEG en utilisant trois valeurs de FQ (100 %, 50 % et 20 %). Il apparaît que le JPEG avec un FQ de 100 % ou de 50 % offre un meilleur taux de compression que notre approche. Ce résultat est attendu, car la compression dans le cas du JPEG est appliquée uniformément sur l'ensemble de l'image, tandis que notre méthode cible des régions spécifiques pour préserver les informations essentielles. Cependant, lorsque le JPEG utilise un FQ de 20 %, l'approche proposée obtient un taux de compression nettement supérieur. Cela s'explique par la forte compression appliquée à certaines parties de l'image, ce qui permet de réduire efficacement la taille tout en maintenant la qualité des zones critiques.

Image Originale		*JPEG (FQ = 50%)*		*JPEG (FQ = 20%)*		*JPEG (FQ = 100%)*		*Approche Proposée*	
Taille		*Taille (Kb)*	*CR*	*Taille (Kb)*	*CR*	*Taille (Kb)*	*CR*	*Taille (Kb)*	*CR*
1	791 Kb	14.85	53,27	25.63	30,86	8.88	89,08	*18*	*43,94*
2	791 Kb	18.94	41,76	32.96	24,00	9.47	83,53	*24.9*	*31,76*
3	791 Kb	25.12	31,49	43.31	18,26	10.61	74,55	*32.2*	*24,57*
4	791 Kb	20.26	39,04	34.83	22,71	9.86	80,22	*26.3*	*30,07*
5	791 Kb	20.19	39,18	35.46	22,31	9.36	84,51	*26.7*	*29,63*
6	791 Kb	23.58	33,55	39.72	19,91	10.54	75,05	*30.7*	*25,77*
7	791 Kb	17.39	45,49	30.98	25,53	8.75	90,40	*23.3*	*33,95*
8	791 Kb	20.53	38,53	36.18	21,86	9.57	82,65	*27.3*	*28,97*
9	791 Kb	20.10	39,35	34.73	22,78	9.63	82,14	*26.1*	*30,31*
10	791 Kb	23.40	33,80	40.06	19,75	10.68	74,06	*30.5*	*25,93*
Moyenne			**39,55**		**22,80**		**81,62**		***30,49***

Tableau 6.6 : Taille de stockage et CR de Jpeg avec différentes valeurs et l'approche proposée.

D’après la figure 6.4, on constate que le PSNR de la méthode proposée est inférieur à celui obtenu avec la compression JPEG standard. Cela indique une dégradation plus importante de la qualité de l’image dans son ensemble lorsque la méthode proposée est utilisée. Cette baisse du PSNR global peut s'expliquer par le fait que la méthode concentre davantage les ressources de codage sur des zones spécifiques de l’image, au détriment des autres régions. Cependant, lorsqu’on s’intéresse uniquement à la zone d’intérêt (ROI), les résultats montrent un PSNR nettement supérieur pour la méthode proposée par rapport à JPEG. Cela signifie que, malgré une qualité globale réduite, la qualité de restitution des zones importantes (ROI) est mieux préservée, ce qui constitue l’objectif principal de cette méthode. Ce compromis entre qualité globale et qualité ciblée est particulièrement pertinent dans les applications où certaines régions de l’image sont plus critiques que d'autres.

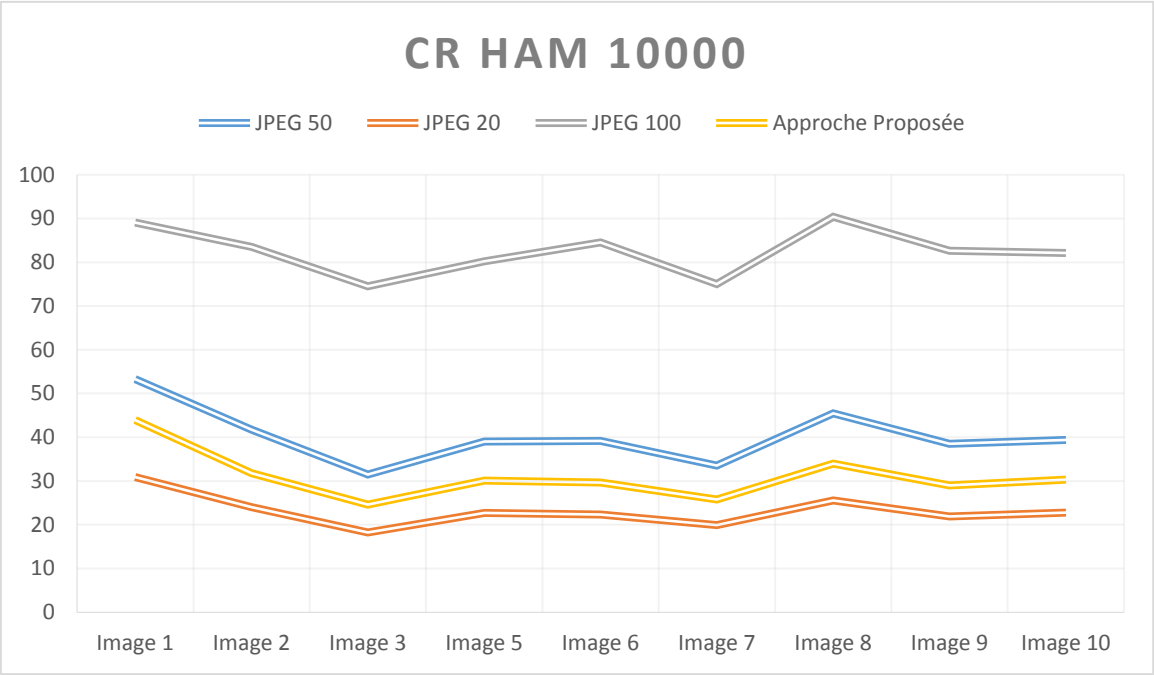


Figure 6.5 : Comparaison des CRs des images médicales dermatologique.

La figure 6.5 présente une comparaison des taux de compression obtenus avec la méthode JPEG pour différents facteurs de qualité (FQ = 20, 50 et 100), ainsi qu'avec la méthode proposée. On observe que l'utilisation de JPEG avec un facteur de qualité élevé (FQ = 100) permet d'atteindre des taux de compression supérieurs. Cependant, cela s'accompagne d'une dégradation notable de la qualité visuelle globale des images, notamment en raison d'un lissage excessif qui peut altérer les détails fins.

En revanche, la méthode proposée offre un compromis intéressant. Les résultats obtenus se situent entre ceux du JPEG avec FQ = 20 et ceux avec FQ = 50, tant en termes de taux de compression que de qualité d'image. Surtout, cette méthode se distingue par une meilleure préservation de la qualité dans les zones d'intérêt, particulièrement cruciales dans le contexte des images médicales dermatologiques. Cela signifie que les régions cliniquement pertinentes, telles que les lésions ou anomalies cutanées, conservent une meilleure netteté et fidélité visuelle, ce qui peut favoriser une analyse plus précise par les professionnels de santé [Lemnadjlia et al., 2024]..

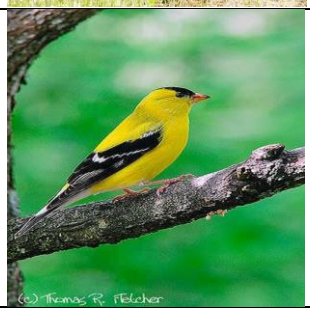
6.3.1.1.2 Implémentation de l'approche proposée sur la base de données CALTECH-BMP

L'implémentation de l'approche proposée a été réalisée sur la base de données CALTECH-BMP [Al-Jarrah, 2018], qui contient un ensemble de 1 500 images d'oiseaux. Les résultats obtenus sont synthétisés dans le tableau 6.7, ainsi que dans les figures 6.6 et 6.7. L'analyse des performances de l'approche proposée en comparaison avec la compression JPEG avec un Facteur de Qualité (FQ) de 50 a révélé des résultats globalement positifs. D'un point de vue

stockage, la méthode a permis une réduction modérée du volume des fichiers, bien que cet avantage soit moins marqué que dans d'autres types d'images. Toutefois, l'efficacité réelle de la méthode ne se limite pas uniquement au gain en espace mémoire. En ce qui concerne la qualité globale des images compressées, les mesures de PSNR ont montré des valeurs presque identiques entre les images restituées par l'approche proposée et celles obtenues avec JPEG ($FQ = 50$). Cela indique que, à l'échelle de l'image complète, les deux méthodes produisent une qualité visuelle similaire.

Le véritable point fort de l'approche proposée réside dans sa capacité à préserver avec une grande précision les zones d'intérêt (ROI) des images d'oiseaux. Contrairement à la compression JPEG standard, qui applique une dégradation uniforme sur toute l'image, l'approche proposée a permis de maintenir une qualité nettement supérieure dans les régions essentielles, comme les plumes, les contours du bec, les yeux et les détails fins du plumage. Cette préservation est confirmée par une valeur élevée du PSNR spécifiquement calculé sur les zones de ROI, démontrant que les informations essentielles sont conservées avec une très faible perte de qualité. Cette particularité est particulièrement précieuse pour des applications où l'identification et l'analyse des détails sont cruciales, comme la reconnaissance automatique des espèces d'oiseaux et les études en ornithologie assistées par ordinateur.

L'implémentation de l'approche proposée sur la base de données CALTECH-BMP a montré une performance notable, notamment grâce à la forte préservation des informations dans les zones d'intérêt des images. Bien que le gain en stockage par rapport au JPEG reste modéré, la qualité des zones essentielles a été nettement mieux préservée, ce qui représente un avantage majeur pour les applications nécessitant une analyse détaillée des images. Ces résultats confirment donc l'efficacité de la méthode proposée pour les images de ce type.

Image Originale (768 KO)	JPEG	Approche Proposée	Taille/ PSNR
			JPEG 48.53 KO/ 24.92 **Approche Proposée** 64 KO/ 24.24 PSNR ROI= 28.76
			**JPEG** 33.60 KO/ 28.38 **Approche Proposée** 47.5 KO/ 28.83 PSNR ROI= 32.96

			JPEG 18.33 KO/ 35.18 **Approche Proposée** 23.2 KO / 34.34 PSNR ROI=39.42
			JPEG 25.42 KO/ 30.38 **Approche Proposée** 34.2 KO / 30.18 PSNR ROI= 43.57
			JPEG 26.63 KO/ 30.14 **Approche Proposée** 33.3 KO / 30.20 PSNR ROI=33.21
			JPEG 49.77 KO/ 28.42 **Approche Proposée** 71.6 KO / 29.99 PSNR ROI= 34.67
			JPEG 20.27 KO/ 31.49 **Approche Proposée** 33.3 KO / 33.90 PSNR ROI= 37.01

			JPEG 15.43 KO/ 31.53 **Approche Proposée** 18.6 KO / 31.02 PSNR ROI= 33.61
			JPEG 16.27 KO/ 32.88 **Approche Proposée** 30.9 KO / 33.83 PSNR ROI= 37.19
			JPEG 16.13 KO/ 32.90 **Approche Proposée** 28.3 KO / 33.46 PSNR ROI= 35.75

Tableau 6.7 : Comparaison de la qualité des images oiseau restaurée.

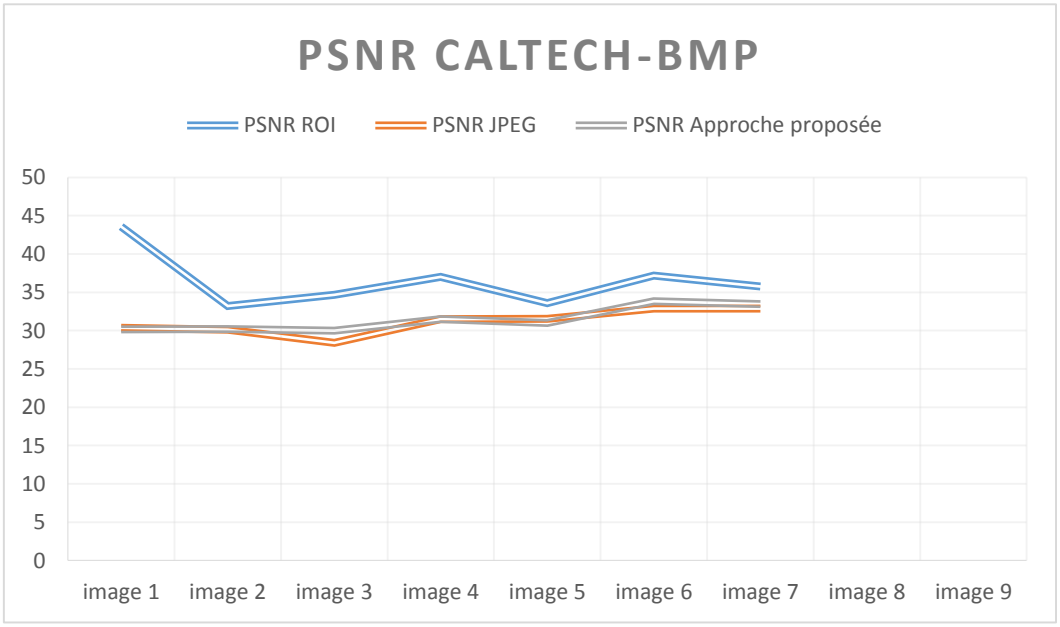


Figure 6.6 : Comparaison de PSNRs des images Caltech-BMP.

La figure 6.6 met en évidence que le PSNR des zones d'intérêt (ROI) obtenues avec l'approche proposée est supérieur à celui obtenu avec la compression JPEG classique et l'approche proposée (image globale). Cela indique que la méthode proposée permet une meilleure restitution de la qualité visuelle dans les régions importantes, démontrant ainsi son efficacité pour préserver les détails essentiels des images. Cependant, la figure 6.7 démontre que la méthode JPEG permet un meilleur gain en termes d'espace de stockage comparé à l'approche proposée

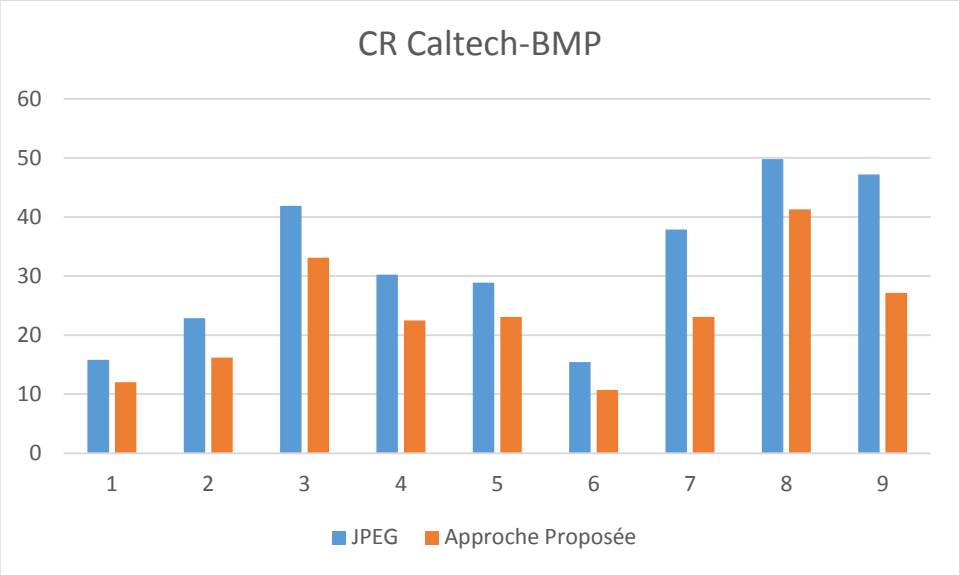


Figure 6.7 : Comparaison des CRs des images de Caltech-BMP.

6.3.1.1.3 Implémentation de l'approche proposée sur la base de données Cars Dataset

Après l'implémentation de l'approche proposée sur la base de données Cars Dataset, une analyse a été réalisée sur un échantillon spécifique composé d'images de la marque « Audi ». Cet échantillon comprend 814 images destinées à l'apprentissage et 199 images réservées aux tests. Les résultats obtenus sont détaillés dans le tableau 6.8, ainsi que dans les figures 6.8 et 6.9.

Les expérimentations ont démontré un grand succès de la méthode appliquée à ces images. Ce succès se manifeste particulièrement dans la préservation élevée de la qualité des zones d'intérêt (ROI) après compression (voir la figure 6.8). L'approche proposée a permis de réduire efficacement la taille des fichiers tout en maintenant une fidélité visuelle exceptionnelle dans les régions cruciales des images, notamment les parties caractéristiques des véhicules, telles que les logos, plaques d'immatriculation, phares et contours de la carrosserie. Cette préservation a été validée par des mesures quantitatives, mettant en avant des valeurs de PSNR

élevées, prouvant ainsi une faible perte de qualité dans les zones essentielles à l'analyse et à la reconnaissance des véhicules.

En comparaison avec la compression JPEG traditionnelle, l'approche proposée a montré un net avantage en termes de restitution des détails dans les zones d'intérêt, Bien que la réduction du volume de stockage soit moins efficace que celle obtenue avec JPEG, l'approche proposée parvient tout de même à diminuer la taille des fichiers, comme l'illustre la figure 6.9. Ce compromis optimal entre qualité et compression démontre la pertinence de la méthode pour les applications nécessitant une analyse fine des images compressées, comme la reconnaissance automatique des véhicules, la surveillance et la gestion de bases de données d'images automobiles.

Image originale (149 KO)	JPEG (FQ = 50)	Approche Proposée	Taille/ PSNR
			JPEG 8.87 KO/ 29.20 **Approche Proposée** 12.3 KO / 21.41 PSNR ROI= 41.12
			JPEG 7.73 KO/ 30.90 **Approche Proposée** 10 KO / 23.42 PSNR ROI= 43.42
			JPEG 6.39 KO/ 32.04 **Approche Proposée** 8.62 KO / 22.22 PSNR ROI=44.86

			JPEG 8.56 KO/ 29.36 **Approche Proposée** 12.1 KO / 19.42 PSNR ROI= 41.83
			JPEG 10.10 KO/ 26.97 **Approche Proposée** 14.1 KO / 18.79 PSNR ROI=40.98
			JPEG 6.5 KO/ 31.45 **Approche Proposée** 9 KO / 30.32 PSNR ROI= 42.53
			JPEG 7.17 KO/ 30.38 **Approche Proposée** 10 KO / 30.74 PSNR ROI= 43
			JPEG 9.11 KO/ 28.34 **Approche Proposée** 12.9 KO / 23 PSNR ROI= 43.61

			JPEG 7.94 KO/ 29.62 **Approche Proposée** 11.2 KO / 24.34 PSNR ROI= 44.69
			JPEG 8.18 KO/ 30.06 **Approche Proposée** 11.5 KO / 19.43 PSNR ROI= 43.24

Tableau 6.8 : Comparaison de la qualité des images de Cars Dataset restaurée.

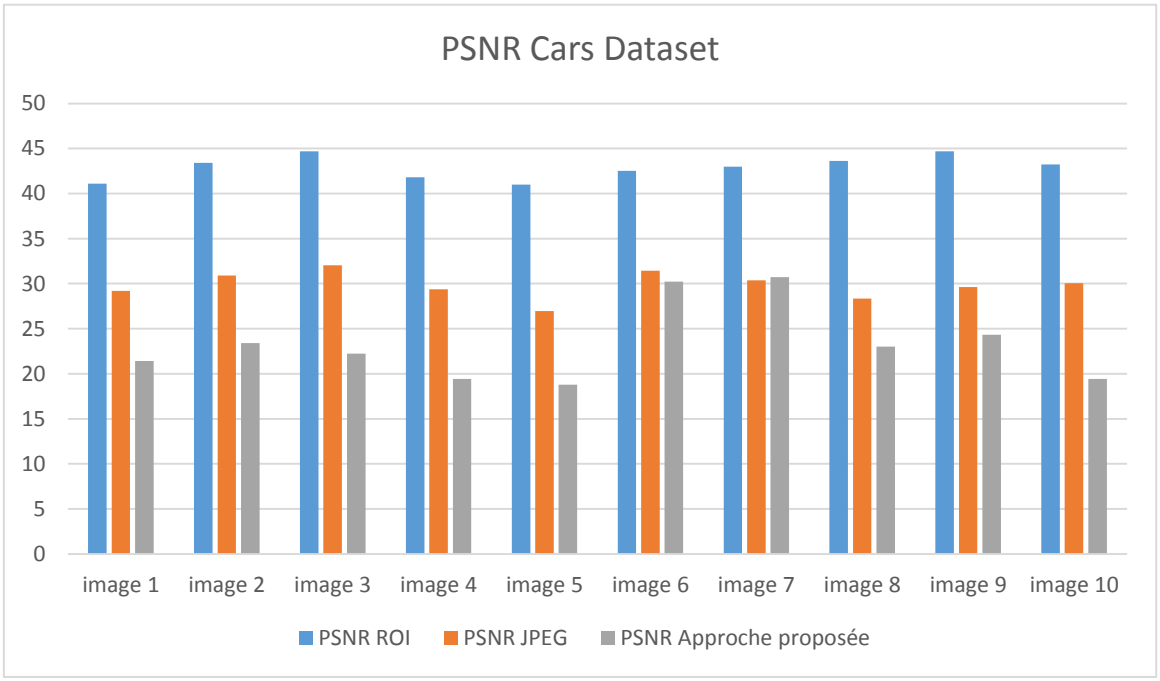


Figure 6.8 : Comparaison des PSNRs de Cars Dataset.

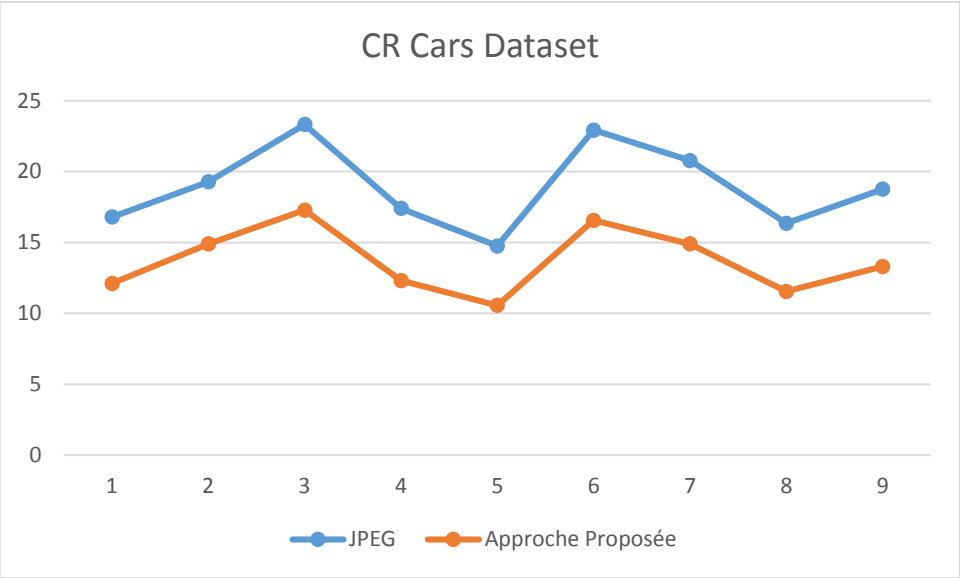


Figure 6.9 : Comparaison des CRs de Cars Dataset.

6.3.1.2 L'implémentation de l'approche proposée sur des images naturelles

Les tests expérimentaux, illustrés par le tableau 6.9 et la figure 6.10, ont révélé les limites de la méthode proposée lorsqu'elle est appliquée à des images naturelles. Deux principales faiblesses ont été identifiées : une qualité d'image restituée insuffisante et une inefficacité en termes de compression.

D'une part, la qualité visuelle des images reconstruites a été jugée insatisfaisante, avec des valeurs de PSNR inférieures à 30, indiquant une dégradation perceptible par rapport aux images originales. Un PSNR inférieur à ce seuil signifie généralement une perte notable de détails, une altération des textures et une augmentation du bruit dans l'image restituée, compromettant ainsi la fidélité visuelle attendue.

D'autre part, l'efficacité de la compression s'est avérée sous-optimale en comparaison avec le standard JPEG. En particulier, lorsque le Facteur de Qualité du JPEG est fixé à 50, la méthode proposée n'apporte aucun gain significatif en termes de réduction du volume de stockage. En d'autres termes, l'espace mémoire requis pour stocker les images compressées reste comparable, voire excessif par rapport à celui de JPEG pour un niveau de qualité similaire. Ce phénomène résulte de la nature même des images naturelles, riches en détails et en textures, qui limitent les possibilités de compression sans perte perceptible.

Ces résultats mettent en évidence que, bien que la méthode proposée vise à préserver une certaine fidélité visuelle, elle ne parvient ni à garantir une qualité suffisante des images restituées, ni à optimiser efficacement l'espace de stockage, la rendant ainsi moins performante que les techniques de compression classiques comme JPEG dans le cas des images naturelles.

	Image originale	JPEG 50%	Approche Proposée
1			
	1.68 MO	43.63 KO PSNR= 36.57	57.9 KO PSNR= 24.86
2			
	1.19MO	35.86 KO PSNR= 35.90	45.6 KO PSNR= 23.03
3			
	1.67MO	64.89 Ko PSNR= 32.10	82.3 KO PSNR= 25.30
4			
	1.67 MO	76.46 Ko PSNR= 31.86	98.2 Ko PSNR= 31.04
5			
	929 KO	40.90 Ko PSNR= 32.99	51.7 KO PSNR=23.49

Tableau 6.9 : Comparaison de la qualité et d'espace de stockage des images de nature restaurée.

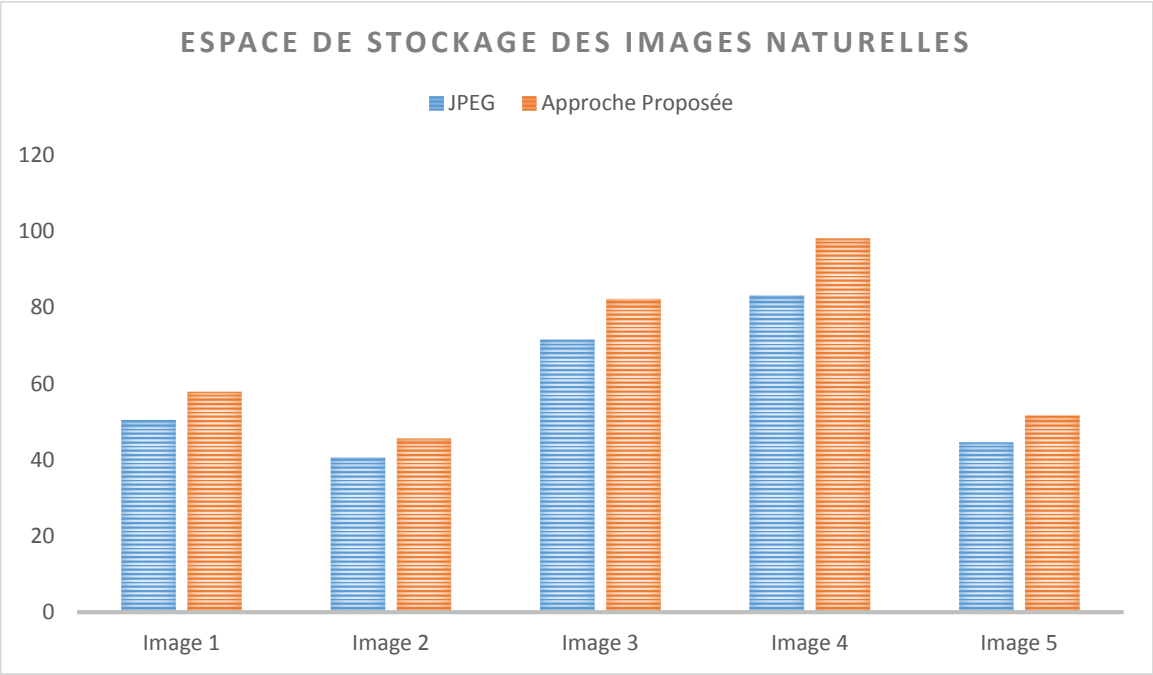


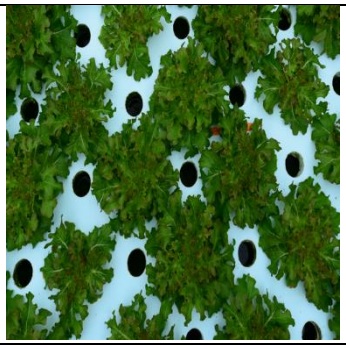
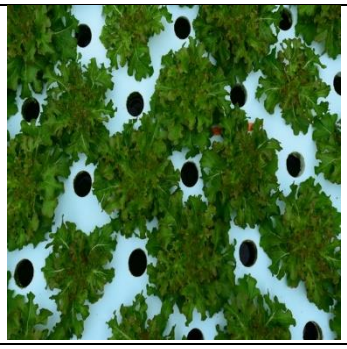
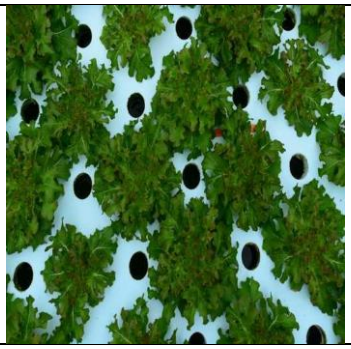
Figure 6.10 : Comparaison des espaces de stockage des images naturelles.

6.3.1.3 L'implémentation de l'approche proposée sur des images texturées

Le tableau 6.10 et la figure 6.11 ont également mis en évidence les limites de l'approche proposée lorsqu'elle est appliquée aux images texturées. Deux observations majeures ressortent de ces résultats : d'une part, une perte significative en termes d'espace de stockage, et d'autre part, une qualité d'image restituée similaire à celle obtenue avec la compression JPEG pour un Facteur de Qualité (FQ) de 50.

En ce qui concerne l'efficacité de la compression, les résultats montrent que la méthode proposée ne permet pas une réduction efficace du volume de données. L'espace de stockage requis reste relativement élevé (voir la figure 6.11), ce qui constitue un inconvénient majeur, notamment dans les applications où l'optimisation de l'espace mémoire est essentielle. Cette inefficacité s'explique par la complexité des images texturées, caractérisées par des variations de motifs répétitifs et des détails fins répartis uniformément sur l'ensemble de l'image. Contrairement aux images contenant des zones homogènes, ces textures complexes ne permettent pas d'exploiter pleinement les techniques de compression qui reposent sur la suppression de redondances spatiales. Concernant la qualité des images restituées, les valeurs du PSNR obtenues sont presque identiques à celles du JPEG avec un FQ de 50, indiquant que les deux méthodes entraînent un niveau de dégradation similaire. Cela signifie que, bien que la méthode proposée maintienne une fidélité visuelle équivalente à celle de JPEG, elle n'offre aucun avantage supplémentaire en matière de restitution de la qualité. En d'autres termes, l'approche ne permet ni une amélioration notable de la qualité de l'image compressée, ni une meilleure réduction de la taille du fichier.

Ces résultats démontrent ainsi que, pour les images texturées, la méthode proposée n'apporte pas de bénéfice tangible par rapport à une compression standard comme JPEG. Elle se révèle inefficace en termes d'optimisation du stockage tout en produisant une qualité d'image équivalente, ce qui limite considérablement son intérêt pour ce type de données.

	Image originale (KO)	JPEG 50% (KO)	Approche proposée (KO)
1			
	768	41.51 PSNR= 29.27	56.4 PSNR= 28.24
2			
	768	40.06 PSNR= 28.41	54.8 PSNR= 28.10
3			
	768	31.82 PSNR= 31.10	41.1 PSNR= 30.04

4			
	768	30.37 PSNR= 31.48	42.7 PSNR=32.08

Tableau 6.10 : Comparaison de la qualité et d'espace de stockage des images texturées restaurée.

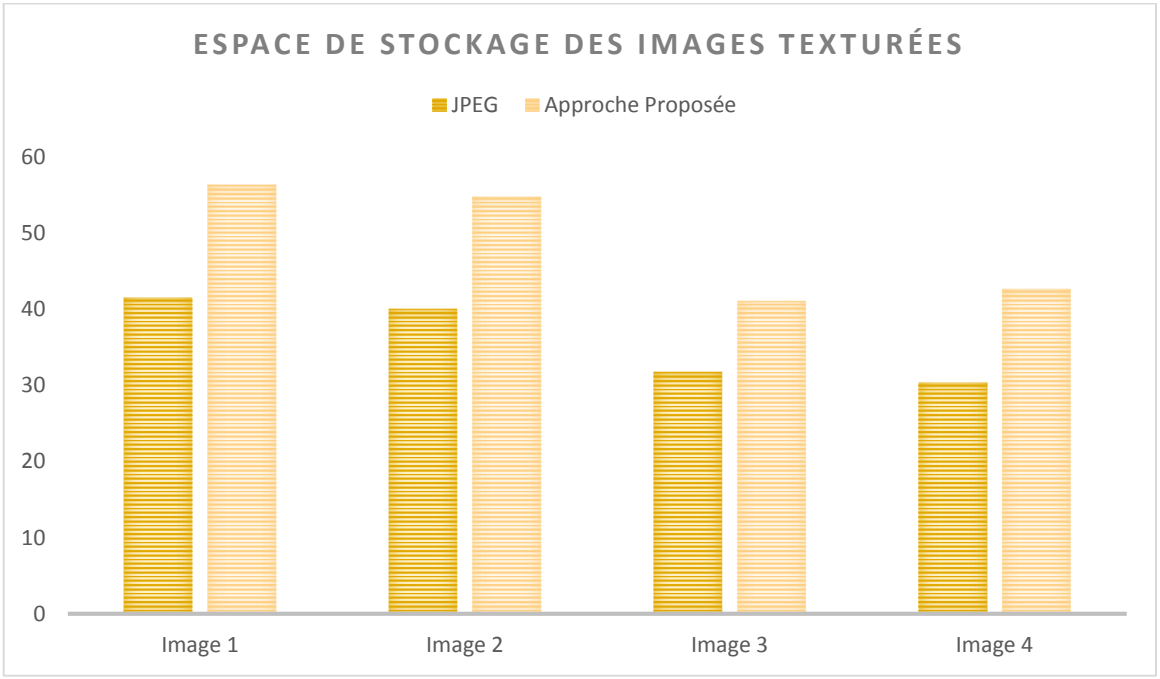


Figure 6.11 : Comparaison des espaces de stockage des images texturées.

6.3.1.4 L'implémentation de l'approche proposée sur des images satellitaires

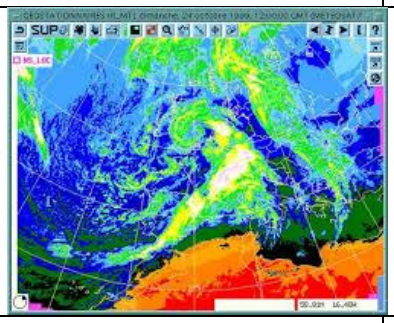
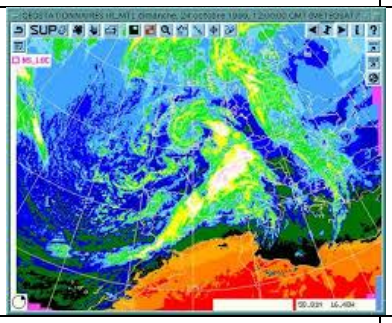
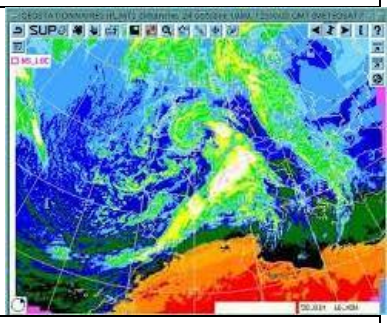
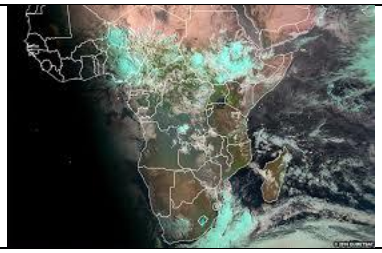
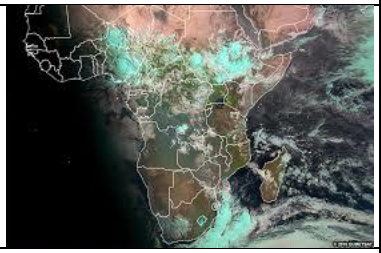
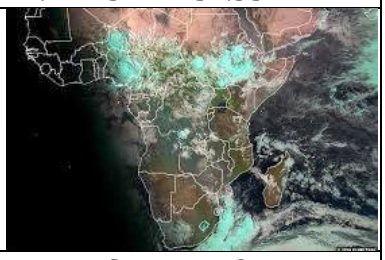
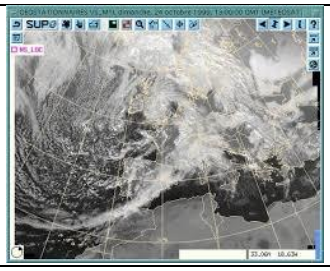
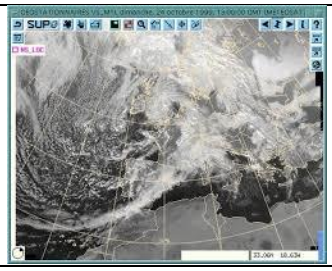
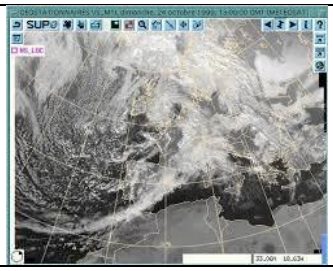
	Image originale (KO)	JPEG 50%(KO)	Approche proposée (KO)
1			
	147	12.92 PSNR= 25.49	18.2 PSNR=25.12
2			
	148	14.61 PSNR= 20.93	20.2 PSNR= 18.39
3			
	147	8.53 PSNR= 27.77	12.7 PSNR= 31.95
4			
	147	9.38 PSNR= 26.69	14.5 PSNR= 21.84
5			
	148	11.28 PSNR= 26.99	16.2 PSNR= 22.09

Tableau 6.11 : Comparaison de la qualité et d'espace de stockage des images satellitaires restituées.

Les tests expérimentaux ont mis en lumière une limite majeure de la méthode proposée lorsqu'elle est appliquée aux images satellitaires. Les résultats présentés dans le tableau 6.10 et la figure 6.12 soulignent un échec significatif en termes d'efficacité, à la fois sur le plan de la qualité visuelle et de la compression des données. Les valeurs obtenues du PSNR sont particulièrement basses, ce qui indique une dégradation sévère de la qualité des images après compression. Un PSNR faible signifie que les images restituées présentent une perte importante de détails, des distorsions visibles, et une altération des textures, rendant parfois difficile l'interprétation des éléments contenus dans l'image. Ce phénomène est particulièrement problématique pour les images satellitaires, qui nécessitent une grande précision pour des applications telles que la cartographie, la télédétection et l'analyse environnementale.

En parallèle, les résultats montrent que l'espace de stockage requis reste élevé malgré la compression appliquée. Cela signifie que la méthode proposée ne parvient pas à optimiser efficacement la réduction du volume de données. Contrairement aux images contenant des zones d'intérêt, les images satellitaires sont souvent très riches en textures complexes, en variations de couleurs et en détails fins répartis sur l'ensemble de la scène. Ces caractéristiques rendent difficile l'élimination des redondances spatiales sans entraîner une perte de qualité significative.

Ainsi, bien que la méthode ait montré une certaine efficacité sur d'autres types d'images, son application aux images satellitaires s'avère peu concluante. Elle ne garantit ni une préservation satisfaisante de la qualité visuelle, ni une réduction efficace du stockage. Cette double faiblesse rend l'approche proposée inadaptée aux besoins des applications nécessitant à la fois une grande fidélité des détails et une optimisation des ressources de stockage.

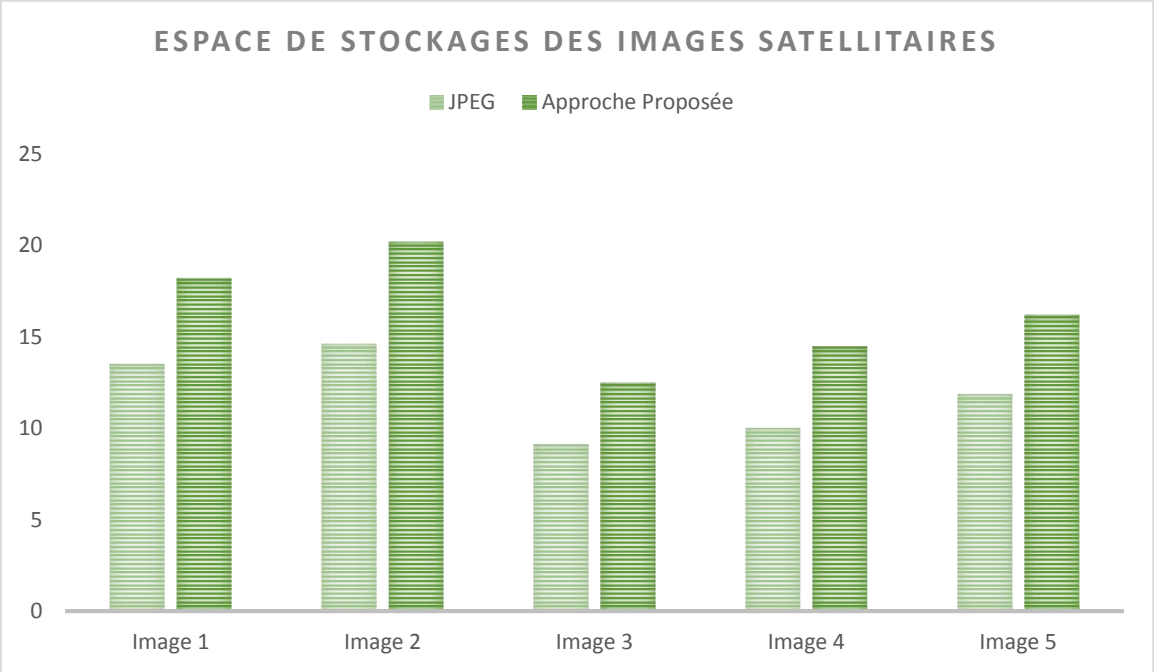


Figure 6.12 : Comparaison des espaces de stockage des images satellitaires.

Type d'Image		JPEG		Approche Proposée		
		CR	PSNR	CR	PSNR	PSNR ROI
Zone d'intérêt	Ham 10000	39.55	35.96	30.49	31.43	39.03
	Caltech-BMP	33.75	30.62	22.46	30.76	35.62
	Cars Dataset	18.95	29.83	13.64	23.3	42.91
	nature	28.97	33.88	22.52	25.54	/
	texturée	21.77	30.06	16.08	29.62	/
	satellitaire	13.51	25.57	6.87	23.88	/

Tableau 6.12 : Tableau comparatif des résultats du différent type d'image.

Les résultats expérimentaux, résumés dans le tableau comparatif 6.12, permettent d'évaluer les performances de l'approche proposée par rapport à la compression JPEG avec un Facteur de Qualité (FQ) de 50. L'analyse des tests menés met en évidence un comportement contrasté de la méthode selon le type d'images traitées. D'une part, l'approche proposée a donné des résultats satisfaisants lorsqu'elle est appliquée aux images contenant des zones d'intérêt (ROI). Dans ces cas, elle a permis un gain significatif en termes de stockage, tout en assurant une meilleure préservation de la qualité visuelle dans les zones essentielles de l'image après compression. Cela confirme l'efficacité de la méthode lorsqu'elle est utilisée dans des contextes où il est primordial de maintenir la fidélité des détails dans des régions spécifiques, comme en imagerie médicale ou dans des documents contenant des informations cruciales.

D'autre part, les tests ont révélé une limitation majeure de la méthode lorsque celle-ci est appliquée à des images ne présentant pas de zones d'intérêt bien définies, notamment les images naturelles, texturées et satellitaires. Pour ces catégories d'images, la méthode a échoué à offrir un avantage significatif, tant en termes de réduction du volume de données qu'en préservation de la qualité visuelle. En effet, le gain en espace de stockage a été négligeable voire inexistant, tandis que la qualité des images compressées était comparable, voire inférieure, à celle obtenue avec JPEG pour un FQ de 50. Cette inefficacité s'explique par la répartition homogène des détails et des textures sur l'ensemble de l'image, empêchant une compression sélective efficace sans altérer la qualité globale.

En conclusion, bien que l'approche proposée démontre une pertinence accrue pour les images comportant des zones d'intérêt, elle n'apporte pas d'amélioration notable pour d'autres types d'images plus complexes en termes de texture et de contenu détaillé. Ces résultats mettent ainsi en évidence les forces et les limites de la méthode, suggérant qu'elle est particulièrement adaptée aux applications où certaines régions spécifiques doivent être préservées avec une haute qualité, mais qu'elle reste moins efficace pour les images globalement riches en détails.

6.3.2 Comparaison de l'Approche proposée avec des méthodes similaires

Dans cette section, nous avons recours à quatre indicateurs clés pour évaluer les performances de l'approche proposée : le taux de compression (CR), l'erreur quadratique moyenne (MSE), le rapport signal-bruit maximal (PSNR) et le gain d'espace (SS). Chacune de ces mesures apporte une perspective spécifique et complémentaire sur l'efficacité et la qualité de la compression

obtenue, permettant ainsi une évaluation approfondie et une comparaison rigoureuse avec d'autres techniques de compression existantes.

L'utilisation combinée de ces quatre mesures offre ainsi une analyse complète de la performance de l'algorithme proposé. Elle permet non seulement d'évaluer la capacité de l'algorithme à compresser efficacement les images tout en conservant les détails cruciaux pour le diagnostic, mais aussi de positionner cette méthode par rapport aux approches de compression traditionnelles. Cette évaluation nous aide à vérifier dans quelle mesure la méthode parvient à atteindre un équilibre optimal entre réduction de taille et préservation de la qualité d'image, répondant ainsi aux exigences des applications critiques.

Le tableau 6.13 et la figure 6.13 illustrent clairement la supériorité de la méthode proposée en matière de compression d'images médicales dermatologiques, notamment lorsqu'on examine le taux de compression (CR) et les autres métriques de performance. Avec un CR de 30.49, notre approche dépasse nettement celle de méthodes comparatives telles que celles de [Muthalaguraja et Kalavathi, 2000], ainsi que les travaux de [Mukhopadhyay et al. 2019], [Sabbavarapu et al. 2021], [Manimekalai et Vasanthi, 2019] et [Hasan et al., 2023], qui affichent des CR de 2,05, 0,66, 23,34, 0,91 et 9,35 respectivement.

En ce qui concerne le rapport signal-bruit de crête (PSNR), la méthode proposée atteint une valeur moyenne de 31.40, indiquant une qualité d'image restaurée bien supérieure à celle obtenue par les approches de [Ghadami et Rahebi, 2023] [Muthalaguraja et Kalavathi, 2000], [Mukhopadhyay et al., 2019], [Manimekalai et Vasanthi, 2019], [Li et al., 2022], [Ghadami et Rahebi, 2023], et [Zerva et al., 2023], dont les PSNR se situent entre 0.49 et un maximum de > 40.

L'erreur quadratique moyenne (MSE) obtenue par notre méthode, avec une valeur de 60.97, témoigne également de la capacité de l'approche proposée à limiter les distorsions visuelles et à préserver les informations essentielles de l'image. En comparaison, les valeurs MSE rapportées par [Hasan et al., 2023] et [Mukhopadhyay et al., 2019] s'élèvent respectivement à 103.9 et 61,08, confirmant que notre méthode est plus efficace pour réduire les pertes de données critiques.

La mesure de similarité structurelle (SS) montre également des résultats très favorables pour notre approche, avec une valeur moyenne de 96,72. Celle-ci surpasse les valeurs obtenues dans les études de [Gaudio et al., 2023], [Ghadami et Rahebi, 2023], [Zerva et al., 2023], [Sabbavarapu et al., 2021] et [Biswas et al.,2021], qui varient entre 71,70 et 95,71. Cette mesure souligne l'excellence de la méthode proposée en termes de conservation des structures essentielles dans les images compressées.

Ces résultats mettent en évidence l'efficacité de notre approche pour préserver les détails et la clarté des zones d'intérêt (ROI) des images. Les valeurs élevées de PSNR et de SS confirment sa capacité à maintenir une qualité d'image élevée malgré la compression. En revanche, l'augmentation de la valeur du MSE reflète une hausse des pertes d'information non essentielles (les zones Non_ROI), c'est-à-dire des détails moins pertinents pour l'analyse Par ailleurs, Grâce à un taux de compression nettement supérieur à celui des méthodes de référence, notre approche

se distingue comme une solution avancée et prometteuse pour la compression d'images. Elle parvient à concilier réduction significative de la taille des fichiers et préservation de la qualité visuelle dans les zones critiques, offrant ainsi une optimisation efficace du stockage sans compromettre la perception visuelle.

Approche	**CR**	**PSNR**	**MSE**	**SS%**
[Gaudio et al., 2023]	-	-	-	71.70
[Ghadami et Rahebi, 2023]	-	27.24	-	80.92
[Hasan et al., 2023]	9.35	74.88	103.9	-
[Sebai et Shah, 2023]	-	-	-	-
[Zerva et al., 2023]	-	> 40	-	92
[Muthalaguraja et Kalavathi, 2000]	2.05	29.56	-	-
[Mukhopadhyay et al., 2019]	0.66	33.05	61.08	-
[Sabbavarapu et al., 2021]	23.34	35.24	20.97	95.71
[Manimekalai et Vasanthi, 2019]	0.91	0.49	0.08	-
[Sran et al., 2020]	-	35.05 → 60.98	-	-
[Ma et al., 2021]	-	-	-	-
[Santosh Kumar et Venkata Ramanaiah, 2021]	-	-	-	-
[Gao et Gao, 2022]	-	-	-	-
[Zhang et Wu, 2021]	-	-	-	-
[Biswas et al., 2021]	-	37.64	-	80
[Li et al., 2022]	-	7.99	-	-
Approche Proposée	***30.49***	***31.40***	***60.97***	***96,72***

Tableau 6.13 : Comparaison de CR, PSNR, MSE et SS de l'approche proposée avec d'autres méthodes.

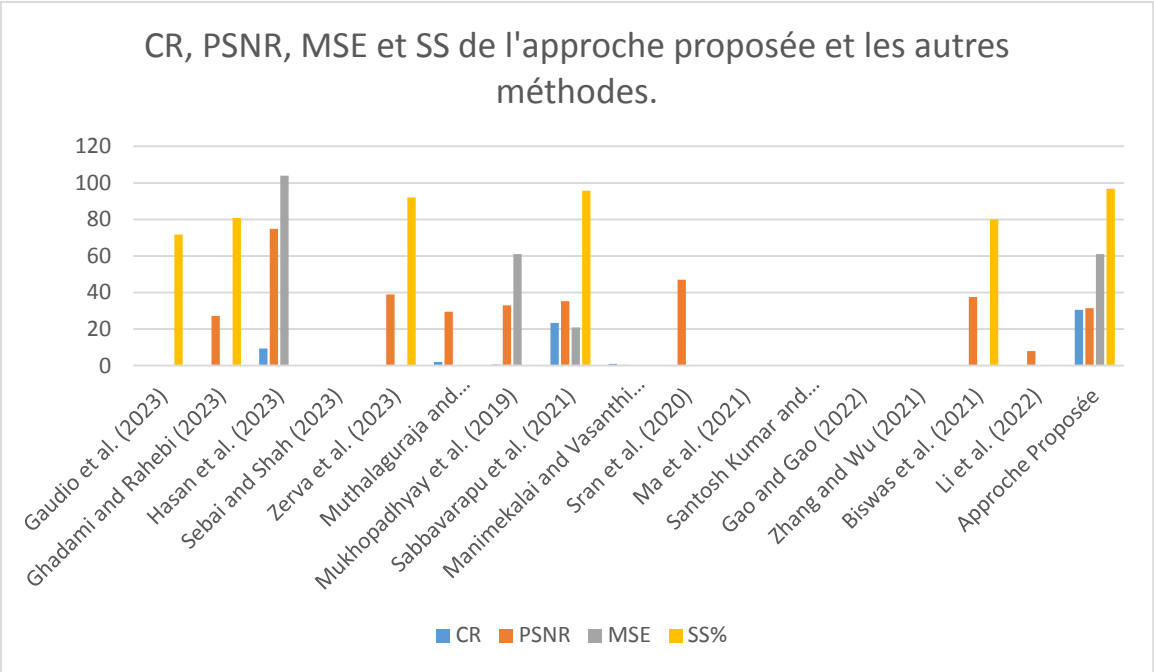


Figure 6.13 : Comparaison de l'approche proposée avec les autres méthodes.

6.4 Conclusion

Dans ce projet, nous avons développé une méthode de compression d'images axée sur la notion de retour sur investissement (ROI). Cette méthode adapte le taux de compression (CR) en fonction de la pertinence de chaque région de l'image. En d'autres termes, une valeur de facteur de qualité (FQ) est spécifiquement sélectionnée pour chaque zone de l'image en fonction de sa pertinence, déterminée par une approche combinant l'apprentissage profond (DL) et la segmentation en superpixels. Ce processus permet d'appliquer différentes valeurs de FQ au sein d'une même image, afin d'optimiser la compression tout en préservant les informations essentielles.

La méthode de compression proposée a été évaluée sur différents types d'images, ce qui a permis de vérifier son efficacité, en particulier pour les images contenant des zones d'intérêt (ROI). Bien que cette approche entraîne une certaine perte d'informations dans les zones de l'arrière-plan, les résultats globaux sont très satisfaisants et répondent aux exigences de qualité d'image et de performance en matière de compression. Ces résultats sont quantifiés à l'aide de métriques telles que la Compression Ratio (CR), le Peak Signal-to-Noise Ratio (PSNR), le Mean Squared Error (MSE) et le Structural Similarity Index (SS), qui montrent clairement l'efficacité de la méthode. Ces métriques confirment que la méthode proposée permet de réaliser une compression efficace tout en préservant une qualité d'image suffisante dans les zones importantes des images.

Les implications de ces résultats sont particulièrement importantes dans des contextes où la préservation des informations présentes dans les zones d'intérêt diagnostiques (ROI) est cruciale. Par rapport à des techniques de compression standard telle que JPEG, l'approche proposée montre une capacité supérieure à préserver les détails essentiels dans les régions ROI, tout en réduisant efficacement la taille du fichier. Cette supériorité est encore renforcée par des comparaisons avec d'autres méthodes de compression utilisées dans le domaine de l'imagerie, où notre méthode se distingue par une meilleure préservation des informations critiques et une qualité d'image globale supérieure.

Ainsi, cette recherche contribue de manière significative au domaine de la compression d'images, en introduisant une approche novatrice qui prend en compte la pertinence des différentes régions de l'image pour optimiser la compression. Ce travail ouvre de nouvelles perspectives pour le stockage et la transmission d'images, en particulier dans des contextes où la conservation des détails diagnostiques est primordiale.

Conclusion générale

Au terme de ce travail de recherche, il est essentiel de revenir sur l'ensemble des réflexions menées, des résultats obtenus, des contributions apportées ainsi que des perspectives qu'elles ouvrent. Cette thèse s'est inscrite dans un contexte technologique et scientifique marqué par une croissance exponentielle des données visuelles, notamment les images numériques, dont le traitement, la transmission et le stockage posent des défis majeurs. Dans de nombreux secteurs, tels que l'industrie, la sécurité, la médecine ou les médias, la compression d'image joue un rôle stratégique pour assurer une gestion efficace de ces flux de données, sans compromettre la qualité de l'information essentielle.

Face aux limites des approches classiques de compression notamment celles qui n'intègrent aucune notion de signification ou de hiérarchie visuelle, cette recherche a proposé une méthode innovante, fondée sur une meilleure compréhension du contenu de l'image. Ce document conclut donc en revenant successivement sur la problématique et les objectifs, les contributions principales, les résultats expérimentaux, les apports scientifiques, les limites du travail, et les perspectives futures qu'il permet d'entrevoir.

1. Rappel de la problématique et des objectifs

Les méthodes de compression d'image classiques, telles que JPEG, reposent sur des principes génériques d'encodage de l'information visuelle, sans distinction du contenu sémantique ou de l'importance des différentes régions. Ces approches, bien que robustes et largement adoptées, traitent toutes les zones d'une image de manière équivalente, ce qui peut entraîner des pertes d'information critiques, notamment dans des contextes où certaines zones ont une valeur stratégique ou diagnostique.

L'objectif de cette thèse a donc été de développer une approche de compression intelligente et adaptative, capable de :

- ✓ Structurer l'image en régions homogènes afin de faciliter un traitement différencié ;
- ✓ Analyser le contenu visuel pour distinguer les zones importantes de celles qui le sont moins ;
- ✓ Préserver la qualité visuelle des zones jugées cruciales tout en maximisant la réduction de données dans les zones périphériques ou secondaires.

Il s'agissait ainsi de concevoir une méthode de compression centrée sur la signification du contenu, combinant des techniques d'apprentissage automatique et des méthodes de segmentation visuelle.

2. Synthèse des contributions et des résultats

Pour répondre à cet objectif, cette recherche a proposé une approche hybride combinant :

- La segmentation par super-pixels, permettant de regrouper les pixels en régions cohérentes du point de vue textural et spatial, facilitant ainsi une analyse contextuelle ;
- Les réseaux de neurones convolutifs (CNN), capables d'apprendre automatiquement des représentations sémantiques à partir des données d'image, afin de classifier les régions selon leur importance ou leur fonction.

Cette combinaison permet d'introduire une hiérarchisation du contenu visuel, où les régions d'intérêt (ROI), identifiées comme pertinentes, sont compressées de manière plus conservatrice, tandis que les zones à faible valeur sémantique sont davantage compressées.

Les résultats expérimentaux, réalisés sur divers jeux de données standards, ont montré :

- Une préservation significative de la qualité visuelle dans les zones importantes ;
- Une réduction efficace de la taille des fichiers ;
- Des scores compétitifs sur des métriques objectives telles que le PSNR (Peak Signal-to-Noise Ratio), le MSE (Mean Squared Error), et le taux de compression.

De plus, des évaluations qualitatives ont confirmé une meilleure lisibilité et fidélité des zones critiques, ce qui est fondamental dans des contextes comme la télémédecine ou la vidéosurveillance.

3. Apports scientifiques et technologiques

Les apports de cette thèse sont multiples et se situent à plusieurs niveaux :

➤ Sur le plan méthodologique :

- Proposition d'un cadre hybride associant des techniques classiques (segmentation) et modernes (deep learning), applicable à différents types de contenus visuels.
- Introduction d'une logique de compression orientée contenu, s'écartant des approches uniformes au profit d'une adaptation dynamique à la structure sémantique de l'image.

➤ Sur le plan théorique :

- Démonstration de la faisabilité d'une compression guidée par la perception visuelle et le contexte.
- Mise en évidence de l'intérêt de la segmentation contextuelle préalable à la compression, souvent négligée dans les approches traditionnelles.

➤ Sur le plan applicatif :

- Développement d'une solution de compression potentiellement exploitable dans des contextes à forte exigence qualitative, comme le diagnostic médical, la robotique visuelle, ou la surveillance automatisée.
- Ouverture vers des systèmes intelligents capables d'auto-ajustement en fonction de l'usage ou de la criticité de l'image.

4. Limites du travail et perspectives de recherche

Bien que les résultats soient prometteurs, plusieurs limites ont été identifiées :

- La qualité de la segmentation par super-pixels, bien qu'efficace, peut être sensible au bruit ou aux variations locales d'intensité. Des méthodes de segmentation plus robustes ou guidées par des modèles attentionnels pourraient améliorer cette étape.
- Les réseaux convolutifs utilisés, bien que performants, restent relativement lourds en termes de calcul. Il serait pertinent d'explorer des architectures plus légères pour des applications en temps réel ou sur appareils embarqués.
- L'ensemble de cette recherche a été spécifiquement dédié aux images intégrant des régions d'intérêt (ROI), excluant ainsi les images sans contenu sémantique significatif.

- L’approche est actuellement limitée aux images 2D statiques. Son extension à d’autres formats visuels (vidéos, images 3D, scènes immersives) constitue une perspective naturelle et riche de promesses.

Parmi les perspectives futures, on peut envisager :

- L’intégration de mécanismes d’attention ou de transformers visuels pour une meilleure compréhension contextuelle du contenu d’image.
- L’application de cette approche dans des systèmes interactifs ou évolutifs (réalité augmentée, interfaces cerveau-machine, etc.).
- L’adaptation dynamique du niveau de compression en fonction du profil utilisateur ou du contexte d’utilisation (télémédecine à distance, transmission mobile, etc.).

5. Conclusion finale

En définitive, cette thèse a montré qu’il est possible de dépasser les paradigmes classiques de la compression d’image en y intégrant des principes d’intelligence artificielle. En analysant et comprenant le contenu visuel avant de le compresser, il devient possible de concevoir des systèmes plus intelligents, plus efficaces et surtout plus respectueux de l’information essentielle.

Le travail présenté ouvre la voie à une nouvelle génération de systèmes de compression adaptatifs, capables de s’aligner sur les exigences contextuelles et sur la nature du contenu. Ce changement de paradigme, de la compression uniforme à la compression sémantique, pourrait profondément transformer la manière dont les images sont stockées, transmises et exploitées à l’avenir, notamment dans des environnements critiques où la qualité ne peut être sacrifiée.

Mes contributions scientifiques

1- Lemnadjlia, S., Melouah, A., Slim, A., « ROI adaptive lossy compression in dermatology using deep learning and superpixels algorithm», International Journal of Computational Vision and Robotics, 2024.
DOI: 10.1504/IJCVR.2024.10065654

2- Lemnadjlia, S., Melouah, A., «Image compression by adaptive JPEG approach », 4th International Conference on Embedded Systems in Telecommunications and Instrumentation, ICESTI'19, Annaba, Algeria, 2019.

3- Lemnadjlia, S., Melouah, A., «Image compression by adaptive JPEG approach », 1st Conference on Innovative Trends in Computer Science, CITCS 2019, Guelma, Algeria, 2019.

4- Saadi, H., Merouani, H. F., Melouah, A., Guessoum, Z., Lemnadjlia, S., & Boukabach, N. (2022). Multi-agents system for breast tumour detection in mammography by deep learning pre-processing and watershed segmentation. *International Journal of Computational Vision and Robotics*, *12*(6), 632-661.

Bibliographie

[Lemnadjlia et al., 2024] Lemnadjlia, S., Melouah, A., Slim, A., « ROI adaptive lossy compression in dermatology using deep learning and superpixels algorithm», International Journal of Computational Vision and Robotics, 2024. DOI: 10.1504/IJCVR.2024.10065654.

[Keshri et al., 2022] Keshri, S., Lal, S., & Shukla, K. K. (2022). Picture quality and compression analysis of multilevel legendre wavelet transformation based image compression technique. *Multimedia Tools and Applications*, *81*(21), 29799-29845.

[Zeybek, 2011] Zeybek, E. (2011). *Compression multimodale du signal et de l'image en utilisant un seul codeur* (Doctoral dissertation, Université Paris-Est).

[Liu, 2015] Liu, Y. (2015). *Codage d'images avec et sans pertes à basse complexité et basé contenu* (Doctoral dissertation, Rennes, INSA).

[Sahir, 2018] Sahir, M. (2018). *Compression des images numériques par la technique des ondelettes* (Doctoral dissertation).

[Hamano et al., 2023] Hamano, G., Imaizumi, S., & Kiya, H. (2023). Effects of jpeg compression on vision transformer image classification for encryption-then-compression images. *Sensors*, *23*(7), 3400.

[de Oliveira, 2022] de Oliveira, V. A. (2022). *Deep learning for embedded image compression in Earth Observation* (Doctoral dissertation, Institut National Polytechnique de Toulouse-INPT).

[Haddad, 2020] Haddad, S. (2020). *Protection of encrypted and/or compressed medical images by means of watermarking* (Doctoral dissertation, Ecole nationale supérieure Mines-Télécom Atlantique).

[Ungureanu et al., 2024] Ungureanu, V. I., Negirla, P., & Korodi, A. (2024). Image-Compression Techniques: Classical and "Region-of-Interest-Based" Approaches Presented in Recent Papers. *Sensors*, *24*(3), 791.

[Liu et al., 2024] Liu, Y., Yang, W., Bai, H., Wei, Y., & Zhao, Y. (2025). Region-adaptive transform with segmentation prior for image compression. In *European Conference on Computer Vision* (pp. 181-197). Springer, Cham.

[Duan et al., 2024] Duan, Q., Hua, Z., Liao, Q., Zhang, Y., & Zhang, L. Y. (2024, March). Conditional Backdoor Attack via JPEG Compression. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, No. 18, pp. 19823-19831).

[Verma et al., 2024] Verma, V., Singh, D., & Khanna, N. (2024). Block-level double JPEG compression detection for image forgery localization. *Multimedia Tools and Applications*, *83*(4), 9949-9971.

[Teng et al., 2024] Teng, G., Jiang, R., Liu, X., Zhou, F., & Chen, Y. (2024). EARN: Toward efficient and robust JPEG compression artifact reduction. *The Visual Computer*, *40*(5), 3033-3053.

[Jin et al., 2024] Jin, X., Cheng, Z., Jiang, Q., Wei, P., Dong, Y., Wu, L., & Miao, S. (2024, July). RIHNet: a robust image hiding method for JPEG compression. In

International Conference on Intelligent Computing (pp. 29-42). Singapore: Springer Nature Singapore.

[Yang et Mandt, 2024] Yang, R., & Mandt, S. (2024). Lossy image compression with conditional diffusion models. *Advances in Neural Information Processing Systems, 36*.

[Makkaoui, 2012] Makkaoui, L. (2012). *Compression d'images dans les réseaux de capteurs sans fil* (Doctoral dissertation, Université de Lorraine).

[Voinson, 2003] Voinson, T. (2003). *Quantification vectorielle algébrique avec zone morte: application à la compression d'images à bas débit et au tatouage d'images* (Doctoral dissertation, Université Henri Poincaré-Nancy 1).

[Berthet, 2022] Berthet, A. (2022). *Deep learning methods and advancements in digital image forensics* (Doctoral dissertation, Sorbonne Université).

[Mazumdar, 2018] Mazumdar, A., J. Singh, Y. S. Tomar, and P. K. Bora. 2018. "Universal Image Manipulation Detection using Deep Siamese Convolutional Neural Network." ArXiv abs/1808.06323.

[Mandelli et al., 2020] Mandelli, S., D. Cozzolino, P. Bestagini, L. Verdoliva, and S. Tubaro. 2020. "CNNBased Fast Source Device Identification." IEEE Signal Processing Letters27:1285–1289. https://doi.org/10.1109/LSP.2020.3008855.

[Rao, 2020] Rao, Y., J. Ni, and H. Zhao. 2020. "Deep Learning Local Descriptor for Image Splicing Detection and Localization." IEEE Access 8:25611–25625. https:// doi.org/10.1109/ACCESS.2020.2970735.

[Meslet-Millet, 2023] Meslet-Millet, F. (2023). *Apprentissage profond pour le trafic réseau: classification, génération et compression* (Doctoral dissertation, Institut National Polytechnique de Toulouse-INPT).

[Zhang et Wallace 2016] Y. Zhang and B. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification," Apr. 2016.

[Fawaz et al., 2019] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification : a review," Data Mining and Knowledge Discovery, vol. 33, July 2019.

[Touzet, 1992] Touzet, C. (1992). *les réseaux de neurones artificiels, introduction au connexionnisme*. Ec2.

[Schmitt et al., 2001] A.Schmitt, B. Le Blanc, M.-M. Corsini, C. Lafond et J. Bruzek, Artificial neural network. A relevant data processing tool for anthropology, 2001.

[Goodfellow et al., 2016] I, Goodfellow, Yoshua Bengio,Aaron Courville. Deep Learning; MIT Press, 2016, pp. [7-9].

[Deng et Yu, 2013] L. Deng and D. Yu. Deep Learning: Methods and Applications. Foundations and Trends, in Signal Processing, vol. 7, nos. 3–4, pp. 197–387, 2013.

[Williams et Zipser, 1989] Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, *1*(2), 270-280.

[Parizeau, 2006] M. Parizeau, RESEAUX DE NEURONES, GIF-21140 et GIF-64326.Université LAVAL , Hiver 2006, P. [2-4].

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems, 26*.

[Sutskever et al., 2014] Sutskever, I. Vinyals, O. & Le. Q. V. Sequence to sequence learning with neural networks. In Proc. Advances in Neural Information Processing Systems 27 3104–3112 2014.

[Cho et al., 2014] Cho, K. et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proc. Conference on Empirical Methods in Natural Language Processing 1724–1734 2014.

[Bahdanau et al., 2015] D. Bahdanau, K. Cho & Y. Bengio, Neural machine translation by jointly learning to align and translate. In Proc. International Conference on Learning Representations http://arxiv.org/abs/1409.0473, [En ligne], 2015.

[Krizhevsky et al., 2012] A. Krizhevsky, I. Sutskever & G. Hinton, ImageNet classification with deep convolutional neural networks. In Proc. Advances in Neural Information Processing Systems 25 1090–1098 2012.

[Hin et al., 2012] G. Hinton et al. Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine 29, 82–97 2012.

[Ma et al., 2015] J. Ma, R. P. Sheridan, A. Liaw, E. G. Dahl & V. Svetnik Deep neural nets as a method for quantitative structure-activity relationships. J. Chem. Inf. Model. 55, 263–274, 2015.

[Leung et al., 2014] M. K. Leung, H. Y. Xiong., L. J. Lee & J. B. Frey, Deep learning of the tissue-regulated splicing code. Bioinformatics 30, i121–i129 2014.

[Xiong et al., 2015] H. Y. Xiong et al. The human splicing code reveals new insights into the genetic determinats of disease. Science 347, 6218, 2015.

[Helmstaedter et al., 2013] M. Helmstaedter et al. Connectomic reconstruction of the inner plexiform layer in the mouse retina. Nature 500, 168–174, 2013.

[Collobert et al., 2011] R. Collobert et al. Natural language processing (almost) from scratch. J. Mach. Learn. Res. 12, 2493–2537, 2011.

[Bordes et al., 2014] A. Bordes, S. Chopra & J. Weston, Question answering with subgraph embeddings. In Proc. Empirical Methods in Natural Language Processing http://arxiv.org/abs/1406.3676v3, 2014.

[Jean et al., 2015] S. Jean, K. Cho, R. Memisevi & Y. Bengio On using very large target vocabulary for neural machine translation. In Proc. ACL-IJCNLP http://arxiv.org/abs/1412.2007, 2015.

[Ravi et al., 2016] Ravì, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., & Yang, G. Z. (2016). Deep learning for health informatics. *IEEE journal of biomedical and health informatics*, *21*(1), 4-21.

[Schmidhuber, 2015] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85-117.

[Hinton, 2006] Hinton, G. E. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*.

[Liu et al., 2017] Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, *234*, 11-26.

[Saideni, 2022] Saideni, W. (2022). *Optimization of video surveillance networks in smart cities using video compressive sensing and deep learning techniques* (Doctoral dissertation, Université de Limoges).

[Deng, 2012] Deng, L. (2012). Three classes of deep learning architectures and their applications: a tutorial survey. *APSIPA transactions on signal and information processing*, *57*, 58.

[Bakir, 2020] Bakir, N. (2020). *Light Field Image and Video Compression* (Doctoral dissertation, INSA de Rennes).

[Simard, 2021] Simard, J. G. (2021). *Learned Image Compression for Machine Visual Perception*. Ecole Polytechnique, Montreal (Canada).

[Ayres et al., 2024] Ayres, L. C., de Almeida, S. J. M., Bermudez, J. C. M., & Borsoi, R. A. (2024). Hierarchical homogeneity-based superpixel segmentation: application to hyperspectral image analysis. *International Journal of Remote Sensing*, *45*(17), 6004-6042.

[Achanta et al., 2012] Achanta, Radhakrishna, A. Shaji, K. Smith, A. Lucchi, P. Fua, and Sabine S¨usstrunk. 2012. "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods." IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (11): 2274–2282. https://doi.org/10.1109/TPAMI.2012.120.

[Machairas et al., 2015] Machairas, Va¨ıa, Matthieu Faessel, David C´ardenas-Pe˜na, Th´eodore Chabardes, Thomas Walter,and Etienne Decenciere. 2015. "Waterpixels." IEEE Transactions on Image Processing 24 (11): 3707–3716.

[Wang et al., 2017] Wang, Murong, Xiabi Liu, Yixuan Gao, Xiao Ma, and Nouman Q. Soomro. 2017. "Superpixel segmentation: A benchmark." Signal Processing: Image Communication 56: 28–39. https://doi.org/https://doi.org/10.1016/j.image.2017.04.007.

[Stutz et al., 2018] Stutz, David, Alexander Hermans, and Bastian Leibe. 2018. "Superpixels: An evaluation of the state-of-the-art." Computer Vision and Image Understanding 166: 1–27. https://doi.org/https://doi.org/10.1016/j.cviu.2017.03.007.

[Ren et Reid, 2011] Ren, C. Y., & Reid, I. (2011). gSLIC: a real-time implementation of SLIC superpixel segmentation. *University of Oxford, Department of Engineering, Technical Report*, 1-6.

[Kohonen, 2001] Kohonen, T. (2001). Self-organizing Maps.-Springer Series in Information Sciences, V. 30, Springer.

[Cousty et al., 2009] Cousty, J., Bertrand, G., Najman, L., and Couprie, M. (2009). Watershed cuts: Minimum spanning forests and the drop of water principle. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(8):1362–1374.

[Felzenszwalb et Huttenlocher, 2004] Felzenszwalb, P. and Huttenlocher, D. (2004). Efficient graph-based image segmentation. International Journal of Computer Vision, 59:167–181.

[OpenCV, 2019] OpenCV (2019). Open Source Computer Vision. https://docs.opencv.org/4.2.0/index.html.

[Meyer, 1992] Meyer, F. (1992). Color image segmentation. In 1992 International Conference on Image Processing and its Applications, pages 303–306.

[Roerdink et Meijster, 2000] Roerdink, J. B. and Meijster, A. (2000). The watershed transform: Definitions, algorithms and parallelization strategies. Fundam. Inf., 41(1,2):187–228.

[Wassenberg et al.,2009] Wassenberg, J., Middelmann, W., and Sanders, P. (2009). An efficient parallel algorithm for graph-based image segmentation. In Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns, CAIP '09, page 1003–1010, Berlin, Heidelberg. Springer-Verlag.

[Shi et Malik, 2000] J. Shi, J. Malik, Normalized cuts and image segmentation, Pattern Analysis and Machine Intelligence, IEEE Transactions on 22 (8) (2000) 888{ 905.

[Shen et al., 2014] J. Shen, Y. Du, W. Wang, X. Li, Lazy random walks for superpixel segmentation, Image Processing, IEEE Transactions on 23 (4) (2014) 1451{1462.

[Zhang et al., 2011] Y. Zhang, R. Hartley, J. Mashford, S. Burn, Superpixels via pseudo-boolean optimization, in: Computer Vision (ICCV), 2011 IEEE Inter-National Conference on, IEEE, 2011, pp. 1387{1394.

[Perbet et Maki, 2011] F. Perbet, A. Maki, Homogeneous superpixels from random walks., in: MVA, 2011, pp. 26{30.

[Tang et al., 2012] D. Tang, H. Fu, X. Cao, Topology preserved regular superpixel, in: Multimedia and Expo (ICME), 2012 IEEE International Conference on, IEEE, 2012, pp. 765{768.

[Liu et al., 2011] M.-Y. Liu, O. Tuzel, S. Ramalingam, R. Chellappa, Entropy rate superpixel segmentation, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 2097{2104.

[Liu et al, 2016] Y.-J. Liu, C.-C. Yu, M.-J. Yu, Y. He, Manifold slic: A fast method to compute content-sensitive superpixels, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 651{659.

[Vincent et Soille, 1991] L. Vincent, P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulations, IEEE Transactions on Pattern Analysis & Machine Intelligence (6) (1991) 583{598.

[Comaniciu et Meer, 2002] D. Comaniciu, P. Meer, Mean shift: A robust approach toward feature space analysis, Pattern Analysis and Machine Intelligence, IEEE Transactions on 24 (5) (2002) 603{619.

[Vedaldi et Soatto, 2008] A. Vedaldi, S. Soatto, Quick shift and kernel methods for mode seeking, in: Computer vision{ECCV 2008, Springer, 2008, pp. 705{718.

[Levinshtein et al., 2009] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, K. Siddiqi, Turbopixels: Fast superpixels using geometric ows, Pattern Analysis and Machine Intelligence, IEEE Transactions on 31 (12) (2009) 2290{2297.

[Conrad et al., 2013] C. Conrad, M. Mertz, R. Mester, Contour-relaxed superpixels, in: Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer, 2013, pp. 280{293.

[Li et Chen, 2015] Z. Li, J. Chen, Superpixel segmentation using linear spectral clustering, in: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, IEEE, 2015, pp. 1356{1363.

[Wang et Wang, 2012] J. Wang, X. Wang, Vcells: Simple and efficient superpixels using edge-weighted centroidal voronoi tessellations, Pattern Analysis and Machine Intelligence, IEEE Transactions on 34 (6) (2012) 1241{1247.

[Veksler et al., 2010] O. Veksler, Y. Boykov, P. Mehrani, Superpixels and supervoxels in an energy optimization framework, in: Computer Vision{ECCV 2010, Springer, 2010, pp. 211{224.

[Van den Bergh et al., 2012] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, L. Van Gool, Seeds: Superpixels extracted via energy-driven sampling, in: Computer Vision{ECCV 2012, Springer, 2012, pp. 13{26.

[Papon et al., 2013] J. Papon, A. Abramov, M. Schoeler, F. Worgotter, Voxel cloud connectivity segmentation-supervoxels for point clouds, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2027{2034.

[Moore et al., 2008] A. P. Moore, J. Prince, J. Warrell, U. Mohammed, G. Jones, Superpixel lattices, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1{8.

[Moore et al., 2010] A. P. Moore, S. J. Prince, J.Warrell, lattice cut-constructing superpixels using layer constraints, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 2117{2124.

[Moore et al., 2009] A. P. Moore, S. J. Prince, J. Warrell, U. Mohammed, G. Jones, Scene shape priors for superpixel segmentation, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 771{778.

[Xu et al., 2014] L. Xu, L. Zeng, Z. Wang, Saliency-based superpixels, Signal, Image and Video Processing 8 (1) (2014) 181{190.

[Wang et al., 2013] P. Wang, G. Zeng, R. Gan, J. Wang, H. Zha, Structure-sensitive superpixels via geodesic distance, International journal of computer vision 103 (1) (2013) 1{21.

[Xie et al., 2014] Y. Xie, L. Xu, Z. Wang, Automated co-superpixel generation via graph matching, Signal, Image and Video Processing 8 (4) (2014) 753{763.

[Tschandl et al., 2018] Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, *5*(1), 1-9.

[Al-Jarrah, 2018] Al-Jarrah, M. (2018). Rgb-bmp steganalysis dataset. *Mendeley Data*, *1*.

[Gaudio et al., 2023] Gaudio, A., Smailagic, A., Faloutsos, C., Mohan, S., Johnson, E., Liu, Y., ... & Campilho, A. (2023). DeepFixCX: Explainable privacy-preserving image compression for medical image analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, e1495.

[Ghadami et Rahebi, 2023] Ghadami, R., & Rahebi, J. (2023). Compression of images with a mathematical approach based on sine and cosine equations and vector quantization (VQ). *Soft Computing*, 1-21.

[Hasan et al., 2023] Hasan, A. D., Kadhim, A. S., & Ali, M. A. (2023). Medical Image Compression Using Hybrid Compression Techniques. *International Journal of Intelligent Systems and Applications in Engineering*, *11*(4s), 634-648.

[Sebai et Shah, 2023] Sebai, D., & Shah, A. U. (2023). Semantic-oriented learning-based image compression by Only-Train-Once quantized autoencoders. *Signal, Image and Video Processing*, *17*(1), 285-293.

[Zerva et al., 2023] Zerva, M. C., Christou, V., Giannakeas, N., Tzallas, A. T., & Kondi, L. P. (2023). An improved medical image compression method based on wavelet difference reduction. *IEEE Access*, *11*, 18026-18037.

[Sabbavarapu et al., 2021] Sabbavarapu, S. R., Gottapu, S. R., & Bhima, P. R. (2021). A discrete wavelet transform and recurrent neural network based medical image compression for MRI and CT images. *Journal of Ambient Intelligence and Humanized Computing*, *12*(6), 6333-6345.

[Manimekalai et Vasanthi, 2019] Manimekalai, M. A. P., & Vasanthi, N. A. (2019). Hybrid Lempel–Ziv–Welch and clipped histogram equalization based medical image compression. *Cluster Computing*, *22*(5), 12805-12816.

[Sran et al., 2020] Sran, P. K., Gupta, S., & Singh, S. (2020). Segmentation based image compression of brain magnetic resonance images using visual saliency. *Biomedical Signal Processing and Control*, *62*, 102089.

[Muthalaguraja et Kalavathi, 2000] V. Muthalaguraja and P. Kalavathi "Significance of ROI based Coding on Lossy and Lossless Medical Image Compression", Third National Conference on Advanced Computing, NCAC'2020, Coimbatore Tamil Nadu, India, pp. 370-374, 2000.

[Mukhopadhyay et al., 2019] Mukhopadhyay, A. P., Mohapatra, S., & Bhattacharya, B. (2019, March). Roi based medical image compression using dwt and spiht algorithm. In *2019 international conference on vision towards emerging trends in communication and networking (ViTECoN)* (pp. 1-5). IEEE.

[Ma et al., 2021] Ma, Y., Zhai, Y., Yang, C., Yang, J., Wang, R., Zhou, J., ... & Wang, R. (2021). Variable Rate ROI Image Compression Optimized for Visual Quality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1936-1940).

[Santosh Kumar et Venkata Ramanaiah, 2021] Santosh Kumar, B. P., & Venkata Ramanaiah, K. (2021). Region of interest-based adaptive segmentation for image compression using hybrid Jaya–Lion mathematical approach. *International Journal of Computers and Applications*, *43*(10), 1035-1046.

[Zhang et Wu, 2021] Zhang, X., & Wu, X. (2021). Attention-guided image compression by deep reconstruction of compressive sensed saliency skeleton. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13354-13364).

[Biswas et al., 2021] Biswas, A. K., Karmakar, S., & Sharma, S. (2021). Performance analysis of a new fractal compression method for medical images based on fixed partition. *International Journal of Information Technology*, 1-9.

[Li et al., 2022] Li, Q., Fu, Y., Zhang, Z., Fofanah, A. J., & Gao, T. (2022). Medical images lossless recovery based on POB number system and image compression. *Multimedia Tools and Applications*, *81*(8), 11415-11440.

[Chavan et al., 2020] Chavan, P. P., Rani, B. S., Murugan, M., & Chavan, P. (2020). A novel image compression model by adaptive vector quantization: modified rider optimization algorithm. *Sādhanā*, *45*, 1-15.

[Dimililer, 2022] Dimililer, K. (2022). DCT-based medical image compression using machine learning. *Signal, Image and Video Processing*, *16*(1), 55-62.

[Gao et Gao, 2022] Gao, H., & Gao, T. (2022). A secure lossless recovery for medical images based on image encoding and data self-embedding. *Cluster Computing*, 1-19.

[Howard et Vitter, 1994] Howard, P. G., & Vitter, J. S. (1994). Arithmetic coding for data compression. *Proceedings of the IEEE*, *82*(6), 857-865.

[Zhang et Zhang, 2014] Zhang X., Zhang W., "Semantic image compression based on data hiding", The Institution of Engineering and Technology 2014, volume 9, pp. 54-61, 2014.

[Vrindavanam et al., 2012] Vrindavanam J., Chandran S., Mahanti G.K., "A survey of image compression methods", International Conference & Workshop on Recent Trends in Technology, TCET'2012, Mumbai, Maharashtra, India, pp. 12-17, 2012.

[Cogranne, 2018] Cogranne R., "Determining JPEG Image Standard Quality Factor from the Quantization Tables.", arXiv:1802.00992, 2018.

[Kadhim, 2016] Kadhim Q.K.," Image Compression Using Discrete Cosine Transform Method", International Journal of Computer Science and Mobile Computing (IJCSMC),volume 5, pp.186-192, 2016.

[Jin et al., 2017] Jin Z., Luo L., Tang Y., Zou W., Li X., "A CNN cascade for quality enhancement of compressed deph images ", Visual Communications and Image Processing, VCIP'2017, Florida, USA, pp. 1-4, 2017.

[Ramous et Hemami, 1996] Ramous M.G, Hemami S.S, "Edge-adaptive JPEG image compression", Visual Communications and Image Processing, volume 2727,, 1996.

[Ma et al., 2019] Ma, S., Zhang, X., Jia, C., Zhao, Z., Wang, S., & Wang, S. (2019). Image and video compression with neural networks: A review. *IEEE Transactions on Circuits and Systems for Video Technology*, *30*(6), 1683-1698.

[Bachard, 2024] Bachard, T. (2024). *Semantic compression of large image collections* (Doctoral dissertation, Université de rennes).

[Dassi, 2023] DASSI, S. K. (2023). *Neural Network Parameter Estimation for Covid-19 with Vaccinated and Timely-Delayed Diagnosis* (Doctoral dissertation, DEPARTMENT OF MATHEMATICS Neural Network Parameter Estimation for Covid-19 with Vaccinated and Timely-Delayed Diagnosis BY SAMUEL KWAME DASSI (PS/MEC/19/0039) MAT 499: PROJECT WORK A PROJECT WORK SUBMITTED TO THE DEPARTMENT OF MATHEMATICS OF THE SCHOOL OF PHYSICAL SCIENCES, COLLEGE OF AGRICULTURE AND NATURAL SCIENCES, UNIVERSITY OF CAPE COAST).

[Ji et al., 2023] Ji, J., Liu, S., Zhang, F., Liao, X., Wang, S., & Liao, J. (2023). Hyperspectral image classification based on unsupervised regularization. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *16*, 1871-1882.

[Fourure, 2017] Fourure, D. (2017). *Réseaux de neurones convolutifs pour la segmentation sémantique et l'apprentissage d'invariants de couleur* (Doctoral dissertation, Université de Lyon).

[Diaz-Zapata, 2019] Diaz-Zapata, M. A. (2019). *Contribution to Panoptic Segmentation* (Doctoral dissertation, Inria; Universidad Autónoma de Occidente).

[Audebert et al., 2018] Audebert, N., Boulch, A., Le Saux, B., & Lefèvre, S. (2018, June). Segmentation sémantique profonde par régression sur cartes de distances signées. In *Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP)*.

[Arnab et Torr, 2017] A. Arnab and P. H. S. Torr. Pixelwise Instance Segmentation With a Dynamically Instantiated Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 441–450, 2017

[Audebert et al., 2017] N. Audebert, B. Le Saux, and S. Lefèvre. Segment-before- Detect : Vehicle Detection and Classification through Semantic Segmentation of Aerial Images. *Remote Sensing*, 9(4) :368, Apr. 2017.

[Badrinarayanan et al., 2017] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet : A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12) :2481–2495, 2017.

[Brostow et al., 2009] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video : A high-definition ground truth database. *Pattern Recognition Letters*, 30(2) :88–97, Jan. 2009.

[Che et al., 2016] H. Chen, X. Qi, L. Yu, and P.-A. Heng. DCAN : Deep Contour-Aware Networks for Accurate Gland Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2487–2496, 2016

[Cordts et al., 2016] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, June 2016.

[Everingham et al., 2014] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes Challenge : A Retrospective. *International Journal of Computer Vision*, 111(1) :98–136, June 2014

[He et al., 2017] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask RCNN. In *International Conference on Computer Vision (ICCV)*, Mar. 2017. arXiv : 1703.06870

[Che et al., 2018] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab : Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99) :1–1, 2018.

[Lin et al., 2014] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO : Common Objects in Context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision ECCV 2014*, number 8693 in Lecture Notes in Computer Science, pages 740–755. Springer International Publishing, Sept. 2014. DOI : 10.1007/978-3-319-10602-1_48.

[Long et al., 2015] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, June 2015.

[Maggiori et al., 2017] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. Can Semantic Labeling Methods Generalize to Any City ? The Inria Aerial Image Labeling Benchmark. In *IEEE International Symposium on Geoscience and Remote Sensing (IGARSS)*, July 2017.

[Ronneberger et al., 2015] O. Ronneberger, P. Fischer, and T. Brox. U-Net : Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer International Publishing, Cham, 2015.

[Rottensteiner et al., 2012] F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf. The ISPRS benchmark on urban object classification and 3d building reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1 :3, 2012.

[Sommer et al., 2017] L. Sommer, K. Nie, A. Schumann, T. Schuchert, and J. Beyerer. Semantic labeling for improved vehicle detection in aerial imagery. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, Aug. 2017.

[Ulman et al., 2017] V. Ulman, M. Maska, K. E. G. Magnusson, O. Ronneberger, C. Haubold, N. Harder, P. Matula, P. Matula, D. Svoboda, M. Radojevic, I. Smal, K. Rohr, J. Jaldén, H. M. Blau, O. Dzyubachyk, B. Lelieveldt, P. Xiao, Y. Li, S.-Y. Cho, A. C. Dufour, J.-C. Olivo-Marin, C. C. Reyes-Aldasoro, J. A. Solis-Lemus, R. Bensch, T. Brox, J. Stegmaier, R. Mikut, S. Wolf, F. A. Hamprecht, T. Esteves, P. Quelhas, . Demirel, L. Malmström, F. Jug, P. Tomancak, E.

Meijering, A. Muñoz-Barrutia, M. Kozubek, and C. Ortiz-de Solorzano. An objective comparison of cell-tracking algorithms. *Nature Methods*, advance online publication, Oct. 2017.

[Zhao et al., 2017] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid Scene Parsing Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.

[Gaudio et al., 2023] Gaudio, A., Smailagic, A., Faloutsos, C., Mohan, S., Johnson, E., Liu, Y. and Campilho, A. (2023) 'DeepFixCX: explainable privacy-preserving image compression for medical image analysis', Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, Vol. 13, No. 4, p.e1495.