



MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH
UNIVERSITE FRERES MENTOURI
CONSTANTINE 1
TECHNOLOGICAL SCIENCES FACULTY
ELECTRONICS DEPARTMENT

وزارة التعليم العالي والبحث العلمي
جامعة الإخوة منتوري- قسنطينة 1
كلية علوم التكنولوجيا
قسم الإلكترونيك



N° d'ordre : 06/DEC/2024

N° Série :01/elect/2024

Thèse présentée pour l'obtention du diplôme de :
Doctorat 3ème Cycle (LMD) en Automatique
Option : Automatique et Informatique Industrielle

Thème :

**Caractérisation de la texture par les champs
aléatoires :
Application à la synthèse et à la classification des textures**

Présenté par :

Ayoub Abderrazak MAAROUF

Soutenu le 16/04/2024

Devant le jury composé de :

Pr. Faouzi SOLTANI	Université des Frères Mentouri Constantine 1	Président
Pr. Fella HACHOUF	Université des Frères Mentouri Constantine 1	Rapporteur
Pr. Toufik BOUDEN	Université Mohamed Seddik Benyahia, Jijel	Examineur
Pr. Abdelouaheb MOUSSAOUI	Université Ferhat Abbass, setif1	Examineur
Pr. Zianou AHMED SEGHIR	Université Abbes Laghrour Khenchela	Examineur
Dr. Meriem HACINI	Université des Frères Mentouri Constantine 1	Examineur

Remerciements

Aujourd'hui, je tiens à exprimer ma profonde gratitude envers toutes les personnes qui ont contribué à la réalisation de ma thèse de doctorat. Ce parcours n'aurait pas été possible sans le soutien, la guidance et l'inspiration que j'ai reçus de nombreuses personnes tout au long de cette aventure académique.

Tout d'abord, je tiens à remercier ma directrice de thèse, Professeure Fella HACHOUF, pour sa confiance en moi, sa patience et sa précieuse direction tout au long de ces années. Votre expertise et vos conseils ont été inestimables, et je ne saurais jamais assez vous remercier.

Je remercie vivement le Prof. Faouzi SOLTANI pour avoir accepté de présider ce jury et de juger ce travail. Les professeurs Toufik BOUDEN, Abdelouaheb MOUSSAOUI, ZIANOU Ahmed Sghir et Meriem HACINI, qui ont accepté de juger cette thèse, trouveront ici l'expression de ma profonde reconnaissance. Je tiens aussi à remercier mes collègues du laboratoire pour leur collaboration, leurs discussions stimulantes et leur amitié. Vos idées et vos encouragements ont été essentiels pour mon développement en tant que chercheur. Ma famille et mes amis méritent une mention spéciale pour leur soutien inconditionnel, leurs mots d'encouragement et leur compréhension tout au long de ces années parfois exigeantes. Je ne pourrais pas demander de meilleurs piliers dans ma vie.

ملخص

يعد توصيف النسيج وتحليله من المجالات الرئيسية في معالجة الصور ورؤية الكمبيوتر. ومع ذلك، فإن تركيب النسيج وتصنيفه لا يزالان من المشاكل التي لم يتم حلها على الرغم من إجراء الكثير من الأبحاث في هذا المجال. يعد التعلم العميق أداة البحث الأكثر استخداماً ونجاحاً في العديد من المجالات، لكنه لم يحقق بعد نفس النجاح في تركيب النسيج. لعلاج هذه المشكلة، قمنا باستكشاف استخدام نقل التعلم، وهي تقنية شائعة الاستخدام في التصنيف، لتركيب الملمس. على الرغم من أننا حصلنا على نتائج واعدة باستخدام الشبكة، إلا أن تعقيدها وقدرتها على الحفاظ على بنية النسيج لا تزال تحديات لم يتم حلها. ولمعالجة هذه المشكلات، قمنا بتطوير طريقة إحصائية جديدة تعتمد على استخدام نماذج الانحدار الذاتي للمعامل العشوائي المفهرس ٢٠-ضرا وقد أثبتت هذه النماذج فعاليتها في الكشف عن معلومات الجوار في الأنسجة، مما يجعلها مناسبة للتوليف. . بالإضافة إلى ذلك، لمعالجة تصنيف النسيج، اقترحنا ثلاث طرق متميزة. الأول يعتمد على تصنيف أقرب جار باستخدام نماذج الانحدار الذاتي ذات المعاملات العشوائية. أما الطريقة الثانية فتتكون من اختيار الميزات باستخدام الخوارزميات الجينية، بالإضافة إلى ثلاث مصنفات. وأخيراً، لحل مشكلة تصنيف الأنسجة، وخاصة تلك المتعلقة بشقوق الطرق وأنماط السطح، قمنا بتطوير طريقة تعتمد على نقل التعلم وتجميع عدة شبكات. أظهرت النتائج التي توصلنا إليها أن نهجنا تفوق على أحدث الأساليب وحقق نتائج واعدة.

Abstract

Texture characterisation and analysis are key areas in image processing and computer vision. However, texture synthesis and classification remain unsolved problems despite much research in this area. Deep learning is the most widely used and successful search tool in many areas, but it has not yet achieved the same success in texture synthesis. To remedy this, we explored the use of transfer learning, a technique commonly used in classification, for texture synthesis. Although we have obtained promising results using the ResNet-50 network, its complexity and its ability to preserve texture structure are still unresolved challenges. To address these issues, we developed a new statistical method based on the use of 2D indexed random coefficient autoregressive (2D-RCA) models. These models have proved effective in detecting neighbourhood information in textures, making them suitable for synthesis. In addition, to address texture classification, we proposed three distinct methods. The first is based on nearest-neighbour classification using autoregressive models with random coefficients. The second consists of feature selection using genetic algorithms, combined with three classifiers: KNN, SVM and random forest. Finally, to solve the problem of classifying textures, particularly those related to road cracks and surface patterns, we developed a method based on transfer learning and the aggregation of several networks. Our results showed that our approach outperformed state-of-the-art methods and delivered promising results.

Keywords : KNN, 2D-RCA, RM, SVM, classification, texture, Synthesis

Résumé

La caractérisation et l'analyse des textures sont des domaines clés du traitement d'images et de la vision par ordinateur. Cependant, la synthèse et la classification des textures demeurent des problèmes non résolus malgré les recherches approfondies dans ce domaine. Le deep learning, l'outil de recherche le plus utilisé et le plus performant dans de nombreux secteurs, n'a pas encore connu le même succès dans le domaine de la synthèse de textures.

Pour remédier à cela, nous avons exploré l'utilisation du transfert d'apprentissage, une technique fréquemment employée en classification, appliquée à la synthèse de textures. Bien que nous ayons obtenu des résultats prometteurs avec le réseau ResNet-50, sa complexité et sa capacité à préserver la structure des textures restent des défis non résolus. Pour surmonter ces problèmes, nous avons développé une nouvelle méthode statistique basée sur l'utilisation des modèles autorégressifs à coefficients aléatoires indexés (2D-RCA). Ces modèles ont prouvé leur efficacité pour détecter les informations de voisinage dans les textures, ce qui les rend particulièrement adaptés à la synthèse. De plus, pour aborder la classification des textures, nous avons proposé trois méthodes distinctes. La première repose sur la classification par les plus proches voisins en utilisant les modèles autorégressifs à coefficients aléatoires. La deuxième méthode consiste en la sélection de caractéristiques à l'aide d'algorithmes génétiques, combinée à l'utilisation de trois classifieurs : KNN, SVM et random forest.

Enfin, pour résoudre le problème de la classification des textures, notamment celles liées aux fissures routières et aux motifs de surface, nous avons développé une méthode basée sur l'apprentissage par transfert et l'agrégation de plusieurs réseaux. Nos résultats ont démontré que notre approche surpasse les méthodes de l'état de l'art et offre des perspectives prometteuses.

Mots clés : KNN, 2D-RCA, RM, SVM, la classification, texture, la Synthèse

Table des matières

Table des figures	i
Liste des tableaux	iii
Liste des Algorithmes	iv
Introduction Générale	1
1 État de l’art sur la synthèse de la texture	3
1.1 Introduction	3
1.2 Notions fondamentales	4
1.2.1 Notion de texture	4
1.2.2 Différents types de texture	5
1.2.3 Usage des textures en traitement d’image	6
1.3 Synthèse de texture	7
1.3.1 Le concept fondamental de la synthèse texturale	7
1.3.2 Applications de la synthèse de texture	8
1.4 Topologie des techniques de synthèse de texture	10
1.4.1 Les méthodes procédurales	10
1.4.2 Les approches de synthèse à partir d’un exemple	10
1.4.3 Méthodes de synthèse sur la base d’un modèle	12
1.5 Approches de synthèse à partir d’un exemple	12
1.5.1 Méthodes non paramétriques	12
1.5.2 Méthodes de synthèse basées sur l’intelligence artificielle	20
1.5.3 Synthèse de texture de couleur	22
1.6 Motivation	22
1.6.1 Limite des méthodes existantes	22
1.7 Approche de synthèse proposée : vue globale	23
1.8 Conclusion	24

2	Extraction des caractéristiques et classification des textures : Etat de l'art	25
2.1	Introduction	25
2.2	Méthodes d'analyse de textures	26
2.2.1	Approche structurelle	26
2.2.2	Approche fréquentielle	26
2.2.3	Approche statistique	32
2.2.4	Approches basées sur un modèle	36
2.3	Classification de la texture	39
2.4	Apprentissage automatique	40
2.4.1	Apprentissage supervisé	40
2.4.2	Apprentissage non supervisé	45
2.4.3	Les réseaux de neurones convolutifs	45
2.5	Conclusion	46
3	Approches développées pour la synthèse de la texture	48
3.1	Introduction	48
3.2	Synthèse de texture par le transfert learning	49
3.2.1	Modélisation de la texture	50
3.2.2	Génération de la texture	53
3.2.3	Les reseaux pré-entraînés pour la synthèse de textures	53
3.2.4	Analyse des résultats	57
3.3	Synthèse de la texture en utilisant les modèles autoregressifs a coefficients aléatoires	59
3.3.1	Modèles RCA spatiaux et estimation des paramètres	60
3.3.2	Algorithme de synthèse de texture utilisant le modèle 2D-RCA . . .	62
3.3.3	Analyse des résultats	63
3.4	Conclusion	73
4	Approches développées pour la classification de la texture	75
4.1	Introduction	75
4.2	Classification de textures par K-NN utilisant un modèle autorégressif à coefficients aléatoires bidimensionnel (2D-RCA)	76
4.2.1	Analyse des resultat	78
4.3	Classification des textures basée sur la construction d'un vecteur de caractéristiques robuste	82
4.3.1	la selection des carachteristiques par les algorithmes genitiques . . .	83

4.3.2	la classifiacion de la texture	84
4.3.3	Analyse des résultats	84
4.4	Apprentissage par transfert basé sur les réseaux de neurones profonds pour la détection des fissures routières	88
4.4.1	Architectures de CNN pré-entraînées pour la classification d'images	89
4.4.2	Méthodes d'ensemble	91
4.5	Test et Résultats	91
4.5.1	Bases de données	91
4.5.2	Prédiction de l'exactitude de la classification	92
4.5.3	Test et résultats	93
4.6	conclusion	95
Conclusion Générale		97
Bibliographie		98

Table des figures

1.1	Textures structurelles	5
1.2	Textures aléatoires	6
1.3	Textures directionnelles	6
1.4	Démonstration du processus de synthèse texturale : A partir d'un exemple (à gauche), il s'agit de synthétiser une nouvelle texture (à droite), supposée être créée par le même processus que l'exemple	7
1.5	Exemples de retouche d'images : image originale	8
1.6	Exemples de retouche d'images : image corrigée par synthèse	8
1.7	Principe des méthodes de synthèse de texture par paramètres	11
1.8	Principe de synthèse d'Efros et Leung [1]	14
1.9	Représentation schématique du processus de synthèse de Wei et Levoy [2] utilisant un voisinage causal en forme de L et un balayage lexicographique	15
1.10	Illustration de l'approche de synthèse de structure/texture en deux étapes [3]	17
1.11	Synthèse de texture utilisant des CNN selon [4]. a : une texture exemplaire. b : résultat synthétisé en utilisant le modèle CNN.	22
1.12	Synthèse de texture utilisant des CNN selon [4]. a : une texture exemplaire. b : résultat synthétisé en utilisant le modèle CNN.	23
2.1	Caractéristiques de texture basées sur le spectre de puissance de l'image.	27
2.2	Calcul du nombre LBP et du contraste C.	35
2.3	les types de machine learning	41
2.4	KNN	43
2.5	Schéma classique d'un CNN	46
3.1	Schéma descriptif de la méthode de synthèse	50
3.2	la méthode de synthèse [4]	51
3.3	schéma de la generation de la texture par le transfert par apprentissage	55
3.4	shéma de modelisation de la texture par le transfert par apprentissage	56
3.5	Synthèse de textures utilisant differents architectures pré entrainais	58

3.6	Méthode proposée	64
3.7	Image d'origine (a), (b) image générée par les modèles 2D-RCA, (c) différence entre les deux images.	66
3.8	Résultats de synthèse des bases de données DTD et Brodatz par la méthode proposée. Dans chaque panneau, à gauche : texture d'origine, à droite : résultat synthétisé	67
3.9	Comparaison des résultats de la synthèse de textures par différentes méthodes.	69
3.10	Comparaison des résultats de la synthèse de textures par différentes méthodes	70
3.11	Les lignes 1 à 5 représentent les images originales et obtenues par différents algorithmes. Les colonnes a à g décrivent respectivement les images originales, c-cgCNN-Gram, c-cgCNN-Mean, gCNN, CoopNet, la méthode d'auto-ajustement et l'algorithme proposé.	72
3.12	Résultats de la synthèse de textures en utilisant la méthode proposée . . .	73
4.1	Diagramme de la méthode proposée	77
4.2	k-nn	78
4.3	brodatz database	79
4.4	(a) images originales (b) algorithme de résultats 2D-RCA Knn, (c) résultats svm (d) résultats de l'algorithme de distance.	81
4.5	Diagramme de la méthode proposée	85
4.6	base de données d'images.	85
4.7	matrice de confusion utilisant quatre classes avec classificateurs Rf	87
4.8	matrice de confusion utilisant cinq classes avec classificateurs Rf	87
4.9	L'approche proposée de détection des fissures routières basée sur l'apprentissage profond d'ensemble	89
4.10	image database	92
4.11	Résultats expérimentaux de la classification des fissures	93
4.12	ROC Curve	95
4.13	Confusion matrices	95

Liste des tableaux

2.1	Ensemble de neuf masques Laws de dimensions 3×3	29
2.2	Caractéristiques statistiques de premier ordre	33
2.3	Matrice de cooccurrence $P(1,0)$ de l'image Im ayant 4 niveaux de gris, pour une distance $d = 1$ et la direction $\theta = 0^\circ$	34
2.4	Caractéristiques déduites de la matrice de cooccurrence	35
3.1	précision de la synthèse en utilisant différentes architectures d'apprentis- sage par transfert pour différentes itérations	58
3.2	Temps de calcul en minutes en utilisant différentes architectures d'appren- tissage par transfert pour différentes itérations	58
3.3	Comparaison de la qualité d'image en utilisant différentes tailles de fenêtres	65
3.4	SIM, MSE et PSNR pour les résultats de synthèse des bases de données DTD et Brodatz.	68
3.5	Comparaison des mesures de qualité donnée par la figure 3.11	71
3.6	"Résultat de la synthèse de texture montré dans la FIG. 3.11 en utilisant la SSIM."	72
3.7	SSIM, MSE pour les résultats de synthèse de la Figure 3.12.	73
4.1	Taux de classification de textures	80
4.2	Classes et méthodes pour l'analyse des textures.	82
4.3	Taux de classification	88
4.4	Comparaison de différentes architectures CNN pour l'analyse d'images . . .	90
4.5	Ensemble de données d'images de fissures	92
4.6	Taux classification pour différentes époques en utilisant différentes archi- tectures d'apprentissage par transfert	94
4.7	Taux de classification en utilisant la méthode d'ensemble pour différentes époques	94

Liste des Algorithmes

3.1	La synthèse des textures par l'apprentissage par tranfert	57
3.2	Synthèse de texture par le modèles 2D-RCA	63
4.1	Classification KNN utilisant des modèles 2D_RCA	79

Introduction Générale

La texture, telle que perçue par nos sens, est omniprésente dans notre environnement quotidien. Elle définit la rugosité d'une écorce d'arbre, la douceur d'une étoffe de soie, la granularité d'une roche ou encore la régularité d'une structure architecturale. La texture est une caractéristique fondamentale qui enrichit notre expérience visuelle et tactile du monde qui nous entoure. Cependant, bien que la reconnaissance et la description de la texture soient des tâches triviales pour les êtres humains, elles posent des défis considérables pour les systèmes de traitement de l'information automatisés.

La compréhension et la caractérisation des textures ont une importance cruciale dans plusieurs domaines, allant de la vision par ordinateur à l'analyse d'images médicales, en passant par la sécurité, le design graphique, l'industrie manufacturière et bien d'autres. En particulier, la synthèse et la classification des textures sont des domaines de recherche actifs, visant à recréer artificiellement des textures réalistes ou à identifier automatiquement des textures dans des images.

L'une des avancées les plus significatives dans l'étude des textures est l'utilisation des champs aléatoires. Les champs aléatoires sont des modèles mathématiques puissants qui permettent de représenter et d'analyser des structures complexes, telles que les textures. Cette thèse se concentre sur la caractérisation des textures en utilisant les champs aléatoires comme outil principal. Plus précisément, elle explore comment les propriétés statistiques et structurales des textures peuvent être capturées, modélisées et manipulées à l'aide de techniques basées sur les champs aléatoires.

L'objectif ultime de cette recherche est de contribuer à la création de méthodes avancées pour la synthèse réaliste des textures et pour l'amélioration des techniques de classification des textures dans des applications telles que la reconnaissance d'objets, la détection de défauts ou la recherche d'images. Cette thèse propose une approche interdisciplinaire en combinant des concepts issus de la statistique, de la vision par ordinateur, de l'apprentissage automatique et de la modélisation probabiliste.

Au cours des prochains chapitres, nous explorerons en détail les fondements théoriques, les méthodologies de recherche, les expérimentations et les applications pratiques qui

contribuent à cette quête pour comprendre et exploiter la texture par le biais des champs aléatoires. À travers cette recherche, nous espérons apporter une contribution significative à la compréhension des textures et ouvrir de nouvelles perspectives pour des applications variées.

Organisation de la thèse

Cette thèse est structurée en quatre chapitres distincts :

Le premier chapitre établit une brève définition de la texture, discutant des divers types de textures et offrant un état de l'art sur les différentes méthodes de synthèse des textures. Cette revue va des méthodes classiques aux approches plus récentes basées sur le deep learning, tout en mettant en évidence les limites inhérentes à ces méthodes.

Le deuxième chapitre se penche sur les caractéristiques des textures et leur classification, en explorant les différents classifieurs utilisés pour résoudre ce problème complexe.

Le troisième chapitre est dédié à nos contributions dans le domaine de la synthèse de textures. Nous exposons d'abord notre méthode de synthèse de textures qui repose sur le concept de transfert d'apprentissage, une approche généralement utilisée uniquement pour la classification. Ensuite, nous présentons une deuxième méthode pour la synthèse de textures qui repose sur des modèles autorégressifs à coefficients aléatoires, ayant donné des résultats supérieurs à l'état de l'art.

Le quatrième chapitre se concentre sur nos contributions dans le domaine de la classification des textures, englobant plusieurs aspects. Notre première méthode implique la classification des textures en utilisant un modèle autorégressif associé au classifieur KNN. Notre deuxième méthode présente un vecteur de caractéristiques robuste qui intègre diverses types de caractéristiques. Enfin, nous abordons la classification d'images de fissures routières enrichies de texture. Cette méthode repose sur le transfert d'apprentissage pour identifier le meilleur réseau pour ce domaine, en plus d'une fusion de différents modèles pré-entraînés, une approche que l'on qualifie d'"ensemble learning" et qui a produit de meilleurs résultats.

Ce plan de thèse s'articule autour de la texture, de sa synthèse et de sa classification, en couvrant une variété de méthodes et d'approches novatrices pour améliorer notre compréhension et notre utilisation des textures dans divers domaines d'application.

Chapitre 1

État de l’art sur la synthèse de la texture

1.1 Introduction

La synthèse de texture est un domaine clé de la vision par ordinateur et du traitement d’images. Elle présente beaucoup d’applications dans de nombreux domaines, notamment la réalité virtuelle, les jeux vidéo, la conception graphique, la cartographie, la robotique, la médecine, etc. La synthèse de texture consiste à générer des images ou des modèles tridimensionnels de surfaces qui ont une apparence et une structure similaires à celles de matériaux réels, tels que la pierre, le bois, le marbre, la peau, etc. La synthèse de texture est un défi complexe, car les textures réelles ont des propriétés intrinsèques telles que la régularité, la répétabilité, la variation et la complexité qui doivent être reproduites de manière efficace et réaliste.

Au fil des années, différentes méthodes de synthèse de texture ont été proposées pour surmonter ces défis. Les premières approches utilisaient des modèles mathématiques simples tels que les fractales, les autosimilarités et les modèles de Markov pour générer des textures. Cependant, ces modèles ne permettaient pas de reproduire des textures complexes et réalistes. Avec l’avènement de la puissance de calcul et des techniques d’apprentissage automatique, de nouvelles approches de synthèse de texture ont émergé, telles que la synthèse basée sur des exemples et la synthèse avec des réseaux de neurones artificiels.

Dans ce chapitre, nous présentons un état de l’art de la synthèse de texture. Nous commençons par présenter les trois principales approches de synthèse de texture : la synthèse basée sur des exemples, la synthèse basée sur des modèles mathématiques, et la synthèse de texture utilisant les réseaux de neurones. Pour chaque classe, nous discutons de ses avantages, de ses limites, et de ses applications dans différents domaines. Nous présente-

rons également des études de cas et des exemples de synthèse de texture pour illustrer les différents aspects de chaque approche. Enfin, nous discuterons des défis et des opportunités futures dans le domaine de la synthèse de texture, ainsi que des directions prometteuses pour la recherche future.

1.2 Notions fondamentales

1.2.1 Notion de texture

En traitement d'images, il est difficile de définir clairement ce qu'est une texture, car il existe de nombreux types de textures et de nombreuses applications qui les utilisent. Dans des dictionnaires courants comme celui de Merriam-Webster [5], la texture est décrite comme la façon dont les fils s'entremêlent dans un tissu, ainsi que les caractéristiques visuelles et tactiles d'un objet. Les deux définitions sont différentes, l'une pour le tissage et l'autre pour l'apparence de l'objet.

En revanche, dans le contexte de l'analyse d'images numériques, la texture peut être définie comme la manière dont les niveaux de gris des pixels varient dans l'espace et délimite une variété de phénomènes naturels qui présentent des motifs répétitifs avec une dose d'aléatoire. Il est important de souligner que cette définition exclut les textures déterministes.

Dans le domaine du traitement d'images, les textures sont considérées comme des phénomènes complexes qui peuvent être décomposés en deux aspects fondamentaux : les éléments de base qui composent la texture, appelés les primitives, et l'organisation spatiale de ces primitives. Différentes définitions ont été proposées pour décrire ce concept [6]. Selon Unser [7], une texture correspond à une région de l'image pour laquelle il est possible d'établir une fenêtre de dimensions minimales telle que toutes les translations possibles de cette fenêtre à l'intérieur de la région donnent une perception visuelle identique.

Germain [8] quand à lui, propose une définition prenant un compte le caractère déterministe ou aléatoire des motifs et de l'arrangement, ainsi que leur échelle variable et le concept de stabilité. Il décrit une texture comme étant une composition hiérarchique de motifs déterministes ou aléatoires arrangés selon des règles qui peuvent être soit déterministes soit aléatoires. Cette définition est applicable aussi bien dans un contexte 2D que 3D.

En infographie, le terme "texture" est fréquemment utilisé pour décrire une image numérique appliquée à la surface d'un modèle tridimensionnel en mailles afin de lui conférer une apparence plus réaliste.

Ces différentes définitions de la texture dépendent du contexte et des auteurs. Cer-

tains considèrent qu'une texture doit être locale et stationnaire, tandis que d'autres se concentrent davantage sur les motifs répétitifs avec une variation aléatoire en niveaux de gris.

1.2.2 Différents types de texture

La notion de texture est complexe, notamment lorsqu'il s'agit de la classification de l'ensemble des textures disponibles. À cet égard, on peut distinguer trois principales familles de textures : les textures structurelles, les textures aléatoires et les textures directionnelles.

Texture structurelles Les textures macrospatiales, également appelées textons, se composent d'un motif de base qui se répète spatialement selon des règles de direction et de placement. Elles peuvent même présenter une certaine périodicité ou cyclo stationnarité (un processus aléatoire plaqué sur un processus périodique). Les approches fréquentielles où structurelles sont particulièrement adaptées pour décrire cette famille de textures. Des exemples courants de ce type de textures sont la texture d'un mur en briques, de certains tissus ou d'un grillage (voir Fig. 1.1).

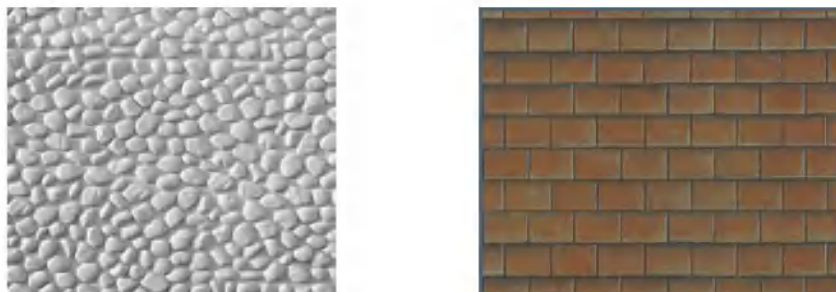


Figure 1.1: Textures structurelles

Textures aléatoires

Dans les textures de ce type, aucun motif particulier n'est localisable ou détectable, et la distribution des intensités ne présente aucune régularité apparente. L'aspect aléatoire joue donc un rôle spécifique dans ce type de texture. Ces textures sont décrites en termes de statistiques ; de densité spectrale, de corrélations ou d'isotropie. Toutefois, leur apparence visuelle reste globalement homogène. Les différentes régions d'une image aérienne, les bois, les champs, sont des exemples courants de textures aléatoires.

Textures Directionnelles Les textures présentées ne sont pas entièrement aléatoires et comportent des orientations ou des directions distinctes. Bien qu'elles ne possèdent pas

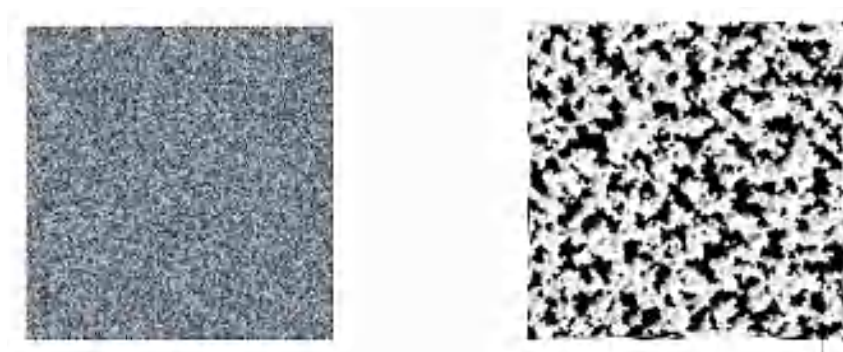


Figure 1.2: Textures aléatoires

d'éléments structurants de base, leur apparence est caractérisée par ces propriétés. La figure 1.3 illustre ces textures.



Figure 1.3: Textures directionnelles

1.2.3 Usage des textures en traitement d'image

La notion de texture est largement exploitée dans de nombreuses applications en traitement d'images. Parmi ces applications, on peut citer la classification, la reconnaissance de formes et la segmentation des images, ainsi que la compression, le rehaussement et la détection de défauts d'images.

Différentes approches pour traiter et analyser les images texturées sont utilisées dans de nombreux domaines d'application, telles que la sécurité, le diagnostic en imagerie médicale, ainsi que l'étude de la structure de divers tissus ou matières. Pour analyser ces textures, il est nécessaire de trouver des attributs descriptifs pour quantifier des notions telles que la finesse, la directionnalité ou la granularité, qui permettent de caractériser les textures. Par ailleurs, les textures naturelles peuvent être modélisées et reproduites efficacement grâce à des techniques de synthèse d'images.

La synthèse de texture est employée afin d'évaluer la généralité et l'adéquation des modèles de texture, et cela peut parfois s'avérer encore plus complexe que la classification

ou la segmentation, du fait de son exigence envers une représentation extrêmement minutieuse et fidèle (du point de vue perceptuel) de la texture. Les textures ont également leur utilité dans les applications graphiques pour reproduire des images de situations réelles observées sous différentes conditions d'éclairage. En résumé, au cours de nombreuses décennies, la pertinence des textures dans le domaine de l'analyse d'images a été confirmée à travers divers contextes.

1.3 Synthèse de texture

1.3.1 Le concept fondamental de la synthèse texturale

Au cours des quinze dernières années, la synthèse de texture est devenue un domaine actif de la recherche en vision par ordinateur. Elle consiste à remplir une région plane ou un volume en utilisant une texture observée sur un échantillon-modèle. Cependant, si l'échantillon est trop petit pour couvrir la région considérée, il doit être étendu. Dans ce cas, copier simplement la texture de départ pour obtenir une nouvelle image de taille supérieure aboutit souvent à un aspect artificiel de pavage régulier. C'est là qu'intervient la synthèse de texture.

L'objectif de la synthèse de texture dans le domaine du traitement d'image consiste à générer une nouvelle texture de dimensions variées qui présente des similitudes avec l'échantillon original. Idéalement, la texture synthétisée devrait s'intégrer harmonieusement en tant qu'élément d'un matériau homogène, duquel la texture d'origine a également été extraite. Cette nouvelle texture ne devrait présenter ni répétitions non désirées, ni imperfections visibles. En termes de perception, la texture synthétisée devrait donner l'impression qu'elle a été créée à travers le même processus que celui-ci ayant donné naissance à l'échantillon initial.

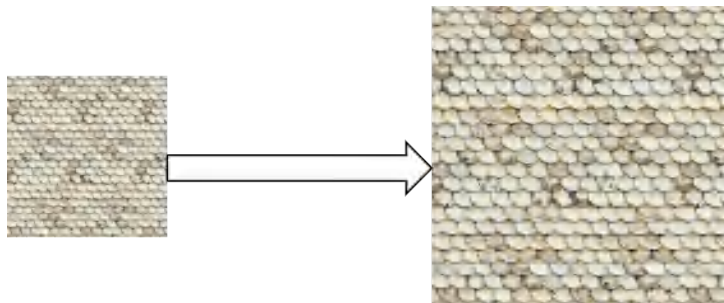


Figure 1.4: Démonstration du processus de synthèse texturale : A partir d'un exemple (à gauche), il s'agit de synthétiser une nouvelle texture (à droite), supposée être créée par le même processus que l'exemple

1.3.2 Applications de la synthèse de texture

Les applications principales de la synthèse de texture sont :

1.3.2.1 Le plaquage (Mapping)

Le plaquage de texture [9] est une technique couramment utilisée en infographie pour synthétiser des textures à partir de petits échantillons, permettant ainsi de créer des textures de grande taille. Ces textures synthétisées peuvent ensuite être appliquées pour habiller la surface d'objets en deux ou trois dimensions.

1.3.2.2 image inpainting

Ce sont des techniques utilisées pour corriger les défauts d'une image. Au fil du temps ou en raison d'une manipulation incorrecte, les photographies ou les films peuvent se dégrader, présentant des rayures, des taches et autres altérations. Ils peuvent également nécessiter des modifications en raison de l'environnement ou de la composition, tels que la présence d'objets ou de personnes indésirables. Dans ces cas, la synthèse de texture peut être utilisée pour corriger ces imperfections [10].

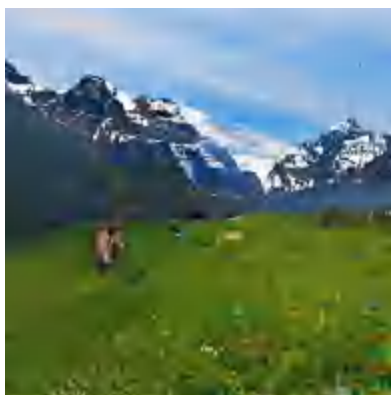


Figure 1.5: Exemples de re-touche d'images : image originale

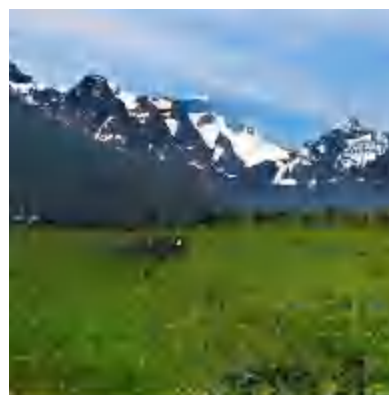


Figure 1.6: Exemples de re-touche d'images : image corrigée par synthèse

1.3.2.3 Animation et synthèse vidéo

Le concept de texture est généralement associé aux images en deux dimensions, mais il peut être adapté aux structures volumétriques telles que celles obtenues à partir des techniques d'imagerie en trois dimensions (par exemple, imagerie médicale, tomographie, imagerie sismique, etc.). L'expansion vers trois dimensions, initialement destinée à la

création de textures volumétriques, peut également s'appliquer aux textures temporelles (2D+T). Ces textures temporelles englobent des zones homogènes sans limites nettement définies, qui évoluent dans l'espace et dans le temps. Par exemple, elles peuvent représenter des phénomènes tels que le feu, la fumée ou les vagues, avec des arrangements spatiaux changeants au fil du temps [2]. D'autres exemples de textures temporelles incluent les animations générées par ordinateur, qui peuvent représenter des mouvements de fond plus ou moins cycliques, comme les vagues, les chutes d'eau ou les mouvements des feuillages. L'introduction du concept de texture temporelle étend ainsi les outils de retouche pour englober la synthèse vidéo.

1.3.2.4 Rendu volumique

La synthèse de texture 3D à partir d'échantillon 2D peut être employée dans le traitement d'images sans avoir besoin de mappage de texture. Il s'agit de synthétiser une texture volumétrique puis de découper des objets à partir du volume résultant [11].

1.3.2.5 Compression d'images

Les images représentant des scènes naturelles comprennent souvent de grandes zones texturées. Il arrive que des algorithmes de compression d'images standard risquent parfois de dégrader ces textures. Avant d'entrer dans la phase de décompression des images, il est nécessaire de segmenter et de modéliser d'abord les textures présentes dans l'image. Cette étape préliminaire permet d'acquérir une représentation condensée des textures, laquelle peut ensuite être exploitée pour reconstituer les zones texturées lors de la phase suivante de décompression des images [12].

1.3.2.6 Simulation de matériaux et synthèse 2D/3D

La synthèse virtuelle de matériaux est essentielle dans les sciences des matériaux, notamment pour la conception de matériaux composites basée sur l'imagerie 3D. Il est parfois impossible d'utiliser des techniques d'imagerie 3D. À titre d'exemple (Wei, 2001), les caractéristiques tridimensionnelles des matériaux sont fréquemment capturées en deux dimensions à l'aide de techniques de microscopie optique ou électronique. Ces techniques fournissent des résolutions spatiales qui dépassent celles atteintes par les tomographes en trois dimensions. Dans cette situation, une méthode de synthèse volumétrique qui repose sur l'inférence 2D/3D représente la seule approche viable pour obtenir un aperçu de la configuration tridimensionnelle du matériau [13].

1.4 Topologie des techniques de synthèse de texture

Au cours des deux dernières décennies, de nombreuses techniques ont été développées en synthèse de texture, qui peuvent être regroupées en quatre grandes familles : les méthodes procédurales, les approches basées sur un exemple et les méthodes basées sur un modèle et récemment les méthodes basées sur l'intelligence artificielle (deep learning) . Il faut noter que les frontières entre ces catégories ne sont pas strictes et que certaines approches peuvent se situer à la jonction de plusieurs d'entre elles. Par exemple, il est possible de synthétiser une texture à partir d'un exemple en utilisant une approche procédurale ou en se basant sur une analyse fondée sur un modèle, comme dans le cas de la synthèse par bruits de Gabor [14]. En raison de l'importante recherche menée sur la synthèse de textures au cours des 30 dernières années, il serait impossible de recenser toutes les méthodes publiées. Toutefois, les recherches les plus récentes et les méthodes les plus importantes proposées ces dernières années peuvent être mentionnées.

1.4.1 Les méthodes procédurales

les méthodes de synthèse procédurale englobent la conception de textures au moyen de fonctions mathématiques ou d'algorithmes possédant un coût de calcul constant. Ces outils se révèlent particulièrement adaptés à la génération de textures pour des éléments au sein d'environnements virtuels, à l'exemple des univers de jeux vidéo [15]. Divers auteurs ont entrepris de reproduire des phénomènes du monde réel, comme les flux d'eau, la corrosion ou la dispersion de particules dans les matériaux, en se fondant sur des propriétés physiques ou chimiques observées [16]. Ces approches reposent sur la transformation de signaux préexistants en textures recherchées. Elles servent fréquemment à engendrer des textures hautement structurées, telles que des motifs de pavage, ou, en revanche, à créer des textures entièrement dépourvues de structure, à l'instar du bruit de Worley [16]. D'autres formes de bruit ont également été avancées, notamment les bruits de Gabor [17]. L'avantage principal des méthodes procédurales est leur flexibilité, permettant la création de textures variées et réalistes. De plus, ces méthodes sont efficaces en termes de temps de calcul, car leur coût est fixe, ce qui les rend particulièrement utiles dans des environnements en temps réel, tels que les jeux vidéo.

1.4.2 Les approches de synthèse à partir d'un exemple

Ces méthodes requièrent une ou plusieurs textures exemplaires en tant qu'éléments d'entrée. La majorité d'entre elles génèrent des textures en reproduisant directement les pixels ou les portions (sous-images) des images d'origine. Par conséquent, ces approches

visent à produire une nouvelle texture aussi similaire que possible à la texture modèle (d'origine) ou exemplaire. Elles se divisent en trois principales catégories :

1.4.2.1 Synthèse paramétrique de texture par analyse

Cette catégorie d'approches se fonde sur une représentation statistique de la texture originale. La synthèse d'une nouvelle texture s'effectue en respectant un ensemble de contraintes statistiques (telles que des signatures texturales ou des paramètres) qui sont déduites à partir de l'image d'origine et appliquées à la texture générée [18, 19].

Le principe de la synthèse paramétrique par analyse est résumé dans la Figure 1.7. Les caractéristiques de l'échantillon initial sont extraites pour obtenir les paramètres nécessaires au processus de synthèse. La texture synthétique est ensuite comparée à l'exemple original pour garantir leur similitude visuelle.



Figure 1.7: Principe des méthodes de synthèse de texture par paramètres

1.4.2.2 Synthèse de texture pixel par pixel

Ces techniques engendrent la texture de pixel en pixel tout en préservant la cohérence entre la texture locale et ses voisins. Elles reposent habituellement sur les principes des champs de Markov, lesquels décrivent la texture comme une réalisation d'un processus local et constant. Afin de maintenir des temps de calcul raisonnables, la plupart de ces méthodes emploient des algorithmes de recherche performants, conjugués à une implémentation multi-échelle capable de représenter le modèle à diverses résolutions, sans imposer une charge de calcul considérablement accrue [20].

1.4.2.3 Synthèse de texture par patch

La synthèse basée sur les patches est apparue comme une amélioration, en termes de complexité de calcul, par rapport aux approches basées sur les pixels qui souffrent d'un coût de calcul élevé, en particulier lorsqu'il s'agit de reproduire des images hautement structurées. L'approche générale consiste à sélectionner dans l'exemple le patch le plus similaire au voisinage actuel dans la texture de sortie, puis à réduire les artefacts et les défauts de contour en utilisant des techniques d'optimisation.

Il convient de noter que la famille des approches basées sur l'exemple comprend la plupart des techniques de synthèse [21].

1.4.3 Méthodes de synthèse sur la base d'un modèle

Ces méthodes consistent à générer un modèle probabiliste pouvant être utilisé pour décrire la texture et également pour la synthétiser. Les paramètres du modèle devraient capturer les caractéristiques visuelles importantes de la texture. Différents modèles existent dans la littérature, tels que [19, 22]. Notez que le modèle peut être obtenu par le biais d'un processus d'apprentissage à partir d'une image d'exemple, ce qui rend certaines de ces approches très proches des algorithmes de synthèse basés sur l'exemple.

Il est important de mentionner que certaines méthodes de synthèse peuvent appartenir à différentes catégories de cette classification. Par exemple, la synthèse peut être réalisée à partir d'un exemple, en utilisant une approche procédurale, basée sur l'analyse d'un modèle (par exemple, la synthèse par convolution de Gabor [17]).

Dans ce qui suit, nous nous concentrons sur les méthodes de synthèse de texture basées sur l'exemple nous établissons un état de l'art exhaustif.

1.5 Approches de synthèse à partir d'un exemple

Diverses méthodes sont employées pour la synthèse des textures en se fondant sur des exemples. Ces méthodes exigent une ou plusieurs textures exemples afin de fonctionner. La majorité de ces techniques repose sur la reproduction des pixels ou des sous-images (patches) présents dans l'image d'origine. Cette duplication vise à créer une nouvelle texture qui imite au mieux le modèle ou l'exemple initial. Ces méthodes facilitent la génération d'une vaste gamme de textures et sont classées en trois catégories principales :

1.5.1 Méthodes non paramétriques

Au cours des décennies récentes, un éventail d'algorithmes pour l'analyse et la création de textures ont vu le jour. En particulier, les techniques fondées sur les champs aléatoires de Markov et les approches basées sur les fragments d'images qui ont prouvé leur efficacité pour générer des textures structurées.

L'une des premières méthodes [23], classée comme un algorithme de synthèse non paramétrique basé sur les champs de Markov, consiste à ordonner les pixels selon un chemin particulier (type de balayage), par exemple un balayage lexicographique, et de synthétiser chaque pixel à partir d'une représentation nonparamétrique de la fonction de probabilité

conditionnelle. Cette fonction probabiliste est estimée à partir d'échantillons de la texture d'entrée, à l'aide d'un modèle de mélange gaussien. Les principaux problèmes associés à cette approche sont la taille réduite du voisinage et la causalité artificielle induite par l'existence d'un chemin prédéfini. Des améliorations ont été proposées en utilisant l'estimateur de densité non paramétrique "fenêtre de Parzen" [24]. La méthode de Paget and Longstaff [24], qui modélise la texture par un champ aléatoire de Markov, consiste à synthétiser la texture, pixel par pixel, en maximisant la vraisemblance de chaque pixel en termes de fonction de densité de probabilité conditionnelle locale (LCPDF). Plus précisément, elle attribue à chaque pixel la valeur la plus probable, connaissant son voisinage. Ce processus nécessite la connaissance du LCPDF, qui représente un modèle statistique unique décrivant les interactions entre les valeurs des pixels. Le LCPDF est estimé à partir de l'image d'entrée, en utilisant l'estimateur de fenêtre de Parzen. La méthode utilise un algorithme multi-échelle permettant d'obtenir des réalisations plus grandes de la texture. Cela consiste à synthétiser le niveau de résolution le plus bas, puis à augmenter progressivement la résolution à synthétiser pour construire les autres niveaux de la pyramide multi-échelle. Il convient de noter que cette méthode est capable de synthétiser des textures complexes et structurées. La technique de synthèse de texture basée sur l'échantillonnage non paramétrique, introduite par Efros et Leung [1], ne requiert pas l'utilisation d'un estimateur de densité de probabilité. Elle repose plutôt sur un processus de recherche du voisin le plus proche, permettant ainsi de créer une texture en prélevant directement de nouvelles valeurs à partir de l'échantillon original. Cette méthode s'avère performante pour générer des textures hautement structurées. La Figure 1.8 présente l'approche non paramétrique d'Efros et Leung (1999). La texture est construite de pixel en pixel en identifiant tous les voisins causaux de l'échantillon d'entrée (les carrés à gauche) qui présentent des similitudes avec l'environnement du pixel en cours (carré à droite). Ensuite, l'un de ces voisins est choisi au hasard, et son pixel central est reproduit. L'algorithme d'Efros et Leung, qui est une approche mono-échelle, a été le précurseur de la méthode proposée par Wei et Levoy [2], qui est l'une des approches les plus polyvalentes. Cet algorithme non paramétrique est basé sur une hypothèse de Markov modélisant la texture comme un processus probabiliste dans lequel toutes les interactions sont locales (localité) et les interactions à plus grande échelle peuvent être reproduites par un effet de propagation (stationnarité). La méthode vise à obtenir une texture de sortie qui est localement similaire à un patch de l'échantillon d'entrée. Par conséquent, chaque pixel synthétisé est déterminé de manière à reproduire la similarité locale. Le processus de synthèse consiste à modifier progressivement l'image de sortie pour qu'elle ressemble à l'échantillon d'entrée [2]. La texture est synthétisée pixel par pixel, de manière itérative, en suivant un type de balayage prédéfini (généralement lexicographique ou marche aléatoire) et une forme de

voisinage prédéfinie (généralement causale ou carrée).

Pour chaque pixel de sortie, son voisinage est capturé, le voisinage le plus similaire est recherché dans l'exemplaire et le pixel correspondant est copié à la position cible dans la texture de sortie, comme illustré dans la Figure 1.8. La distance euclidienne est utilisée pour mesurer la similarité entre deux voisinages.

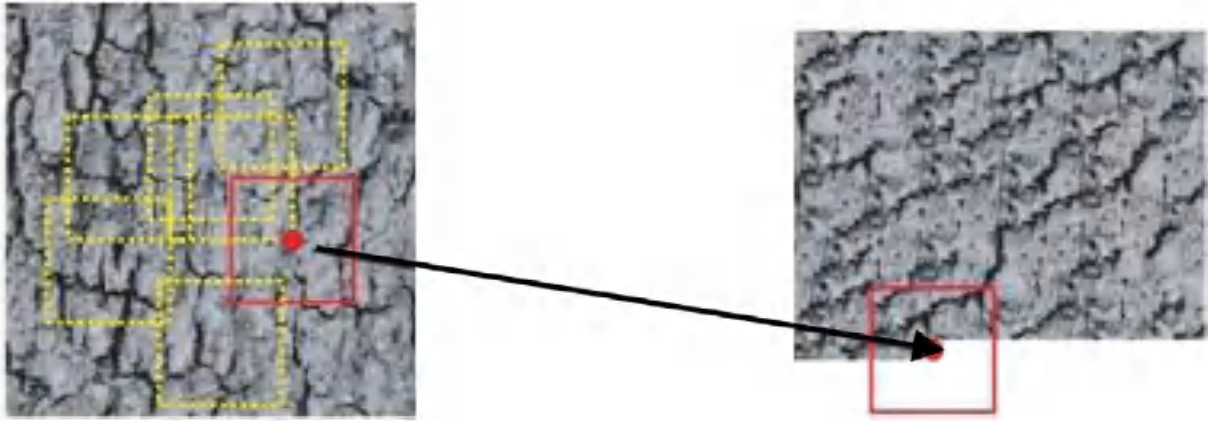


Figure 1.8: Principe de synthèse d'Efros et Leung [1]

Pour éviter l'utilisation de grands voisinages, ce qui rend l'algorithme lent, une approche multi-échelle est utilisée. Elle consiste à construire des pyramides de textures multi-résolutions afin de capturer les structures de manière plus compacte, en utilisant moins de pixels dans les échelles de résolution inférieure de la pyramide [2]. La synthèse commence à l'échelle de résolution la plus basse et s'arrête à l'échelle de résolution la plus élevée, de sorte que chaque échelle est synthétisée à partir de l'échelle supérieure la Figure 1.9.

Afin de réduire la complexité computationnelle de la recherche exhaustive, les voisinages sont considérés comme des points dans un espace multidimensionnel. Ensuite, le processus de correspondance des voisins peut être vu comme un problème de recherche du plus proche voisin en plaçant les voisins dans un modèle de probabilité de regroupement basé sur l'étude réalisée par [23]. La recherche du plus proche voisin utilise la recherche binaire fournie par la quantification vectorielle à structure arborescente (TSVQ) [25].

Afin d'accélérer la recherche du voisinage le plus similaire, une autre approche efficace consiste à restreindre, pour chaque pixel à créer, le nombre de voisinages à confronter, conformément à [26]. Ces voisinages sont définis comme les environnements des échantillons déjà exploités pour la synthèse des pixels adjacents au pixel de sortie actuel.

Une autre catégorie de méthode qui exploite des similitudes est exposée dans [27]. Cet algorithme fusionne les concepts décrits dans [28] et les étend à des cas d'appariement

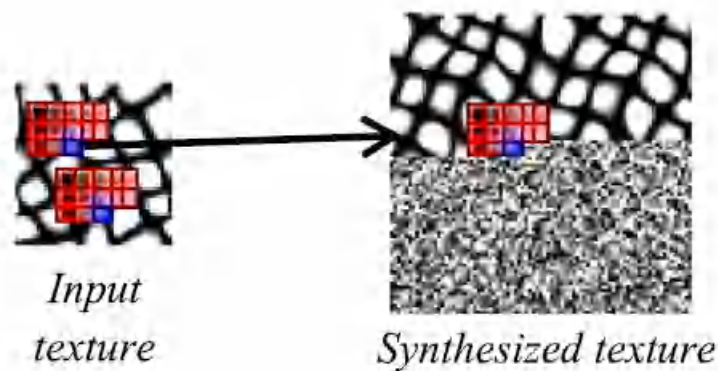


Figure 1.9: Représentation schématique du processus de synthèse de Wei et Levoy [2] utilisant un voisinage causal en forme de L et un balayage lexicographique

d'images. D'autres approches visent à éviter les comparaisons avec le voisinage le plus proche, ce qui entraîne une charge de calcul considérable [29]. Dans l'objectif de simplifier la charge de calcul, une méthode basée sur la k -cohérence a été présentée par [20]. Cette approche permet de créer des fonctions de texture qui rendent compte de l'apparence d'une surface dans le monde réel, en prenant en compte les conditions d'éclairage et les angles d'observation. Le principe sous-jacent consiste à anticiper l'identification, pour chaque pixel de la texture originale, des k correspondances les plus proches dans cette même texture. En d'autres termes, il s'agit des k pixels dont les voisinages présentent la plus grande similitude avec le voisinage du pixel en question, en utilisant une mesure de distance spécifique pour évaluer ces similitudes. Lors de la phase de synthèse, les pixels environnants du pixel cible servent à créer un groupe de candidats parmi lesquels la valeur du pixel optimal est déterminée puis reproduite. La valeur de k varie en fonction du type de texture à synthétiser : une valeur élevée est généralement privilégiée, tandis qu'une valeur réduite est plus adaptée aux textures comportant des détails à haute fréquence.

L'une des premières approches de synthèse de texture basée sur les patches consiste à sélectionner aléatoirement, à partir de l'image d'entrée, des patches qui ressemblent, en termes d'une certaine mesure, aux patches environnants dans l'image de sortie, puis à réduire les artefacts ou les défauts de contour dans les régions de chevauchement [1]. Plus précisément, chaque bloc de sortie est choisi de manière à correspondre à ses voisins dans la région de chevauchement, et l'effet de bloc est réduit en utilisant des coupes d'erreurs minimales entre les patches. Cette méthode a démontré son efficacité dans le transfert de texture, c'est-à-dire dans le rendu d'un objet avec une texture prise à partir d'un autre objet. Une extension de cet algorithme avec des caractéristiques profondes est proposée

dans [30]. Cette extension consiste à combiner des modèles de champ aléatoire de Markov génératif et des réseaux neuronaux convolutifs profonds entraînés de manière discriminative (dCNNs). Le champ aléatoire de Markov agit sur des niveaux supérieurs d'une pyramide de caractéristiques dCNN et contrôle la disposition de la texture, ce qui permet d'augmenter la plausibilité visuelle. Cette méthode est capable à la fois de correspondre et d'adapter des caractéristiques locales avec une variabilité considérable. La synthèse de texture basée sur les patchs a également été réalisée sous la forme de mosaïques chaotiques [31], qui consiste à distribuer aléatoirement les patchs de texture d'entrée sur la texture de sortie et à lisser les bordures des patchs qui se chevauchent. L'inconvénient associé à cette approche est que le processus de synthèse produit de grandes régions où la texture d'entrée est copiée sans aucune variation par rapport à l'exemple, ce qui n'est pas visuellement satisfaisant. Un algorithme de recherche de voisinage rapide basé sur une structure pyramidale (quad-tree) de la texture d'entrée a été introduit. Cela assure une technique d'ajustement séquentiel optimal en fixant un patch après l'autre [32]. Une version améliorée [33] utilise la technique de coupe de graphe afin de déterminer la région optimale du patch, indépendamment du décalage entre l'exemple et la texture de sortie. Il a été démontré que la synthèse d'une couche géométrique peut aider à la synthèse de la texture [34]. La méthode de synthèse en deux étapes proposée par [34] consiste à générer, dans la première étape, un flux d'orientation synthétique qui est utilisé, dans une deuxième étape, pour contraindre la synthèse de la texture. En fait, la première étape consiste à synthétiser une carte d'orientation à partir de la carte d'orientation de la texture d'entrée. La carte d'orientation est déduite de l'analyse du tenseur de structure locale de l'exemple [35]. L'algorithme de synthèse est basé sur l'algorithme de Heeger et Bergen [36]. Ce dernier reproduit les distributions marginales des coefficients issus de la décomposition multi-échelle de la carte d'orientation. La deuxième étape consiste à ajuster les distributions marginales des coefficients d'une décomposition grouplet conditionnée par la carte d'orientation. Toute la procédure de synthèse est très efficace lorsqu'il s'agit de textures localement parallèles (par exemple, les rayures et les tourbillons), mais elle échoue à reproduire les textures structurées et les textures comprenant des motifs croisés considérés comme des singularités du flux d'orientation.

Un autre algorithme de synthèse basé sur la structure est proposé par [3]. Une première étape de cet algorithme consiste à reproduire la couche de structure de l'exemple. La couche de structure est analysée et représentée par le tenseur de structure local, sans se limiter au flux d'orientation comme dans Peyré (2009), mais en tenant compte de l'orientation locale, de l'énergie et du degré d'anisotropie [35]. À cette fin, l'algorithme non paramétrique de Wei et Levoy (2000) est adapté aux spécificités du tenseur de structure. Plus précisément, le champ de tenseur de structure de la texture d'entrée est calculé, puis

un champ de tenseur de structure de sortie est synthétisé avec le processus de synthèse de Wei et Levoy, tandis que quatre métriques d'espace de tenseurs sont utilisées pour comparer les tenseurs de structure [3]. Le champ de tenseur de structure synthétisé est ensuite utilisé dans une deuxième étape pour contraindre la synthèse de la texture (Fig. 1.10). Le processus de synthèse est également similaire à celui de Wei et Levoy. Cependant, chaque voisinage est divisé en deux composantes : un voisinage d'intensité dans les textures d'entrée et de sortie, et un voisinage de tenseur dans le champ de tenseur de la texture d'entrée et dans le champ de tenseur synthétisé. La métrique utilisée pour mesurer la ressemblance entre les voisinages combine la somme des distances au carré pour la composante de voisinage d'intensité et l'une des quatre métriques d'espace de tenseurs pour la composante de tenseur. Afin de réduire la charge de calcul, des pyramides multi-résolution sont utilisées en plus d'une interpolation bilinéaire qui remplace la synthèse exhaustive du niveau de pyramide le plus bas [37]. Une extension de cette approche, qui consiste à synthétiser des textures de forme arbitraire, a été proposée dans [38].

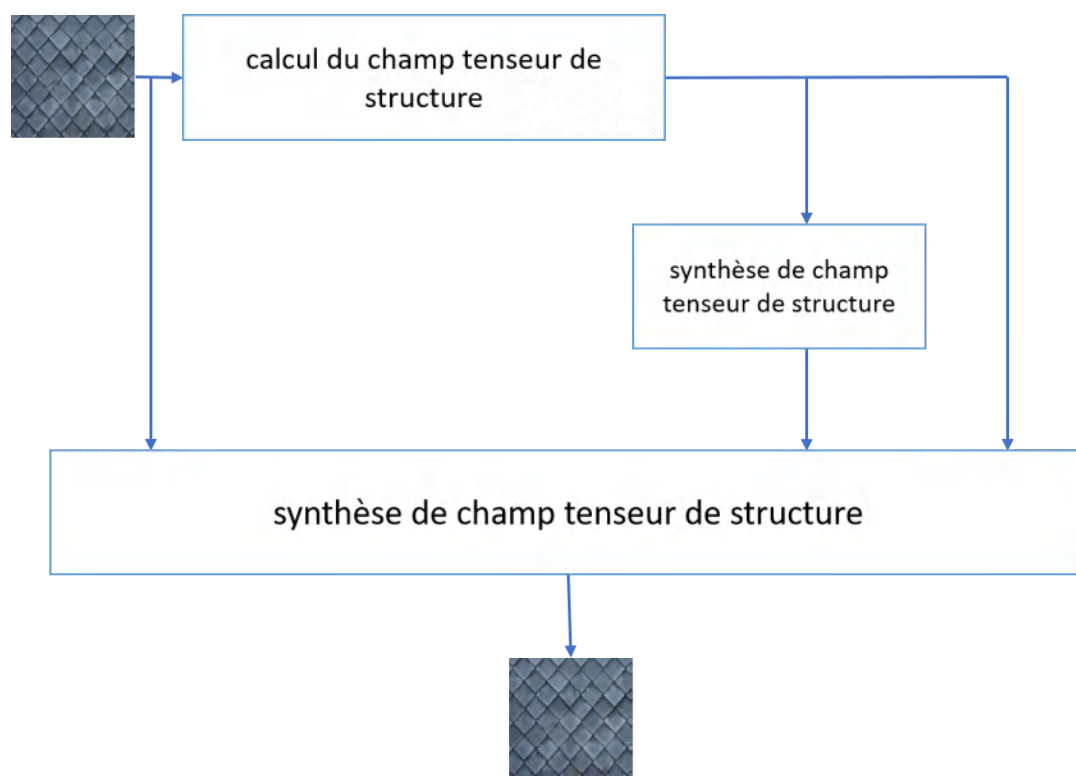


Figure 1.10: Illustration de l'approche de synthèse de structure/texture en deux étapes [3]

1.5.1.1 Les méthodes paramétriques

Les approches de synthèse paramétrique constituent une autre famille d'algorithmes de synthèse de textures. La première approche paramétrique remarquable a été proposée

par [36]. Cette méthode contraint la synthèse en utilisant des histogrammes des réponses des filtres à différentes échelles et orientations d'une pyramide orientable. Il s'agit de l'une des premières méthodes de synthèse de textures en couleur. La méthode consiste en une décomposition itérative de l'image en un ensemble de sous-bandes de fréquence, l'ajustement des histogrammes et la reconstruction de l'image. La méthode proposée dans [39] surpasse le processus itératif. Elle consiste à modifier l'image d'entrée de manière à restreindre la distribution conditionnelle de la sortie du filtre sur plusieurs résolutions. Une amélioration substantielle a été réalisée par [19] en utilisant la transformée en ondelettes pour paramétrer le modèle avec un ensemble de descripteurs statistiques calculés à différentes orientations et échelles. Cette méthode étend la modélisation statistique au deuxième ordre et met à jour la réponse du filtre, non pas par égalisation d'histogramme, mais par un algorithme de descente de gradient. Plusieurs améliorations de cette méthode ont été proposées [18]. Cependant, de telles approches conduisent à des formulations complexes difficiles à optimiser. Les méthodes stochastiques ont également montré leur capacité à produire des résultats impressionnants avec des micro-textures stochastiques. Par exemple, le bruit de taches ou le bruit de tir bidimensionnel a été introduit par [40] pour synthétiser de nouvelles textures à partir d'images de taches générées en assemblant un grand nombre de fonctions d'intensité à des positions aléatoires. Dans cette approche, l'asymptote du bruit discret de taches (DSN) est obtenue par randomisation des phases de tous les coefficients de la transformée de Fourier. Dans [41] les auteurs proposent un algorithme de bruit de phase aléatoire (RPN) basé sur la randomisation de la phase de la transformée de Fourier. Cet algorithme est capable de reproduire des textures caractérisées par le module de leur transformée de Fourier, plus précisément des micro-textures stochastiques. Les textures à phase aléatoire supposent une superposition linéaire. Lorsque la superposition linéaire n'est pas adaptée, comme c'est le cas pour la formation de certaines images naturelles (par exemple, les images satellite), plusieurs auteurs préfèrent le principe d'occlusion qui a conduit au modèle de feuilles mortes stochastiques [42]. Un algorithme de synthèse de textures gaussiennes flexibles est proposé dans [43]. Il consiste à résumer un échantillon de texture en un texton orienté vers la synthèse, c'est-à-dire une petite image pour laquelle la simulation de bruit discret de taches est plus efficace que l'algorithme de convolution classique. Dans [44] les auteurs ont récemment proposé une approche intégrant des méthodes basées sur des patches et des méthodes stochastiques. Ils proposent un cadre unifié qui impose des contraintes statistiques (de premier ordre) sur l'utilisation d'atomes de dictionnaire adaptatifs, ainsi que des contraintes de second ordre sur les valeurs des pixels. Comme mentionné précédemment, les méthodes de synthèse de textures basées sur un modèle paramétrique consistent à construire un modèle décrivant l'image, qui peut ensuite être utilisé pour analyser la texture et la synthétiser. Les modèles autorégressifs

(AR) supposent une interaction locale entre les pixels de texture. L'intensité d'un pixel est décrite par une somme pondérée des intensités des pixels voisins [45]. Il faut noter que les méthodes basées sur ces modèles ne fonctionnent pas bien pour la synthèse de textures naturelles [46].

Le modèle de Wold est une méthode plus récente utilisée pour modéliser les textures. Il décompose la texture en trois composantes orthogonales : une composante aléatoire, une composante harmonique et une composante évanescente (c'est-à-dire aléatoire dans une direction et déterministe dans la direction orthogonale). Par conséquent, ils sont plus flexibles que les modèles AR, mais leurs paramètres sont souvent difficiles à estimer [47]. Les modèles fractals [48] sont utilisés en infographie pour créer des textures d'images réalistes. Cela est possible car la dimension fractale est relativement insensible à l'échelle des images et car elles présentent une forte corrélation avec l'impression visuelle causée par la rugosité de surface. Les fractales sont capables de décrire des objets présentant un degré élevé d'irrégularité. De nombreuses méthodes de synthèse de textures basées sur des fractales ont été proposées, utilisant un décalage moyen et un filtrage de Fourier [49]. Les méthodes fractales représentent fidèlement les textures naturelles aléatoires, mais elles échouent à décrire les structures locales de l'image car elles manquent de sélectivité en termes d'orientation.

De nombreux autres modèles ont également été proposés, tels que les champs aléatoires de Markov et de Gibbs [50], les modèles de Gabor et les modèles d'ondelettes [51]. Dans ces approches, la texture est modélisée par un modèle probabiliste ou par une combinaison linéaire d'un ensemble de fonctions de base.

Les difficultés résident dans le choix du modèle approprié pour la texture cible et dans l'estimation de ses paramètres, ce qui entraîne une perte de généralité. Une méthode de synthèse basée sur les champs aléatoires de Markov et utilisant l'optimisation est proposée dans [33]. Elle repose sur une mesure de similarité permettant de mesurer la qualité d'une texture synthétisée par rapport à un échantillon d'entrée en minimisant une fonction d'énergie, optimisée à l'aide d'un algorithme de type Expectation-Maximization. Dans [52] les auteurs ont proposé une méthode de synthèse paramétrique basée sur des patches qui utilise un algorithme aléatoire pour trouver rapidement des correspondances approximatives entre les patches d'image les plus proches. La méthode de Simakov [53] utilise une mesure de similarité bidirectionnelle pour la synthèse visuelle de données. Cette approche consiste à capturer fidèlement les informations visuelles des données d'entrée et à introduire, autant que possible, de nouveaux artefacts visuels qui ne sont pas présents dans les données d'entrée tout en préservant la cohérence. Une approche basée sur la coupe de graphe utilisant des décalages entre les caractéristiques répétitives de l'image est proposée dans [54]. Cette méthode permet d'obtenir de meilleures structures synthétisées en

associant des patches d'image similaires et en déterminant leurs positions relatives.

1.5.2 Méthodes de synthèse basées sur l'intelligence artificielle

La synthèse de texture par Deep Learning (CNN) et GAN est une méthodologie de synthèse de texture utilisant les réseaux de neurones convolutionnels (CNN) et des réseaux antagonistes génératifs (GAN) pour générer des textures réalistes. La méthode commence par l'apprentissage d'un CNN à partir d'un ensemble de données d'apprentissage de textures. Le CNN apprend à extraire les caractéristiques des textures en utilisant des filtres convolutionnels et des couches de pooling. Une fois que le CNN est entraîné, il peut être utilisé pour générer de nouvelles textures. Cependant, les textures générées par un CNN peuvent manquer de réalisme et de variété. C'est là que les réseaux antagonistes génératifs (GAN) entrent en jeu. Les GAN utilisent deux réseaux de neurones : un générateur et un discriminateur. Le générateur est entraîné à générer des textures, tandis que le discriminateur est entraîné à distinguer les textures générées par le générateur des textures réelles. Les deux réseaux sont entraînés de manière antagoniste, de sorte que le générateur apprend à produire des textures plus réalistes, tandis que le discriminateur apprend à distinguer les textures réelles des textures générées.

En combinant les deux techniques, la synthèse de texture par Deep Learning (CNN) et GAN permet de générer des textures réalistes et variées. Elle a de nombreuses applications dans le domaine de l'animation 3D, de la réalité virtuelle et augmentée, de la conception graphique et de la création de jeux vidéo.

Un nouveau modèle de textures naturelles basé sur les espaces de caractéristiques des réseaux de convolution optimisés pour la reconnaissance d'objets est présenté dans [4]. Ils utilisent le réseau de convolution VGG-19 [55], qui utilise de petits filtres pour trouver des caractéristiques très localisées et une architecture de pointe pour la reconnaissance d'objets. Les caractéristiques de différentes tailles sont extraites des différentes couches du réseau de convolution entraîné. Les corrélations entre les réponses de ces différentes couches sont utilisées comme statistique de résumé spatial. De nouvelles images sont générées en utilisant la descente de gradient sur une image aléatoire pour produire la même description stationnaire dérivée de la statistique de résumé spatial. Cela est similaire à [19], mais en utilisant des réseaux de convolution pour la génération automatique de caractéristiques et de descripteurs spatiaux. Ils démontrent comment leur méthode surpasse [19], montrant les avantages potentiels des caractéristiques auto-apprises.

[56] démontre que les filtres des réseaux de convolution peu profonds peuvent être efficacement utilisés comme modèle pour les textures naturelles. Il montre que des réseaux moins profonds peuvent produire des synthèses de textures d'une qualité perceptive comparable

aux méthodes de pointe qui nécessitent des réseaux de convolution profonds et multicouches. Ils comparent leurs résultats à [4], qui utilise un réseau VGG à 19 couches, en présentant des résultats visuellement similaires à l'aide d'un réseau monocouche à échelle multiple, bien qu'ils commentent qu'ils montrent moins de variabilité.

Le travail de Gatys [4] a été approfondi dans [57] par l'exploration de la dimension temporelle dans la synthèse de textures. Ils substituent un réseau de convolution 3D au réseau de convolution 2D utilisé dans le cadre de Gatys. En utilisant des ConvNets 3D pré-entraînés [58], ils sont capables de calculer des statistiques de corrélation sur les réponses des caractéristiques dans la procédure de synthèse avec la dimension temporelle ajoutée. Les résultats présentés démontrent que des textures dynamiques réalistes peuvent être synthétisées. Dans [59], ils examinent l'exploitation des connaissances acquises à partir de grands ensembles de données et leur utilisation dans des ensembles de données plus petits. En utilisant le transfert d'apprentissage, en transmettant les pondérations du réseau appris pour les utiliser comme configuration initiale, ils ont réussi à obtenir des résultats dépassant les méthodes faites à la main sur des ensembles de données avec peu de données d'entraînement. Cela démontre la capacité des CNN à apprendre les caractéristiques les plus pertinentes pour l'analyse de textures et leur capacité de généralisation.

Récemment, dans [60], un nouveau modèle de texture, appelé Conditional Generative ConvNet (cgCNN), a été proposé en combinant les statistiques profondes de texture avec le cadre probabiliste de Generative ConvNet (gCNN) [61]. Étant donné une texture, cgCNN utilise les statistiques profondes d'un ConvNet entraînable pour construire une distribution conditionnelle basée sur l'énergie, qui est ensuite entraînée via l'estimation de la vraisemblance maximale (MLE). En échantillonnant à partir de la distribution conditionnelle apprise, de nouvelles textures peuvent être créées. contrairement aux modèles de texture précédents qui s'appuyaient sur des ConvNets pré-entraînés. Par conséquent, il présente deux avantages majeurs : 1) Il permet la création d'images unifiées. 2) La méthode d'échantillonnage peut facilement échapper aux minima locaux. Ainsi, il peut synthétiser des textures avec des structures non locales sans utiliser de termes de pénalité supplémentaires.

Cependant, la synthèse de texture par Deep Learning (CNN) et GAN présente également des défis et des limites, tels que la complexité de l'apprentissage, le besoin d'un grand ensemble de données d'apprentissage et les difficultés à générer des textures complexes. Des recherches supplémentaires sont nécessaires pour améliorer la qualité des textures générées et pour développer des méthodes plus efficaces pour la synthèse de textures complexes comme le montre la figure 1.11.

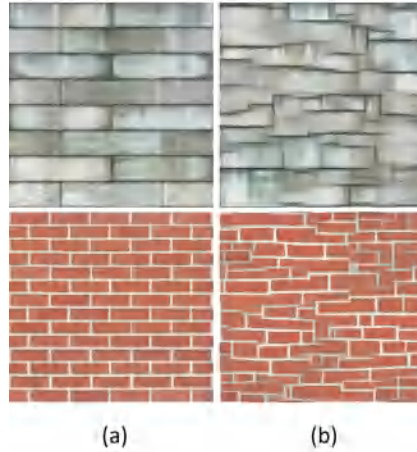


Figure 1.11: Synthèse de texture utilisant des CNN selon [4]. a : une texture exemplaire. b : résultat synthétisé en utilisant le modèle CNN.

1.5.3 Synthèse de texture de couleur

Plusieurs applications graphiques nécessitent la synthèse de textures en couleur pour représenter des textures du monde réel observées dans différentes conditions d'éclairage. Par exemple, en rendu d'images, les textures peuvent imiter les détails de surface d'objets réels. La plupart des algorithmes de synthèse existants conviennent aux textures en couleur en considérant simplement les trois canaux (rouge, vert et bleu) de la texture couleur dans le modèle RVB [41]. Dans [28], les auteurs considèrent les composantes de couleur comme des dimensions supplémentaires pour la représentation des pixels et des voisinages. En revanche, dans [41] génèrent la magnitude de la transformée de Fourier des trois composantes de manière indépendante et elles attribuées à chacune une phase aléatoire unique. Un problème important dans les méthodes de synthèse basées sur la structure est le calcul de la couche de structure de la texture en couleur. [62] propose une formulation pour le gradient d'une image en couleur avec l'extraction d'un seul vecteur (direction et magnitude de la variation maximale). D'autres approches basées sur les tenseurs [3] reposent sur la composante de luminance de la texture en couleur et/ou ajoutent les composantes de tenseur calculées pour chaque canal de couleur dans un modèle RVB.

1.6 Motivation

1.6.1 Limite des méthodes existantes

Les méthodes existantes ont fait leurs preuves en matière de traitement d'une grande variété de textures et ont donné des résultats très satisfaisants dans de nombreux domaines

d'application. Cependant, la plupart d'entre elles sont incapables de traiter les textures présentant de grandes variations directionnelles, comme le montrent les études [63], par exemple, dans le cas de textures anisotropes structurées d'échantillons de matériaux. Dans ces situations, la plupart des approches existantes ont tendance à produire des textures plus régulières que l'exemple d'origine. De plus, elles éprouvent des difficultés à synthétiser des textures présentant des structures laminaires complexes.

Il existe également des cas critiques où les textures ont des motifs multi-échelles que les approches classiques ne parviennent pas à reproduire. Cela est illustré à la Fig 1.12.



Figure 1.12: Synthèse de texture utilisant des CNN selon [4]. a : une texture exemplaire. b : résultat synthétisé en utilisant le modèle CNN.

1.7 Approche de synthèse proposée : vue globale

Comme il a été mentionné dans les titres précédents, les méthodes existantes n'ont pas rencontré un franc succès dans la synthèse de texture. De plus, les méthodes basées sur les réseaux de neurones convolutifs (CNN) n'ont pas atteint le même niveau de réussite dans la représentation des textures que dans d'autres tâches. Par conséquent, dans la présente étude, nous nous intéressons à une nouvelle méthode qui a réussi à représenter de manière remarquable une large gamme de textures, à savoir les modèles 2D-RCA. En effet, les auteurs dans [64], ont généré à partir de plusieurs modèles 2D-RCA de premier ordre certaines textures synthétiques et ont prouvé qu'elles sont tout à fait similaires aux textures naturelles appropriées.

Les motivations pour considérer certaines textures comme étant une réalisation d'un processus autoregressif bidimensionnel à coefficients aléatoires 2D-RCA du premier ordre reposent sur trois arguments solides : Des tests de simulation ont démontré que ce modèle est capable de représenter efficacement une vaste gamme de textures. Il convient de noter que les modèles 2D-RCA comportent de nombreux avantages significatifs. Ce modèle se distingue par des zones regroupées présentant des variations intenses d'échelle de gris, suivies de zones présentant des changements plus faibles.

Le premier ordre a été choisi car les modèles du premier ordre sont largement utilisés

dans la pratique. Ils permettent de représenter la valeur d'un site spécifique en ne tenant compte que des pixels voisins immédiats.

Ces arguments soutiennent fermement l'emploi du procédé stationnaire 2D-RCA du premier ordre pour la représentation des textures. Notre méthode dans cette étude implique la génération de textures à partir d'un exemple, en segmentant chaque image originale en blocs modélisés via le modèle 2D-RCA. Pour approximer les blocs à partir de l'image source, nous employons la méthode généralisée des moments (GMM) pour estimer les paramètres des modèles en utilisant diverses tailles de fenêtres. Cette recherche apporte des insights essentiels sur les images créées.

Les résultats satisfaisants que nous avons obtenus ont été comparés à ceux fournis par des méthodes bien établies. L'algorithme proposé se révèle supérieur aux approches les plus récentes, démontrant ainsi son efficacité et sa performance.

1.8 Conclusion

Ce chapitre présente une récapitulation des principes essentiels de la synthèse de texture, de ses diverses applications et des différents types d'algorithmes proposés dans la recherche existante. L'examen approfondi de la littérature démontre clairement que la synthèse de texture est un domaine en constante évolution, témoignant de l'engagement actif de la communauté scientifique envers cette discipline.

Cependant, les méthodes actuelles décrites dans la littérature rencontrent des difficultés lorsqu'il s'agit de synthétiser des textures présentant une structure complexe, comme celles qui présentent des variations directionnelles, des motifs multi-échelles ou des non-stationnarités.

Chapitre 2

Extraction des caractéristiques et classification des textures : Etat de l'art

2.1 Introduction

Le traitement et l'analyse des textures jouent un rôle essentiel dans de nombreux domaines tels que la vision par ordinateur, la reconnaissance de formes et l'analyse d'images. La texture représente les variations spatiales des niveaux de gris ou des couleurs dans une image, fournissant des informations cruciales pour la perception et la compréhension visuelle. La classification précise des textures est une tâche complexe, mais d'une importance primordiale pour de nombreuses applications pratiques telles que la détection, l'imagerie aérienne et médicales.

Dans ce chapitre, nous explorons les méthodes et les caractéristiques de classification de la texture. Tout d'abord, nous examinons les caractéristiques statistiques, qui exploitent les propriétés de distribution des niveaux de gris dans une région texturée. Ensuite, nous abordons les caractéristiques structurelles, qui capturent les relations spatiales et les motifs récurrents présents dans la texture. Nous discuterons également les méthodes de classification classiques, allant des classificateurs bayésiens aux machines à vecteurs de support (SVM) et aux réseaux de neurones, qui permettent de prendre des décisions de classification précises et fiables.

En explorant ces concepts et ces techniques, nous cherchons à fournir un aperçu approfondi des méthodes de classification de la texture, mettant l'accent sur les caractéristiques les plus efficaces et les approches de classification les plus prometteuses. Cette compréh-

sion approfondie contribuera à améliorer les performances de classification dans diverses applications, telles que la détection d'objets, la reconnaissance de motifs et la surveillance visuelle.

2.2 Méthodes d'analyse de textures

L'utilité de la texture dans le domaine du traitement d'image est indéniable, bien que définir précisément ce qu'est la texture présente une difficulté qui a donné lieu à plusieurs méthodes d'analyse de celle-ci [65]. Le but de ces méthodes est de capturer un ensemble d'attributs ou de paramètres visant à décrire les caractéristiques de la texture de manière représentative, pertinente et discriminatoire. Cette approche permet ainsi de distinguer une texture parmi d'autres. On peut les regrouper en cinq approches principales : l'approche structurelle, l'approche fréquentielle, l'approche statistique, l'approche basée sur les modèles et l'approche basée sur la morphologie mathématique.

2.2.1 Approche structurelle

Cette méthode vise à extraire et localiser les primitives des textures [66]. Ces approches se basent sur des techniques d'auto-corrélation pour retrouver l'emplacement de la primitive initialement extraite dans la texture, permettant ainsi de déduire une règle de placement. Ce qui distingue ces méthodes, c'est qu'elles suivent toutes un processus en deux étapes : d'abord l'extraction de la primitive, puis la recherche de la règle de placement.

2.2.2 Approche fréquentielle

Dans cette approche particulière, l'objectif est de décrire une texture en se basant sur ses caractéristiques fréquentielles.

Une texture dite "douce", où l'on ne perçoit pas de variations significatives du niveau de gris, se traduira par un spectre de puissance présentant des amplitudes élevées dans le domaine des basses fréquences. En revanche, pour une texture "grossière" et "granuleuse", où les variations du niveau de gris sont abruptes et rapprochées, le spectre de puissance affichera des amplitudes plus élevées dans le domaine des hautes fréquences.

2.2.2.1 Transformée de Fourier

Cette approche vise à extraire les caractéristiques fréquentielles de la texture. Pour une image numérique d'une taille $M \times N$, on définit la transformée de Fourier Discrète (TFD)

par l'équation suivante [66] :

$$F(\mu, v) = \frac{1}{M \times N} \sum_{u=0}^{v-1} \sum_{l=0}^{M-1} I(u, v) e^{2\pi i (\frac{u}{N} \mu + \frac{v}{M} v)} \quad (2.1)$$

L'expression de la transformée de Fourier inverse est :

$$I(u, v) = \frac{1}{M \times N} \sum_{\mu=0}^{N-1} \sum_{v=0}^{M-1} f(\mu, v) e^{2\pi i (\frac{\mu}{N} k + \frac{v}{M} l)} \quad (2.2)$$

Cette méthode offre la transformation d'une représentation de l'image dans le domaine spatial (coordonnées u et v) à une représentation dans le domaine fréquentiel (coordonnées μ et v), μ et v étant les fréquences spatiales des variations des niveaux de gris. Les termes de basse fréquence de cette expression indiquent les légères variations des niveaux de gris de l'image, les termes de hautes fréquences réfèrent à la variation abrupte comme dans le cas des contours d'objets ou un bruit.

Donc les caractéristiques de texture peuvent être extraites à l'aide de la transformée de Fourier, telles que l'énergie calculée dans des couronnes ou en fonction de certaines directions. Dans ce cas, le domaine des fréquences est subdivisé en anneaux ou secteurs angulaires, et l'énergie calculée dans ces zones détermine une caractéristique spécifique de la texture (voir Figure 2.1).

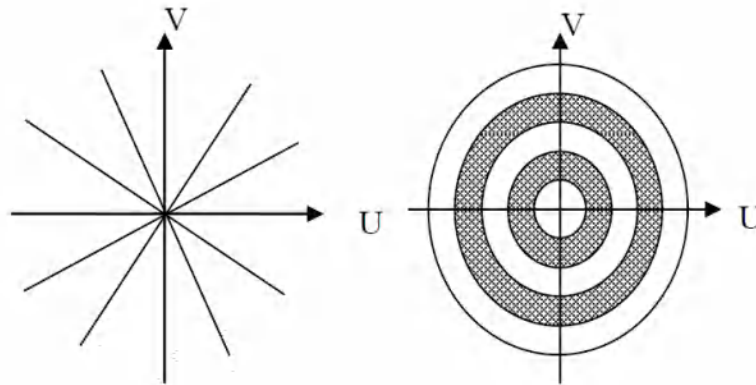


Figure 2.1: Caractéristiques de texture basées sur le spectre de puissance de l'image.

Les caractéristique déduites à partir de la figure 2.1 sont les suivantes :

énergie : Cet attribut représente l'énergie dans une couronne E_{r_1, r_2} . Il évalue la grossièreté ou la finesse de la texture. Ce paramètre est défini de la manière suivante :

$$f_{r_1, r_2} = \int_0^{2\pi} \int_{r_2}^{r_1} |F(u, v)|^2 dr d\theta \quad \text{avec} \quad r = \sqrt{u^2 + v^2} \quad \text{et} \quad \theta = \arctan\left(\frac{v}{u}\right) \quad (2.3)$$

L'énergie angulaire : θ_1, θ_2 représente l'énergie calculée dans chaque secteur angulaire et représentent l'orientation de la texture. Cette énergie est définie comme suit :

$$f_{\theta_1, \theta_2} = \int_{\theta_1}^{\theta_2} \int_0^\infty |F(u, v)|^2 dr d\theta \quad \text{avec} \quad r = \sqrt{u^2 + v^2} \quad \text{et} \quad \theta = \arctan\left(\frac{v}{u}\right) \quad (2.4)$$

2.2.2.2 Fonction d'Auto-corrélation

Une caractéristique cruciale que l'on trouve dans certaines textures est la répétition des motifs structurants à l'intérieur de l'image. On peut quantifier cette régularité, qui varie de la subtilité à la robustesse, en utilisant la fonction d'autocorrélation (ACF : autocorrelation function) de l'image. La fonction d'autocorrélation, symbolisée par ρ , d'une image $I(x, y)$ est définie de la manière suivante :

$$\rho(x, y) = \frac{\sum_{u=0}^N \sum_{v=0}^N I(u, v) I(u + x, v + y)}{\sum_{u=0}^N \sum_{v=0}^N I^2(u, v)} \quad (2.5)$$

Cette fonction établit une mesure de la similitude moyenne entre les pixels pour une distance donnée (x, y) . L'évolution de cette mesure en fonction des variations de distance (x, y) permet d'évaluer la régularité de la texture, fournissant ainsi une évaluation des relations spatiales entre les éléments constitutifs de la texture. Cette fonction est intimement liée à la taille des éléments structurants de la texture, comme la finesse de la texture par exemple. Lorsque les éléments structurants sont de grande taille, la fonction varie lentement avec l'augmentation de la distance, tandis qu'elle varie rapidement pour de plus petites structures. Si la fonction d'autocorrélation présente des variations périodiques de croissance et de décroissance en fonction de la distance, cela suggère que les éléments structurants sont périodiques. Dans le cas de textures régulières, la fonction d'autocorrélation affiche des pics et des creux.

2.2.2.3 Les masques de Laws

En 1980, Laws a présenté une nouvelle méthode visant à caractériser l'énergie d'une texture [67]. Pour identifier les pixels de texture à haute énergie dans une image, il a employé trois filtres : le filtre gaussien, le détecteur de contour et le filtre de Laplace. Les

masques de Laws sont créés par une convolution conjointe de trois masques de base 1×3 suivants :

$$L_3 = (1, 2, 1)$$

$$E_3 = (-1, 0, 1)$$

$$S_3 = (-1, 2, -1)$$

Les lettres initiales de ces masques désignent respectivement : la moyenne locale des niveaux de gris (Local averaging), la détection de contours (Edge detection) et la détection de points (Spot detection). Ces masques de base, en réalité, englobent tout le sous-espace 1×3 et forment un ensemble complet de neuf masques (voir Tableau 2.1). De manière similaire, les masques 1×5 constituent un ensemble exhaustif de 25 masques 5×5 qui sont obtenus en appliquant une convolution sur ces masques 1×5 . Dans ce cas, les initiales des masques sont identiques à celles mentionnées précédemment, avec l'ajout de la détection d'ondulation (Ripple detection) et la détection d'onde (Wave detection)

$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -1 \\ -2 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -1 \\ -2 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix}$	$\begin{bmatrix} -1 \\ -2 \\ -1 \end{bmatrix}$
$E3^T L3$			$E3^T E3$			$E3^T S3$		
$\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -2 \\ 0 \\ 2 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$	$\begin{bmatrix} -2 \\ 0 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 17 \\ 0 \\ -1 \end{bmatrix}$
$S3^T L3$			$S3^T E3$			$S3^T S3$		
$\begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}$	$\begin{bmatrix} -2 \\ 4 \\ -2 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -2 \\ 4 \\ -2 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$

Tableau 2.1: Ensemble de neuf masques Laws de dimensions 3×3 .

Sur les images filtrées $f(x, y)$, obtenue après convolution de l'image $I(x, y)$ avec les masques, est estimée une énergie locale pour chaque pixel sur une fenêtre de voisinage de largeur $2w+1$. Les caractéristiques spectrales détectées sont des haute fréquence et les orientations relevées sont les directions horizontales, verticales et obliques. Les propriétés statistiques locales (moyenne et variance) de l'énergie sont estimées pour caractériser les différentes textures.

Moyenne

$$V_{\text{moy}}(x, y) = \frac{1}{(2w+1)^2} \sum_{i=x-w}^{x+w} \sum_{j=y-w}^{y+w} I_f(i, j)$$

Variance

$$\text{Var}(x, y) = \frac{1}{(2w + 1)^2} \sum_{i=x-w}^{x+w} \sum_{j=y-w}^{y+w} |I_f(i, j) - V_{\text{moy}}(x, y)|$$

2.2.2.4 Les filtres de Gabor

La transformée de Fourier d'une image est un outil précieux pour révéler les motifs réguliers au sein de la texture en explorant son domaine fréquentiel. Cependant, elle présente un inconvénient majeur, à savoir qu'elle traite l'ensemble de l'image sans prendre en compte la localisation spatiale. Une solution à ce problème consiste à recourir à une transformation alternative appelée la transformée de Fourier à fenêtre glissante. Le principe de cette approche est d'appliquer la transformée de Fourier sur une fenêtre d'observation qui se déplace le long de l'image. Le choix de la taille de la fenêtre et de l'intervalle de déplacement dépend des caractéristiques spatiales spécifiques des textures à analyser.

Il existe différentes fenêtres d'observation disponibles, notamment la fenêtre triangulaire, la fenêtre de Hamming, la fenêtre de Hanning et la fenêtre de Gauss. Lorsque la fenêtre de Gauss est utilisée dans ce contexte, on parle de transformation de Gabor. Les filtres de Gabor sont des filtres orientés passe-bande qui décrivent une propriétés de localisation fréquentielle et spatiale. Leur efficacité dans le domaine de l'analyse de la texture a été démontrée dans plusieurs travaux [68]. Un filtre de Gabor h , à deux dimensions est représenté comme une fonction gaussienne modulée par une onde plane sinusoïdale :

$$h(x, y) = \exp \left(-\frac{1}{2} \left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right] \right) \cos (2\pi (u_0 x + v_0 y) + \Phi) \quad (2.6)$$

où u_0, v_0 et Φ sont respectivement les fréquences et la phase de l'onde plane sinusoïdale. σ_x^2 et σ_y^2 caractérisent l'étendue spatiale du filtre.

Un filtre de Gabor d'orientation arbitraire θ peut être ensuite obtenu en faisant subir une rotation au système d'axes (x, y) .

$$\begin{aligned} x' &= x \cos \theta + y \sin \theta \\ y' &= -x \sin \theta + y \cos \theta \end{aligned}$$

En ajustant les différents paramètres $(\theta, \sigma_x, \sigma_y, \mu_0$ et $v_0)$, on obtient plusieurs filtres de réponse impulsionnelle h_j . Pour une image I donnée, le résultat I_j est déterminé à la sortie de chacun de ces filtres. Ensuite, plusieurs caractéristiques peuvent être extraites de ces images, telles que la moyenne absolue de la déviation est :

$$m_j = 1/n^2 \sum_{x,y}^{n-1} |I_j(x,y)| \quad (2.7)$$

2.2.2.5 La transformé en ondelette

La représentation de Gabor repose sur l'utilisation d'une fenêtre d'observation de longueur fixe, ce qui peut être limitatif car certaines textures présentent des caractéristiques à différentes échelles. Pour surmonter cette limitation, la transformée en ondelettes adopte une approche multi-échelles pour analyser la texture, en utilisant des fenêtres d'analyse de différentes tailles [69]. À partir des transformées en ondelettes d'une image, il est possible d'extraire de nombreux attributs statistiques ou de cooccurrences, tels que les "Wavelet Statistical Features (WSF)" et les "Wavelet Cooccurrence Features (WCF)". De plus, les énergies calculées pour les différents filtres sont utilisées pour caractériser les textures en vue de la classification.

L'avantage majeur de cette transformée réside dans sa capacité à caractériser la texture à plusieurs échelles, en intégrant à la fois les informations globales et locales présentes dans l'image.

Les ondelettes sont des fonctions réelles ou complexes oscillantes. Elles sont construites par une diladation et une translation d'une ondelette de base $\Psi(t)$ appelé « ondelette mère », en outre ces fonctions sont de moyenne nulle :

$$\int_{-\infty}^{+\infty} \Psi(t) dt = 0 \quad (2.8)$$

Et la norme unitaire s'écrit comme suit :

$$\int_{-\infty}^{+\infty} |\Psi(t)|^2 dt = 1 \quad (2.9)$$

La transformée en ondelettes d'un signal $f(t)$ correspond à une fonction à la fois du temps et de l'échelle, résultant de la projection du signal sur une famille de fonctions ondelettes :

$$Wf(u, s) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} f(t) \Psi^* \left(\frac{t-u}{s} \right) dt \quad (2.10)$$

Dans cette formulation, le paramètre d'échelle ou de dilatation est représenté par S , le paramètre de translation par u et $\sqrt{s}$ est désigné comme le facteur de normalisation. On utilise la notation $\Psi_s(u) = \frac{1}{\sqrt{s}} \Psi \left(\frac{t-u}{s} \right)$ pour décrire la version translatée et dilatée de l'ondelette mère $\Psi_s(u)$. Le symbole « $*$ » représente le complexe conjugué. L'extension de

la transformation en ondelettes au cas 2D (image) est réalisée avec succès. Son utilisation en analyse de la texture implique une décomposition itérative de l'image en quatre sous-bandes de résolution inférieure. Ces sous-bandes sont obtenues par convolution avec des filtres passe-bas et passe-haut. Chaque pixel est ensuite associé à un vecteur d'attributs de texture correspondant aux coefficients d'ondelettes obtenus dans chaque sous-bande [70]. Toutefois, ces coefficients d'ondelettes ne sont pas directement discriminants car leur réponse n'est pas uniforme pour une fréquence donnée. Pour extraire l'information texturale, une estimation locale est réalisée dans une fenêtre de voisinage.

2.2.3 Approche statistique

Dans ce contexte, l'analyse de la texture est élaborée par les approches statistiques. Cela est dû à la nature statistique de la texture. En effet à l'appréhension que la texture a une structure aléatoire ne comportent pas de contours forts et ne possède pas un motif de base unique. Cette approche prend en compte le domaine spatial et englobe un large éventail de méthodes, dont certaines seront présentées en détail dans les sections suivantes de ce chapitre.

2.2.3.1 Méthode statistique du 1er Ordre

Dans cette approche la texture est caractérisée par les estimateurs du premier ordre au niveau d'un pixel isolé et en terme de niveaux de gris de leur distribution. Contrairement aux estimateurs d'ordre supérieur, qui reposent sur la corrélation des niveaux de gris de deux pixels ou plus en fonction de leur position dans l'image, les estimateurs du premier ordre utilisent principalement les paramètres calculés à partir de l'histogramme de l'image. Considérons $g(x,y)$ comme le niveau de gris du pixel de coordonnées (x,y) . L'histogramme normalisée de l'image $p(i)$ associé à $g(x,y)$ est défini par :

$$p(i) = p(g(x, y) = i) = \frac{h(i)}{N}, \quad 0 \leq i \leq G - 1 \quad (2.11)$$

La formule ci-dessous exprime la relation où N représente le nombre de pixels dans l'image, $h(i)$ est la fréquence d'apparition du niveau de gris i , et G correspond au nombre total de niveaux de gris.

Les attributs de texture calculés à partir de l'histogramme de l'image sont consignés dans le tableau 2.2.

Tableau 2.2: Caractéristiques statistiques de premier ordre

caractéristiques	Formule
Moyenne	$S_M = \frac{1}{N} \sum_{i=0}^{G-1} ih(i)$
Variance	$\sigma^2 = \frac{1}{N} \sum_{i=0}^{G-1} (i - S_M)^2 h(i)$
Energie	$ASM = \frac{1}{N} \sum_{i=0}^{G-1} h(i)^2$
Entropie	$ENT = -\frac{1}{N} \sum_{i=0}^{G-1} h(i) \log[h(i)]$
Skewness	$SKW = \frac{1}{N\sigma^3} \sum_{i=0}^{G-1} (i - S_M)^3 h(i)$
Kurtosis	$Kur = \frac{1}{N\sigma^4} \sum_{i=0}^{G-1} (i - S_M)^4 h(i)$

2.2.3.2 Méthode des matrices de cooccurrence

Haralik [71] a développé la matrice de cooccurrence GLCM (grey level co-occurrence matrix). C'est une méthode statistique du deuxième ordre car elle étudie la relation de niveau de gris entre deux pixels. Cette approche est perçue comme la référence dans le domaine d'analyse de la texture.

la matrice de coocurrence est introduite selon les étapes suivantes : Considérons la fonction image $I(x,y)$ définie sur un domaine D , telle que $I(x,y)=0, 1, 2, \dots, G-1$, où G représente le nombre de niveaux de gris dans l'image.

De plus, supposons qu'il existe une relation spatiale, notée R , qui lie deux pixels dans ce domaine. Dans ce contexte, la matrice de cooccurrence du domaine D , relative à la relation R , est définie par le terme général suivant :

$$t(i, j, d, \theta) = \text{card} \left\{ \begin{array}{l} ((x, y), (x', y')) \in D, \quad \text{vérifiant } R \\ I(x, y) = i, I(x', y') = j \end{array} \right.$$

avec $i = 0, \dots, G - 1$ et $j = 0, \dots, G - 1$. dans une déplacement d et une orientation θ . Chaque élément $t(i,j)$ de la matrice représente le nombre de paires de points du domaine D qui sont liées par la relation R , où l'un a le niveau de gris i et l'autre a le niveau de gris j .

Une autre approche consiste à définir une matrice de cooccurrence normalisée en divisant chaque élément $t(i,j)$ par le nombre total de paires de points du domaine D qui vérifient

la relation R .

En résumé, la matrice de cooccurrence capture les fréquences d'apparition des différentes combinaisons de niveaux de gris entre deux pixels liés par la relation spatiale R , et la version normalisée permet d'obtenir des probabilités d'occurrence relatives pour ces combinaisons. Cette approche est largement utilisée dans le traitement d'images pour extraire des informations utiles sur la texture et les relations entre les différents niveaux de gris présents dans l'image.

$$T(i, j, d, \theta) = \frac{t(i, j, d, \theta)}{N}$$

tel que

$$N = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} T(i, j, d, \theta)$$

$T(i, j, d, \theta)$ est la fréquence relative. elle présente la probabilité de chercher un couple de point du domaine D , le premier a un niveau de gris i et le deuxième de niveau j , vérifiant R . Haralik a défini cette relation spatiale entre les deux points par leur distance d et leur orientation θ par rapport à l'horizontale.

Tableau 2.3: Matrice de cooccurrence $P(1,0)$ de l'image Im ayant 4 niveaux de gris, pour une distance $d = 1$ et la direction $\theta = 0^\circ$.

$Im =$	1	4	2	3
	3	1	4	2
	3	2	1	3
	1	3	4	1

 $P(1, 0^\circ) =$

N_i, N_j	1	2	3	4
1	0	1	3	3
2	1	0	2	2
3	3	2	0	1
4	3	2	1	0

La matrice de cooccurrence contient une grande quantité d'informations, mais son exploitation intégrale peut être difficile. Une approche consiste à réduire le nombre de niveaux de gris, permettant ainsi l'utilisation des éléments de la matrice de cooccurrence comme attributs de texture [72]. Une autre méthode courante consiste à extraire plusieurs attributs afin de résumer l'ensemble des informations apportées par la matrice de cooccurrence [73]. Parmi ceux-ci, les six attributs du tableau 2.4 sont les plus fréquemment utilisés

2.2.3.3 Les motifs binaires locaux (LBP)

Les motifs binaires locaux, introduits par Ojala [74] dans le but d'analyser les textures au sein des images en niveaux de gris, consistent à évaluer la structure locale autour de chaque pixel P de l'image $I(i,j)$ à étudier. Ces évaluations sont obtenues en comparant le

Tableau 2.4: Caractéristiques déduites de la matrice de cooccurrence

Contraste	$CON = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i-j)^2 T^2(i, j, d, \theta)$
Variance	$VAR = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i-j)^2 T(i, j, d, \theta)$
Second moment angulaire	$SMA = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} T^2(i, j, d, \theta)$
Entropie	$ENT = - \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} T(i, j, d, \theta) \log T(i, j, d, \theta)$
Corrélation	$\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{(i - \mu_x)(j - \mu_y) T(i, j, d, \theta)}{\sigma_x \sigma_y}$
Homogénéité locale ou moment angulaire inverse	$IDM = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{1}{1 + (i-j)^2} T(i, j, d, \theta)$

niveau de gris du pixel central P avec ceux de ses pixels voisins.

Le principe des motifs binaires locaux (LBP) est une notion simple, qui consiste à attribuer un code binaire à chaque pixel selon son voisinage. Ce code, qui décrit la texture locale d'une region, est obtenu en comparant les niveaux de gris des pixels voisins avec celui du pixel central. Pour créer un motif binaire, chaque voisin se voit attribuer la valeur "1" s'il est supérieur ou égal au niveau de gris du pixel central, sinon "0". Ensuite, ces pixels du motif binaire sont pondérés et additionnés pour produire le code LBP du pixel central. Dans l'ensemble de l'image, les pixels résultants auront des intensités comprises entre 0 et 255, comme dans une image classique en 8 bits. Au lieu de représenter l'image par la séquence des motifs LBP, il est possible d'utiliser un histogramme à 255 dimensions comme descripteur de texture.

6	5	2
7	6	1
9	3	7
(a)		
1	0	0
1		0
1	0	1
(b)		
1	2	4
8		16
32	64	128
(c)		
1	0	0
8		0
32	0	128
(d)		

Figure 2.2: Calcul du nombre LBP et du contraste C.

Dans l'exemple de la figure 2.2, nous avons : $LBP = 1 + 8 + 32 + 128 = 169$ et $C = (6 + 7 + 9 + 7) * 4 - (5 + 2 + 1 + 3) * 4 = 4.5$.

Pour obtenir un code LBP dans un voisinage de P pixels avec un rayon R, il nous suffit de compter combien de pixels présentent une valeur de niveau de gris (gp) supérieure ou

égale à celle du pixel central.

$$\text{LBP}(m, r) = \sum_{i=0}^{p-1} u(g_i - g_c) 2^i \quad (2.12)$$

où g_c représente le niveau de gris du pixel central, g_i allant de 0 à $p-1$ est le niveau de gris du pixel voisin, et $u(x)$ désigne la fonction définie de la manière suivante :

$$u(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases} \quad (2.13)$$

Ojala [74] a introduit deux dérivations de la méthode LBP . La première est adaptée à des voisinages de tailles variables, ce qui permet de prendre en compte la texture à différentes échelles. La seconde, quant à elle, définit ce que l'on nomme les "LBP uniformes".

2.2.4 Approches basées sur un modèle

Les approches basées sur un modèle consistent à créer un modèle pour la génération de la texture. Ce modèle est défini par des paramètres qui permettent soit de caractériser une texture existante, soit de créer une nouvelle texture de manière synthétique. En termes de classification, on peut distinguer principalement trois types de modèles : les modèles linéaires, plus précisément les modèles autorégressifs, les modèles fractals et les modèles Markoviens.

2.2.4.1 Modélisation linéaire de la texture

Cette approche de modélisation implique que la valeur du niveau de gris de chaque pixel est déterminée par une combinaison linéaire des valeurs des pixels environnants [75].

Soit l'image $I = \{I(i, j) / 0 \leq i, j \leq M - 1\}$ modélisée par :

$$I(i, j) = \sum_{(k, l) \in R} \alpha(k, l) I(i - k, j - l) + \sum_{(k', l') \in R} \beta(k', l') e(i - k', j - l') \quad (2.14)$$

$e(i, j)$ représente un champ aléatoire, R est considéré comme une région de prédiction. M représente les tailles horizontale et verticale de l'image.

Au sein des différentes modélisations linéaires de la texture, le modèle autorégressif

Modèle autorégressif

Le modèle autorégressif (AR) repose sur l'idée qu'il existe des interactions locales entre les pixels d'une image, et cette interaction est mathématiquement exprimée en décrivant l'intensité de chaque pixel donné comme une combinaison pondérée des intensités des

pixels voisins précédents.

$$I_s = \sum_{r \in N_s} \theta_r I_r + e_s \quad (2.15)$$

Où I_s : Intensité du site s ,

I_r : Intensités des sites r voisins de s ,

N_s : Voisinage du site s ,

θ_r : Paramètres du modèle,

e_s : Bruit indépendant, uniformément distribué.

Le but est donc de déterminer, pour chaque texture, les paramètres θ_r qui la modélisent. Ces paramètres peuvent être alors estimés par différentes techniques dont les moindres carrés ou la méthode des autocorrélations .

La segmentation texturale basée sur les modèles autorégressifs (AR) implique l'identification des paramètres du modèle spécifiquement pour une région donnée de l'image. Ces paramètres sont ensuite utilisés pour discerner les caractéristiques de texture. Ces modèles ont été étendus à diverses autres applications, notamment la création de textures, la compression, la restauration d'images, la classification et la segmentation.

2.2.4.2 Champs de Markov

Les champs de markov est le concept le plus utilisé dans le domaine du traitement d'images, et précisément l'analyse de la texture. Ils permettent de modéliser les propriétés globales en se basant sur des contraintes locales. Parmi les modèles les plus utilisés figurent le champ de Markov Gaussien ou GMRF (Gaussian Markov Random Field) [76] ainsi que le modèle binomial [50]. Les paramètres de ces modèles sont considérés comme des attributs de texture. Dans le cas du GMRF classique, le niveau de gris d'un pixel est déterminé par la somme pondérée, en utilisant les paramètres du modèle, des niveaux de gris de ses voisins, auquel est ajouté un bruit Gaussien de moyenne nulle. La variance de ce bruit constitue également l'un des paramètres du modèle. Concernant le deuxième modèle, la probabilité de chaque pixel de la texture, conditionnellement à ses voisins, suit une distribution binomiale dépendant du voisinage et du nombre de niveaux de gris de l'image.

Considérons un ensemble de pixels S et une famille de variables aléatoires $X = (x_s)_{s \in S}$ définie sur cet ensemble, où chaque x_s prend des valeurs dans un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_k\}$. Cette famille sera désignée par la suite comme un "champ aléatoire". De plus, supposons l'existence d'un système de voisinage noté V . Les différentes lois de probabilité associées à X seront notées par la lettre p . Un champ X est considéré comme

un champ de Markov par rapport à V si et seulement si il satisfait les deux propriétés suivantes :

- Positivité

$$\forall x \in \Omega d(s), Px > 0 \text{ (contrainte de positivité)}$$

- Markovianité :

$$p(x_s/x_t, t \neq s) = p(x_s/x_t, t \in V_s)$$

L'utilisation des champs de Markov en traitement d'images repose principalement sur le théorème de Hammersley-Clifford et l'application de la mesure de Gibbs [77].

2.2.4.3 Champs de Gibbs :

Avant d'introduire le concept de champ de Gibbs (ou "Gibbs Random Fields" en anglais), nous devons d'abord expliquer ce qu'est une mesure de Gibbs. La mesure de Gibbs associée à une fonction d'énergie $U : \Omega \rightarrow \mathbb{R}$ est une probabilité P définie sur l'ensemble Ω de la manière suivante :

$$P[X = x] = Z^{-1} e^{-U(x)/T}$$

Où : $U(x) = \sum_{c \in C} U_c(x)$

Dans cette expression, C représente le système de cliques associé au système de voisinage $V(s)$, T est un paramètre de contrôle appelé la température, et Z est une constante de normalisation connue sous le nom de fonction de partition de Gibbs, définie comme suit :

$$Z = \sum_{Y \in \Omega} e^{-U(y)/T}$$

Il est important de noter que le calcul de Z est pratiquement impossible en raison du très grand nombre de configurations possibles. Par exemple, dans le cas d'une image de 512×512 pixels avec 4 classes possibles, l'ensemble Ω contient $4^{262,144}$ configurations possibles !

Maintenant, pour définir le champ de Gibbs, nous pouvons dire que :

"Le champ de Gibbs associé au potentiel du système de voisinage $V(s)$ est un processus aléatoire X dont la probabilité est déterminée par une mesure de Gibbs associée à $V(s)$." En résumé, un champ de Gibbs est caractérisé par un ensemble de potentiels $\{U_c\}$ indexés par les cliques c .

2.3 Classification de la texture

La classification des textures est un domaine de recherche important en vision par ordinateur. Un grand effort a été déployé dans la recherche d'une description efficace des textures. Les approches de recherche récentes peuvent être largement catégorisées en deux branches principales. La première branche consiste à apprendre des caractéristiques extraites d'un réseau neuronal convolutionnel (CNN). Elles ont été largement acceptées en raison de leur puissance de modélisation et de leur expressivité. Cependant, elles présentent également certains inconvénients. Par exemple, le coût computationnel élevé des CNN nécessite un matériel puissant tel qu'une unité de traitement graphique (GPU) pour entraîner le modèle, et elles nécessitent beaucoup de données d'entraînement pour construire un tel modèle puissant. En revanche, la deuxième branche de description des textures, les caractéristiques conçues à la main, ne sont pas toujours inférieures à celles des CNN. Elles nécessitent moins de données d'entraînement et ont un coût computationnel plus faible par rapport aux équivalentes d'apprentissage.

Cependant, la discrimination des textures est généralement un problème difficile en raison de la diversité et de la complexité des textures naturelles. Et il n'existe toujours pas de méthode unique capable d'extraire des caractéristiques discriminantes pour tous les types de données de texture. Les tendances de recherche récentes suivent les deux branches. Il existe des travaux qui combinent les deux types de caractéristiques dans un système de représentation d'image.

La classification joue un rôle essentiel dans la partition d'un ensemble d'observations, comme des images, en différentes classes. Elle consiste à regrouper dans une même classe les observations (images) présentant des caractéristiques similaires. La classification d'images est une étape cruciale dans des applications de reconnaissance des formes telles que la reconnaissance des visages, des empreintes, etc., ainsi que dans l'indexation d'images pour rechercher une image dans une base de données.

Dans ces applications, chaque image peut être décrite par un ensemble d'attributs, généralement liés aux caractéristiques de texture. Les méthodes de classification se distinguent en deux catégories : supervisées et non supervisées.

Les méthodes de classification supervisées utilisent un ensemble d'images préalablement étiquetées pour entraîner un modèle. Ce modèle est ensuite utilisé pour classer de nouvelles images en leur attribuant une classe prédéfinie.

D'un autre côté, les méthodes de classification non supervisées ne nécessitent pas d'étiquettes préalables. Elles identifient automatiquement des structures ou des similitudes entre les images et les regroupent en classes, sans qu'on leur ait fourni d'informations sur les classes attendues. En somme, la classification d'images est un processus fondamental

pour organiser et identifier des motifs dans les données visuelles, facilitant ainsi la reconnaissance et l'indexation dans diverses applications.

Dans un contexte supervisé, l'ensemble d'observations est composé d'échantillons, où chaque observation est décrite par un ensemble d'attributs. On connaît à l'avance l'appartenance de chaque observation à une classe spécifique, et ces observations sont appelées prototypes de classe.

L'objectif de la classification est de définir des modèles mathématiques pour chaque classe à partir de ces prototypes. Ces modèles peuvent prendre la forme d'un simple centre de gravité, d'une fonction de densité de probabilité ou encore d'une surface de séparation entre les classes.

Les étapes de caractérisation et de classification des observations prototypes des classes sont regroupées sous le terme de "phase d'apprentissage". Cette phase est distincte de la "phase de décision", où l'on attribue une nouvelle observation à l'une des classes définies lors de l'apprentissage.

Il existe de nombreuses méthodes de classification supervisée. Parmi ces méthodes, on retrouve la classification par la règle des K plus proches voisins, le support vecteur machine et la classification par les CNN.

2.4 Apprentissage automatique

L'apprentissage automatique (ML) fait partie du domaine de l'intelligence artificielle qui a connu un énorme progrès au cours des dernières décennies et s'est ouvert à de nombreux domaines différents, y compris l'analyse d'images et le traitement d'images. Les algorithmes d'apprentissage automatique sont organisés en plusieurs catégories, telles que l'apprentissage supervisé et l'apprentissage non supervisé.

2.4.1 Apprentissage supervisé

L'apprentissage supervisé, une approche bien connue de l'apprentissage automatique, repose sur l'utilisation d'exemples pré-classifiés. Chaque entrée de ces exemples est associée à une classe déjà définie. L'objectif majeur est de généraliser cette connaissance à l'ensemble des données, permettant ainsi au système de créer un modèle performant pour toutes les entrées possibles. Les données d'apprentissage consistent en des objets accompagnés de leurs catégories respectives, formant ainsi l'ensemble d'apprentissage. Le système ajuste ensuite ses paramètres internes pour minimiser l'erreur, puis évalue la qualité du modèle en utilisant un second ensemble d'exemples pré-classifiés, appelé ensemble de test. Cette évaluation permet de mesurer le taux de classification correcte et celui des erreurs de

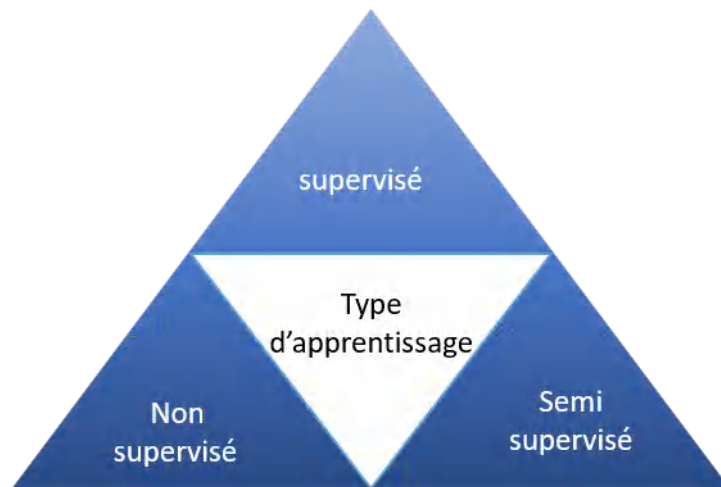


Figure 2.3: les types de machine learning

classification.

Dans le cadre de l'apprentissage supervisé, l'objectif peut être la prédiction de catégories, ce qui conduit à la notion de classificateur. Cela se traduit par la résolution de problèmes de classification, où chaque entrée se voit attribuer une étiquette lorsque les valeurs de sortie sont des nombres entiers. En revanche, il est également possible d'apprendre des fonctions capables de prédire des valeurs numériques lorsque la sortie recherchée est continue, ce qui correspond à la fonction de régression. Ainsi, divers algorithmes d'apprentissage supervisé ont été développés pour aborder à la fois la classification et la régression.

2.4.1.1 Classification

Un classificateur est un mécanisme à n entrées représentant les n entités $e_1, e_2, \dots$, en calculé à partir d'un objet à classer, et une (1) sortie. Un classificateur de classe S générera l'un des symboles $w_1, w_2, \dots, w_S$ comme sortie, et l'utilisateur interprète cette sortie comme une décision concernant la classe de l'objet traité. Les symboles générés w_s sont les identificateurs de classes. Les algorithmes de classification tels que SVM, K-NN, Naive bayes sont des algorithmes pour classifier des données. Ces données peuvent être des mots, des couleurs, des sons, etc. Dans un contexte de classification d'images, ces algorithmes peuvent être utilisés comme classificateurs basés sur des informations extraites des pixels de ces images.

2.4.1.2 les séparateurs à vaste marge (SVM)

Les Machines à Vecteurs de Support (SVM) représentent une catégorie d'algorithmes de classification, qui ont été introduits par Boser, Guyon et Vapni [78], puis perfectionnés

par [79]. Le SVM est un outil puissant pour résoudre des problèmes de classification, c'est-à-dire pour déterminer à quelle catégorie appartient un échantillon donné. Cette approche repose sur la création d'un hyperplan dans un espace de dimension élevée, dans le but de séparer efficacement les données appartenant à différentes classes. L'hyperplan en question est spécialement conçu pour maximiser la séparation entre les points de données les plus proches appartenant à différentes classes, ces points étant dénommés la "marge". Il est important de noter que la performance de généralisation du classificateur SVM dépend étroitement de la taille de cette marge, comme l'ont souligné Boser [78]. Plus précisément, la marge correspond à la distance entre la frontière de séparation et les points de données les plus proches. L'algorithme SVM construit un modèle en se basant sur un concept appelé la "marge fonctionnelle", ce qui en fait un classificateur binaire linéaire. Ce modèle ainsi obtenu est polyvalent, pouvant être utilisé pour effectuer des classifications, qu'elles soient linéaires ou non linéaires. Pour réaliser des classifications non linéaires, le SVM peut être facilement adapté en utilisant une fonction de noyau, qui permet de projeter les données d'entrée dans un espace de caractéristiques de haute dimension, comme expliqué par [80]. Dans cet algorithme, chaque point de données d'apprentissage appartient à l'une des deux classes, et l'objectif est de déterminer à quelle classe appartient un nouveau point de données.

2.4.1.3 Algorithme du plus proche voisin (KNN)

KNN est un type de méthode d'apprentissage supervisé qui est connu pour être très efficace en termes de capacité de classification sur de petits ensembles de données. Dans un classifieur KNN, toutes les caractéristiques des données d'entraînement et les étiquettes de classe correspondantes sont nécessaires pour classer de nouvelles données d'entrée. KNN détermine la similarité des caractéristiques en calculant la distance (comme la distance euclidienne) entre les données de test et les caractéristiques des données d'entraînement. Ensuite, l'appartenance à une classe est prédite par un processus de vote entre les K voisins les plus proches. Un classifieur KNN est illustré dans la Figure 2.6, où le point jaune représente un nouveau point de données de test. Si nous supposons que $K=3$, le processus de vote inclut le nouveau point de données dans la classe B et si nous supposons que $K=7$, le nouveau point de données est inclus dans la classe A. Dans l'algorithme KNN, il y a principalement deux paramètres à régler ; la valeur de K, qui est le nombre de voisins comme expliqué ci-dessus, et le choix de la métrique de distance (comme les distances euclidienne, cosinus et de Manhattan).

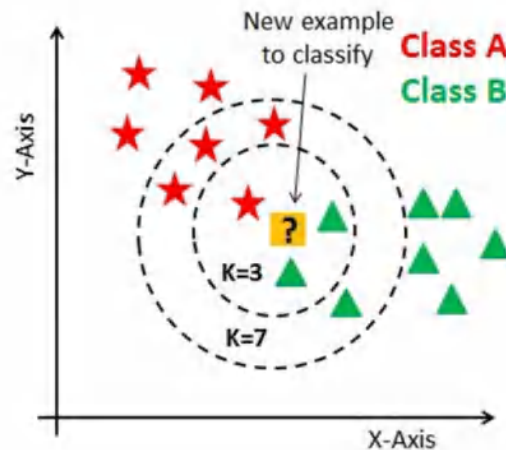


Figure 2.4: KNN

2.4.1.4 Forêts aléatoires des arbres décisionnels

Les "Forêts Aléatoires" ou "Random Forests" (RF) en anglais, sont parmi les algorithmes couramment utilisés en classification supervisée. Cet algorithme a été introduit par Breiman en 2001 [81]. Il demeure l'un des choix les plus efficaces, même lorsqu'il est appliqué à des volumes considérables de données. Il combine de manière complexe plusieurs mécanismes, parmi lesquels figure le Bagging. Les forêts aléatoires fusionnent des prédicteurs ou estimateurs issus d'arbres de décision, créant ainsi une approche désignée sous le nom d'"arbres de décision" à l'heure actuelle. En substance, ces algorithmes établissent une collection de prédicteurs pour ensuite agréger l'ensemble de leurs prédictions. En ce qui concerne la classification, cette agrégation implique généralement un vote majoritaire parmi les étiquettes de classe proposées par les différents prédicteurs. Pour la régression, la valeur de la cible prédite pour un échantillon d'entrée est calculée comme la moyenne des valeurs de régression prédites par les différents arbres de la forêt.

2.4.1.5 Exactitude et validation croisée

Pour évaluer l'efficacité d'un algorithme d'apprentissage automatique supervisé dans le domaine du traitement d'images, une mesure populaire est l'exactitude (Accuracy), qui est basée sur les résultats de prédiction obtenus en appliquant le modèle d'apprentissage automatique aux données de test. Par exemple, les résultats de prédiction pour la classification binaire sont formulés à l'aide de quatre mesures : vrais positifs (tp), faux positifs (fp), vrais négatifs (tn) et faux négatifs (fn). La mesure d'exactitude est ensuite donnée

par le ratio suivant :

$$Exactitude = (tp + tn) / (tp + fp + tn + fn) * 100 \quad (2.16)$$

La validation croisée (cross-validation) est une approche statistique largement utilisée pour déterminer à quel point un modèle d'apprentissage automatique développé peut généraliser la solution sur de nouvelles données. Pendant la validation croisée, l'ensemble complet des données d'entraînement est divisé de manière aléatoire en n plis (ou "folds") égaux. Le modèle d'apprentissage automatique est ensuite entraîné sur $n-1$ plis tandis que le pli restant est conservé à des fins de test. Cette opération est répétée de manière itérative n fois, chaque itération traitant un pli de test nouvellement sélectionné. Par conséquent, nous obtenons n valeurs différentes spécifiant l'exactitude du modèle d'apprentissage automatique. La moyenne des n mesures obtenues est alors considérée comme la performance globale du modèle d'apprentissage automatique. Cette approche permet une évaluation plus réaliste du modèle au prix d'un coût de calcul plus élevé.

Dans ce travail, nous nous concentrons sur l'apprentissage automatique supervisé appliqué aux problèmes de classification. En utilisant des algorithmes d'apprentissage automatique, on s'attend aux avantages suivants : les algorithmes d'apprentissage automatique peuvent traiter de grandes quantités de données et découvrir des tendances et des motifs spécifiques qui ne seraient pas évidents pour les humains.

Avec l'apprentissage automatique, il n'est pas nécessaire de suivre un projet à chaque étape, car cela donne à la machine la capacité d'apprentissage, lui permettant de faire des prédictions. Les algorithmes d'apprentissage automatique acquièrent de l'expérience. Ils s'améliorent en termes d'exactitude et d'efficacité à mesure qu'ils sont nourris de nouvelles données. Par conséquent, ils peuvent prendre de meilleures décisions. Les algorithmes d'apprentissage automatique sont efficaces pour traiter des données multidimensionnelles et multivariées. . Le processus de prédiction (décision) se déroule des observations (caractéristiques sur les branches) aux conclusions (étiquettes de classe sur les feuilles). Cette analyse de l'algorithme est facilement explicable et présente une complexité computationnelle faible, ce qui en fait une méthode simple et pratique en apprentissage automatique. Les classificateurs discriminants linéaires et les classificateurs discriminants quadratiques tentent de déterminer respectivement des frontières linéaires et quadratiques/surfaces entre les classes existantes. En général, l'analyse discriminante linéaire suppose que les données de chaque classe appartiennent à une distribution gaussienne spécifique. Au cours du processus d'entraînement, les paramètres (moyenne, variance) pour la distribution gaussienne associée à chaque classe sont estimés.

2.4.2 Apprentissage non supervisé

Dans l'apprentissage non supervisé, un ensemble d'entrées est présenté au système de classification pendant la phase d'apprentissage, à la différence de l'apprentissage supervisé. Cet ensemble ne comporte pas d'étiquettes de classe correspondantes. L'objectif principal de l'apprentissage non supervisé est de modéliser la structure des données en entrée, permettant ainsi d'obtenir davantage d'informations sur la distribution des données sous-jacentes.

L'apprentissage non supervisé trouve une application particulièrement pertinente dans les problèmes de regroupement, où l'objectif est de comprendre et d'identifier les relations entre des objets au sein d'une collection donnée. Une méthode courante dans ce type d'apprentissage consiste à définir une mesure de similitude entre deux objets, puis à regrouper les objets qui sont plus similaires entre eux par rapport à ceux d'autres groupes. Ce processus d'apprentissage non supervisé peut être utilisé pour diverses tâches, notamment le regroupement, l'estimation de la densité, la réduction de dimension, et bien d'autres encore.

2.4.3 Les réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (Convolutional Neural Network, CNN) ont émergé dès la fin des années 1990 [LBBH98]. Cependant, leur popularité n'a réellement explosé qu'à partir de 2012, grâce à l'extraordinaire succès de ces méthodes pour la classification à grande échelle [KSH12] sur le jeu de données ImageNet (ILSVRC 2012) [82].

Les réseaux de neurones convolutifs (Convolutional Neural Networks, CNN) étendent la notion de neurones en incluant les filtres de convolution, communément utilisés pour traiter les images. L'objectif recherché est double : (1) favoriser le partage des poids d'un réseau de neurones entre tous les pixels d'une image pour réduire le nombre de paramètres à optimiser, et (2) apprendre des filtres de convolution optimisés pour accomplir une tâche spécifique et ainsi extraire automatiquement des caractéristiques.

Tout comme les réseaux de neurones classiques (par exemple, les perceptrons multicouches, MLP), les réseaux de neurones convolutifs sont composés de couches, désignées sous le terme de couches de convolution. La plupart des architectures modernes sont profondes, formant ainsi des Réseaux de Neurones Convolutifs Profonds (Deep Convolutional Neural Networks, DCNN). Cependant, au lieu de se baser exclusivement sur des perceptrons, certaines de ces couches sont composées d'opérateurs issus du traitement du signal et de l'image, comme les filtres de convolution (où la convolution prend la place du neurone, en remplacement du perceptron) et le pooling (une opération de sous-échantillonnage non linéaire). Plus précisément, les filtres de convolution sont utilisés pour filtrer l'image en vue

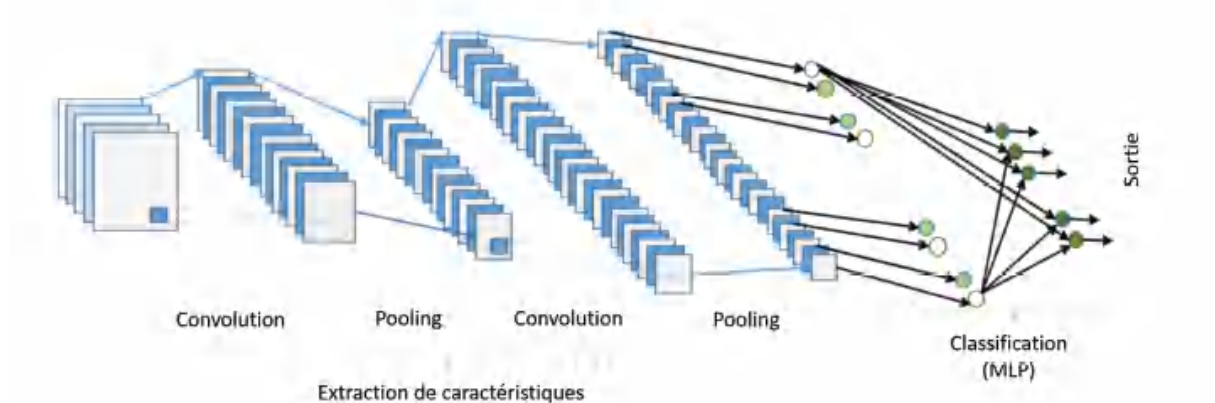


Figure 2.5: Schéma classique d'un CNN

d'extraire des caractéristiques, qui peuvent ensuite servir d'entrées pour des couches totalement connectées (Fully Connected, FC), équivalant ainsi à un perceptron multicouche (dans le contexte de la classification).

L'avantage majeur de cette approche réside dans sa capacité à optimiser simultanément l'algorithme de classification ainsi que les filtres d'extraction de caractéristiques (consultez la Figure 2.11 pour plus d'illustration).

2.5 Conclusion

En conclusion, ce chapitre a examiné les caractéristiques et les méthodes de classification de la texture dans le domaine de l'analyse d'images. Les caractéristiques statistiques et structurelles ont été présentées comme des outils essentiels pour capturer les propriétés visuelles et les relations spatiales des textures. En utilisant ces caractéristiques, les méthodes de classification telles que les classificateurs bayésiens, les machines à vecteurs de support (SVM) et les réseaux de neurones ont été discutées pour effectuer des tâches de classification précises.

Il est important de noter que le choix des caractéristiques et des méthodes de classification dépendra du contexte spécifique de l'application et des données disponibles. De plus, l'évolution rapide des techniques d'apprentissage automatique et des réseaux de neurones profonds ouvre de nouvelles perspectives pour la classification de la texture, avec des performances améliorées et des capacités d'adaptation accrues.

En somme, la compréhension approfondie des caractéristiques et des méthodes de classification de la texture joue un rôle fondamental dans l'amélioration de la reconnaissance et de l'analyse des textures. Les futures recherches dans ce domaine continueront de chercher des approches novatrices pour extraire des caractéristiques discriminantes et développer

des méthodes de classification robustes, contribuant ainsi à l'avancement des domaines tels que la vision par ordinateur, la robotique et la perception

Chapitre 3

Approches développées pour la synthèse de la texture

3.1 Introduction

Dans ce chapitre, nous présentons les méthodes de synthèse de texture développées dans cette thèse. La première méthode présente la synthèse de la texture par le deep learning (apprentissage profond) . Les méthodes existantes dans la littérature utilisent seulement le transfert learning dans le cadre de la classification. Pour cela nous avons proposé d'utiliser le concept de transfert learning dans le cadre de la synthèse pour trouver le meilleur réseau adapté pour ce problème. Différents réseaux neuronaux convolutionnels profonds pré-entraînés ont été pris en compte. Dans un deuxième volet, nous avons développé une nouvelle approche utilisant les statistiques spatiales pour la synthèse de la texture. Elle est basée sur un modèle autorégressif bidimensionnel 2D-RCA pour l'extraction des caractéristiques des textures basée sur les modèles. Ces modèles ont une bonne capacité à détecter les informations de voisinage. Les simulations ont démontré que les modèles 2D-RCA sont très appropriés pour représenter les textures. Ainsi, dans ce travail, pour générer des textures à partir d'un exemple, chaque image originale est divisée en blocs qui sont modélisés par le modèle 2D-RCA. L'algorithme proposé produit des approximations des blocs d'images obtenus à partir de l'image originale en utilisant la méthode généralisée des moments (GMM). Différentes tailles de fenêtres ont été utilisées. Cette étude offre des informations importantes sur l'image nouvellement générée. Les résultats satisfaisants obtenus ont été comparés à ceux fournis par des méthodes bien établies. L'algorithme proposé surpasse les approches de pointe.

3.2 Synthèse de texture par le transfert learning

La modélisation et la synthèse de texture sont d'excellentes approches d'étudier la représentation mathématique des images et de mettre en évidence certaines de leurs limites. Les représentations d'images utilisées pour la synthèse de texture vont des champs aléatoires de Markov aux décompositions en ondelettes ou à la modélisation markovienne non paramétrique. Depuis Gatys [4] les CNN (réseaux de neurones convolutionnels) sont le principal fournisseur de représentations d'images. Ce travail propose d'utiliser les statistiques du second ordre des caractéristiques extraites d'un CNN pré-entraîné au lieu d'une décomposition en caractéristiques classiques. Bien que ces méthodes produisent des résultats de pointe, une limitation réside dans la difficulté à capturer efficacement les structures à grande échelle pour la synthèse à haute résolution.

La synthèse de la texture à travers les réseaux de neurones convolutionnels se divise en deux parties essentielles, chacune jouant un rôle vital dans la création de textures réalistes et esthétiquement agréables.

La première étape consiste en la modélisation de la texture. Dans cette phase, le réseau de neurones convolutionnels est chargé d'extraire les caractéristiques essentielles de la texture désirée à partir d'un ensemble de données d'apprentissage. Ces caractéristiques peuvent inclure des motifs, des structures, des couleurs et d'autres aspects visuels de la texture. L'une des étapes cruciales de cette modélisation est la création de la matrice de Gram. La matrice de Gram capture les corrélations entre les différentes caractéristiques extraites, ce qui permet de représenter de manière compacte et significative la structure de la texture. La création de cette matrice est un processus essentiel, car elle agit comme la pierre angulaire pour la génération ultérieure de la texture.

La deuxième étape est la génération proprement dite de la texture. Dans cette phase, la matrice de Gram obtenue précédemment est utilisée comme point de départ. En appliquant des transformations et des opérations spécifiques sur cette matrice, le réseau de neurones est capable de produire une texture synthétique qui présente les caractéristiques visuelles et structurelles de la texture modélisée dans la première étape. Cette transformation de la matrice de Gram en une image texturée permet de recréer la texture de manière cohérente et réaliste.

Dans notre travail [83], nous avons introduit le concept d'apprentissage par transfert, une technique souvent utilisée principalement dans le domaine de la classification, mais que nous avons étendue à la synthèse.

Dans ce travail, nous avons appliqué cinq réseaux profonds pré-entraînés, à savoir, alexnet, GoogleNet, VGG16, VGG19 et ResNet50, pour la synthèse de textures en utilisant le transfert d'apprentissage.

3.2.1 Modélisation de la texture

La synthèse de la texture par les réseaux de neurones est basée sur les caractéristiques extraites des réseaux neuronaux convolutionnels [84]. Ces réseaux sont créés en combinant plusieurs processus fondamentaux, tels que les convolutions, les poolings et les fonctions d'activation non linéaires. Dans ce travail, la sortie de la couche précédente est soit convoluée, soit regroupée dans chaque couche du réseau. Le modèle de texturation discuté dans cette section est basé sur celui proposé dans [19]. Pour créer une texture à partir d'une image source particulière, nous extrayons d'abord de manière homogène des caractéristiques de différentes tailles à partir de celle-ci. Ensuite, pour obtenir une description stationnaire de l'image source, nous calculons une statistique sommaire spatiale des réponses des caractéristiques (Fig 3.1). Enfin, nous effectuons une descente de gradient sur une image aléatoire initialisée avec du bruit blanc pour trouver une nouvelle image ayant la même description stationnaire (Fig 3.1). La Figure 3.1 fournit une description schématique de l'algorithme de synthèse :

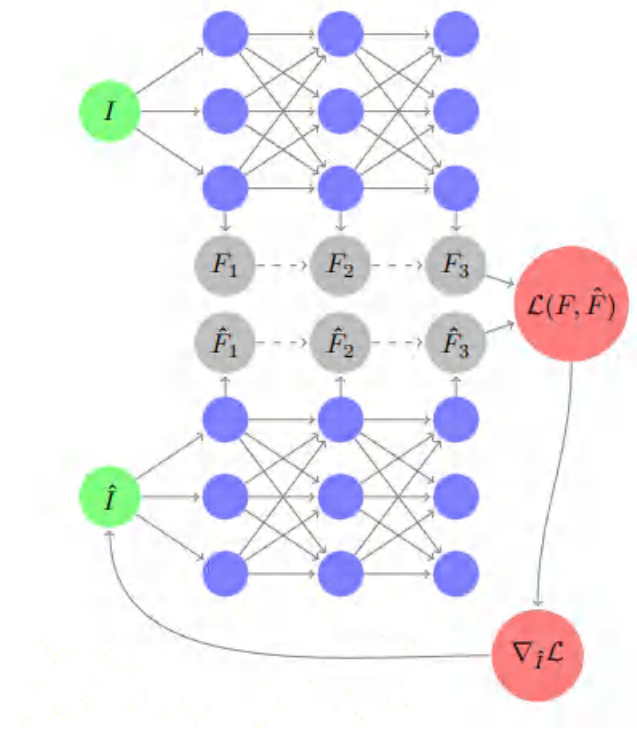


Figure 3.1: Schéma descriptif de la méthode de synthèse

La différence principale par rapport au travail de [19] est que nous utilisons l'espace des caractéristiques fourni par un réseau neuronal profond performant et des statistiques sommaires spatiales. Il s'agit des corrélations entre les réponses des caractéristiques dans

chaque couche du réseau, plutôt qu'un banc de filtres linéaires et un ensemble de statistiques sommaires soigneusement choisies. Nous le faisons passer d'abord par un réseau neuronal convolutionnel pour caractériser une texture vectorisée donnée dans la méthode proposée. De plus, nous calculons les activations pour chaque couche. Ces activations, qui consistent en la réponse du réseau à une image, génèrent un ensemble d'images filtrées. Chaque couche peut être considérée comme une banque de filtres non linéaires (appelés cartes de caractéristiques).

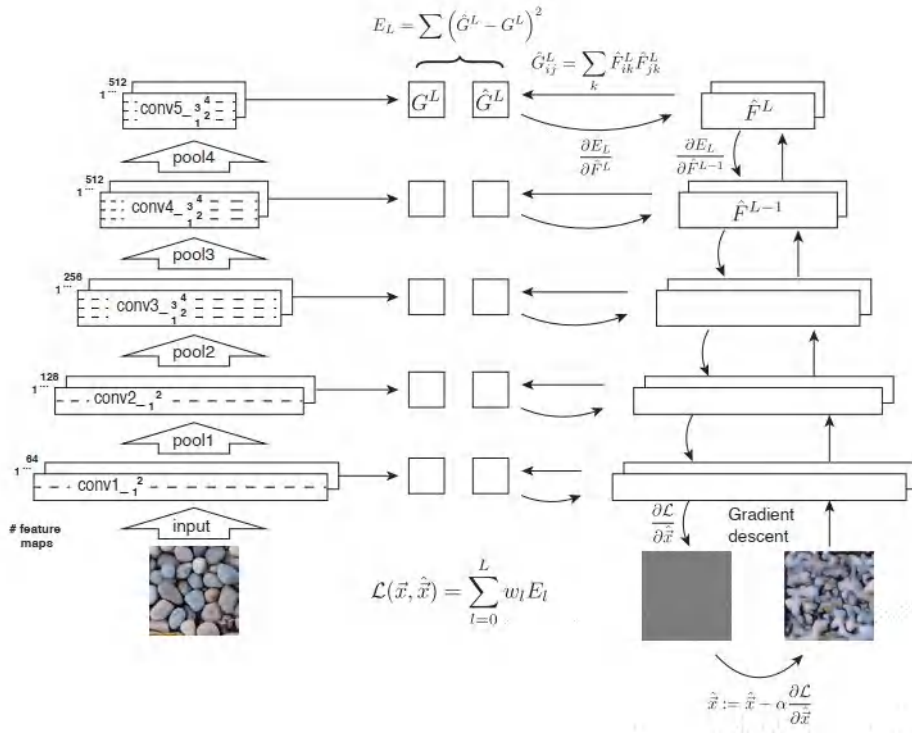


Figure 3.2: la méthode de synthèse [4]

La synthèse de texture (à droite) fig(3.2) gauche. Les réponses des caractéristiques de plusieurs couches sont calculées. Une image de bruit blanc x est passée à travers le CNN et une fonction de perte E_l est calculée pour chaque couche incluse dans le modèle de texture. La fonction de perte totale L est une somme pondérée des contributions E_l de chaque couche. En utilisant la descente de gradient sur la perte totale par rapport aux valeurs des pixels, une nouvelle image est trouvée qui produit les mêmes matrices de Gram G_l que la texture originale. Les textures sont stationnaires, donc un modèle de texture doit être agnostique des informations spatiales. Une statistique sommaire qui ignore les informations spatiales dans les cartes de caractéristiques est donnée par les corrélations entre les réponses des différentes caractéristiques. Ces corrélations entre les caractéristiques sont, à une constante de proportionnalité près, données par la matrice de

Gram.

Les matrices de Gram ($G_1, \dots, G_L$) fournissent une collection de statistiques que les auteurs affirment être une excellente représentation.

3.2.1.1 Les matrices de Gram

La matrice de Gram est un outil important dans la synthèse de texture, en particulier dans le contexte de la synthèse basée sur des réseaux de neurones convolutionnels (CNN). Le principe de la matrice de Gram dans la synthèse de texture repose sur l'utilisation de cette matrice pour capturer les caractéristiques statistiques de textures existantes, ce qui permet de générer de nouvelles textures de manière réaliste.

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l \quad (3.1)$$

$$\mathcal{L}(G, \hat{G}) = \sum_{l=1}^L w_l E_l \quad (3.2)$$

Le principe de fonctionnement de la matrice de Gram dans la synthèse de texture est le suivant :

1.Extraction de caractéristiques : Tout d'abord, un modèle CNN est utilisé pour extraire des caractéristiques à partir d'une image de texture existante. Ces caractéristiques sont généralement des activations de certaines couches du réseau, qui capturent des informations sur les motifs, les textures, les couleurs, etc., présents dans l'image.

2.Calcul de la matrice de Gram : Une fois que les caractéristiques ont été extraites, on calcule la matrice de Gram à partir de ces caractéristiques. Pour ce faire, on prend les activations de différentes couches du réseau (par exemple, les activations de plusieurs filtres dans une couche convolutive) et on calcule le produit scalaire entre elles pour chaque paire de caractéristiques. Cela crée une matrice symétrique où chaque élément représente la similarité entre deux caractéristiques.

3.Utilisation de la matrice de Gram pour la synthèse : La matrice de Gram ainsi calculée capture les relations statistiques entre les caractéristiques de l'image de texture d'origine. Cette matrice est ensuite utilisée comme référence pour générer de nouvelles textures. En ajustant les caractéristiques extraites d'une manière qui correspond à la matrice de Gram de l'image de texture d'origine, vous pouvez générer une nouvelle texture qui partage certaines des caractéristiques statistiques de l'original.

Le principe sous-jacent est que la matrice de Gram capture les informations sur les motifs et les textures, mais pas sur les détails spécifiques de l'image. Par conséquent, en utilisant

cette matrice comme guide, nous pouvons créer de nouvelles textures qui ressemblent à l'originale sans reproduire exactement les mêmes détails.

3.2.2 Génération de la texture

Pour générer une nouvelle texture à partir d'une image donnée, la descente de gradient est utilisée pour identifier une autre image qui correspond à la représentation de la matrice de Gram de l'image originale à partir d'une image de bruit blanc. Pour ce faire, on réduit la distance moyenne au carré entre les éléments de la matrice de Gram de l'image originale et la matrice de Gram de l'image en cours de création.

représentations de la matrice de Gram dans la couche l (Eq.3.1). La contribution de la couche l à la perte totale est alors :

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G_{ij}^l - \hat{G}_{ij}^l \right)^2 \quad (3.3)$$

et la perte totale est de :

$$\mathcal{L}(\vec{x}, \hat{\vec{x}}) = \sum_{l=0}^L w_l E_l \quad (3.4)$$

où w_l sont les facteurs de pondération de la contribution de chaque couche à la perte totale. La dérivée de E_l par rapport aux activations de la couche l peut être calculée analytiquement :

En utilisant la rétro-propagation à erreur standard [85], les gradients de E_l , et donc le gradient de $\mathcal{L}(\vec{x}, \hat{\vec{x}})$, peuvent être facilement déterminés par rapport aux pixels $\vec{x}$. Certaines stratégies d'optimisation numérique peuvent utiliser le gradient $\nabla_{\vec{x}} \mathcal{L}$ en tant qu'entrée.

3.2.3 Les reseaux pré-entraînés pour la synthèse de textures

Le réseau neuronal convolutionnel a été largement utilisé dans le traitement des images et la classification des textures et a obtenu de meilleurs résultats que les méthodes classiques, en particulier dans la synthèse des textures. Les textures sont représentées par les corrélations entre les cartes de caractéristiques dans plusieurs couches du réseau. Cette corrélation montre que les représentations des textures à travers les couches capturent de plus en plus les propriétés statistiques des images naturelles tout en rendant les informations sur les objets plus explicites. Ce modèle fournit un nouvel outil pour générer des stimuli pour les neurosciences. En outre, il pourrait permettre de mieux comprendre les représentations profondes apprises par les réseaux neuronaux convolutifs. Ce travail a ap-

pliqué quatre réseaux profonds pré-entraînés, dont alexnet, GoogleNet, VGG16, VGG19 et ResNet50, pour la synthèse de textures en utilisant l'apprentissage par transfert.

3.2.3.1 AlexNet

L'architecture d'AlexNet [82] est nettement plus complexe et étendue que celle de LeNet [86]. Il comporte huit couches, principalement des couches convolutionnelles, suivies de max-pooling et de trois couches entièrement connectées. La sortie est une fonction softmax à 1000 classes. Il est entraîné sur deux GPU GTX 580 de 3 Go qui ne communiquent que dans des couches particulières. Cette méthode permet de réduire les taux d'erreur des cinq meilleures prédictions. L'architecture ZFNet [37] améliore l'AlexNet en visualisant les actions d'AlexNet dans les différentes couches, ce qui facilite le débogage et améliore les résultats. De plus, elle permet d'observer la progression des caractéristiques pendant l'entraînement et, dans les couches intermédiaires, de traduire les activités en espace pixel.

3.2.3.2 GoogleNet

L'architecture Inception-v1 de GoogleNet maximise l'utilisation des ressources informatiques à l'intérieur du réseau [87]. Le réseau Inception est plus rapide que le réseau non Inception. Le module Inception du design de GoogleNet utilise une fonction d'activation rectifiée linéaire, une couche de pooling moyenne et une couche non entièrement connectée, avec suppression aléatoire après la suppression de la couche entièrement connectée. Ioffe et Szegedy [88], qui ont tenté de surmonter le décalage covariant interne, ont amélioré Inception-v1 pour créer Inception-v2. Ils ont obtenu un taux d'erreur des cinq meilleures prédictions de 4,82 %. Ce résultat est de 3,5 % meilleur que le précédent.

3.2.3.3 ResNet

ResNet [89] a utilisé l'apprentissage résiduel pour faciliter l'entraînement en profondeur du réseau et réduire les erreurs liées à la profondeur. Cette architecture présente plusieurs structures, dont 18 couches, 34 couches, 50 couches, 101 couches et 152 couches. Cette dernière est supérieure aux autres. IN [90] a fusionné l'architecture Inception avec des connexions résiduelles car elle est moins complexe et plus profonde que VGG et présente des performances similaires au réseau Inception-v3. Les trois architectures ResNet-Inception et Inception-v4 ont été testées : Inception-ResNet-v1 se comporte de manière similaire à Inception-v3, tandis que ResNet-Inception-v2 surpasse ResNet-Inception-v1. Inception-v4 est plus simple que Inception-v3. De plus, il présente davantage de modules Inception, bien qu'il ait des performances similaires à ResNet-Inception-v2.

3.2.3.4 VGGNet

Karen Simonyan et Andrew Zisserman [55] ont étudié l'effet de la profondeur d'un réseau convolutionnel neuronal sur sa précision en matière de reconnaissance d'images. Ils utilisent de très petits filtres de convolution (3×3) pour augmenter la profondeur des 11 à 19 couches de poids du réseau VGGNet construit. Les configurations VGG16 et VGG19, qui utilisent 16 et 19 couches de poids, sont les plus performantes. Avec l'augmentation de la profondeur, l'erreur de classification diminue jusqu'à ce qu'elle atteigne 19 couches, où elle devient saturée. L'importance de la profondeur dans les représentations visuelles est confirmée par les auteurs.

3.2.3.5 l'utilisation d'apprentissage par transfert dans la synthèse

Comme mentionné précédemment, l'apprentissage par transfert est généralement utilisé pour la classification, mais dans notre travail, nous l'adaptions également à la synthèse de manière similaire. Dans notre étude, nous avons transmis des images de textures simples à travers quatre modèles pré-entraînés afin d'extraire différentes caractéristiques de tailles variées, comme illustré dans la Figure 3.3. Après cette étape, nous avons obtenu cinq vecteurs de caractéristiques distincts, un pour chaque réseau, et avons calculé cinq matrices de Gram différentes, chacune avec ses propres paramètres.

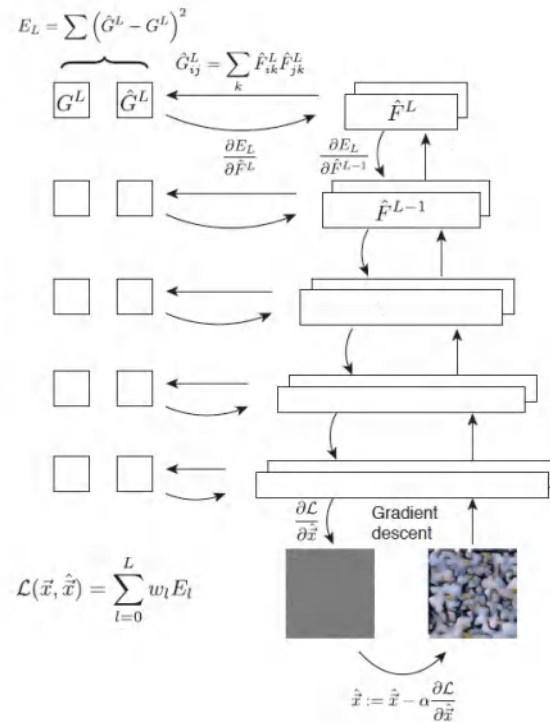


Figure 3.3: schéma de la generation de la texture par le transfert par apprentissage

Après avoir créé les matrices de Gram pour les cinq réseaux (AlexNet, GoogLeNet, ResNet et VGG) avec différents paramètres, nous passons à l'étape de génération de la texture. Au cours de cette étape, nous suivons les étapes suivantes pour chaque réseau

Génération de textures :

1. Nous commençons par une image de bruit blanc.
2. Nous la faisons passer à travers chaque réseau
3. Ensuite, nous calculons la matrice de Gram.
4. Nous calculons la fonction de perte.
5. Nous effectuons une rétro-propagation pour déterminer les gradients.
6. Enfin, nous incorporons le gradient dans notre stratégie d'optimisation.

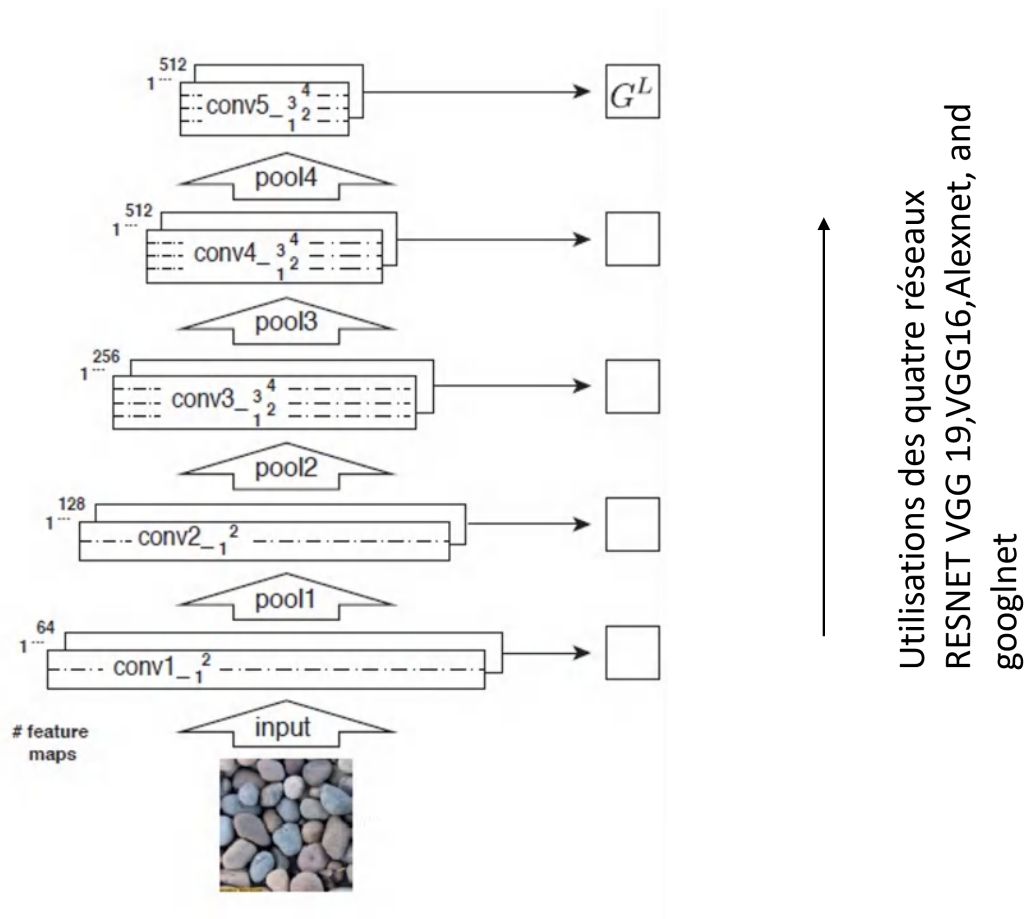


Figure 3.4: schéma de modelisation de la texture par le transfert par apprentissage

Algorithme 3.1 : La synthèse des textures par l'apprentissage par tranfert

entrée : texture original, image bruit blanc

sortie : texture synthétisé

— modélisation de la texture :

1- Extraire des caractéristiques de tailles différentes // Passer une image d'échantillon à travers lrs pré entraîné model

// Extraire des caractéristiques de différentes tailles

2-Calculer une statistique de résumé spatiale // "Corrélations de caractéristiques en tant que SSS

// Les textures sont stationnaires par définition

// Calculer le produit interne entre les cartes de caractéristiques dans chaque couche

// Un ensemble de matrices Gram G1, G2, ..., GL spécifie complètement une texture.

— generation de la texture

1-Commencer avec une image de bruit blanc

2-La faire passer à travers un ConvNet (réseau de convolution)

3-Calculer une matrice Gram

4-Calculer une fonction de perte

5-Effectuer la rétropropagation pour trouver les gradients

6-Injecter le gradient dans une stratégie d'optimisation"

3.2.4 Analyse des résultats

Le modèle de synthèse proposé a été mis en œuvre en Python à l'aide de la bibliothèque PyTorch pour réaliser les expériences, tandis que la méthode originale utilisait le cadre Caffe. La principale différence avec l'implémentation de [4] concerne la gestion de l'intensité des pixels lors de l'optimisation.

Les poids du réseau de générateurs ont été initialisés en utilisant la méthode de Xavier [91]. L'étape d'apprentissage a été mise en œuvre en utilisant l'optimiseur d'Adam [91], pour 150, 300 et 500 itérations. Le taux d'apprentissage initial est de 0,1. Il a été réduit à une valeur de 0,7 à l'itération 300 et toutes les 200 itérations. La taille du lot a été fixée à 16, de la même manière que pour [4].

Pour tester les différentes architectures proposées, le Describable Textures Dataset (DTD) a été utilisé [92]. Il se compose de 5140 images de textures en couleur ; il comprend 47 catégories de textures contenant 120 images de taille 300*300 et 640*640.

La figure 3.5 montre les résultats des différents réseaux convolutifs CNN utilisés. D'une manière générale, on constate que tous les réseaux ont donné des résultats acceptables.

Cependant, ces résultats seront classés en fonction de la précision et du temps de calcul des architectures proposées.

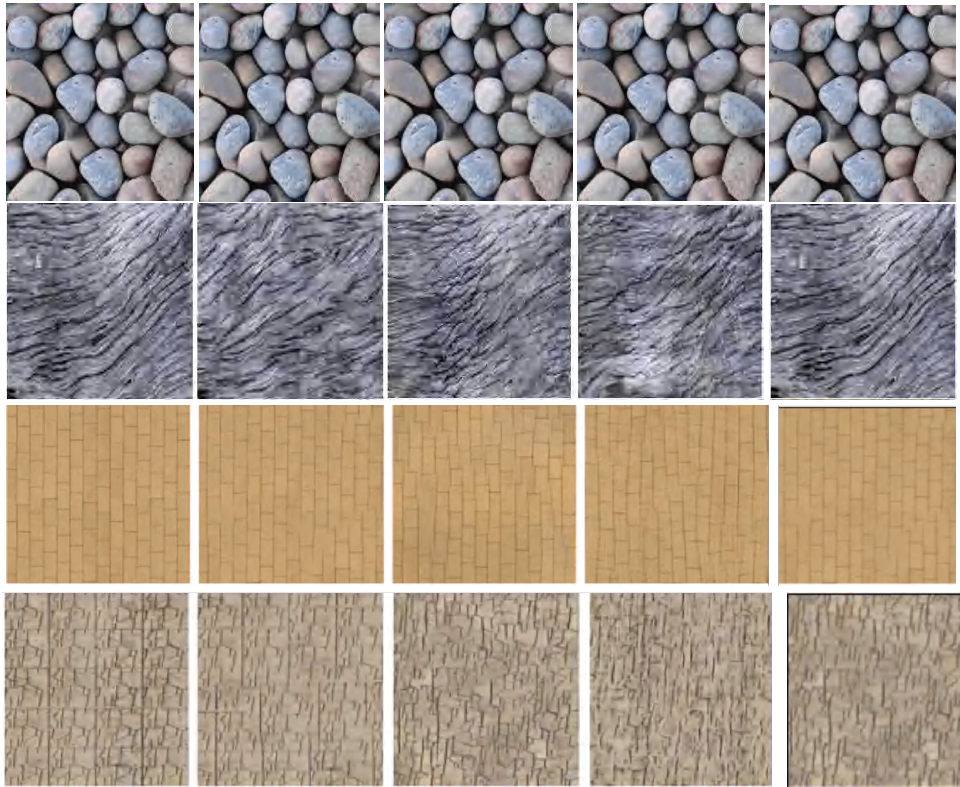


Figure 3.5: Synthèse de textures utilisant différents architectures pré entraînés

Tableau 3.1: précision de la synthèse en utilisant différentes architectures d'apprentissage par transfert pour différentes itérations

Architectures	iteration = 150	iteration = 300	iteration = 500
AlexNet	94.5	95.65	95.85
GoogleNet	95.69	95.8	96
VGG-16	96.06	96.14	96.71
VGG-19	96.97	96.55	96.47
ResNet-50	96.14	97.22	97.55

Tableau 3.2: Temps de calcul en minutes en utilisant différentes architectures d'apprentissage par transfert pour différentes itérations

Architectures	iteration= 150	iteration = 300	iteration = 500
AlexNet	44	98	101
GoogleNet	79	158	281
VGG-16	95	107	175
VGG-19	53	117	197
ResNet-50	55	110	183

L'objectif de cette étude est d'augmenter la précision, d'accélérer le temps de calcul en

utilisant différentes architectures d'apprentissage par transfert. Le nombre d'itérations sur la précision de la synthèse de texture est une autre caractéristique relevée dans les études. Cet effet est visible dans les tableaux 3.1 et 3.2 Le temps de calcul augmente progressivement avec le nombre total d'itérations; le tableau 3.2 montre que la majorité des réseaux pré-entraînés obtiennent de bons résultats avec une itération de 500 et des résultats acceptables avec une itération de 150 malgré le nombre inférieur de couches d'AlexNet. VGG16 et VGG19 sont aussi performants que resnet .

Resnet50 atteint la meilleure précision de 98,14 %, 98,22 % et 98,55%, respectivement, pour 150, 300 et 500 itérations. En outre, il a été observé qu'avec 500 itérations, les meilleurs résultats ont été obtenus pour les quatre architectures. Cependant, les réseaux VGG16 nécessitent une formation plus longue que les autres architectures. En considérant les résultats donnés dans le tableau 3.1 le tableau 3.2 et les images données par la fig 3.5, nous pouvons conclure que le ResNet-50 est l'architecture la plus adaptée pour la synthèse de texture.

3.3 Synthèse de la texture en utilisant les modèles autoregressifs a coefficients aléatoires

La synthèse de texture basée sur des exemples est constituée des algorithmes de création de texture les plus utilisés et les plus étudiés. Ces approches prennent une image du monde réel comme entrée. Elles tentent de produire une nouvelle texture à partir de celle-ci. L'objectif de ces méthodes est de générer une nouvelle texture qui soit perceptuellement similaire à l'originale.

Depuis plus de vingt ans, la synthèse de textures à partir d'exemples est un sujet de recherche actif. Jusqu'à présent, la génération d'une texture basée sur un exemple est un défi. Dans ce travail [93], nous présentons une nouvelle technique de synthèse de texture basée sur les modèles 2D-RCA.

Il est important de souligner que les paramètres 2D-RCA ont été utilisés dans [64] comme caractéristiques discriminantes dans la classification d'images et dans une petite simulation prouvant leur capacité à représenter une texture. Aucune modélisation réelle de texture n'a été effectuée pour générer de véritables images de texture. Dans le cadre de cette these, nous traitons ce problème. Avant de présenter la méthode proposée, nous donnons un aperçu de la construction des modèles 2D-RCA.

Glossaire

Les variables en gras sont des vecteurs dans Z^2 .

Les variables soulignées sont des vecteurs.

Pour chaque vecteur $x(t)$, $\hat{X}(N)$ est le moment empirique.

Pour chaque vecteur :

$\mathbf{s} = (s_1, s_2)$ et $\mathbf{t} = (t_1, t_2)$, on écrit $\mathbf{s} \ll \mathbf{t}$

si et seulement si $\left[(s_1 < t_1) \vee (s_1 = t_1) \wedge (s_2 \leq t_2) \right]$

pour $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^2$ tels que $\mathbf{a} = (a_1, a_2)$; $\mathbf{b} = (b_1, b_2)$; et $\mathbf{a} \ll \mathbf{b}$,

$S[a, b]$ est un ensemble indexé défini par :

$S[\mathbf{a}, \mathbf{b}] = (l, m) \in \mathbb{Z}^2 / a_1 \leq l \leq b_1, a_2 \leq m \leq b_2$ en arrangeant ses termes selon l'ordre lexicographique $\ll$.

3.3.0.1 Champs aléatoire

Considérons un espace probabiliste $(\Omega, \mathcal{A}, \mathcal{P})$ et un domaine D qui appartient à l'espace $\mathbb{R}^d$. Un champ aléatoire sur D , prenant des valeurs dans E , est une collection de variables aléatoires définies sur l'espace probabiliste $(\Omega, \mathcal{A}, \mathcal{P})$. Chaque élément s du domaine D est associé à une variable aléatoire X_s , et nous notons cette collection sous la forme $\{X_s, s \in D\}$.

Les caractéristiques d'un champ aléatoire sont entièrement définies par les distributions fini-dimensionnelles $(X_{s_1}, X_{s_2}, \dots, X_{s_n})$ pour toute sélection finie $\{s_1, s_2, \dots, s_n\}$ de points dans D . Ces distributions doivent se conformer à des conditions minimales de cohérence appelées conditions de Kolmogorov. Un champ aléatoire discret X réfère à un processus où les valeurs aléatoires sont associées à des positions sur le réseau $\mathbb{Z}^d$. Les emplacements, généralement représentant des points dans des réseaux structurés liés à un graphe de voisinage, sont caractérisés par les valeurs X_s , qui peuvent être considérées comme des données "intégrées" sur les régions s . Dans le contexte bidimensionnel ($d = 2$), le réseau est souvent appelé une "grille".

3.3.1 Modèles RCA spatiaux et estimation des paramètres

S.Kharfouchi [94] a introduit le modèle spatial non linéaire qui généralise les modèles autorégressifs standard à coefficients aléatoires (RCA) à deux dimensions; 2D-RCA. Il est généré par :

$$X(t) = \sum_{\mathbf{s} \in]0; P]} a_{\mathbf{s}}(\mathbf{t}) x(\mathbf{t} - \mathbf{s}) + e(\mathbf{t}), \mathbf{t} \in \mathbb{Z}^2 \quad (3.5)$$

Où $a_{\mathbf{s}}(\mathbf{t}) = \alpha_s + \beta_{\mathbf{s}}(\mathbf{t})$

Pour ces modèles, nous avons besoin des hypothèses suivantes :

Les modèles 2D-RCA proposés dans [64] sont définis sur un réseau régulier. Ils sont unila-

téraux par construction ; seulement la plupart des images que nous avons acquises avec des pixels irréguliers. Heureusement, avec l'utilisation croissante de la technologie au moins dans certaines situations, les données avec des pixels peuvent être remplacées par une grille régulière en utilisant des techniques d'interpolation d'images et des programmes de rééchantillonnage. Comme dans la pratique, la plupart des modèles spatiaux sont du premier ordre ; nous nous concentrerons sur les modèles 2D-RCA du premier ordre donnés par :

$$X(i, j) = \alpha X(i, j - 1) + \beta X(i - 1, j) + \gamma X(i - 1, j - 1) + a_1(i, j)X(i, j - 1) + a_2(i, j)X(i - 1, j) + a_3(i, j)X(i - 1, j - 1) + \varepsilon(i, j) \quad (3.6)$$

Où $\varepsilon(i, j)$, (i, j) dans Z^2 est une séquence de variables aléatoires indépendantes et identiquement distribuées (i.i.d) avec une moyenne nulle et une variance σ^2 .

α ; β et γ sont des constantes.

$\{a_l(i, j); (i, j) \text{ dans } Z^2\}$ sont des séquences indépendantes de variables aléatoires centrées sur Z^2 .

$E[a_l^2(i, j)] = \eta_l^2$ et $a_l(i, j)$ sont indépendantes de $\{\varepsilon(i, j); (i, j) \in Z^2\}$ pour tous $l = 1, 2, 3$.

La condition de stationnarité du second ordre est donnée par :

$$\begin{cases} \Delta > 0 \\ \frac{1}{2} (\alpha^2 + \beta^2 + \eta_1^2 + \eta_2^2 + \sqrt{\Delta}) < 1 \end{cases}$$

Où :

$$\Delta = (\alpha^2 + \beta^2 + \eta_1^2 + \eta_2^2)^2 + 4(\gamma^2 + \eta_3^2 + 2\alpha\beta\gamma) \geq 0 \quad (3.7)$$

Dans des conditions stationnaires, l'estimation du modèle 2D-RCA donnée par l'équation (3.8) est réalisée par la méthode généralisée des moments (GMM) [95]. Sur la base des observations $(X(i, j), 1 \leq i \leq n, 1 \leq j \leq m)$, l'estimateur GMM de $\underline{\theta} = (\alpha, \beta, \gamma)$ est donné par :

$$\hat{\underline{\theta}}_{N \times M} = (\hat{\alpha}_N, \hat{\beta}_N, \hat{\gamma}_N) = \left(\sum_{i=1}^N \sum_{j=1}^M [\underline{x}(i, j) \underline{x}'(i, j)] \right)^{-1} \times \sum_{i=1}^N \sum_{j=1}^M \underline{x}(i, j) X(i, j) \quad (3.8)$$

Où :

$$\underline{x} = (X(i, j - 1); X(i - 1, j); X(i - 1, j - 1)). \quad (3.9)$$

Les caractéristiques (2D-RCA) sont représentées par la matrice $\hat{\theta}_{N \times M}$ de dimensions $n \times m$ (où $N \times M$ correspond à la taille des images utilisées).

3.3.2 Algorithme de synthèse de texture utilisant le modèle 2D-RCA

Les modèles 2D-RCA présentent deux caractéristiques essentielles. Tout d'abord, des expériences de simulation ont montré que ces modèles peuvent représenter une large gamme de textures. Deuxièmement, pour décrire de nombreuses images de texture, les modèles 2D-RCA ne nécessitent pas un grand nombre de paramètres.

Cette section présente un nouvel algorithme de synthèse des textures. Cet algorithme produit une approximation des images en utilisant les modèles 2D-RCA. L'algorithme est basé sur le fait que il peut représenter n'importe quelle image de texture à l'aide de processus 2D-RCA unilatéraux du premier ordre.

L'image originale est divisée en sous-images quadrillées de taille $k \times k$. Les modèles 2D-RCA du premier ordre sont ajustés à chaque bloc. Ensuite, pour chaque modèle local ajusté, une sous-image approximative est formée en utilisant la méthode générale des moments. Pour toutes les sous-images de chaque modèle local ajusté, les intensités sur les limites sont présentées en lissant le bord entre les blocs.

Soit :

$$X = X(m, n) \quad 0 \leq m \leq M - 1, 0 \leq n \leq N - 1 \quad (3.10)$$

l'image originale.

Considérons l'image approximative $\hat{X}$ de X , de la forme :

$$\hat{X}(i, j) = \hat{\alpha}X(i - 1, j) + \hat{\beta}X(i, j - 1) + \hat{\gamma}(1, 1)X(i - 1, j - 1) \quad (3.11)$$

où $\hat{\alpha}$, $\hat{\beta}$ et $\hat{\gamma}$ sont les estimations GMM de α , β et γ méthode.

Soit

$$4 \leq k \leq \min(M, N)$$

Une taille de bloc inférieure à ($k \times k = 4 \times 4$) donne des résultats non significatifs.

Par souci de simplicité, nous supposons que les images à traiter sont ordonnées de telle sorte que le nombre de colonnes moins une et le nombre de lignes moins une sont tous deux des multiples de $k - 1$; c'est-à-dire,

$$X = X(m', n') \quad 0 \leq m' \leq M' - 1, 0 \leq n' \leq N' - 1 \quad (3.12)$$

où :

$$M' = \left\lceil \frac{M-1}{k-1} \right\rceil (k-1) + 1,$$

$$N' = \left\lceil \frac{N-1}{k-1} \right\rceil (k-1) + 1.$$

Pour tout $i_b = 1, \dots, \left\lceil \frac{M-1}{k-1} \right\rceil$, et pour tout $j_b = 1, \dots, \left\lceil \frac{N-1}{k-1} \right\rceil$, nous définissons le bloc $(k-1) \times (k-1)$ (i_b, j_b) de l'image X par :

$$B_X(i_b, j_b) = Z(r, s) \quad (3.13)$$

$$(k-1)(i_b-1) + 1 \leq r \leq (k-1)i_b$$

$$(k-1)(j_b-1) + 1 \leq s \leq (k-1)j_b$$

La méthode est reprise par l'algorithme suivant et la Fig.3.6

Algorithme 3.2 : Synthèse de texture par le modèles 2D-RCA

entrée : texture original X , taille de la fenêtre $(k \times k)$

sortie : texture approximée $\hat{X}$

- Diviser l'image originale en blocs $B_X(i_b, j_b)$.
- Modéliser chaque bloc d'image $B_X(i_b, j_b)$ avec les modèles 2D-RCA en utilisant l'équation 3.6 et les estimateurs GMM $(\hat{\alpha}, \hat{\beta}, \hat{\gamma})$ donnés par l'équation 3.8.
- Soit $\hat{Z}$ le bloc approximé $B_X(i_b, j_b)$ par

$$\hat{Z}(r, s) = \hat{\alpha}Z(r-1, s) + \hat{\beta}Z(r, s-1) + \hat{\gamma}(1, 1)Z(r-1, s-1) \quad (3.14)$$

- Concaténer toutes les images de sous-blocs approximées $\hat{Z}(r, s)$.
- Calculer l'image approximée $\bar{X}$ de X comme :

$$\hat{X}_{m,n} = \hat{Z}_{m,n} + \bar{X}$$

3.3.3 Analyse des résultats

Dans cette section, des résultats expérimentaux détaillés sont présentés pour évaluer l'efficacité de l'algorithme proposé. La méthode proposée a été appliquée à un grand nombre

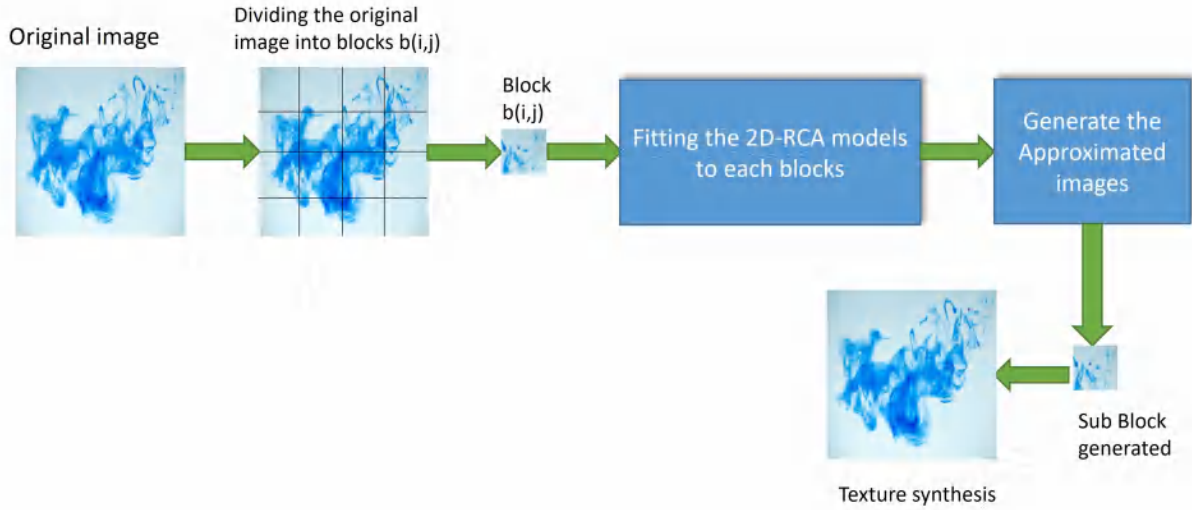


Figure 3.6: Méthode proposée

de textures provenant de trois bases de données différentes, notamment la base de données Broadatz [96], DTD [92] et CG-Textures [97].

Ensemble de données DTD : contient 5640 images. Pour chacune d'entre elles, 47 classes de textures sont disponibles pour les 120 catégories d'images. Les tailles des images varient entre 300×300 et 640×640 . Il s'agit de l'ensemble de données le plus difficile car il contient de grandes images.

Textures Broadatz : est l'ensemble de données de textures le plus utilisé, en particulier dans les domaines de la vision par ordinateur et du traitement image. Étant donné qu'ils ont été si fréquemment utilisés dans des articles précédents sur l'analyse/synthèse de textures, les inclure dans une étude de synthèse de textures est presque inévitable. Il se compose de 112 textures en images en niveaux de gris de différents types de textures.

Bases de données CG-texture : contiennent des images de textures en couleur provenant d'une collection de photos texturées naturelles, étiquetées avec diverses caractéristiques centrées sur l'humain qui sont motivées par les caractéristiques perceptuelles des textures.

La méthode proposée peut synthétiser des textures de haute qualité et de manière stable. Pour évaluer la qualité des images reconstruites par l'algorithme proposé, trois mesures d'évaluation de la qualité d'image ont été utilisées. L'indice de similarité structurelle (SSIM) [98], le rapport signal sur bruit maximal (PSNR) [99] et l'erreur quadratique moyenne (MSE) [100] ont été utilisés pour quantifier la similarité entre l'image originale et celle reconstituée. Les images de meilleure qualité auront une MSE plus faible, un PSNR élevé et un SSIM élevé. Le SSIM, le PSNR et le MSE sont donnés par :

$$\text{MSE} = \frac{1}{MN} \sum_{n=0}^M \sum_{m=1}^N [\hat{g}(n, m) - g(n, m)]^2 \quad (3.15)$$

$$\text{PSNR} = 10 \log_{10} (\text{peakval}^2) / \text{MSE} \quad (3.16)$$

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_x\sigma_y + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3.17)$$

où μ_x et μ_y représentent les moyennes locales, σ_x et σ_y sont les écarts-types, et σ_{xy} est la covariance croisée pour les images x et y respectivement. La première étape de l'algorithme proposé consiste à trouver la meilleure taille de fenêtre pour obtenir les meilleurs résultats. Dans ce but, plusieurs tests ont été réalisés en utilisant plusieurs tailles de fenêtres : 8×8 , 16×16 , 32×32 et 64×64 pixels, respectivement. De plus, l'effet sur les résultats de la synthèse de textures est démontré. Ainsi, différentes textures d'images de la base de données Brodatz ont été testées. Il est intéressant de remarquer qu'une taille de fenêtre inférieure à 5×5 est inappropriée pour caractériser une texture. De plus, l'ajustement des modèles 2D-RCA à l'image entière se compose de trois paramètres qui ne sont pas suffisants pour modéliser toute l'image .

La figure 3.7(a) représente les images originales. Les images reconstruites par la méthode proposée sont présentées dans la figure 3.7(b) en utilisant une taille de fenêtre de 32×32 . La figure 3.7(c) montre la différence entre les images originales et les images reconstituées. Pour différentes images, on observe que les images reconstruites sont visuellement acceptables. Les structures, les contours et les régions de texture sont préservés, comme on peut le remarquer dans la figure 3.7(c).

Tableau 3.3: Comparaison de la qualité d'image en utilisant différentes tailles de fenêtres

Window size	8×8			16×16			32×32			64×64		
	SSim	MSE	PSNR	SSim	MSE	PSNR	SSim	MSE	PSNR	SSim	MSE	PSNR
image1	0.993	4.543	21.71	0.995	9.421	24.121	0.998	4.639	26.712	0.995	15.765	26.712
image2	0.998	5.456	20.581	0.998	5.885	24.121	0.998	4.064	24.121	0.997	2.458	23.89
image3	0.996	1.423	22.141	0.997	1.4	22.818	0.997	0.825	23.79	0.997	4.45	20.08
image4	0.998	0.93	23.26	0.997	1.425	22.11	0.993	0.745	23.01	0.995	3.17	20.08
image5	0.996	2.5	22.81	0.991	1.5	22.18	0.992	0.489	23.224	0.997	4.067	20.78
image6	0.997	1.543	22.72	0.996	1.4320	23.11	0.999	0.065	23.82	0.992	1.459	21.34

Le tableau 3.3 présente les valeurs obtenues pour SSIM, MSE et PSNR pour différentes tailles de fenêtres : 8×8 , 16×16 , 32×32 et 64×64 . Selon les résultats donnés dans le tableau 3.3 et la figure 3.7, on remarque que les meilleurs résultats sont obtenus avec

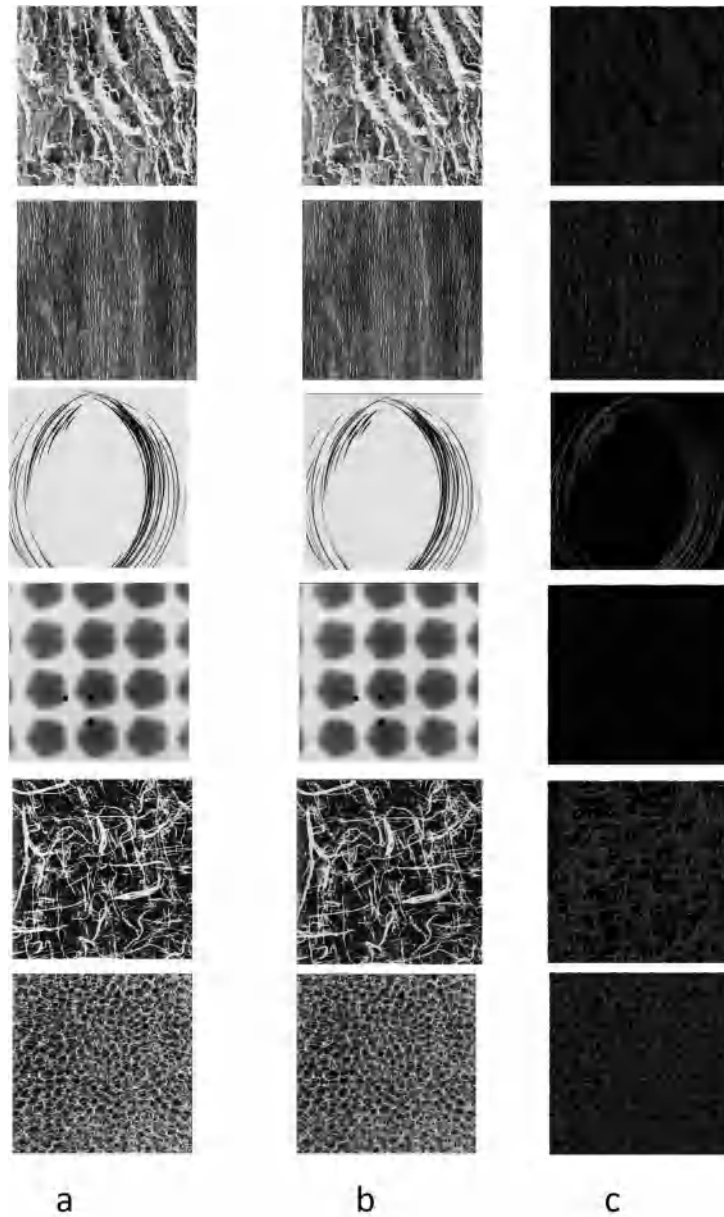


Figure 3.7: Image d'origine (a), (b) image générée par les modèles 2D-RCA, (c) différence entre les deux images.

une taille de fenêtre de 32×32 . Les mauvais résultats sont inhérents à la remarque mentionnée précédemment concernant la taille de la fenêtre et le nombre de paramètres représentant une texture. Ainsi :

Certaines régions semblent homogènes à grande échelle, ce qui implique qu'elles sont une réalisation d'un processus aléatoire stationnaire, mais hétérogènes à petite échelle, ce qui implique que la région n'est pas stationnaire.

Pour de petites fenêtres, les niveaux de gris dans différentes régions sont souvent

comparables. Ainsi, pour les tests suivants, nous adoptons une taille de fenêtre de 32×32 . Diverses textures provenant de différentes bases de données d'images ont été synthétisées pour apporter une valeur supplémentaire à la méthode proposée.

La figure 3.8 présente les résultats de synthèse des bases de données DTD et Brodatz. Chaque panneau de gauche représente la texture originale, et à droite : le résultat de synthèse à l'aide de la méthode proposée. Dans cette expérience, seuls les résultats de 20 textures sont présentés.

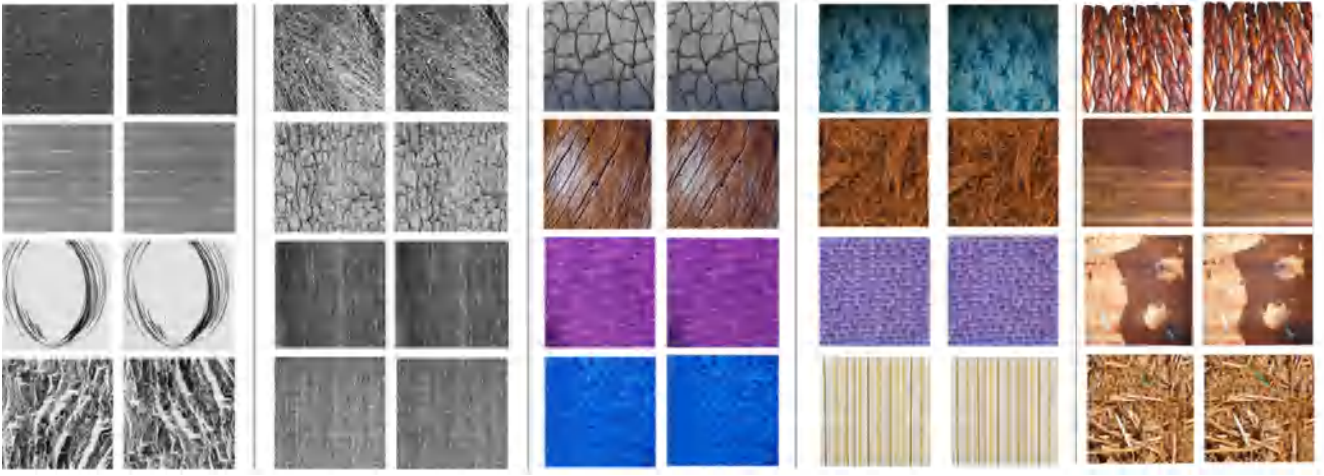


Figure 3.8: Résultats de synthèse des bases de données DTD et Brodatz par la méthode proposée. Dans chaque panneau, à gauche : texture d'origine, à droite : résultat synthétisé

Le tableau 3.4 montre les valeurs obtenues pour SSIM, MSE et PSNR. Les tests effectués ont prouvé que la méthode de synthèse de texture proposée est adaptée pour représenter une diversité de classes de textures, comme le confirment les résultats de la figure 3.8 et du tableau 3.4. Pour certaines textures synthétisées de la figure 3.6, les valeurs de PSNR sont faibles. Cependant, les textures sémantiques sont préservées. Elles sont simplement moins nettes que les originales. Par exemple, pour les images 9, 7 et 5 de la figure 3.8, les valeurs de PSNR sont respectivement de 15,36, 17,25 et 19,42.

L'efficacité de l'approche proposée a été comparée à celle de méthodes bien établies traitant de ce problème dans la littérature. Elles consistent en : Portilla et Simoncelli [19], Gatys et al. [101, 102], Corrélations Profondes pour la Synthèse de Textures (Deep Correlations) [97], c-cgCNN-Gram et c-cgCNN-Mean [60], gCNN [61], CoopNet [103] et la méthode d'optimisation de texture à auto-ajustement [101].

Les figures de la figure 3.9(a) à la figure 3.9(e) et de la figure 3.10(a) à la figure 3.10(e) représentent respectivement les images originales, Portilla et Simoncelli [19],

Tableau 3.4: SIM, MSE et PSNR pour les résultats de synthèse des bases de données DTD et Brodatz.

image	SSIM	PSNR	MSE
image 1	0.9995	27.001	0.0025
image 2	0.9997	26.001	0.0023
image 3	0.9998	27.125	0.0084
image 4	0.9994	23.03	0.0897
image 5	0.9993	19.4248	0.0119
image 6	0.9996	22.0443	0.0087
image 7	0.9990	17.25	0.0793
image 8	0.9991	20	0.1528
image 9	0.9990	15.56	0.0278
image 10	0.9996	21.2821	0.0074
image 11	0.9997	22.5054	0.0056
image12	0.9999	26.6825	0.0021
image 13	0.9998	24.8141	0.0033
image 14	0.9996	21.2436	0.0075
image 15	0.9998	22.89471	0.0717
image 16	0.9998	26.7121	0.0021
image 17	0.9996	21.6248	0.0069
image 18	0.9998	28.8630	0.0670
image 19	0.9995	20.0573	0.0993
image 20	0.9998	20.2697	0.0094

Gatys [102], kaspar [101], Corrélations Profondes pour la Synthèse de Textures (Deep Correlations) [97] et les résultats proposés. Dans ces expériences, les images de test ont été extraites des bases de données CG-Textures et Brodatz. Tous les algorithmes utilisés ont été exécutés avec leurs paramètres par défaut tels que fournis dans leurs articles. Les pires résultats ont été obtenus avec l’algorithme de Gatys [102], suivi en termes de qualité par kaspar [101], Deep Correlations [97] et les résultats proposés. Notre méthode génère autant que possible toutes les textures, qu’elles soient déterministes ou aléatoires. Sur la base des résultats précédents, on peut observer que les méthodes précédentes ont été insatisfaisantes en termes de préservation des structures.

Le tableau 3.5 justifie cette évaluation. Pour les quatre méthodes, le tableau 3.5 montre les valeurs obtenues pour SSIM, MSE et PSNR. Pour toutes les images testées, la méthode proposée obtient le plus haut SSIM dans $[0.9994; 0.9998]$, le plus bas MSE dans $[0.0027; 0.1368]$ et le plus haut PSNR dans $[17.8895; 25,7590]$, tandis que les métriques des méthodes comparées sont très mauvaises, notamment en ce qui concerne les valeurs de MSE et SSIM. Une comparaison qualitative avec des méthodes récentes basées sur l’apprentissage en profondeur a été effectuée. Les résultats obtenus sont comparés à ceux



Figure 3.9: Comparaison des résultats de la synthèse de textures par différentes méthodes.

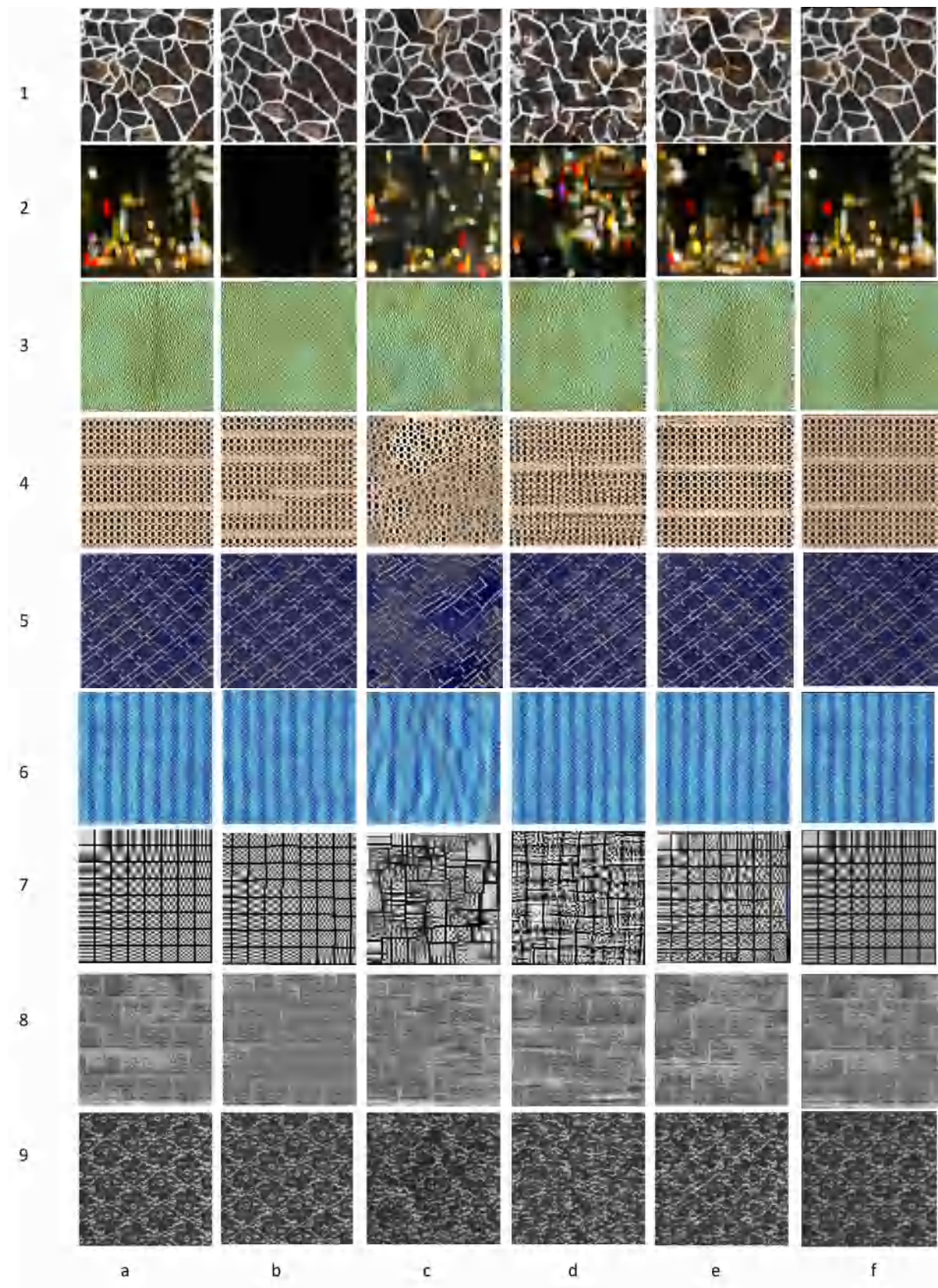


Figure 3.10: Comparaison des résultats de la synthèse de textures par différentes méthodes

fournis par c-cgCNN-Gram [60], c-cgCNN-Mean, gCNN [61], CoopNet [103] et la méthode d’optimisation de texture à auto-ajustement [101].

Tableau 3.5: Comparaison des mesures de qualité donnée par la figure 3.11

Window size	Proposed method			Portilla and Simoncelli's			Kaspar et al's			Gatys et al's			The deep correlation synthesized		
	SSim	MSE	PSNR	SSim	MSE	PSNR	SSim	MSE	PSNR	SSim	MSE	PSNR	SSim	MSE	PSNR
image1	0.9997	0.0047	23.2990	0.1652	9.2866e+03	9.2866e+03	0.1461	8.5796e+03	8.7961	0.1453	9.3568e+03	8.215	8.4195	0.1483	14.7218
image2	0.9996	0.0068	21.6777	0.1369	1.9630e+03	15.2016	0.1278	1.9195e+03	15.2988	0.0789	1.9321e+03	15.2706	0.0970	2.2149e+03	14.6773
image3	0.9994	0.0120	19.2006	0.1786	1.2643e+03	17.1122	0.1989	1.5741e+03	16.1605	0.0895	15.3198	15.21	0.1256	2.1224e+03	14.8626
image4	0.9995	0.0079	21.0428	0.1663	4.1138e+03	11.9884	0.1906	5.0343e+03	11.1114	0.1414	11.9925	0.149	0.5649	1.5204e+03	16.3113
image5	0.9997	0.0067	21.7500	0.2091	1.4312e+03	16.5739	0.2322	1.2710e+03	17.0893	0.2322	1.3937e+03	16.6890	0.1950	1.3416e+03	16.8546
image6	0.9997	0.0069	21.6189	0.1156	1.3393e+03	16.8621	0.1045	1.3393e+03	16.6162	0.1070	1.4173e+03	17.1091	0.2015	1.2242e+03	17.2523
image7	0.9998	0.0027	25.7590	0.1369	1.9630e+03	15.2016	0.1278	1.9195e+03	15.2988	0.2103	15.2706	0.0970	0.2214	0.2562	14.6773
image8	0.9996	0.0084	20.7420	0.0832	1.5484e+03	16.2321	0.0769	1.7870e+03	15.6096	0.0654	1.9013e+03	15.3402	0.0791	1.7552e+03	15.6875
image9	0.9994	0.0163	17.8895	0.0723	1.9301e+03	15.2751	0.0505	2.2445e+03	14.6196	0.0439	2.0796e+03	14.9510	0.0301	2.2230e+03	14.6613

Les images de la figure 3.11(a) à la figure 3.11(g) représentent respectivement les images originales, c-cgCNN-Gram [60], c-cgCNN-Mean, gCNN [61], CoopNet [103], méthode d’auto-ajustement [101] et l’algorithme proposé. Les résultats obtenus avec gCNN [61], CoopNet [103] et la méthode d’auto-ajustement sont les pires, notamment pour les textures aléatoires. La méthode d’auto-ajustement fonctionne mieux que gCNN [61] et CoopNet sur les textures déterministes. Les deux méthodes c-cgCNN-Gram et c-cgCNN-Mean produisent de meilleurs échantillons que les autres méthodes de référence. Cependant, les textures produites ne sont pas identiques à l’exemple, contrairement à la méthode proposée, qui produit des textures visuellement similaires à l’exemple. De plus, elles peuvent également reproduire des détails à petite échelle, même pour des textures fortement structurées.

Nous pouvons constater que la méthode proposée s’adapte à toutes les textures déterministes et aléatoires différentes par rapport aux meilleures méthodes de la littérature. Il est également important de mentionner que les résultats de la méthode proposée sont généralement comparables. Nous évaluons la similarité structurelle à différentes échelles (SSIM) entre la texture synthétisée et l’exemple. Un score élevé indique un degré élevé de similarité visuelle. Le tableau 3.6 résume les résultats quantitatifs. La méthode proposée surpasse les méthodes de référence existantes dans la majorité des cas, selon les résultats obtenus.

Dans le cadre de la conservation et de la restauration du patrimoine, nous avons testé l’algorithme proposé sur des images représentant des vestiges de l’époque romaine. La figure 3.12.a est un aqueduc romain et la figure 3.12.b est une statue de l’empereur Constantin, fondateur de la ville de Constantine en Algérie. Les images reconstruites présentent une très bonne qualité et sont similaires aux originales. Cela est justifié par les valeurs de l’EQM (erreur quadratique moyenne) et du SSIM (indice de similarité structurale) données dans le tableau 3.7.

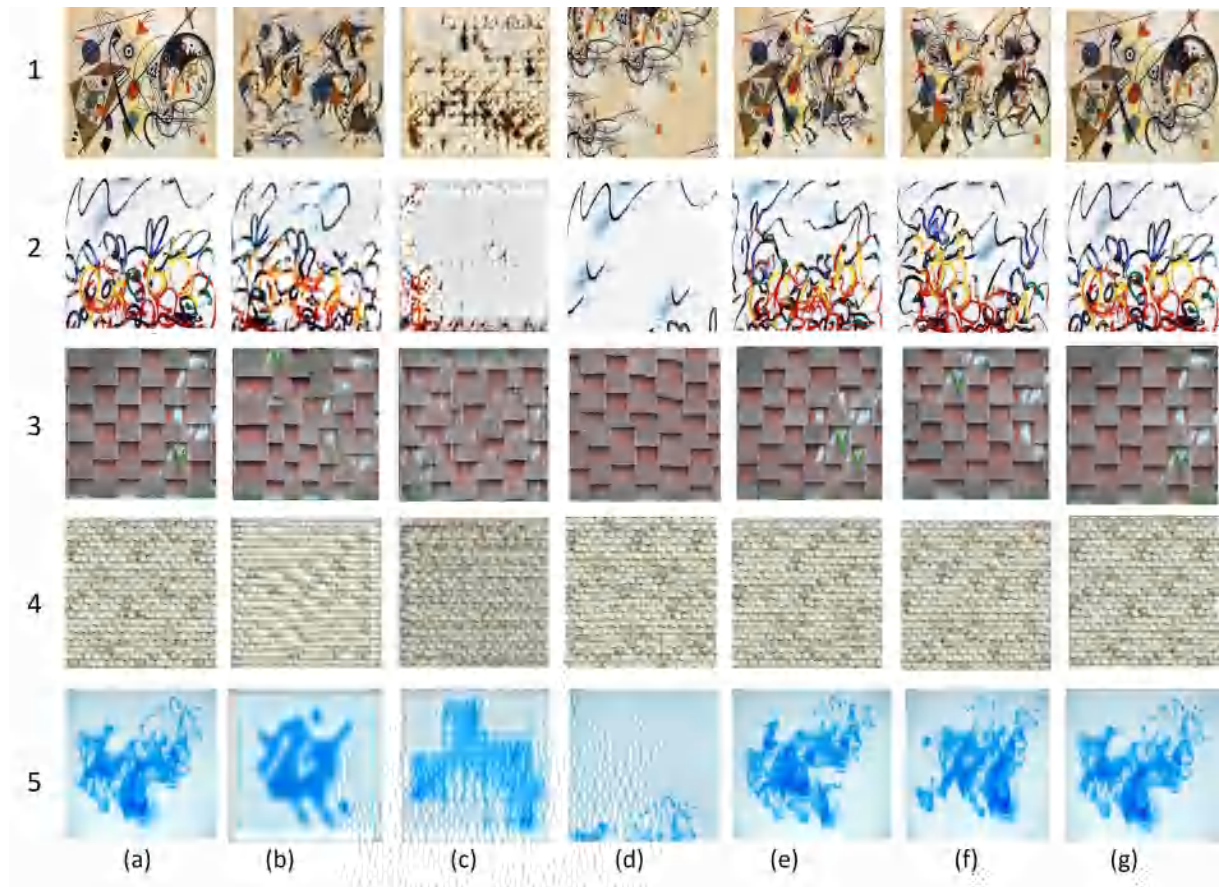


Figure 3.11: Les lignes 1 à 5 représentent les images originales et obtenues par différents algorithmes. Les colonnes a à g décrivent respectivement les images originales, c-cgCNN-Gram, c-cgCNN-Mean, gCNN, CoopNet, la méthode d'auto-ajustement et l'algorithme proposé.

Tableau 3.6: "Résultat de la synthèse de texture montré dans la FIG. 3.11 en utilisant la SSIM."

Méthodes	image 1	image 2	image 3	image 4	image 5
gCNN [61]	0.05	0.05	0.11	0.11	0.35
self-tuning [101]	0.03	0.17	0.07	0.01	0.42
CoopNet [103]	0.05	0.09	0.20	0.08	0.32
c-cgCNN-Gram [60]	0.10	0.09	0.31	0.36	0.43
c-cgCNN-Mean [60]	0.14	0.10	0.10	0.00	0.46
méthode proposé	0.9099	0.927	0.9014	0.9246	0.90251

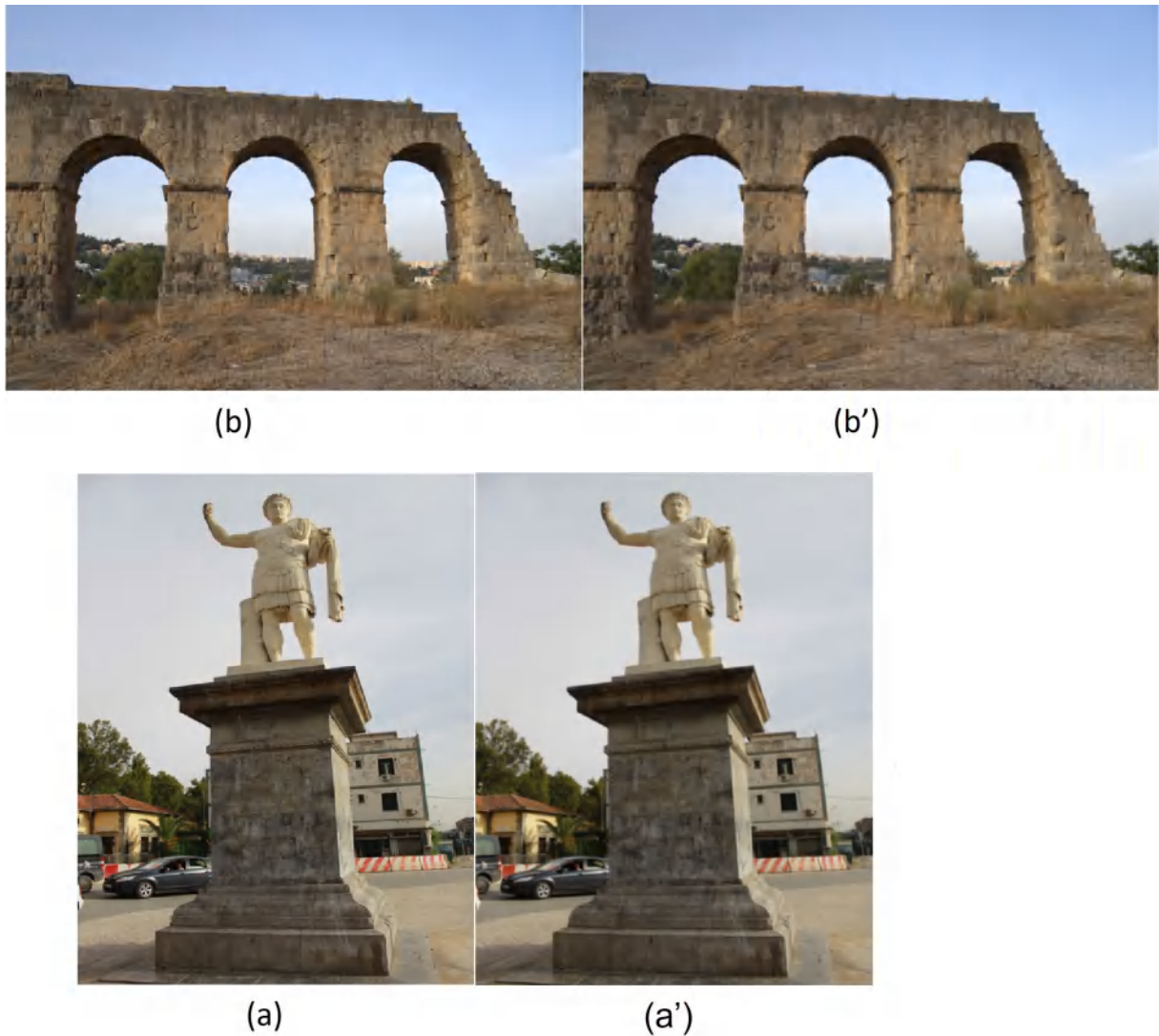


Figure 3.12: Résultats de la synthèse de textures en utilisant la méthode proposée

Tableau 3.7: SSIM, MSE pour les résultats de synthèse de la Figure 3.12.

image	SSIM	MSE
image 1	0.9995	0.0025
image 2	0.9997	0.0023

3.4 Conclusion

Dans ce chapitre, nous avons présenté les méthodes proposées pour la synthèse de textures, ainsi que les résultats obtenus par ces méthodes. La synthèse de textures par les réseaux de neurones donne de bons résultats, mais présente des contraintes, notamment dans la

synthèse de textures structurales et aléatoires. En revanche, la deuxième méthode, basée sur les modèles 2D-RCA, a démontré son efficacité et sa robustesse.

En se basant sur la qualité des résultats obtenus, à la fois les textures déterministes et aléatoires peuvent être modélisées à l'aide de ces paramètres. L'un des avantages de cette méthode est qu'elle ne nécessite que quelques paramètres, présentant une bonne complexité algorithmique.

Pour évaluer les performances de l'algorithme proposé, nous avons comparé les résultats à ceux obtenus par d'autres approches, ce qui a permis de prouver l'efficacité de notre méthode. la deuxième approche se distingue par sa simplicité par rapport aux méthodes comparées. Il est à noter qu'elle s'adapte à toutes les différentes classes de textures, contrairement aux autres processus de l'état de l'art qui ne conviennent qu'à certains types spécifiques.

Chapitre 4

Approches développées pour la classification de la texture

4.1 Introduction

La classification de la texture s'inscrit au cœur même de notre compréhension des images et des informations visuelles. Elle trouve des applications variées, allant de la détection des anomalies médicales à l'analyse des matériaux industriels et à la reconnaissance des motifs dans les œuvres d'art. Cependant, malgré les avancées significatives dans ce domaine, des défis subsistent, notamment la nécessité de traiter des textures complexes, non uniformes et présentant des variations subtiles. C'est dans ce contexte que s'inscrit ma contribution, qui cherche à repousser les limites actuelles de la classification de la texture. La classification de la texture revêt une importance capitale dans une multitude d'applications, allant de la reconnaissance de motifs dans les images médicales à la détection de défauts.

Dans ce chapitre, nous présentons les approches de classification développées dans le cadre de cette thèse, qui englobe un ensemble diversifié de méthodes et de techniques visant à attribuer des étiquettes spécifiques à différentes textures. Premièrement, une nouvelle approche pour la classification automatique des textures en utilisant le classificateur K Plus Proches Voisins (K-NN) est présentée. Cette approche se divise en deux étapes. Tout d'abord, le modèle autorégressif bidimensionnel aléatoire 2D-RCA est appliqué aux images de textures afin d'extraire les paramètres pertinents permettant de représenter cette texture. Ensuite, les paramètres extraits sont estimés à l'aide de la Méthode Généralisée des Moments (GMM). Ces paramètres sont utilisés comme caractéristiques de pixels dans la classification des images de textures en utilisant la méthode des K Plus Proches Voisins (K-NN). Dans un autre volet, une recherche exhaustive de caractéris-

tiques pertinentes pour la classification automatique des textures a été entreprise. Ces caractéristiques ont été extraites de différentes méthodes coopératives traitant de la caractérisation des textures. Un vecteur de caractéristiques optimal a été construit à l'aide d'algorithmes génétiques (GA) pour éviter la redondance des caractéristiques. Ensuite, la classification des textures a été réalisée à l'aide d'algorithmes de classification multi-classes SVM, de K-plus proches voisins et de la forêt aléatoire.

Tout d'abord, les fissures sont l'un des dommages routiers les plus répandus et elles peuvent être dangereuses. La détection des fissures est généralement effectuée par des inspecteurs formés ou des ingénieurs structurels. Cependant, il s'agit d'une tâche chronophage, subjective et exigeante en main-d'œuvre. Jusqu'à présent, la plupart des technologies de détection de fissures reposent sur l'inspection manuelle plutôt que sur l'identification automatique basée sur l'image, ce qui rend le processus à la fois coûteux et chronophage. Dans notre thèse, nous proposons un système d'analyse automatique des dommages aux revêtements basé sur un cadre d'apprentissage par transfert profond. L'objectif de cette recherche est d'améliorer la précision de la classification tout en réduisant le temps d'entraînement.

En utilisant quatre architectures de réseaux de neurones convolutifs (CNN) pré-entraînés, notre système de classification évalue l'apprentissage par transfert profond pour l'extraction de caractéristiques. Ensuite, ces quatre modèles pré-entraînés sont combinés dans ce que nous appelons l'apprentissage en ensemble pour augmenter la précision de prédiction du modèle

4.2 Classification de textures par K-NN utilisant un modèle autorégressif à coefficients aléatoires bidimensionnel (2D-RCA)

Les caractéristiques de texture fournissent des informations cruciales sur les primitives qui composent une texture et leur interrelation [104]. Dans ce travail, nous développons une méthode proposée basée sur les statistiques spatiales et le K Plus Proches Voisins (K-NN). Tout d'abord, une image est modélisée en utilisant deux modèles autorégressifs à coefficients aléatoires [94]. Ensuite, la Méthode Généralisée des Moments (GMM) [95] est employée pour estimer les coefficients de ces modèles. Les paramètres estimés par le 2D-RCA seront utilisés comme entrées pour le K Plus Proches Voisins (K-NN) afin de classer les textures. Le schéma de la méthode proposée est présenté dans la figure 4.1. Dans cette section, nous présentons un algorithme de classification de textures utilisant

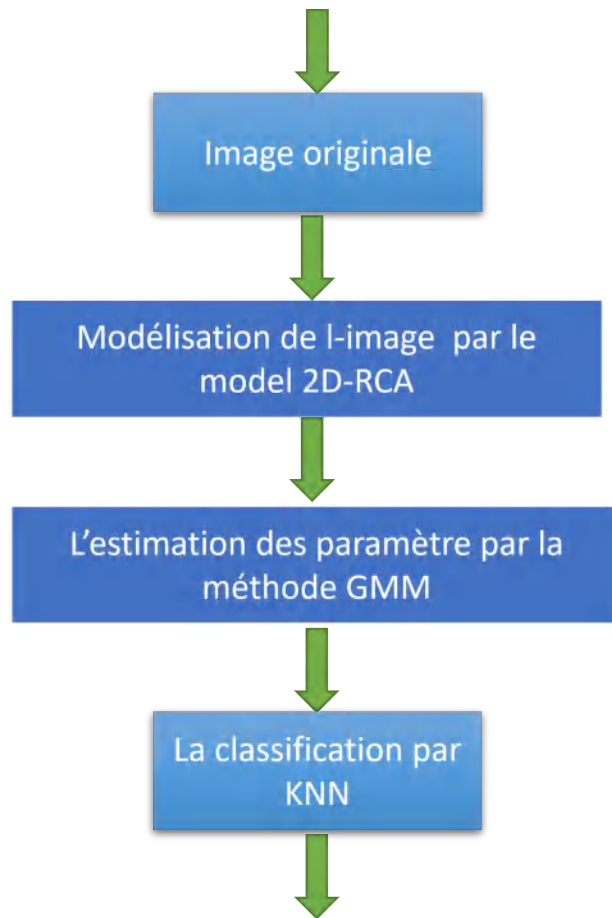


Figure 4.1: Diagramme de la méthode proposée

la modélisation 2D-RCA (modèle autorégressif à coefficients aléatoires bidimensionnel). Nous avons adopté l'approche des K-plus proches voisins (K-NN) comme classifieur pour cette étude.

Les K-plus proches voisins (K-NN) comptent parmi les algorithmes les plus simples en apprentissage automatique pour catégoriser des objets selon les échantillons d'entraînement les plus proches dans l'espace des caractéristiques [105]. Cet algorithme calcule la distance entre l'échantillon de requête et chaque échantillon d'entraînement, puis sélectionne les voisins les plus proches ayant la plus petite distance. Comparativement à d'autres techniques, les K-plus proches voisins sont plus simples à implémenter et ne requièrent pas beaucoup d'entraînement, ce qui les rend particulièrement utiles lorsqu'on travaille avec un petit ensemble de données insuffisant pour un entraînement efficace avec d'autres approches d'apprentissage automatique.

Le principal inconvénient de la méthode des K-plus proches voisins est que la vitesse de calcul de la distance augmente avec le nombre d'échantillons d'entraînement. Le terme 'K' représente le nombre de voisins les plus proches pris en compte lors de la prédiction.

Le plus proche voisin est déterminé par sa proximité avec le pixel étudié.

Par exemple, si $K = 6$, nous prenons en compte les six points les plus proches et utilisons l'étiquette majoritaire parmi ces six points comme prédiction pour l'étiquette. La Figure 4.2 illustre la manière dont les voisins sont estimés à l'aide du K-NN.

L'objectif est de prédire l'étiquette pour le point marqué 'X'. Si $K = 3$, parmi les 3 points voisins de 'X', 2 sont des cercles bleus et 1 est noir. Nous prédirions donc que l'étiquette pour 'X' est un cercle bleu. Si $K = 6$, parmi les 6 points voisins de 'X', 4 sont des cercles noirs et 2 sont bleus. Nous prédirions donc que l'étiquette pour 'X' est un cercle noir.

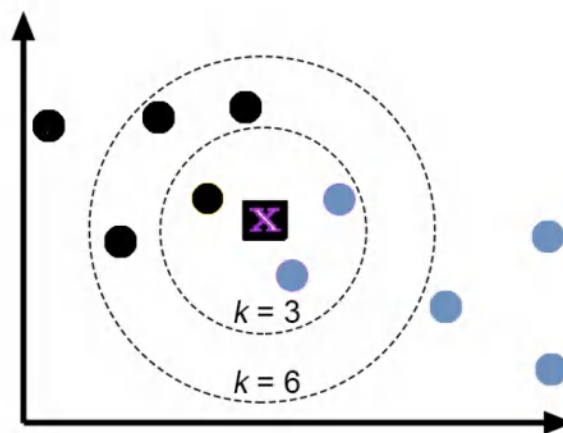


Figure 4.2: k-nn .

À partir de cet exemple, nous pouvons constater que lorsque K varie, l'étiquette prédite change également. Ainsi, K est le paramètre des K -plus proches voisins qui doit être ajusté pour déterminer la valeur optimale. Différentes valeurs de K sont testées sur les données d'entraînement étiquetées, et la meilleure valeur de K est ensuite choisie. Une fois la valeur de K établie, elle peut être utilisée pour prédire les points de données non étiquetés.

L'algorithme proposé se divise en deux sections. Tout d'abord, l'image est modélisée en utilisant le modèle 2D-RCA, puis les paramètres sont estimés à l'aide du processus GMM. Ces paramètres sont ensuite utilisés comme vecteurs caractéristiques pour la classification avec le K-NN

4.2.1 Analyse des resultat

Dans cette section, les résultats des méthodes proposées sont présentés. L'algorithme a été testé sur la base de données Brodatz [106], qui est largement connue. La méthode proposée a été appliquée à différentes images de la base de données d'images USC-SIPI

Algorithme 4.1 : Classification KNN utilisant des modèles 2D_RCA

entrée : IMG image avec N lignes et M colonnes, Nombre de classes c , paramètre k , taille de la fenêtre

sortie : image classifiée

- Extraction des caractéristiques de texture en utilisant le modèle 2D-RCA :
 - Identifier les k classes thématiques pertinentes sur l'image IMG
 - // Classification supervisée, sélection des échantillons pour chaque classe.
 - Pour chaque classe thématique $C_i; i = 1, \dots, k$, sélectionner une fenêtre d'image de taille $n \times n$, puis en utilisant les équations 3.6 et 3.8, estimer les paramètres α, β, γ et σ .
 - Les 4 valeurs obtenues permettent de construire un vecteur caractéristique V_i de dimension 1×4 pour chaque classe thématique $C_i; i = 1, \dots, k$.
 - Étape de classification :
 - Pour chaque pixel pix de IMG, faire :
 - Extraire la fenêtre d'image W_{pix} de taille $n \times n$ centrée sur pix .
 - Construire les vecteurs caractéristiques en suivant les étapes précédentes.
 - Appliquer l'algorithme KNN en utilisant les paramètres estimés du modèle 2D-RCA en tant que vecteur d'entraînement.
-

présentées à la figure (4.3).

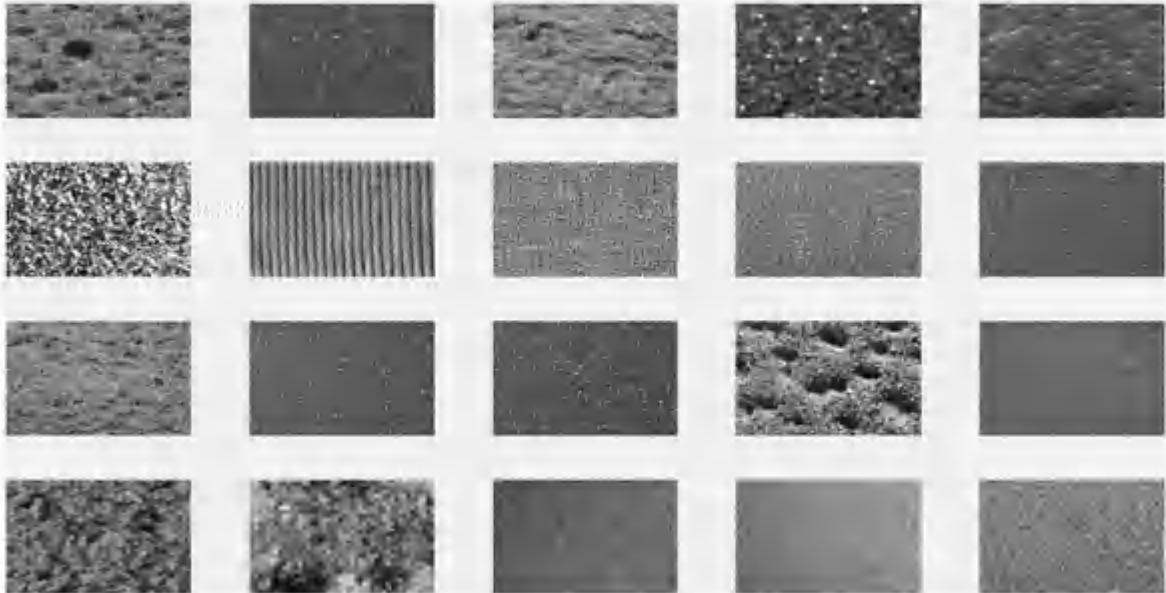


Figure 4.3: brodatz database

La base de données d'images USC-SIPI est une collection d'images numérisées principa-

lement destinée à soutenir la recherche en traitement d'images, analyse d'images et vision par ordinateur. La base de données d'images USC-SIPI a été initialement publiée en 1977 et depuis lors, de nombreuses autres images ont été ajoutées. Elle comprend 155 images en niveaux de gris, dont 130 de taille 512×512 et 25 de taille 1024×1024 .

Dans le modèle de classification K-NN proposé, la meilleure valeur de k optimale est de 2. Les résultats obtenus sont présentés dans la figure (4.4). Différents tests ont été effectués avec une fenêtre de taille 25×25 sur deux et quatre classes avec des régions différentes. Il est remarqué que la plupart des textures fines ont été détectées par l'approche proposée (voir fig. 4.4a).

Pour évaluer les performances de l'approche proposée, les résultats obtenus ont été comparés à ceux donnés par les machines à vecteurs de support (SVM) et l'algorithme basé sur la distance. L'algorithme SVM a utilisé les paramètres 2D-RCA tandis que l'algorithme basé sur la distance a utilisé les quatre premiers paramètres de Haralick, à savoir le contraste, l'homogénéité, la corrélation et l'entropie, extraits de la matrice de co-occurrence.

Les résultats de la classification illustrés à la figure (4.4) montrent que ceux obtenus par la classification K-NN en utilisant les paramètres 2D-RCA sont les meilleurs pour différentes textures par rapport aux autres algorithmes. Le modèle 2D-RCA présente une certaine homogénéité par rapport aux textures, ce qui se traduit par un meilleur taux de classification en fonction de différentes valeurs de K du classifieur K-NN. Ces résultats ont été calculés pour différentes textures et méthodes, et sont présentés dans le Tableau 4.1.

Sur la plupart des images de test, le meilleur taux de classification est donné par l'approche proposée. Pour les différentes images, on observe que la classification des images est visuellement acceptable. Les tests effectués (voir tableau 4.1) ont montré que le meilleur résultat est obtenu par l'algorithme proposé.

Tableau 4.1: Taux de classification de textures
Taux de classification

Images	KNN	SVM	Distance algorithm
image1	91.005	89.89	90.01
image2	90.11	89.84	88.87
image3	90.02	90.11	90
image4	90.1	90	89.8
image5	90	87.51	87.59
image6	94.07	90.02	90

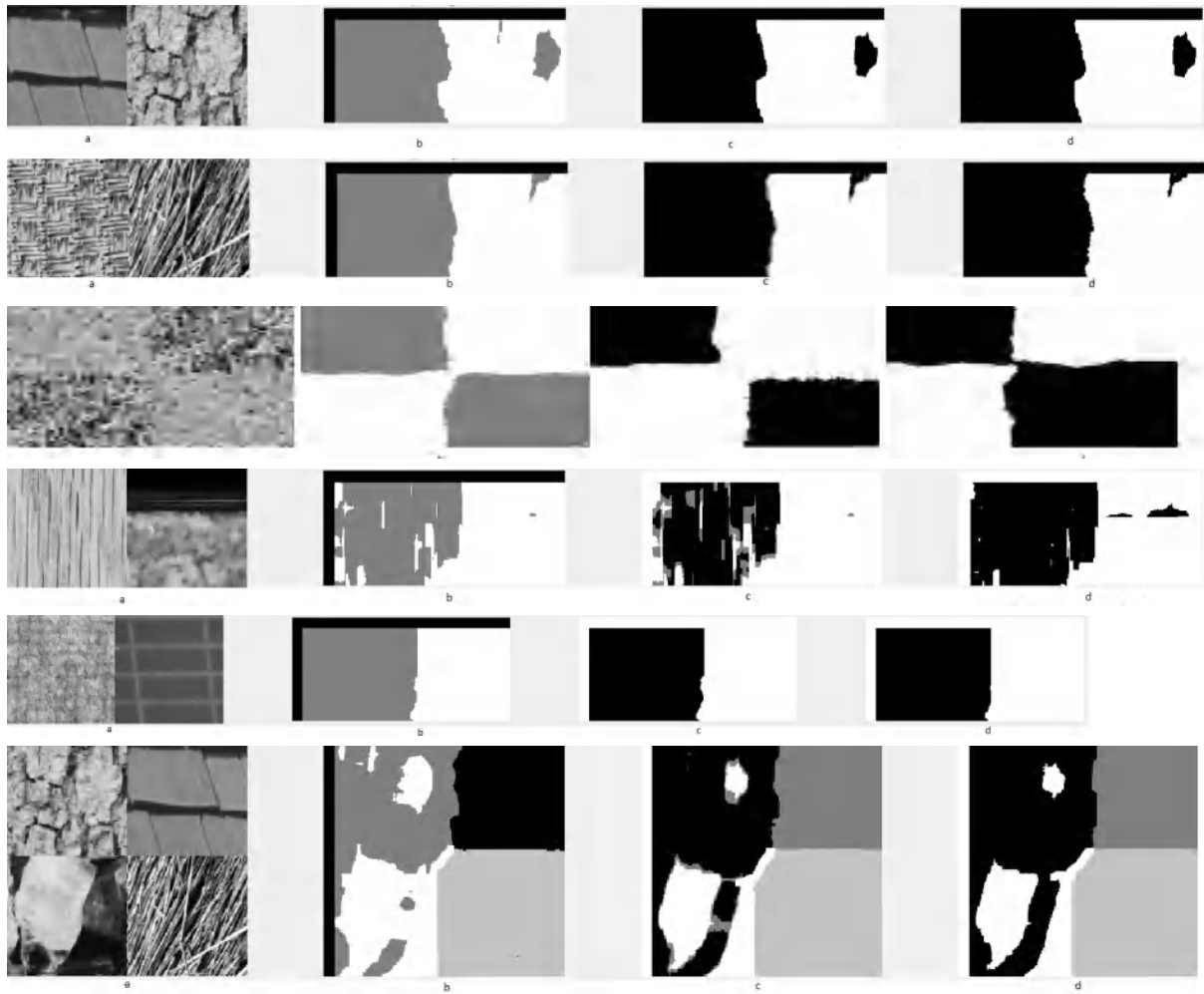


Figure 4.4: (a) images originales (b) algorithmme de résultats 2D-RCA Knn, (c) résultats svm (d) résultats de l’algorithme de distance.

Les résultats de classification illustrés par la figure (4.4) montrent que les résultats obtenus par la classification K-NN, en utilisant les paramètres 2D-RCA, sont les meilleurs pour différentes textures par rapport aux autres algorithmes. Le modèle 2D-RCA présente une certaine homogénéité par rapport aux textures, d’où un meilleur taux de classification a été observé en fonction de différentes valeurs de K en utilisant le classifieur des K-plus proches voisins. Ces résultats ont été calculés pour différentes textures et méthodes et sont présentés dans le Tableau (4.1).

Sur la plupart des images de test, l’approche proposée donne le meilleur taux de classification.

Pour différentes images, on observe que la classification est visuellement acceptable. Les tests effectués (voir Tableau 4.1) ont montré que le meilleur résultat est obtenu par notre algorithme.

4.3 Classification des textures basée sur la construction d'un vecteur de caractéristiques robuste

Dans ce travail [107], différentes méthodes d'extraction de caractéristiques discriminantes ont été envisagées, incluant les approches statistiques, structurales, basées sur des modèles, et sur des graphes. Un vecteur de caractéristiques judicieux a été construit et optimisé à l'aide d'algorithmes génétiques. Étant donné que les caractéristiques utilisées proviennent de méthodes diverses, leurs effets coopèrent pour caractériser une texture, comme illustré dans le tableau 4.2, qui représente les classes de caractéristiques de texture. Cela conduit à un vecteur de caractéristiques optimal. Les caractéristiques locales binaires (LBP), l'histogramme des gradients orientés (HOG), un modèle autorégressif à coefficients aléatoires bidimensionnel (2D-RCA), et le descripteur de Weber (WLD) ont été choisis pour construire notre vecteur de caractéristiques, en raison des diverses informations sur la texture qu'ils fournissent. Des algorithmes de classification multi-classes tels que SVM, K-NN, et Random Forest ont été utilisés pour classer différentes images de texture. Les performances de chaque classifieur utilisant le même vecteur de caractéristiques construit ont été estimées et discutées. Cette étude vise à explorer l'utilité de combiner différentes caractéristiques de texture, incorporant divers types d'informations, pour classer des bases de données d'images.

Tableau 4.2: Classes et méthodes pour l'analyse des textures.

Classes	Méthodes
Approches statistiques	Modèle de motif binaire local (LBP), Descripteur local de Weber
Approches structurales	
Approches basées sur la transformation	Banques de filtres : caractéristiques de texture de Law
Approches basées sur des modèles	Approche basée sur les model 2D-RCA
Approches basées sur des graphes	Structures de graphes locaux
Approches basées sur l'apprentissage	Les CNN

4.3.1 la sélection des caractéristiques par les algorithmes génétiques

Les algorithmes génétiques se sont avérés efficaces pour résoudre des problèmes d'optimisation impliquant des espaces de recherche de grande dimension. Ferri et ses collègues ont introduit l'application des algorithmes génétiques à la résolution de problèmes de sélection de sous-ensembles de caractéristiques [108]. Leur étude a démontré que l'utilisation des algorithmes génétiques est particulièrement adaptée à la sélection de caractéristiques dans des ensembles de taille moyenne (composés de 20 à 49 caractéristiques), où la plupart des méthodes classiques exigent des temps de calcul considérables pour effectuer une sélection.

La quantité d'informations contenue dans le vecteur de caractéristiques est souvent trop importante, ce qui augmente le temps d'exécution et l'espace mémoire requis pour les grandes images. Une solution pour résoudre ce problème consiste à optimiser le nombre de caractéristiques. Ainsi, les caractéristiques non pertinentes sont exclues du vecteur de caractéristiques afin d'éviter la redondance des données et de réduire la consommation de temps et d'espace.

La prochaine étape après l'extraction des caractéristiques consiste à combiner toutes les caractéristiques extraites pour obtenir des résultats plus précis. Les dimensions du vecteur de caractéristiques fusionnées sont de 1×44 pour chaque image. Les caractéristiques à haute dimension augmentent le temps d'exécution du système et les besoins en espace de traitement. Pour cela, des techniques de sélection de caractéristiques sont utilisées pour identifier un sous-ensemble des caractéristiques les plus pertinentes parmi les données non pertinentes, considérant les caractéristiques redondantes comme non pertinentes. Ces caractéristiques fusionnées sont ensuite utilisées dans les algorithmes génétiques pour l'optimisation des caractéristiques [109].

Dans cette étude, une approche basée sur les algorithmes génétiques est utilisée pour la sélection de caractéristiques, avec une fonction de coût basée sur l'erreur quadratique moyenne (MSE). Les meilleures caractéristiques sélectionnées sont utilisées dans tous les classificateurs impliqués dans la méthode présentée pour la classification.

Les composants essentiels des algorithmes génétiques sont :

- **Population initiale de chromosomes :** Soit m le nombre de caractéristiques. La taille de la population est N . Pour créer une population aléatoire P de N

chromosomes, on utilise la formule suivante :

$$P = [C1, C2, \dots, CN]$$

- **Fonction de coût** : L'erreur quadratique moyenne est utilisée pour évaluer l'aptitude de chaque individu de la population.
- **Sélection** : Sélectionner deux parents de la population en fonction de leur meilleure aptitude, qui peuvent générer une nouvelle progéniture. Cela garantit que seules les meilleures solutions les plus aptes sont utilisées pour générer une progéniture.
- **Recombinaison ou croisement** : Recombiner les parents pour former une nouvelle progéniture à partir des chaînes des deux parents, en copiant les bits sélectionnés de chaque parent.
- **Mutation** : Après la réalisation du croisement, muter la nouvelle progéniture à partir d'un seul parent. Cela permet de réduire les optimaux locaux.

4.3.2 la classification de la texture

Les caractéristiques présentées ci-dessus ont été extraites à partir d'images texturées. L'efficacité de chaque caractéristique a été testée à l'aide de trois classificateurs d'apprentissage automatique (SVM multiclasse, K-NN, algorithmes RF). Ensuite, la classification a été réalisée en utilisant la concaténation de ces caractéristiques comme vecteur de données pour chaque classificateur ; la méthode proposée est présentée dans la figure 4.6.

4.3.3 Analyse des résultats

Différents tests ont été effectués sur trois bases de données avec plusieurs classes, notamment la base de données Brodatz [110], KTH-TIPS [111] et UIUC [112].

La base de données Brodatz est une base de données de textures bien connue. Elle est dédiée à la reconnaissance de textures invariantes par rotation. Elle comprend treize catégories de textures issues de l'album initial de Brodatz et contient 1248 images de dimensions 128×128 pixels. Les textures sont fournies avec six angles de rotation différents (0° , 30° , 60° , 90° , 120° , 150°). Le nombre d'échantillons par catégorie et par orientation est de seize images.

KTH est l'abréviation de Kungliga Tekniska Högskolan University and TIPS, qui signifie textures sous éclairage, pose et échelle variables. Les images ont été prises à neuf échelles différentes couvrant deux octaves. À l'échelle significative, la distance entre la caméra et la cible était de 28 cm. Dans cette étude, neuf types différents ont été utilisés, avec 81 images pour chaque classe. Les textures considérées sont du papier de verre, du papier

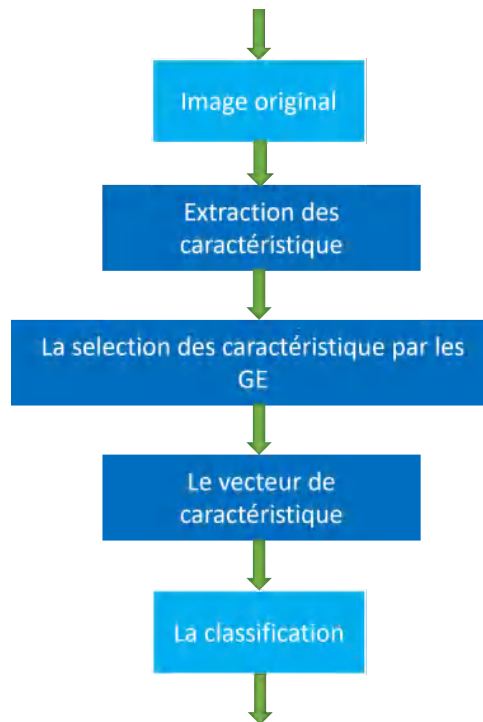


Figure 4.5: Diagramme de la méthode proposée

aluminium froissé, du polystyrène, de l'éponge, du velours côtelé, du lin, du coton, du pain brun, de l'écorce d'orange et des biscuits.

La base de données UIUC contient 25 classes de textures. Chaque classe contient 40 échantillons. Toutes les images sont au format JPG en niveaux de gris avec une résolution de 640×480 pixels. La figure 4.8 montre des exemples de chaque base de données.



Figure 4.6: base de données d'images.

Pour évaluer les performances de la méthode proposée, une technique de validation croisée à 5 plis a été utilisée. Trois bases de données d'images de textures ont servi à constituer

des ensembles d'apprentissage et de test. Chaque pli est divisé en un ensemble d'apprentissage de 80% et un ensemble de test de 20%. L'apprentissage et les tests sont répétés cinq fois, garantissant que chaque point de données se retrouve exactement une fois dans l'ensemble de test de 20%. Le modèle a montré de bonnes performances pour les cinq distorsions.

Les algorithmes génétiques (GA) sont utilisés pour réduire le nombre de caractéristiques. L'ensemble de caractéristiques contenant six caractéristiques est utilisé comme entrées pour trois classifieurs. Une population de 30 chromosomes a été générée de manière aléatoire, chaque chromosome contenant 44 gènes (un gène pour chaque caractéristique). Les opérations génétiques telles que le croisement en un point et la mutation ont été utilisées, avec un taux de croisement de 90% et un taux de mutation de 10%. La méthode de sélection du tournoi a été appliquée pour choisir le pool de reproduction. Les résultats de classification sont présentés en trois phases distinctes. Dans la première phase, la classification est effectuée sur chaque caractéristique individuelle. Puis, dans la deuxième phase, un vecteur de caractéristiques fusionnées est utilisé. La meilleure méthode de sélection des caractéristiques obtenue par les GA est employée. Les résultats de classification ont été obtenus avec des ensembles d'apprentissage et de test de 40/50.

Le tableau 4.3 illustre les performances de chaque caractéristique dans chaque base de données avec plusieurs classifieurs et les résultats de la fusion des caractéristiques de texture. Il est observé que la combinaison des quatre classes de caractéristiques est la méthode la plus efficace, surpassant nettement toutes les autres dans toutes les bases de données. Les résultats de classification pour la sélection des caractéristiques ont également été calculés. Dans le premier test, un rapport apprentissage/test de 40/60 a été sélectionné, atteignant une précision de classification de 98,4%. Il est évident, d'après ce tableau, que la méthode de sélection proposée donne de meilleurs résultats que les deux premières phases.

Une autre approche constructive et utile pour évaluer les performances du classifieur est l'utilisation de la matrice de confusion, qui vise à montrer le nombre d'éléments de la classe A qui sont attribués à la classe B. Par exemple, nous avons choisi de montrer les matrices de confusion pour les trois bases de données.

Nous pouvons observer dans le tableau 4.3 et les figures 4.7 et 4.8 que le meilleur résultat de classification a été obtenu en utilisant le classifieur de forêt aléatoire pour combiner les quatre classes de caractéristiques. Cette supériorité s'explique par plusieurs facteurs :

- Il est inégalé en termes de précision parmi les algorithmes actuels.
- Il fonctionne efficacement sur de vastes bases de données.
- Il peut gérer un grand nombre de facteurs d'information sans nécessiter de suppression de variables.
- Il est plus rapide sur le plan computationnel que d'autres stratégies de regroupe-

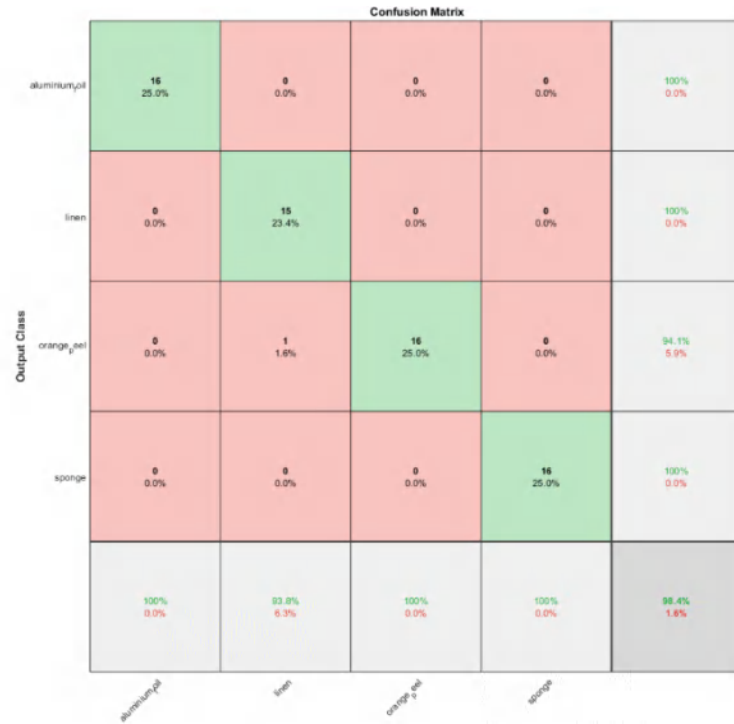


Figure 4.7: matrice de confusion utilisant quatre classes avec classificateurs Rf

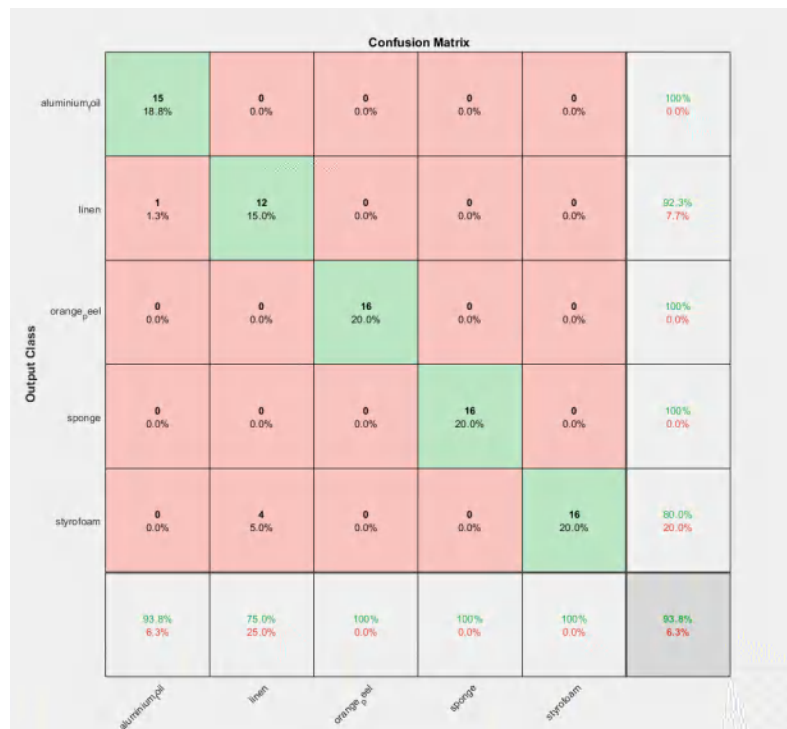


Figure 4.8: matrice de confusion utilisant cinq classes avec classificateurs Rf

ment d'arbres.

Tableau 4.3: Taux de classification

Taux de classification									
features	Brodatz database			Kh tips			UIUC database		
classifiers	SVM	KNN	RF	SVM	KNN	RF	SVM	KNN	RF
FILTER BANKS	77.652	72.45	77.5	79.123	78.463	79.402	79.8	78.45	79.652
LOCAL GRAPH STRUCTURES	78.45	75.785	78.516	78.45	76.45	77.45	78.45	75.45	79.4
2DRCA parameters	89.45	89.45	88.5	89.45	89.45	89.45	73.8	89.45	87.4
HOG	86.75	84.125	86.45	81.5	80.450	82.45	80.4	79.05	82.2
WDT	85.75	81	85.25	82.67	80.125	83.5	79.45	70.75	82.4
LBP	8.45	89.45	87.5	81.02	75.8	80.3	85.8	84.7	85.8
Combination of for features	92.5	88.3	98.2	90.4	80.1	96.32	88.8	84.3	90.75
Combination of for features using gA	94.5	90	99	91.4	80.1	98.4	91.8	86.3	93.75

4.4 Apprentissage par transfert basé sur les réseaux de neurones profonds pour la détection des fissures routières

Les fissures sont l'un des dommages routiers les plus courants et elles peuvent être dangereuses. La détection des fissures est généralement effectuée par des inspecteurs qualifiés ou des ingénieurs structurels. Cependant, il s'agit d'une tâche chronophage, subjective et nécessitant beaucoup de main-d'œuvre. Jusqu'à présent, la plupart des technologies de détection de fissures reposaient sur une inspection manuelle plutôt que sur une identification automatique basée sur l'image, rendant le processus coûteux et chronophage. Dans cet article, nous proposons un système d'analyse automatique des détériorations de chaussée basé sur un cadre d'apprentissage en profondeur par transfert. L'objectif de cette recherche est d'améliorer la précision de classification tout en réduisant le temps d'entraînement. En utilisant quatre architectures de réseaux neuronaux convolutifs (CNN) pré-entraînés, notre système de classification évalue l'apprentissage en profondeur par transfert pour l'extraction de caractéristiques. Ensuite, ces quatre modèles pré-entraînés sont combinés dans ce que nous appelons l'apprentissage par ensemble pour augmenter la précision des prédictions du modèle.

Dans ce chapitre, nous présentons un système de classification automatique conçu pour la détection des fissures routières. Nous utilisons un ensemble de données d'images de fissures de béton capturées dans différents bâtiments du campus de l'Université technique du Moyen-Orient [113]. Cet ensemble de données se compose uniquement de deux types d'images (fissures positives et négatives) à classer. Sur la base de cet ensemble de données, nous avons utilisé quatre architectures de réseaux neuronaux convolutifs (CNN) profonds

pour l'extraction de caractéristiques à partir d'images de route : AlexNet [114], GoogleNet [115], VGG16, VGG19 [55], et Residual Networks ResNet50 [89]. Les scores individuels sont ensuite combinés dans un réseau neuronal profond d'ensemble pour améliorer le score global. Le diagramme de flux de la méthode proposée est représenté à la Figure 4.9. Ce système facilite les interventions des experts et du gouvernement, les aidant à résoudre le problème de classification des fissures routières et à gagner du temps. La précision globale de classification des quatre architectures pré-entraînées est rapportée en fonction du temps d'entraînement et du nombre d'époques. Ce dernier est exploré pour valider son impact sur la réduction du temps de calcul. Les caractéristiques extraites sont réparties sur trois époques. En comparaison avec les travaux connexes, de bons résultats ont été obtenus. De plus, des résultats acceptables ont été obtenus en peu de temps.

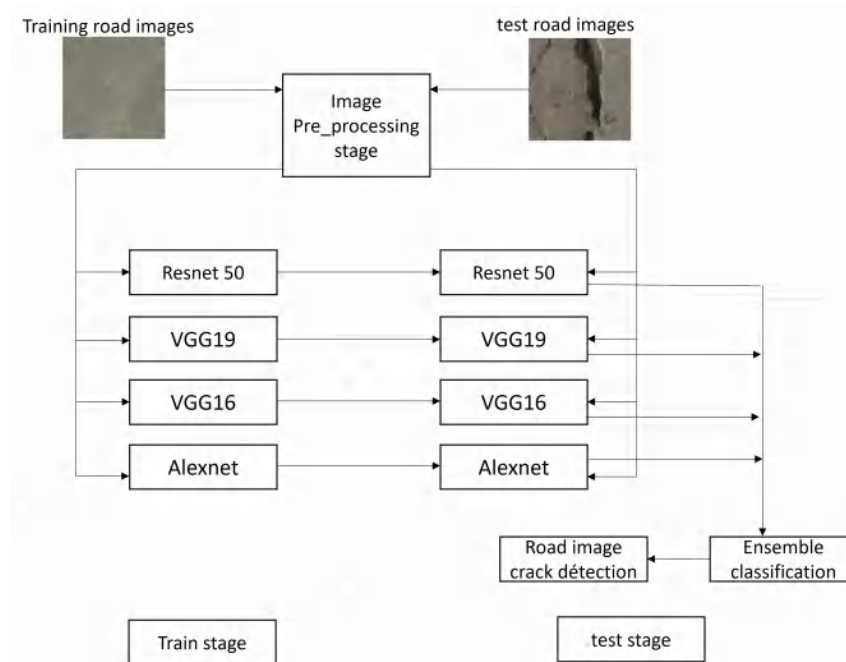


Figure 4.9: L'approche proposée de détection des fissures routières basée sur l'apprentissage profond d'ensemble

Dans cette étude, l'apprentissage par transfert à partir de réseaux de neurones profonds pré-entraînés est utilisé pour la détection des fissures sur les images.

4.4.1 Architectures de CNN pré-entraînées pour la classification d'images

Les architectures de réseaux de neurones convolutifs (CNN) sont construites en utilisant plusieurs blocs pour apprendre les hiérarchies spatiales de caractéristiques : des couches

de convolution équipées d'un ensemble de filtres, des couches de *pooling*, et des couches entièrement connectées (FCLs). La compétition ILSVRC a conduit à la création d'architectures profondes significatives à partir de 2012. L'erreur de classification des gagnants de cette compétition a diminué de 15,3% avec AlexNet (2012) [114] à 2,251% avec SE-Net (2017) [116]. Le nombre de couches a également été augmenté, passant de 8 dans AlexNet à 152 dans des architectures plus récentes. Le Tableau 4.4 compare l'erreur de classification, le nombre de couches, les tâches effectuées, l'environnement d'exécution et les ensembles de données d'entraînement de ces architectures.

Tableau 4.4: Comparaison de différentes architectures CNN pour l'analyse d'images

ILSVRC architectures	Nombre de couches	Les cinq principaux taux d'erreur	Tâches	Dataset	Environnement d'exécution
AlexNet	8	15.3%	Classification	ImageNet	Two GTX 580 GPUs 3Gg (parallel)
Resnet	18-34-50-152	3.57%	Classification	imageNet COCO	Two GPUs
GoogleNet 1	22	6.67%	Detection Classification	ImageNet	CPU
VGGNet	16 – 19	6.8%	Localization Classification	ImageNet	Four NVIDIA

4.4.1.1 AlexNet

L'architecture AlexNet [114] est nettement plus complexe et étendue que celle de LeNet [86]. Elle comporte huit couches, majoritairement des couches de convolution, suivies de couches de max-pooling et de trois couches entièrement connectées. La sortie est une fonction softmax à 1000 classes. Elle est entraînée sur deux GPU GTX 580 de 3 Go qui communiquent uniquement dans certaines couches, réduisant ainsi les taux d'erreur pour les cinq meilleures prédictions.

4.4.1.2 GoogleNet

L'architecture GoogleNet, initialement appelée Inception-v1, est conçue pour une utilisation plus efficace des ressources informatiques [115]. Cette architecture, comprenant le module Inception, utilise une fonction d'activation linéaire rectifiée, une couche de pooling moyenne, et élimine la couche entièrement connectée traditionnelle. Des améliorations successives par Ioffe et Szegedy [88] ont mené à Inception-v2, avec une réduction du taux d'erreur à 4,82%. Szegedy et al. [117] ont ensuite développé Inception-v3, qui a réduit encore l'erreur à 3,5%.

4.4.1.3 ResNet/Inception-v4

ResNet utilise l'apprentissage résiduel pour faciliter l'entraînement de réseaux profonds et améliorer leur précision avec l'augmentation de la profondeur [89]. Cette architecture est proposée en plusieurs variantes avec 18, 34, 50, 101, et 152 couches, la version 152 couches étant la plus performante. En combinant les architectures Inception et ResNet, Szegedy

et al. [90] ont évalué Inception-ResNet-v1 et Inception-v4, notant des performances comparables à celles de InceptionNet avec une structure moins complexe.

4.4.1.4 VGGNet

Karen Simonyan et al. [55] ont étudié l'impact de la profondeur d'un réseau de neurones convolutionnel sur l'exactitude de la reconnaissance d'images. Ils ont utilisé des filtres de convolution très petits (3×3) pour augmenter la profondeur des architectures VGGNet qui comportent entre 11 et 19 couches de poids. Les meilleures performances ont été obtenues avec les configurations VGG16 et VGG19, respectivement avec 16 et 19 couches de poids. L'erreur de classification diminue avec l'augmentation de la profondeur jusqu'à atteindre 19 couches, où elle se stabilise.

4.4.1.5 Paramètres d'apprentissage par transfert

La mise en place de l'apprentissage par transfert consiste à appliquer les connaissances acquises sur un problème spécifique à d'autres problèmes similaires en utilisant un modèle pré-entraîné. AlexNet, VGG16, VGG19, et ResNet-50, qui comprennent 1000 classes, sont utilisés pour évaluer leur précision de classification avec 1,28 million d'images d'entraînement, 100 000 images de test et 50 000 images de validation. Ils sont testés pour leur précision dans la classification d'objets.

4.4.2 Méthodes d'ensemble

L'apprentissage en ensemble, ou ensemble de modèles, utilise divers algorithmes d'apprentissage automatique pour améliorer les performances prédictives [118]. Cette technique est souvent plus précise que les prédictions des modèles individuels. Nous utilisons des stratégies simples pour combiner les prédictions de différents réseaux neuronaux profonds, comme la moyenne simple des sorties softmax, une moyenne pondérée basée sur les performances des modèles, et le vote majoritaire pour déterminer la classe la plus fréquemment prédite.

4.5 Test et Résultats

4.5.1 Bases de données

Pour les tâches de classification et de segmentation, nous utilisons un ensemble de données open-source d'images de fissures de béton provenant de l'Université technique du Moyen-Orient [113] (voir Tableau 4.5). Pour la tâche de classification, un total de 40 000 images

de dimensions 227×227 pixels ont été générées à partir de 458 images originales. Ces images sont réparties de manière équilibrée entre les classes "fissure" et "non-fissure" (voir Fig.4.10). Les images fournies offrent une haute qualité de surface et présentent une large gamme de conditions d'éclairage, sans aucune manipulation telle que la rotation ou le retournement pour augmenter artificiellement la variabilité des données.



Figure 4.10: image database

L'ensemble de données d'images a été divisé de manière aléatoire en 5 ensembles de tailles égales ; 4 ensembles pour l'entraînement, tandis que le reste est utilisé pour les tests.

Tableau 4.5: Ensemble de données d'images de fissures

Tâche	Nombre d'images	la taille	fissure	Non-fissure	Train	Dev set	Test
Classification	40,000	227×227	20,000	20,000	32,000	400	

4.5.2 Prédiction de l'exactitude de la classification

Les images de test ont été classifiées à l'aide des réseaux préalablement entraînés et l'exactitude totale de la classification a été calculée. L'exactitude de la classification, qui mesure la capacité d'un modèle à prédire correctement la valeur d'un attribut pour de nouvelles données, est définie comme le rapport des vrais positifs (VP) et des vrais négatifs (VN) par rapport au nombre total d'essais :

$$\text{Exactitude} = \frac{VP + VN}{VP + VN + FN + FP} \times 100 \quad (4.1)$$

où VP et VN représentent respectivement les résultats corrects pour les classes positives et négatives, tandis que FN et FP représentent les erreurs pour les classes positives et négatives.

Tableau 4.6: Taux classification pour différentes époques en utilisant différentes architectures d'apprentissage par transfert

Architectures	Epoch = 30	Epoch = 50	Epoch = 100
AlexNet	94.5	95	95.2
GoogleNet	94.5	95	94.8
VGG-16	95.5	96.1	97
VGG-19	94.5	95	96.2
ResNet-50	96.5	95	97

Tableau 4.7: Taux de classification en utilisant la méthode d'ensemble pour différentes époques

Architectures	Epoch = 30	Epoch = 50	Epoch = 100
ensemble	96.25	97	98

Les classifieurs sont entraînés avec une taille de lot de 64. Pendant 100 époques, les classifieurs VGG16 et ResNet-50 atteignent presque 0,97 lors de l'entraînement, tandis que le classifieur basé sur VGG19 atteint un maximum de 0,965 (voir Fig. 4.12). Ce schéma est également observé lors des tests. Sur l'ensemble des 4000 images de test, les classifieurs basés sur VGG16 et Inception-v3 obtiennent des scores de classification presque parfaits, avec seulement 8 faux positifs (FP) ou faux négatifs (FN) ou moins (voir Fig. 4.12). En revanche, avec environ 110 FP et 12 FN, le classifieur basé sur AlexNet montre une précision inférieure à celle des deux autres classifieurs (voir Fig. 4.13 et Tableau 4.6). Sur la base des résultats présentés dans le Tableau 4.6, il a été observé que la majorité des réseaux pré-entraînés donnent de bons résultats avec 100 époques, atteignant une exactitude de classification de 96,2%, 97% et 97% pour VGG16, VGG19 et ResNet-50, respectivement. D'après le Tableau 4.7, l'ensemble des CNN proposés permet une meilleure détection des fissures routières, atteignant une exactitude globale de 98%.

La courbe ROC pour les résultats de classification obtenus avec les méthodes d'ensemble suggérées est présentée dans la Figure 4.12. Les cas positifs dans les images routières ont été classés comme vrais positifs (TP) et les cas négatifs comme négatifs corrects (TN) pour créer la courbe ROC. Il est à noter que l'exactitude des instances positives est de 1,00, indiquant que tous les cas positifs ont été correctement identifiés, soit une précision de 100%. Les méthodes d'ensemble affichent également une aire sous la courbe (AUC) de 1,00, démontrant ainsi une performance de classification parfaite.

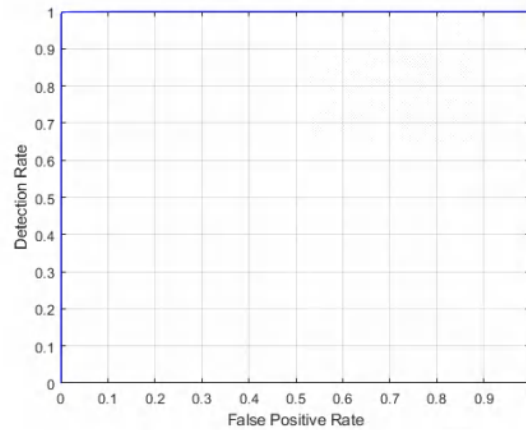


Figure 4.12: ROC Curve

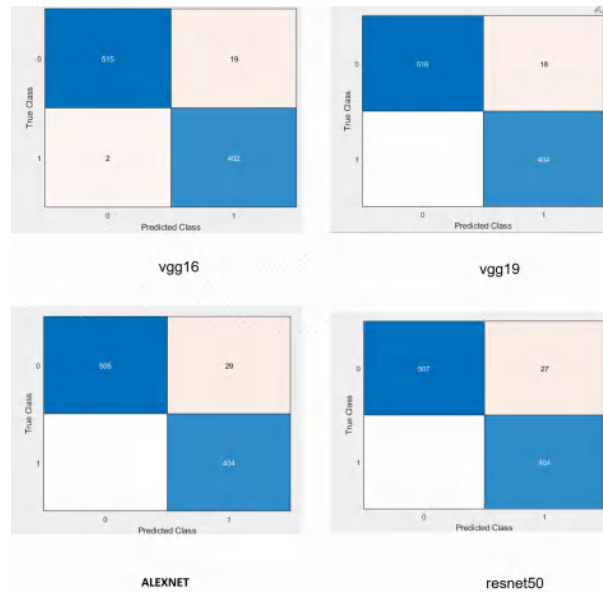


Figure 4.13: Confusion matrices

4.6 conclusion

Dans ce chapitre, nous avons présenté les méthodes de classification des textures ainsi que les résultats obtenus par ces méthodes. La classification à l'aide des paramètres des modèles autorégressifs à coefficient aléatoire (2D-RCA) en utilisant les KNN a démontré son efficacité et sa robustesse, notamment dans le cas de bases de données de petite taille, surpassant ainsi les méthodes basées sur les SVM. La classification des textures, en particulier dans de grandes bases de données, nécessite la sélection des caractéristiques optimales pour capturer l'ensemble des informations. Notre méthode, basée sur la sélection de caractéristiques à l'aide d'un algorithme génétique, s'est avérée performante

pour la classification des textures, produisant de bons résultats sur trois bases de données différentes. De plus, nous avons constaté que la classification à l'aide d'arbres aléatoires surpassait celle basée sur les SVM et les KNN. Dans le cas des images de fissures routières enrichies en textures, notre méthode, fondée sur le transfert d'apprentissage, a été en mesure de détecter toutes les fissures, obtenant des résultats supérieurs à l'utilisation directe de CNN.

Conclusion Générale

Dans cette thèse, nous avons exploré le problème de la synthèse et de la classification des textures, un défi qui demeure persistant malgré de nombreuses recherches dans ce domaine. Bien que l'intelligence artificielle ait connu un grand succès dans divers secteurs du traitement d'images, elle n'a pas encore exploité pleinement son potentiel en matière de synthèse de textures. Les méthodes traditionnelles se sont avérées limitées pour capturer l'ensemble des informations texturées et ne réussissent qu'à générer certains types de textures. Pour notre première approche, nous avons développé une méthode basée sur l'apprentissage par transfert pour la synthèse de textures. Celle-ci a produit des résultats supérieurs à ceux des premières méthodes fondées sur les réseaux de neurones convolutifs (CNN), mais elle n'a pas complètement résolu la complexité de la tâche ni réussi à capturer toutes les informations texturées. Notre deuxième méthode, qui utilise des modèles autorégressifs, a réussi à synthétiser tous les types de textures tout en conservant toutes les informations des images, surpassant ainsi les méthodes existantes, y compris celles basées sur l'intelligence artificielle. Cette méthode s'est avérée supérieure, confirmée visuellement et par des métriques de validation. La classification des textures reste une étape cruciale pour l'exploitation des images, nécessitant la recherche de méthodes efficaces pour caractériser ces textures. Pour améliorer la qualité de la classification, nous avons développé trois approches distinctes. Premièrement, nous avons utilisé les coefficients du modèle autorégressif à coefficients aléatoires comme caractéristiques pour le classifieur KNN, ce qui a donné de bons résultats pour des petites images, bien que moins efficace pour de grandes bases de données que le SVM. Deuxièmement, nous avons proposé un vecteur de caractéristiques robuste contenant plusieurs classes de caractéristiques pertinentes. En appliquant divers algorithmes de classification, nous avons obtenu des résultats supérieurs à ceux des approches utilisant une seule classe de caractéristiques.

En conclusion, notre travail a contribué à des avancées significatives dans la synthèse et la classification des textures, ouvrant ainsi de nouvelles perspectives pour l'intelligence artificielle dans le domaine du traitement d'images.

Bibliographie

- [1] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001.
- [2] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488, 2000.
- [3] Adib Akl, Charles Yaacoub, Marc Donias, Jean-Pierre Da Costa, and Christian Germain. Structure tensor based synthesis of directional textures for virtual material design. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4867–4871. IEEE, 2014.
- [4] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28, 2015.
- [5] Merriam-webster’s collegiate online dictionary. <http://https://www.merriam-webster.com/>. Accessed : 2010-09-30.
- [6] Robert M Haralick et al. Statistical and structural approaches to texture, 1979.
- [7] Michael Unser. *Description statistique de textures : application à l’inspection automatique*. PhD thesis, EPFL, 1984.
- [8] Christian Germain. *Contribution à la caractérisation multi-échelle de l’anisotropie des images texturées*. PhD thesis, Université Sciences et Technologies-Bordeaux I, 1997.
- [9] Paul S Heckbert. Survey of texture mapping. *IEEE computer graphics and applications*, 6(11) :56–67, 1986.
- [10] Hitoshi Yamauchi, Jörg Haber, and H-P Seidel. Image restoration using multiresolution texture synthesis and image inpainting. In *Proceedings Computer Graphics International 2003*, pages 120–125. IEEE, 2003.

- [11] Radu-Drăgoș Urs, Jean-Pierre Da Costa, and Christian Germain. Maximum-likelihood based synthesis of volumetric textures from a 2d sample. *IEEE transactions on image processing*, 23(4) :1820–1830, 2014.
- [12] Martin Köppel, Xi Wang, Dimitar Doshkov, Thomas Wiegand, and Patrick Ndjiki-Nya. Depth image-based rendering with spatio-temporally consistent texture synthesis for 3-d video with global motion. In *2012 19th IEEE International Conference on Image Processing*, pages 2713–2716. IEEE, 2012.
- [13] Jean-Pierre Da Costa and Christian Germain. Synthesis of solid textures based on a 2d example : application to the synthesis of 3d carbon structures observed by transmission electronic microscopy. In *Image Processing : Machine Vision Applications III*, volume 7538, pages 295–304. SPIE, 2010.
- [14] Rafael Fonolla Navarro and Javier Portilla. Robust method for texture synthesis-by-analysis based on a multiscale gabor scheme. In *Human Vision and Electronic Imaging*, volume 2657, pages 86–97. SPIE, 1996.
- [15] David S Ebert, F Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing & modeling : a procedural approach*. Morgan Kaufmann, 2003.
- [16] Steven Worley. A cellular texture basis function. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 291–294, 1996.
- [17] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel. Random phase textures : Theory and synthesis. *IEEE T image process*, 20(1) :257–267, 2010.
- [18] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision : Third International Conference, SSVM 2011, Ein-Gedi, Israel, May 29–June 2, 2011, Revised Selected Papers 3*, pages 435–446. Springer, 2012.
- [19] Javier Portilla and Eero P Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision*, 40 :49–70, 2000.
- [20] Xin Tong, Jingdan Zhang, Ligang Liu, Xi Wang, Baining Guo, and Heung-Yeung Shum. Synthesis of bidirectional texture functions on arbitrary surfaces. *ACM Transactions on Graphics (ToG)*, 21(3) :665–672, 2002.
- [21] Cecilia Aguerrebere, Yann Gousseau, and Guillaume Tartavel. Exemplar-based texture synthesis : the efros-leung algorithm. *Image Processing On Line*, 2013 :223–241, 2013.

- [22] Lara Raad, Axel Davy, Agnès Desolneux, and Jean-Michel Morel. A survey of exemplar-based texture synthesis. *Annals of Mathematical Sciences and Applications*, 3(1) :89–148, 2018.
- [23] Kris Papat and Rosalind W Picard. Novel cluster-based probability model for texture synthesis, classification, and compression. In *Visual Communications and Image Processing'93*, volume 2094, pages 756–768. SPIE, 1993.
- [24] Rupert Paget and I Dennis Longstaff. Texture synthesis via a noncausal nonparametric multiscale markov random field. *IEEE transactions on image processing*, 7(6) :925–931, 1998.
- [25] Allen Gersho and Robert M Gray. *Vector quantization and signal compression*, volume 159. Springer Science & Business Media, 2012.
- [26] Michael Ashikhmin. Synthesizing natural textures. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 217–226, 2001.
- [27] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. *Image Analogies*. Association for Computing Machinery, New York, NY, USA, 1 edition, 2023.
- [28] Li Yi Wei. Texture synthesis by fixed neighborhood searching. phd, 2001.
- [29] Steve Zelinka and Michael Garland. Towards real-time texture synthesis with the jump map. *Rendering Techniques*, 2002 :13th, 2002.
- [30] Yang Lu, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. Exploring generative perspective of convolutional neural networks by learning random field models. *Statistics and Its Interface*, 11(3) :515–529, 2018.
- [31] Y Xu, B Guo, and HY Shum. Fast and memory efficient texture synthesis. *Microsoft Research, Tech. Rep. MSR-TR-2000-32*, 2000.
- [32] Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics (ToG)*, 20(3) :127–150, 2001.
- [33] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures : Image and video synthesis using graph cuts. *Acm transactions on graphics (tog)*, 22(3) :277–286, 2003.
- [34] Gabriel Peyré. Texture synthesis with grouplets. *IEEE transactions on pattern analysis and machine intelligence*, 32(4) :733–746, 2009.
- [35] Josef Bigun. Optimal orientation detection of linear symmetry, 1987.

- [36] David J Heeger and James R Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238, 1995.
- [37] Adib Akl. *Analyse/synthèse de champs de tenseurs de structure : application à la synthèse d’images et de volumes texturés*. PhD thesis, Université de Bordeaux, 2016.
- [38] Adib Akl, Charles Yaacoub, Marc Donias, Jean-Pierre Da Costa, and Christian Germain. Synthesis of arbitrary-shaped textures constrained by the structure tensor field. *Signal, Image and Video Processing*, 12 :41–49, 2018.
- [39] JS DeBonet. Multiresolutionsamplingprocedurefor analysisandsynthesisoftextureimages. *Proceedings of SIGGRAPH’97*, pages 362–368, 1997.
- [40] Jarke J Van Wijk. Spot noise texture synthesis for data visualization. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 309–318, 1991.
- [41] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel. Micro-texture synthesis by phase randomization. *Image processing on line*, 1 :213–237, 2011.
- [42] Charles Bordenave, Yann Gousseau, and Francois Roueff. The dead leaves model : a general tessellation modeling occlusion. *Advances in applied probability*, 38(1) :31–46, 2006.
- [43] Bruno Galerne, Arthur Leclaire, and Lionel Moisan. A texton for fast and flexible gaussian texture synthesis. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, pages 1686–1690. IEEE, 2014.
- [44] Guillaume Tartavel, Yann Gousseau, and Gabriel Peyré. Variational texture synthesis with sparsity and spectrum constraints. *Journal of Mathematical Imaging and Vision*, 52 :124–144, 2015.
- [45] Rama Chellappa and RL Kashyap. Texture synthesis using 2-d noncausal autoregressive models. *IEEE T Acoust Speech*, 33(1) :194–203, 1985.
- [46] Michal Haindl. Texture synthesis. *Department of Computer Science [CS]*, (R 9139), 1991.
- [47] Youssef Stitou, Flavius Turcu, Yannick Berthoumieu, and Mohamed Najim. Three-dimensional textured image blocks model based on wold decomposition. *IEEE Transactions on Signal Processing*, 55(7) :3247–3261, 2007.
- [48] Rama Chellappa, Rangasami L Kashyap, and Bangalore S Manjunath. Model-based texture segmentation and classification. In *Handbook of pattern recognition and computer vision*, pages 277–310. World Scientific, 1993.

- [49] Heinz-Otto Peitgen and Dietmar Saupe. *The science of fractal images*. Springer-Verlag, 1988.
- [50] George R Cross and Anil K Jain. Markov random field texture models, 1983.
- [51] Marianna Clark, Alan C Bovik, and Wilson S Geisler. Texture segmentation using gabor modulation/demodulation. *Pattern Recogn let*, 6(4) :261–267, 1987.
- [52] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patch-match : A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3) :24, 2009.
- [53] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. Summarizing visual data using bidirectional similarity. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.
- [54] Kaiming He and Jian Sun. Statistics of patch offsets for image completion. In *Computer Vision–ECCV 2012 : 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II 12*, pages 16–29. Springer, 2012.
- [55] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*, 2014.
- [56] Ivan Ustyuzhaninov, Wieland Brendel, Leon A Gatys, and Matthias Bethge. Texture synthesis using shallow convolutional networks with random filters. *arXiv preprint arXiv :1606.00021*, 2016.
- [57] Dafydd Ravenscroft. Deep learning methods for texture analysis in medical imaging. 2018.
- [58] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.
- [59] Luiz G Hafemann, Luiz S Oliveira, Paulo R Cavalin, and Robert Sabourin. Transfer learning between texture classification tasks using convolutional neural networks. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2015.
- [60] Zi-Ming Wang, Meng-Han Li, and Gui-Song Xia. Conditional generative convnets for exemplar-based texture synthesis. *IEEE T Image Process*, 30 :2461–2475, 2021.
- [61] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *International Conference on Machine Learning*, pages 2635–2644. PMLR, 2016.

- [62] Silvano Di Zenzo. A note on the gradient of a multi-image. *Computer vision, graphics, and image processing*, 33(1) :116–125, 1986.
- [63] Radu Dragos Urs. *Non-parametric synthesis of volumetric textures from a 2D sample*. PhD thesis, Université Sciences et Technologies-Bordeaux I, 2013.
- [64] Amel Boulemnadjel, Fella Hachouf, and Soumia Kharfouchi. Gmm estimation of 2d-rca models with applications to texture image classification, 2015.
- [65] Andrzej Materka, Michal Strzelecki, et al. Texture analysis methods—a review. *Technical university of lodz, institute of electronics, COST B11 report, Brussels*, 10(1.97) :4968, 1998.
- [66] Sébastien Mavromatis. *Analyse de texture et visualisation scientifique*. PhD thesis, Aix-Marseille 2, 2001.
- [67] Kenneth I Laws. Rapid texture identification. In *Image processing for missile guidance*, volume 238, pages 376–381. Spie, 1980.
- [68] Mahamadou Idrissa and Marc Acheroy. Texture classification using gabor filters. *Pattern Recognition Letters*, 23(9) :1095–1102, 2002.
- [69] Jaideva C Goswami and Andrew K Chan. *Fundamentals of wavelets : theory, algorithms, and applications*. John Wiley & Sons, 2011.
- [70] Mausumi Acharyya and Malay K Kundu. An adaptive approach to unsupervised texture segmentation using m-band wavelet transform. *Signal Processing*, 81(7) :1337–1356, 2001.
- [71] Robert M Haralick, Karthikeyan Shanmugam, and Its’ Hak Dinstein. Textural features for image classification. *IEEE T systems man*, (6) :610–621, 1973.
- [72] Jacky Desachy. *Contribution à l’étude de la texture en télédétection*. PhD thesis, 1980.
- [73] Robert M. Haralick, Karthikeyan Shanmugam, and Its’ Hak Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6) :610–621, 1973. Publisher : Ieee.
- [74] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1) :51–59, 1996.
- [75] Sergios Theodoridis. Konstantinos koutroumbas pattern recognition. In *Beijing : Mechanical*, pages 163–205. Industrial Press, 2003.
- [76] Anil K Jain and Sateesha G Nadabar. Markov random field applications in image analysis. In *Data Analysis in Astronomy IV*, pages 39–50. Springer, 1992.

- [77] Dalila Benboudjema. Champs de markov triplets et segmentation bayésienne non supervisée d’images. *Optimisation et sûreté des systèmes, Université de Technologie de Troyes-Institut National des Télécommunications*, 2005.
- [78] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.
- [79] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.
- [80] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.
- [81] Leo Breiman. Random forests. *Machine learning*, 45 :5–32, 2001.
- [82] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3) :211–252, 2015.
- [83] Ayoub Maarouf Abderrazak and Fella Hachouf. Texture synthesis using improved transfer learning. In *2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pages 1–5. IEEE, 2022.
- [84] Leon Alexander Gatys. *Texture synthesis and style transfer using perceptual image representations from convolutional neural networks*. PhD thesis, Eberhard Karls Universität Tübingen, 2017.
- [85] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks : Tricks of the trade*, pages 9–48. Springer, 2012.
- [86] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.
- [87] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.
- [88] Sergey Ioffe and Christian Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.
- [89] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- [90] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.
- [91] Diederik P Kingma and Jimmy Ba. Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*, 2014.
- [92] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.
- [93] Ayoub Abderrazak Maarouf, Fella Hachouf, and Soumia Kharfouchi. Exemplar-based texture synthesis using two random coefficients autoregressive models. *Image Analysis and Stereology*, 42(1) :37–49, 2023.
- [94] S KHarfouchi. *Statistical inference In spatial bilinear processes*. PhD thesis, PhD thesis, Université Mentouri Constantine, 2012.
- [95] Harry H Kelejian and Ingmar R Prucha. A generalized moments estimator for the autoregressive parameter in a spatial model, 1999.
- [96] Mihran Tuceryan and Anil K Jain. Texture analysis. In *Handbook of pattern recognition and computer vision*, pages 235–276. World Scientific, 1993.
- [97] Omry Sendik and Daniel Cohen-Or. Deep correlations for texture synthesis. *ACM Trans Graph (ToG)*, 36(5) :1–15, 2017.
- [98] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment : from error visibility to structural similarity. *IEEE T Image Process*, 13(4) :600–612, 2004.
- [99] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim : A feature similarity index for image quality assessment. *IEEE T Image Process*, 20(8) :2378–2386, 2011.
- [100] Zhou Wang and Alan C Bovik. A universal image quality index. *IEEE signal proc let*, 9(3) :81–84, 2002.
- [101] Alexandre Kaspar, Boris Neubert, Dani Lischinski, Mark Pauly, and Johannes Kopf. Self tuning texture optimization. In *Comput Graph Forum*, volume 34, pages 349–359. Wiley Online Library, 2015.
- [102] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv :1508.06576*, 2015.
- [103] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. Cooperative training of descriptor and generator networks. *IEEE T Pattern Anal*, 42(1) :27–45, 2018.

- [104] Anne Humeau-Heurtier. Texture feature extraction methods : A survey. *IEEE Access*, 7 :8975–9000, 2019.
- [105] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2) :1883, 2009.
- [106] Phil Brodatz. *Textures : a photographic album for artists and designers*. Dover Pubns, 1966.
- [107] AYOUB ABDERRAZAK MAAROUF and FELLA HACHOUF. Robust characteristics for texture classification. 2021.
- [108] FJ Ferri, V Kadiramanathan, and J Kittler. Feature subset search using genetic algorithms. In *IEE/IEEE Workshop on Natural Algorithms in Signal Processing*, 1993.
- [109] David E. Golberg. Genetic algorithms in search, optimization, and machine learning. *Addion wesley*, 1989(102) :36, 1989.
- [110] FJ Díaz-Pernas, Miriam Antón-Rodríguez, JF Díez-Higuera, Mario Martínez-Zarzuela, D González-Ortega, and Daniel Boto-Giralda. Texture classification of the entire brodatz database through an orientational-invariant neural architecture. In *International Work-Conference on the Interplay Between Natural and Artificial Computation*, pages 294–303. Springer, 2009.
- [111] Barbara Caputo, Eric Hayman, and P Mallikarjuna. Class-specific material categorisation. In *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, volume 2, pages 1597–1604. IEEE, 2005.
- [112] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. A sparse texture representation using local affine regions. *IEEE transactions on pattern analysis and machine intelligence*, 27(8) :1265–1278, 2005.
- [113] FO Çağlar and R Özgenel. Concrete crack images for classification. *Mendeley Data*, 2, 2019.
- [114] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25 :1097–1105, 2012.
- [115] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [116] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

- [117] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
- [118] Omer Sagi and Lior Rokach. Ensemble learning : A survey. *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, 8(4) :e1249, 2018.