N° d'ordre NNT : 2020LYSE2063

THESE de DOCTORAT DE L'UNIVERSITÉ DE LYON

Opérée au sein de

L'UNIVERSITÉ LUMIÈRE LYON 2

École Doctorale : ED 512 Informatique et Mathématiques

Discipline : Informatique

Soutenue publiquement le 19 octobre 2020, par :

Rémi RATAJCZAK

Analyse automatique d'images aériennes historiques : application à une étude épidémiologique.

Devant le jury composé de :

Christophe COLLET, Professeur des universités, Université de Strasbourg, Président

Sébastien LEFEVRE, Professeur des universités, Université de Bretagne Sud, Rapporteur

Nicolas TOGNE, Professeur des universités, Conservatoire National des Arts et Métiers, Rapporteur

Florence TUPIN, Professeure, Télécom Paris Tech, Examinatrice

Carlos Fernando CRISPRIM JUNIOR, Maître de conférences, Université Lumière Lyon 2, Examinateur

Clément MALLET, Ingénieur Divisionnaire Université Gustave Eiffel, Examinateur

Laure TOUGNE, Professeure des universités, Université Lumière Lyon 2, Directrice de thèse

Béatrice FERVERS, Professeure associée, Centre Régional Lutte contre le Cancer L. Bérard, Co-Directrice de thèse

Contrat de diffusion

Ce document est diffusé sous le contrat *Creative Commons* « Paternité – pas d'utilisation commerciale – pas de modification » : vous êtes libre de le reproduire, de le distribuer et de le communiquer au public à condition d'en mentionner le nom de l'auteur et de ne pas le modifier, le transformer, l'adapter ni l'utiliser à des fins commerciales.

UNIVERSITÉ LUMIÈRE LYON 2
École Doctorale Informatique et Mathématiques de Lyon

THÈSE

pour obtenir le grade de

DOCTEUR EN INFORMATIQUE

présentée et soutenue par

Rémi RATAJCZAK

le 19 octobre 2020

Analyse automatique d'images aériennes historiques : application à une étude épidémiologique

Directrices de thèse : **Laure TOUGNE , BÉATRICE FERVERS**
Co-encadrants de thèse : **Carlos CRISPIM-JUNIOR , ÉLODIE FAURE**

Devant le jury composé de:

Rapporteur	Sébastien LEFEVRE	Professeur, Université Bretagne Sud, Vannes
Rapporteur	Nicolas THOME	Professeur, Cnam, Paris
Examinatrice	Florence TUPIN	Professeure, Institut Mines Télécom, Paris
Examinateur	Christophe COLLET	Professeur, Université de Strasbourg, Strasbourg
Examinateur	Clément MALLET	Ingénieur Divisionnaire (HDR), IGN, Saint-Mandé
Directrice de thèse	Laure TOUGNE	Professeure, Université Lyon 2, Lyon
Directrice de thèse	Béatrice FERVERS	Professeure associée, Centre Léon Bérard, Lyon
Encadrant	Carlos CRISPIM-JUNIOR	Maître de Conférences, Université Lyon 2, Lyon
Encadrante (invitée)	Elodie FAURE	Ingénieure, Institut Gustave Roussy, Villejuif

Univ Lyon, Lyon 2, LIRIS, F-69676 Lyon, France
Département Cancer et Environnement, Centre Léon Bérard, Lyon, France
Agence De l'Environnement et de la Maîtrise de l'Energie, Angers, France

Remerciements

Et voilà, on arrive au bout de cette aventure. En écrivant ces quelques lignes, je me rends compte de tout le chemin parcouru ces dernières années. Ce fût une expérience particulièrement intense, avec de nombreux moments de doutes, de travail intense, d'erreurs, de réussites, et de joies partagées. Encore aujourd'hui, et peut-être encore plus qu'avant, je reste admiratif des personnes qui osent se lancer dans le vaste monde qu'est celui de la recherche. Merci à vous, chercheurs de tous temps et de tous poils pour avoir attisé ma curiosité.

Je tiens à remercier mes encadrants, qui n'ont eu de cesse de me guider et de m'épauler tout au long de ce périlleux périple. Merci de m'avoir consacré du temps alors que vous en manquiez. Merci de m'avoir permis d'éviter nombre de faux pas. Merci de m'avoir permis de sortir de ma zone de confort. Merci de m'avoir aidé à accepter et à corriger certaines de mes erreurs. Merci de m'avoir fait prendre conscience que faire de son mieux, c'est déjà pas si mal. Merci pour votre bienveillance. J'ai beaucoup appris grâce à vous.

Je remercie également mes collègues de bureau pour avoir contribué à cette épopée riche en repas copieux et discussions animées. Merci d'avoir rendu ce voyage aussi peu monotone, de m'avoir enrichi de vos connaissances, et de m'avoir supporté de longues semaines durant. J'espère que nos discussions auront été aussi intéressantes pour vous qu'elles l'ont été pour moi.

Je remercie ma famille et mes amis. Merci d'avoir toujours été là pour moi. Je ne saurais le dire autrement. Aurore, merci pour ton support quotidien et tes délicieux gâteaux, je n'en serais pas là sans toi.

Enfin, je remercie le Centre Léon Bérard, l'ADEME et le LIRIS pour avoir participé au financement de mes travaux. Je remercie aussi le LabEX IMU pour son support financier apporté dans le cadre du projet GOURAMIC.

Résumé

Cette thèse, co-financée par l'ADEME, se place dans le cadre d'une collaboration entre le LIRIS et le Centre Léon Bérard autour de l'étude épidémiologique TESTIS. L'étude TESTIS vise à estimer l'impact des pesticides sur le développement de la tumeur germinale du cancer du testicule. Cette maladie ayant un temps de développement long, il est nécessaire d'avoir accès à des informations remontant jusqu'à la naissance des sujets considérés. Dans le cas de TESTIS, les sujets les plus âgés sont nés au début des années 1970. Afin de tenir compte des expositions résidentielles individuelles aux pesticides propagés par les vents, le Centre Léon Bérard a mis au point une métrique se basant sur l'occupation du sol autour des habitations. Malheureusement, aucune base de données d'occupation du sol avant 1990 n'est actuellement suffisamment précise pour être utilisée. Afin d'obtenir ces informations, les géomaticiens du Centre Léon Bérard sont chargés de photo-interpréter des images aériennes historiques en niveaux de gris. Ce processus manuel étant particulièrement long et fastidieux, l'utilisation de méthodes automatiques ou semi-automatiques a été suggérée. L'objectif de cette thèse est de développer des algorithmes pour aider les géomaticiens à obtenir des cartes d'occupation du sol en un temps raisonnable. Pour cela, nous nous sommes intéressés à l'utilisation de méthodes de classification de textures que nous avons intégrées au sein d'un logiciel d'aide à l'annotation. Celui-ci est actuellement utilisé dans le cadre de l'étude TESTIS. Nous nous sommes ensuite intéressés à la colorisation automatique et non-supervisée des images aériennes historiques afin de proposer une visualisation alternative aux géomaticiens. Ces travaux nous ont également menés à étudier l'intérêt des couleurs générées artificiellement pour la classification des données historiques. Enfin, nous avons cherché à améliorer les cartes d'occupation du sol générées par notre logiciel au travers de méthodes de post-traitement, ouvrant la voie au développement de chaines de traitements plus performantes.

Mots clés : Images aériennes, classification, colorisation, post-traitement, texture, occupation du sol

Abstract

This thesis, co-funded by the ADEME, takes place in the context of a collaboration between the LIRIS laboratory and the Centre Léon Bérard as part of the TESTIS epidemiological study. The TESTIS study aims to estimate the impact of pesticides on the development of germ cell tumor of testicular cancer. As this disease has a long development time, it is necessary to have access to data dating back to the birth of the subjects. In the case of TESTIS, the oldest subjects were born in the early 1970s. In order to take into account individual residential exposures to pesticides spread by winds, the Centre Léon Bérard has developed a metric based on land use around dwellings. Unfortunately no land use database before 1990 is sufficiently accurate to be used. In order to obtain this information, the geomatics specialists at the Centre Léon Bérard are tasked with photo-interpreting historical aerial images in grayscale. This manual process is particularly long and tedious. Therefore, the use of automatic or semi-automatic methods has been suggested. The objective of this thesis is to develop algorithms to help geomatics specialists obtain land cover maps in a reasonable time. For that, we were interested in the use of texture classification methods that we have integrated into an annotation assistance software. This software is currently used in the TESTIS study. We then put our focus on the development of unsupervised colorization methods to provide alternative visualizations of the historical aerial images. This work also led us to study the interest of the artificially generated colors for land use classification. Finally, we sought to improve the land use maps generated by our software through post-processing methods, paving the way for the development of more efficient pipelines.

Keywords : Aerial images, classification, colorization, post-processing, texture, land use land cover

Table des matières

Table des matières ix

Introduction générale 1

1 Cadre de travail 3
1.1 Contexte . . . 4
1.1.1 Cancer du testicule . . . 4
1.1.2 Pesticides dans le monde . . . 6
1.1.3 Méthodologie de l'étude TESTIS . . . 8
1.2 Données disponibles . . . 12
1.2.1 Occupation du sol . . . 12
1.2.2 Images satellites . . . 14
1.2.3 Images aériennes . . . 18
1.3 Problématique et positionnement . . . 22

2 Notions de base 25
2.1 Extraction de caractéristiques de textures . . . 26
2.1.1 La texture . . . 26
2.1.2 Description de la texture . . . 27
2.1.3 Application de la texture en télédétection . . . 33
2.2 Sur-segmentation . . . 34
2.2.1 Méthodes courantes . . . 34
2.2.2 Application de la sur-segmentation en télédétection . . . 38
2.3 Algorithmes de classification . . . 39
2.3.1 Définitions . . . 39
2.3.2 Algorithmes communs . . . 40
2.4 Réseaux de neurones à convolutions . . . 44
2.4.1 Blocs de base . . . 46
2.4.2 Application des réseaux de neurones à convolutions en télédétection . . . 49
2.5 Conclusion et positionnement . . . 50

3 Classification de textures 51
3.1 Introduction . . . 52
3.2 HistAerial, un nouveau jeu de données . . . 53
3.2.1 Images sources . . . 53
3.2.2 Propriétés des images sources . . . 53
3.2.3 Génération du jeu de données . . . 54
3.3 Algorithmes évalués sur HistAerial . . . 58
3.3.1 Algorithmes d'extraction de caractéristiques de la littérature . . . 58
3.3.2 Proposition de nouveaux filtres pour la texture . . . 63
3.3.3 Classifieurs utilisés avec les descripteurs de textures . . . 65
3.3.4 Réseaux de neurones profonds à convolutions évalués . . . 65
3.4 Résultats et discussions . . . 67

3.4.1 Mise en place des expériences . . . 67
3.4.2 Comparaison globale . . . 69
3.4.3 Importance du contexte spatial . . . 73
3.4.4 Conclusion partielle . . . 75
3.5 Extension aux images en couleurs : cas des écorces d'arbres . . . 75
3.5.1 Jeux de données . . . 76
3.5.2 Méthodes . . . 77
3.5.3 Expériences et résultats . . . 79
3.5.4 Conclusion partielle . . . 80
3.6 Conclusion . . . 81

4 Colorisation automatique 83
4.1 Introduction . . . 84
4.2 Travaux connexes et notions spécifiques . . . 85
4.2.1 Réseaux de neurones adversaires génératifs (GAN) . . . 85
4.2.2 Réseaux de neurones cycliques . . . 85
4.2.3 Approches pour la colorisation . . . 86
4.3 Vers une colorisation automatique des images aériennes historiques . . . 88
4.3.1 Col-Cycle . . . 89
4.3.2 Reconstruction des images colorisées . . . 91
4.3.3 Données et entraînement . . . 93
4.3.4 Résultats et discussions . . . 93
4.3.5 Application à la classification . . . 94
4.3.6 Conclusion partielle . . . 96
4.4 Vers une amélioration de la colorisation . . . 97
4.4.1 Blocs de base . . . 97
4.4.2 SpyncoGan . . . 100
4.4.3 Mise en place des expériences . . . 104
4.4.4 Résultats et discussions . . . 105
4.4.5 Application à la classification . . . 110
4.4.6 Conclusion partielle . . . 112
4.5 Conclusion . . . 114
4.6 Visualisations supplémentaires . . . 114

5 Segmentation sémantique et post-traitement 119
5.1 Introduction . . . 120
5.2 Travaux connexes . . . 121
5.2.1 Bords et bords profonds . . . 121
5.2.2 Champs aléatoires conditionnels . . . 122
5.3 Méthode . . . 123
5.3.1 Détection de bords profonds et représentations basées superpixels . . . 124
5.3.2 Intégration au sein d'un champ aléatoire conditionnel . . . 126
5.4 Expériences et résultats . . . 127
5.4.1 Mise en place . . . 127
5.4.2 Expériences . . . 129
5.4.3 Apport de la colorisation . . . 133
5.5 Conclusion . . . 136

Conclusion générale et perspectives 137

Références 141

A Gouramic I

Liste des figures **V**

Liste des tableaux **VIII**

Introduction générale

L'impact des modifications de l'environnement et des modes de vie sur l'augmentation de l'apparition de certains cancers est une préoccupation majeure de santé publique. Avec environ 382 000 nouveaux cas de cancers estimés en 2018, le nombre de cancers a plus que doublé sur presque 40 ans [INC19]. Outre les facteurs individuels de risques établis, les variations spatiales et l'évolution rapide de l'apparition de certains cancers dans les populations migrantes sont en faveur d'un rôle des facteurs environnementaux dans le développement de ces maladies. Parmi les facteurs environnementaux, les expositions environnementales aux pesticides sont particulièrement suspectées. Pour la population générale, l'exposition aux pesticides provient de la dérive des pesticides appliqués sur les cultures. Ainsi, plusieurs études ont montré une corrélation entre la taille des surfaces cultivées et la distance des résidences aux cultures, avec l'exposition aux pesticides d'origine agricole. Cependant, le lien avec ces expositions est parfois difficile à établir sur de longues périodes. Or, un délai de latence important (*i.e.*, plusieurs années) est supposé entre les premières expositions et le développement de certains cancers, tels que le cancer du testicule. Les connaissances actuelles sont d'ailleurs en faveur d'un rôle des expositions précoces dans la vie, voire durant le développement *in utero*. Cela nécessite l'accès à des données anciennes pour étudier le cas des malades les plus âgés. Malheureusement, il y a actuellement un manque de données historiques fiables relatives aux expositions. Ce point conduit à une réduction des informations utilisables et peut être responsable de sous-estimations ou de sur-estimations du risque.

Dans ce cadre, le Centre de lutte contre le cancer Léon Bérard étudie, avec l'étude épidémiologique TESTIS, le lien entre cancer du testicule et expositions résidentielles issues de l'épandage des pesticides agricoles à proximité des lieux de vie des sujets inclus dans l'étude. Pour cela, il a besoin de définir les types de cultures à proximité des résidences des sujets pour estimer un score d'exposition individuel aux pesticides, et ce depuis le développement *in utero* des sujets (début des années 1970 pour les plus âgés). Malheureusement, il n'existe pas, à l'heure actuelle, de bases de données géographiques contenant ces informations, et l'annotation manuelle des terrains cultivés à partir d'images aériennes d'archives est une tâche spécialisée particulièrement longue et fastidieuse (plusieurs heures par image). Le département Cancer et Environnement du Centre Léon Bérard s'est ainsi associé à l'équipe IMAGINE du laboratoire LIRIS afin de développer un logiciel de traitement d'images pour accélérer ce travail. Ce partenariat s'est traduit par l'emploi temporaire d'un ingénieur, moi même, qui a permis de mettre en place une preuve de concept afin de produire une couche de données de qualité. Encouragés par de premiers résultats, ces travaux ont pu être continués en thèse via un co-financement de l'Agence De l'Environnement et de la Maîtrise de l'Energie (ADEME) et du Centre Léon Bérard, sous un encadrement partagé avec le LIRIS.

L'objectif principal de cette thèse est ainsi de développer des méthodes permettant la reconnaissance automatique des parcelles de terrains à partir d'images aériennes historiques, et d'intégrer ces avancées au sein d'outils logiciels à destination des géomaticiens travaillant sur le projet TESTIS. Pour cela, nous avons d'abord abordé la problématique de la reconnaissance des occupations du sol via la classification de la texture. Nous avons intégré les chaînes de traitements évaluées au sein d'un logiciel permettant à l'utilisateur de guider la segmentation à l'aide de traces (possibilités de vérification et de correction). Celui-ci est actuellement en cours d'utilisation dans le cadre de l'étude TESTIS (voir Annexe A). Cependant, les images historiques étant principale-

ment disponibles en niveaux de gris, elles sont particulièrement difficiles à interpréter par un être humain par rapport à des images en couleurs. Afin de combler ce fossé visuel et proposer des représentations alternatives aux géomaticiens, nous nous sommes alors intéressés à la colorisation automatique des images aériennes historiques. Enfin, malgré les résultats satisfaisants de notre logiciel, ceux-ci ont tendance ; par construction ; à ne pas respecter la géométrie des parcelles. Afin d'améliorer la qualité des occupations du sol générées, nous avons cherché à utiliser des méthodes de sur-segmentations et à intégrer l'information portée par les segments au sein d'un champ aléatoire conditionnel dans un cadre de post-traitement.

Cette thèse est ainsi composée de 5 chapitres :

— Le chapitre 1 présente le cadre de travail de la thèse du point de vue du projet TESTIS. Il introduit également les problématiques liées aux données à notre disposition.

— Le chapitre 2 présente les notions de base sur lesquelles nos travaux se sont appuyés : des filtres de textures aux réseaux de neurones profonds à convolutions, en passant par la sur-segmentation.

— Le chapitre 3 présente les travaux que nous avons menés sur la classification de textures naturelles à l'aide de méthodes classiques et de réseaux de neurones profonds.

— Le chapitre 4 présente les méthodes que nous avons développées pour la colorisation non supervisée d'images aériennes historiques en nous basant sur des réseaux de neurones générateurs adversaires cycliques et pseudo-cycliques.

— Le chapitre 5 présente nos travaux sur le post-traitement de segmentations sémantiques à l'aide d'un champ aléatoire conditionnel et de superpixels générés à partir de bords détectés par un réseau de neurones entièrement convolutif.

Nos travaux sur la classification de textures ont été présentés dans le cadre de la conférence nationale CFPT 2018 (Conférence Française de Photogrammétrie et de Télédétection) [RCJF+18], du journal international IEEE TIP (*Transactions on Image Processing*) [RCJF+19a] et de la conférence internationale VISAPP 2019 (*International Conference on Computer Vision Theory and Applications*) [RBCJT19]. Nos travaux sur la colorisation ont été exposés à la conférence internationale IGARSS 2019 (*International Geoscience and Remote Sensing Symposium*) [RCJF+19b] et au *Workshop* SUMAC (*Structuring and Understanding of Multimedia heritAge Contents*) mené en conjonction avec la conférence internationale ACM MM (ACM *Multimedia* 2019) [RCJF+19c]. Nos travaux sur le post-traitement de segmentations sémantiques d'images aériennes ont été acceptés pour présentation à la conférence internationale IGARSS 2020 [RCJF+20], et un article a également été soumis à la conférence internationale IPTA 2020 (*International Conference on Image Processing Theory, Tools and Applications*). Notre logiciel a par ailleurs fait l'objet de communications courtes (*abstract proceedings*) dans le cadre d'une conférence nationale et de deux conférences internationales rattachées au domaine de l'épidémiologie [FRCJ+18a; FRCJ+19; FRCJ+18b].

Chapitre 1

Cadre de travail

Le but de ce chapitre est de présenter le cadre de travail dans lequel nos travaux de recherche ont été réalisés afin de donner au lecteur un aperçu des enjeux applicatifs sous-jacents à nos développements, de fournir une vision globale des données disponibles, et d'introduire les problématiques qui ont été traitées. Nous verrons tout d'abord le contexte épidémiologique dans lequel s'inscrit cette thèse au travers de l'étude TESTIS, qui vise à évaluer s'il existe une association entre l'exposition aux pesticides et le risque de cancer du testicule à l'aide d'un Système d'Information Géographique (SIG). Nous ferons ensuite un état des lieux des données disponibles pour déterminer l'occupation des sols à partir d'images aériennes et satellites, avec une accentuation particulière sur les données historiques. Nous introduirons enfin les problématiques qui ont été soulevées et auxquelles nous avons répondu.

Sommaire

1.1 Contexte 4
1.1.1 Cancer du testicule 4
1.1.2 Pesticides dans le monde 6
1.1.3 Méthodologie de l'étude TESTIS 8
1.2 Données disponibles 12
1.2.1 Occupation du sol 12
1.2.2 Images satellites 14
1.2.3 Images aériennes 18
1.3 Problématique et positionnement 22

1.1 Contexte

Cette thèse se place dans le cadre de l'étude épidémiologique TESTIS portée par le département Cancer et Environnement du Centre Léon Bérard. TESTIS est une étude multicentrique d'envergure nationale visant à caractériser l'impact de l'exposition vie entière aux pesticides (domestiques, professionnels et environnementaux) de participants français sur le risque de développement d'une Tumeur Germinale du Testicule (TGT) à l'âge adulte. Le terme germinale est ici associé aux cellules de reproductions présentes dans les testicules et impliquées dans le développement des spermatozoides. Des études sur les TGT dans la littérature ont suggéré qu'une origine précoce des expositions pouvait avoir un fort impact sur le développement de la tumeur (jeune âge des patients). L'étude TESTIS s'intéresse ainsi tout particulièrement à l'hypothèse d'une association entre l'exposition aux pesticides pendant les périodes critiques de développement de l'humain et le risque de développement d'une TGT. Pour cela, 472 hommes (cas), nés entre 1971 et 1997, appariés à 683 témoins sur le centre recruteur et l'âge, ont été recrutés entre 2015 et 2018. La répartition spatiale de ces sujets (cas et témoins) à leur année de naissance est présentée sur la Figure 1.1. A ce jour, TESTIS est l'une des plus larges études cas-témoins portant sur ce type de tumeurs et couvrant le territoire français.

1.1.1 Cancer du testicule

En chiffre

Le cancer du testicule est le cancer le plus fréquemment observé chez l'homme jeune de 15 à 44 ans dans les pays développés, avec un âge moyen de diagnostique estimé à 33 ans [FBB$^+$13; WB18]. Bien que n'affectant qu'une faible partie de la population, son incidence (*i.e.*, sa fréquence d'apparition) augmente de 2,6%/an en moyenne depuis 1980. Il a ainsi été estimé en 2014 qu'une augmentation moyenne de 24% aurait lieu en Europe d'ici 2025 [LCLTF$^+$14]. Parmi les variantes du cancer du testicule, la TGT, étudiée dans le cadre de TESTIS, representerait 98% des cas observés [FBB$^+$13], avec des taux de survie à 5 ans de 95% pour les tumeurs localisées et de 80% pour les tumeurs métastasées (*i.e.*, qui s'étendent au reste du corps) [Fel08]. Il semble également intéressant de constater que le cancer du testicule a un taux d'incidence qui semble être positivement corrélé avec le niveau économique des pays [WB18]. Les taux de mortalité liés au cancer du testicule semblent être, quant à eux, inversement proportionnels aux niveaux économiques des pays. Tandis que ces observations ont été réalisées à partir de données de 2012 [WB18], il est possible de voir sur les Figures 1.2 et 1.3 proposées par le Centre International de Recherche sur le Cancer

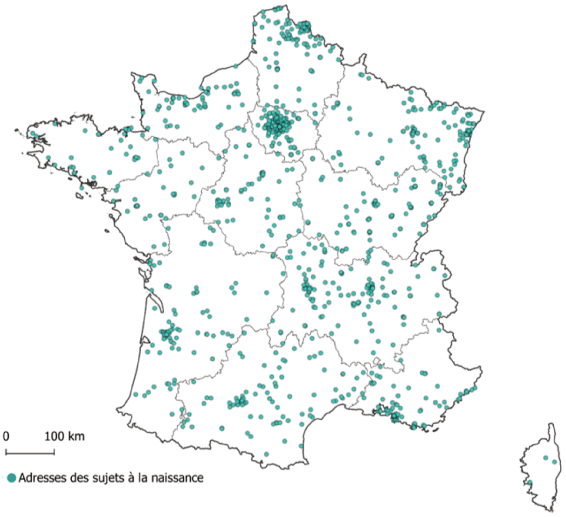


FIGURE 1.1 – Distribution des sujets recrutés dans l'étude TESTIS à leur année de naissance, France métropolitaine (Matthieu Dubuis, 2020).

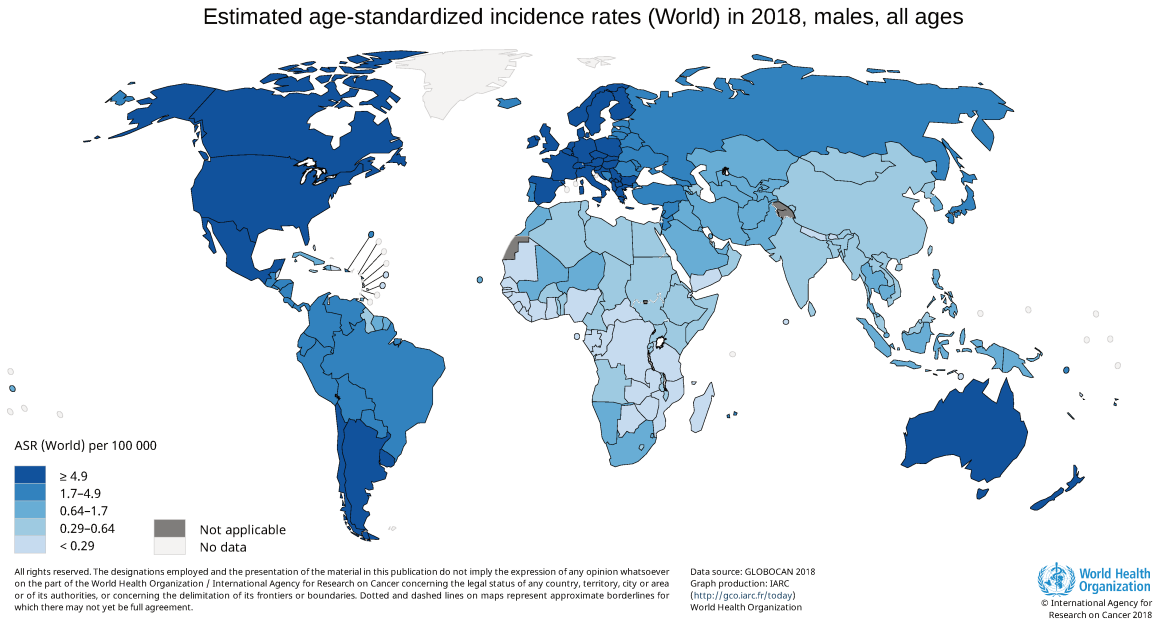


FIGURE 1.2 – Taux d'incidence standardisé pour l'âge (*Age-Standardized Rates*, ASR) du cancer du testicule en 2018. Données fournies par le CIRC [dRslCC20].

(CIRC) - *International Agency for Research on Cancer* (IARC) [dRslCC20], que leur validité semble aussi se vérifier en 2018. Sur ces deux figures, les taux d'incidence et de mortalité standardisés pour l'age (ASR) correspondent aux valeurs qui seraient obtenues pour des populations suivant la distribution des âges de la population mondiale standard fournie par le CIRC [1]. Il est ainsi possible d'observer que les pays développés tels que la France ou les Etats-Unis ont des taux de mortalité relativement faibles en comparaison des taux d'incidence (facteurs 28 et 21 respectivement), tandis que le Mali a un fort taux de mortalité (plus élevé que la France) pour un taux d'incidence relativement faible (dix fois moins élevé que la France). Ces éléments tendent à indiquer que la maladie touche plus les populations des pays économiquement développés que celles des pays en développement, mais qu'elle y est traitée de façon plus efficace (mortalité plus faible).

Facteurs de risques

Les facteurs de risques sont des éléments non directement causaux ayant été identifiés ou étant suspectés de faciliter le développement d'un cancer. Dans le cas d'une TGT, certains facteurs de risques tels qu'un antécédent cancéreux localisé sur les testicules ou la présence de syndromes particuliers ont déjà pu être identifiés [2]. D'autres facteurs de risques font actuellement le sujet de recherches approfondies. Parmi eux, les facteurs environnementaux sont fortement suspectés d'avoir un impact sur le développement de la maladie. Cette hypothèse a été émise de par l'observation de fortes disparités géospatiales dans le monde [WB18]. Elle est renforcée par l'étude des taux d'incidence sur les flux migratoires de populations entre la première génération (adultes migrants) et la deuxième génération (enfants nés sur place), qui montrent que la deuxième génération tend à être plus touchée que la première [SSSJ10]. Ces observations mettent en avant l'influence potentielle d'un facteur environnemental, indépendant de la génomique, intervenant durant les phases de développements de l'humain. En particulier, l'hypothèse du rôle des expositions intra-utérines ; c'est à dire lors du développement du foetus dans le ventre de la mère ; associées à un développement long de la maladie a été émise en 2001 suite à l'observation de pics

1. `www-dep.iarc.fr/WHOdb/glossary.htm` (accès : 2020-01-31)
2. `www.urofrance.org/congres-et-formations/formation-initiale/referentiel-du-college/tumeurs-du-testicule.html` (accès : 2020-01-31)

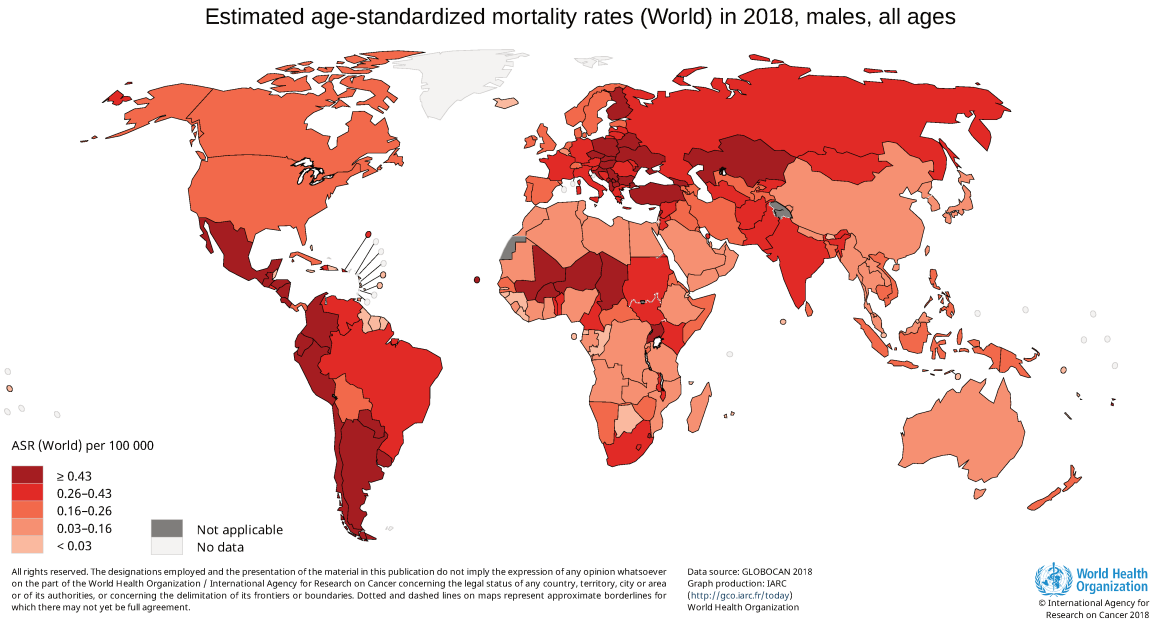


FIGURE 1.3 – Taux de mortalité standardisé pour l'âge (*Age-Standardized Rates*, ASR) lié au cancer du testicule en 2018. Données fournies par le CIRC [dRslCC20].

d'incidence de la TGT chez l'adulte jeune [SRDMM01]. Les sources d'expositions environnementales actuellement suspectées comme étant des facteurs de risques sont des produits ayant des effets perturbateurs endocriniens, dont les polluants chimiques dans l'air rejetés par l'industrie chimique et les pratiques agricoles. Les expositions professionnelles et domestiques sont également suspectées. Dans le cadre de TESTIS, les expositions intra-utérines aux pesticides constituent la principale hypothèse de recherche.

1.1.2 Pesticides dans le monde

Les pesticides sont des produits chimiques qui permettent d'éradiquer les insectes ravageurs ou d'éliminer certaines espèces de végétaux. On distingue plusieurs catégories de pesticides en fonction des êtres vivants qu'ils permettent d'éliminer, tels que les insectes (insecticides), les mauvaises herbes (herbicides), ou encore les champignons (fongicides). Les pesticides étant par nature nocifs pour les êtres biologiques (dont l'humain), ils sont impropres à la consommation et font l'objet de contrôles réglementaires importants dans le domaine de l'agro-alimentaire. On parle alors de limites maximales en résidus de pesticides. Tandis que ces protections protègent *partiellement* le consommateur d'une ingurgitation excessive de pesticides par l'alimentaire, les populations restent exposées aux inhalations des molécules qui sont propagées dans l'air, puis éventuellement déplacées par les vents.

Consommation

La consommation des pesticides dans le monde et dans le temps peut être suivie à l'aide des statistiques récoltées par l'Organisation des Nations Unies pour l'alimentation et l'agriculture (*Food and Agriculture Organization of the United Nations*, FAO). Ces dernières ont été mis en ligne en accès libre et gratuit via de la base de données FAOSTAT [3]. La Figure 1.4 a été générée à partir de ces données. Nous pouvons remarquer sur cette figure que l'utilisation de pesticides dans le monde est en augmentation depuis près de 30 ans, et ce en particulier dans les pays d'Asie et d'Amérique où l'utilisation de pesticides est respectivement passée de 2.12 kg/hectare

3. `http://www.fao.org/faostat/en/` (accès : 2020-02-10)

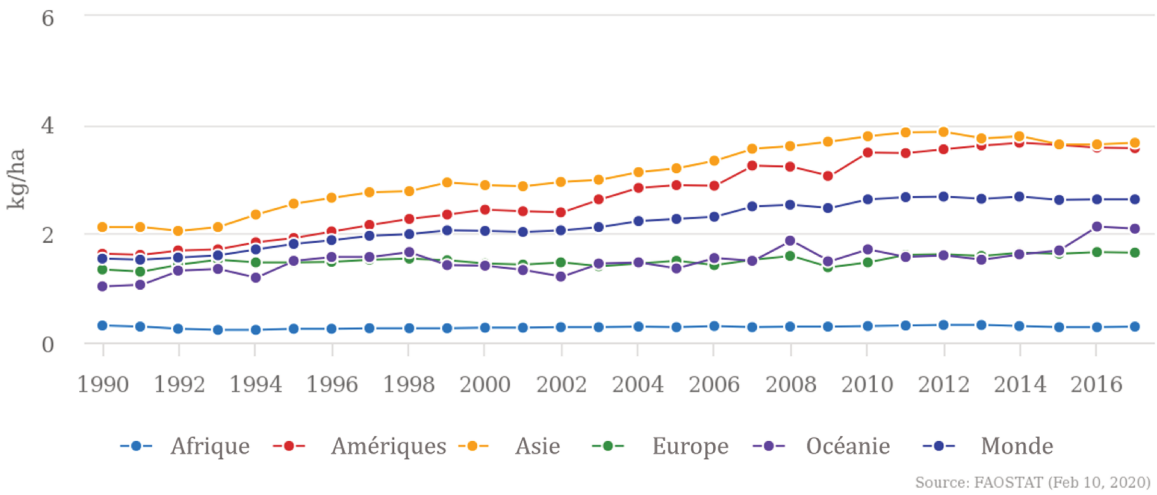


FIGURE 1.4 – Évolution de la consommation moyenne de pesticides dans le monde par hectare de surface cultivée de 1990 à 2017. Données fournies par la FAO.

et de 1.63kg/hectare en 1990 à 3.67 kg/hectare et 3.57 kg/hectare en 2017. En Europe, on observe que la consommation des pesticides sur cette période est restée relativement stable, passant de 1.34 kg/hectare en 1990 à 1.65 kg/hectare en 2017. L'utilisation importante de pesticides en Asie et Amérique est principalement attribuée à la culture du riz. En 2017, tandis que la moyenne mondiale de consommation de pesticides était de 2.63 kg/hectare, la Chine en consommait 13.07 kg/hectare et le Japon 11.76 kg/hectare. A titre comparatif, la France consommait en moyenne 3.63 kg/hectare de pesticides. Il est cependant important de préciser que ces chiffres représentent des données par unité de surface cultivée. Ils ne représentent donc pas la consommation brute des pays (*i.e.*, la quantité totale de pesticides utilisés). En 2010, il avait été ainsi estimé, dans un rapport de l'office parlementaire d'évaluation des choix scientifiques et technologiques sur les pesticides et la santé [4], que la France, plus grand producteur agricole en Europe, avait été le 4ème pays plus grand consommateur de pesticides dans le monde et le plus grand consommateur en Europe pour l'année 2008. Ce rapport indique également que, "rapportée à la consommation moyenne de pesticides par hectare cultivé, la France se place dans une position moyenne".

Impact sur la santé

L'impact des pesticides sur le développement de maladies humaines est depuis longtemps suspecté. Il a été quantitativement étudié pour certaines d'entre elles, telles que pour la maladie de Parkinson [PPC+17]. Les pesticides sont par ailleurs des facteurs de risques avérés pour de nombreux cancers. Selon un rapport de l'INSERM [5], on remarque ainsi une augmentation du risque sur le cancer de la prostate sur les ruraux et les ouvriers (entre 12% et 28% selon les populations). Ce même rapport indique que les pesticides peuvent avoir un impact sur la grossesse et le développement de l'enfant. En particulier, les expositions professionnelles tendent à augmenter le risque de mort foetale ainsi que le risque de réduction des capacités visuelles et motrices chez l'enfant. Les expositions résidentielles en période pré-natale augmenteraient quant à elles les risques de malformations et de leucémies. Dans le cadre de TESTIS, il s'agit d'étudier l'impact de l'exposition vie entière, incluant la période pré-natale, aux pesticides sur le développement de la TGT à l'âge adulte.

4. `http://www.assemblee-nationale.fr/13/pdf/rap-off/i2463.pdf` (accès : 2020-02-10)
5. `http://www3.ligue-cancer.net/docs/fichiers/pesticides.pdf` (accès : 2020-02-10)

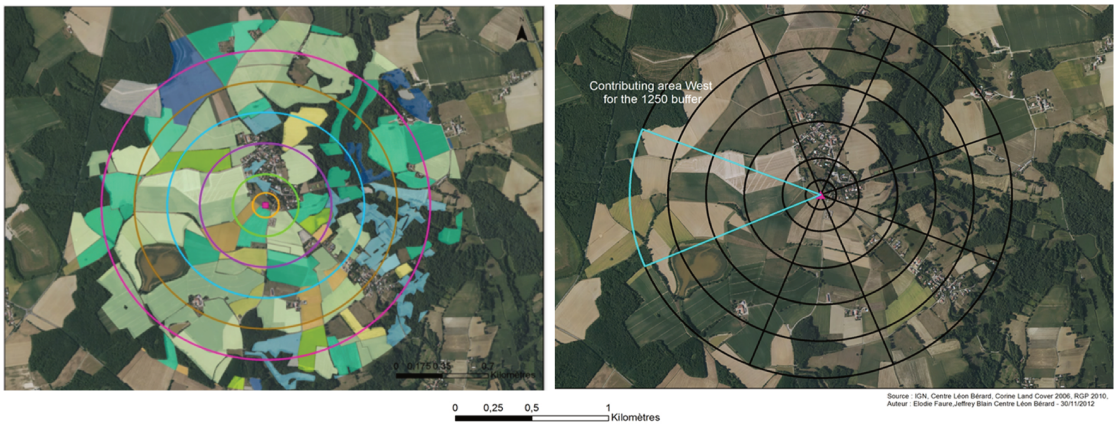


FIGURE 1.5 – Illustration de l'étude SIGEXPO avec l'occupation des sols hors barrières naturelles et artificielles, et modélisation des vents dominants sur un sixième d'arc de cercle [FBF+18].

1.1.3 Méthodologie de l'étude TESTIS

L'étude TESTIS fait suite à deux projets préliminaires, TESTEPERA et SIGEXPO. Ils ont permis d'étudier et de mettre en place les briques de base nécessaires à la réalisation de TESTIS. Ces deux projets ont été réalisés durant la thèse de Rémi Béranger [Bé14]. Nous les exposons succinctement ici afin de mettre en avant la méthodologie appliquée dans le cadre de TESTIS.

TESTEPERA

TESTEPERA est une étude pilote cas témoins qui a été réalisée sur un sous-ensemble représentatif de la population cible de TESTIS [BBB+14]. Elle visait à déterminer l'efficacité de différents modes de recrutement et la capacité de collection de données pertinentes pour la période prénatale. Elle a été réalisée entre 2011 et 2012. Durant cette période, 150 sujets masculins ont été contactés dans la région Rhône-Alpes en France, dont 58 hommes atteint d'un cancer et 92 sujets témoins. Pour l'ensemble des recrutements, il est intéressant de remarquer que seuls les sujets ayant rencontré un recruteur en personne ont accepté de participer à l'étude. Les questionnaires permettant la récolte des données ont permis de déterminer les emplois et la géolocalisation des sujets dans le temps. L'étude a montré que la précision de la géolocalisation des sujets était dépendante du niveau d'urbanisation, en plus d'être dépendante de la précision portée par les réponses des sujets (*e.g.*, précision à l'adresse, à la rue, au lieu-dit). A noter que cette précision ne semble pas biaisée par la période d'étude [BBB+14].

SIGEXPO

SIGEXPO [BBB+13] est une étude lancée en 2012 qui avait pour but d'identifier les facteurs déterminants de l'exposition environnementale aux pesticides agricoles et de développer une nouvelle métrique basée sur un SIG adapté au territoire français à partir de données récentes. Pour cela, l'étude s'est basée sur une campagne de prélèvements de poussières en habitat intérieur dans la région Rhône-Alpes. Il a en effet été montré dans la littérature que les poussières que l'on retrouve au sein des habitations contiennent des traces de pesticides provenant certes d'un usage domestique, mais aussi de l'application de produits sur les cultures proches de l'habitation [GWA+11]. Il a par ailleurs été montré que les connaissances géographiques liées à la localisation des cultures à proximité des habitations permettent de prédire efficacement la présence de pesticides dans les poussières. Les foyers étudiés dans le cadre de SIGEXPO ont ainsi été sélectionnés de par leur proximité avec différents types de cultures représentatifs du territoire français (arboriculture / vergers, vignes, champs céréaliers) à des rayons de taille variable. En pratique, des aires de rayon compris entre 100 mètres et 1250 mètres ont été étudiées pour déterminer les facteurs environnementaux, les études précédentes suggérant des aires de rayon entre 500 mètres

et 1250 mètres en fonction de l'étude [CMZ+11; GWA+11]. A noter que ce rayon a été étendu à 1500 mètres dans le cadre de TESTIS. L'étude SIGEXPO s'est basée sur des données d'Occupation du sol (OCS) disponibles (BD Alti, Institut Géographique National (IGN) ; BD Topo, IGN) ; Registre Parcellaire Géographique (RPG), IGN) [FBF+18] afin de développer un modèle SIG incluant non-seulement les informations spatiales liées aux cultures, mais aussi la présence de barrières topologiques naturelles (*e.g.*, haies) et artificielles (*e.g.*, murs, bâti) propres à bloquer la propagation des particules. La direction des vents dominants a aussi été intégrée dans la méthode développée afin de pondérer l'importance des types de cultures dans une direction particulière selon ce vecteur de diffusion (voir Figure 1.5). Cette information a été obtenue grâce aux données des stations météorologiques de Météo France. Les résultats obtenus ont montré que l'analyse de l'environnement à l'aide du modèle SIG développé pour SIGEXPO permettait d'expliquer une part non négligeable des phytosanitaires retrouvés dans les poussières, et ainsi d'inférer des scores d'exposition individuels pour chaque habitat dont l'OCS alentours est connue [BBB+13].

TESTIS

L'étude TESTIS a été lancée en 2015 et est encore en cours actuellement (03/2020). Cette étude a pour ambition d'étendre les approches développées lors des projets TESTEPERA et SIGEXPO à l'échelle nationale, et ce sur la vie entière des sujets. Le diagramme de flux simplifié de la figure 1.6 permet de visualiser les relations entre TESTIS et ces deux études préliminaires. A noter que ce diagramme exclut les aspects liés aux expositions professionnelles et domestiques aux pesticides, aussi étudiées dans le cadre de TESTIS. Par la suite, nous nous intéresserons tout particulièrement à la partie représentée à gauche de ce diagramme, correspondant à la génération de cartes d'occupation du sol géoréférencées à partir d'images aériennes historiques.

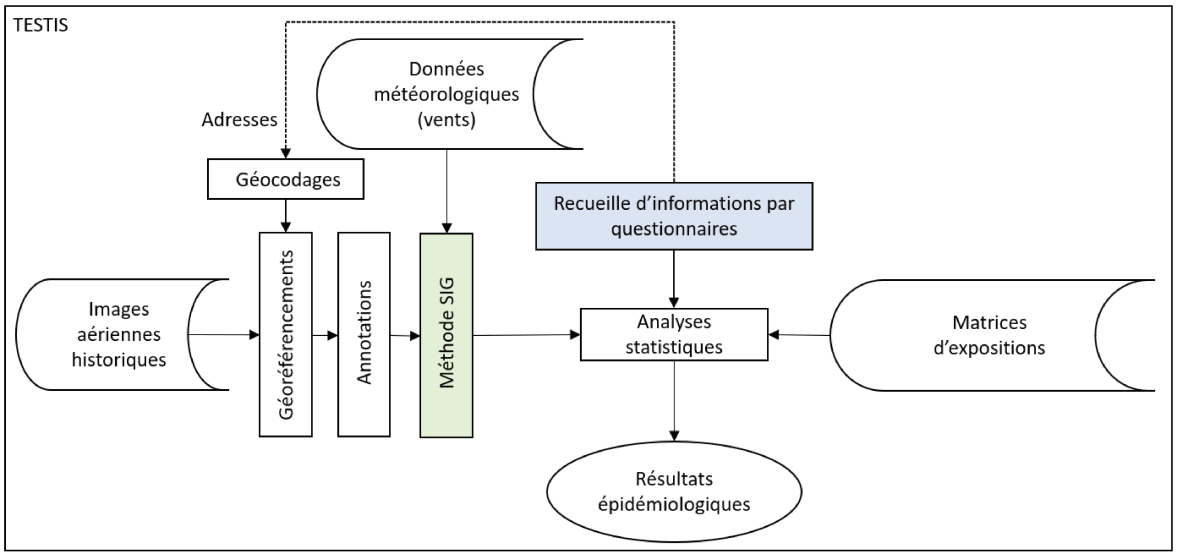


FIGURE 1.6 – Diagramme de flux simplifié de l'étude TESTIS. La méthode SIG (en vert) a été développée dans le cadre de SIGEXPO. L'utilisation de questionnaires pour le recueille d'informations (en bleu) a été validée dans le cadre de TESTEPERA.

Recrutement des sujets. Entre 2015 et 2018, un recrutement de 1155 sujets (sur 1500 prévus initialement) à l'échelle nationale a été réalisé dans 20 Centres Hospitaliers Universitaires en France Métropolitaine sous la coordination du Centre Léon Bérard. Ces recrutements ont permis d'inclure 472 sujets (cas) atteints de TGT et 683 témoins non atteints par la TGT. Les témoins ont été répartis en deux catégories, A et B, correspondants respectivement à des hommes donneurs de sperme mariés à des femmes infertiles, et à des hommes mariés à des femmes ayant une grossesse pathologique. Afin de n'inclure que des sujets ayant un âge dans la fourchette correspondant au pic d'incidence de la TGT, seuls des sujets adultes âgés au plus de 44 ans au moment du recrute-

ment ont été inclus dans l'étude. Le protocole détaillé du recrutement est présenté dans l'article de Béranger et *al.* [BPB+14]. Les participants et leurs mères (N=50% de répondantes) ont répondu à un entretien téléphonique afin de collecter des données concernant les lieux d'habitation, les métiers et les usages domestiques de produits chimiques. Ces questionnaires permettent d'inférer les informations nécessaires à l'estimation des expositions domestiques et professionnelles aux pesticides au travers de matrices d'expositions et d'un codage des métiers réalisé par une hygiéniste industrielle. L'estimation des expositions aux pesticides d'origine agricole requiert quant à elle des étapes de traitements de données particulières afin d'exploiter l'approche SIG développée durant le projet SIGEXPO. L'approche qui a été retenue dans le cadre de TESTIS pour obtenir des OCS consiste à photo-interpréter les images aériennes panchromatiques (en niveaux de gris) disponibles pour la période d'intérêt de l'étude. On remarquera que cette période d'intérêt s'étend, de par l'âge des sujets recrutés, du début des années 1970 à la fin des années 1990, et jusqu'en 2018 pour les exposition vie entière. Nous décrivons ci-après le processus suivi pour générer des OCS.

Géocodage des sujets. Avant d'estimer l'OCS autour d'une habitation, il est nécessaire de connaitre la position géographique de celle-ci. Pour cela, il est nécessaire de géocoder les sujets, c'est à dire de les replacer sur la carte de France, et ce pour chacune de leurs adresses. En fonction de la qualité des informations recueillies avec les questionnaires, cette étape de géocodage peut être plus ou moins automatisée à l'aide de la Base Adresse Nationale [6]. Dans le cas où les adresses ne sont que peu précises (*e.g.*, nom de rue mais pas de numéro), les géomaticiens peuvent décider de suivre des règles arbitraires afin de réaliser le géocodage (*e.g.*, placer le sujet au milieu de la rue) [FDCC+17]. Dans le cadre de TESTIS, l'ensemble du géocodage est réalisé dans le repère géographique français Lambert93. Il a par ailleurs été observé qu'un sujet de l'étude TESTIS aura eu 6,6 adresses en moyenne au cours de sa vie.

Génération de l'occupation du sol. Une fois le géocodage réalisé, il est nécessaire d'accéder aux données d'OCS autour des lieux d'habitation des sujets aux dates correspondantes. Cependant, aucune base de données annotées disponibles avant 1990 existe aux degrés de précisions spatiale et temporelle désirés (voir section 1.2), et le Recensement Statistique Agricole français, contenant des statistiques instantanées décennales au niveau communal, ne permet pas d'estimer un score individuel d'exposition autour d'une habitation particulière (*e.g.*, les champs peuvent se trouver de l'autre côté de la commune par rapport à l'habitation considérée). Face à ce constat, l'approche qui a été retenue dans le cadre de TESTIS consiste en une photo-interprétation (*i.e.*, annotation) des images aériennes historiques panchromatiques disponibles autour d'une habitation à date donnée. Ces images ont été choisies dû à leur disponibilité et à leurs hautes résolutions permettant une annotation à la parcelle près (voir section 1.2). On remarquera que l'estimation de différents types de cultures à partir d'images dans un contexte épidémiologique a déjà montré son intérêt par le passé [MAN10], où les auteurs proposaient l'utilisation de données satellites pour estimer l'exposition aux pesticides d'origine agricole en Californie, États-Unis. Nous décrivons ici le processus générique suivi par les géomaticiens travaillant sur l'étude TESTIS pour générer des cartes d'OCS par photo-interprétation.

- Pour une date donnée et pour un sujet donné, il est dans un premier temps nécessaire d'acquérir les images d'archives intersectant une aire de rayon 1.5 kilomètres autour du lieu d'habitation du sujet. Dans le cadre de TESTIS, le choix s'est porté sur les images aériennes historiques archivées par IGN (voir section 1.2). Celles-ci possèdent une résolution spatiale élevée et se sont révélées facilement accessibles.
- Si les images obtenues ne sont pas géoréférencées, c'est à dire que la transformation affine entre le plan image et le référentiel géographique n'est pas connue (*i.e.,* on a l'image, mais on ne sait pas la situer sur la carte), il est alors nécessaire d'effectuer ce géoréférencement pour pouvoir les intégrer convenablement dans un SIG. Pour cela, l'approche standard consiste à indiquer des points de contrôles sur l'image, qui seront ensuite mis en cor-

6. `https://www.data.gouv.fr/en/datasets/base-adresse-nationale/` (accès : 2020-03-23)

respondance avec des points de contrôles connus (*e.g.*, intersection de routes). Des approches automatiques sont proposées par certains logiciels tels que ArcGIS [7], mais la documentation indique que cette approche ne fonctionne pas correctement avec les données numérisées ou historiques. Des approches automatiques expérimentales prometteuses développées par l'IGN français apparaissent peu à peu pour les photographies aériennes historiques [GLBM18] en se basant sur le logiciel MicMac [8]. L'approche manuelle reste néanmoins privilégiée dans le cadre de TESTIS.

- Une fois l'image géoréférencée sur la carte, il est nécessaire de la segmenter en plusieurs classes d'OCS afin d'obtenir la donnée désirée. Dans le cadre de TESTIS, 7 classes d'OCS ont été identifiées, à savoir les prairies, les champs de grandes cultures, les zones urbaines, les forêts, les vignes, les vergers et les eaux. En particulier, les vignes, vergers et grandes cultures représentent des utilisations de phytosanitaires différents dont l'impact intéresse particulièrement les épidémiologistes. Les zones urbaines et les forêts permettent quant à elles de définir des barrières naturelles ou artificielles lors de l'intégration des vents dominants (voir section 1.1.3). Nous rappelons qu'afin de réaliser cette étape, l'approche choisie pour TESTIS est la photo-interprétation des images aériennes historiques par un géomaticien expérimenté. Cette approche a été appliquée lors d'une étude préliminaire sur les images qui correspondent aux sujets issus de l'étude TESTEPERA. Il a alors été estimé que le temps d'annotation était de 6 à 10 heures par image avec une approche basée sur le détourage des parcelles. Cette durée d'annotation particulièrement longue s'explique par la difficulté d'interprétation des données historiques, ainsi que par la taille (*e.g.*, 12 000 x 12 000 pixels) et la haute résolution des images exploitées (0.6 mètres en moyenne sur l'année de naissance). Une approche alternative basée sur une annotation d'un maillage régulier a été proposée en 2015 dans le cadre du stage d'Amélie Machelart, permettant de réduire le temps d'annotation manuelle à environ 2h30 par image contre une perte de données liée à un territoire représenté de façon morcelée (grille régulière). Un exemple visuel est présenté sur la Figure 1.7 au niveau de la commune de Die, France, en 1978, avec 9 classes d'OCS (distinction entre forêts denses et forêts peu denses, et entre prairies et parcs / jardins non exploitée par la suite). A noter que cette deuxième approche a motivé le développement du logiciel semi-automatique Gouramic, basé sur nos travaux et présenté en annexes (Annexe A). Gouramic permet de réduire le temps d'annotation à environ 20 minutes par image. L'utilisation de Gouramic pour la photo-interprétation est celle qui a été retenue pour l'étude TESTIS.

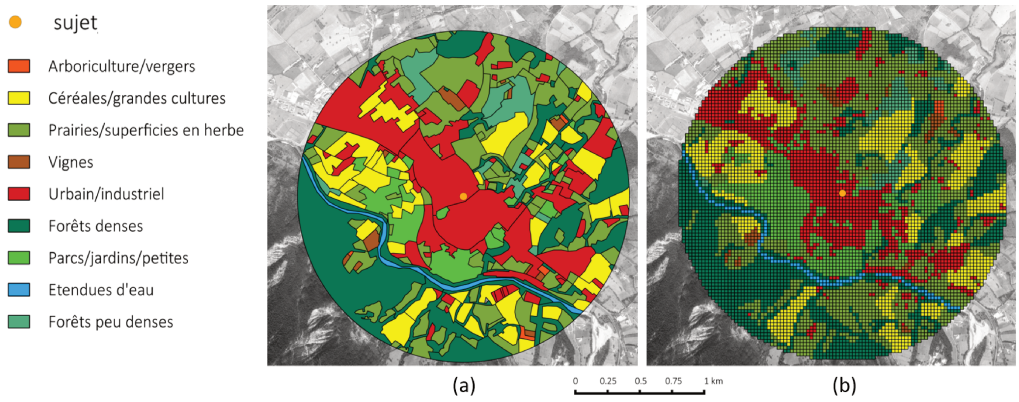


FIGURE 1.7 – Exemples de cartes d'occupation du sol générées manuellement à partir d'images aériennes historiques au niveau de la commune de Die, France, en 1978. Comparaison d'une approche (a) par détourage de parcelles, et d'une approche (b) par annotation de cellules sur une grille régulière.

7. `https://desktop.arcgis.com/fr/arcmap/10.3/manage-data/raster-and-images/` (accès : 2020-03-18)
8. `https://micmac.ensg.eu/index.php/IGN` (accès : 2020-03-23)

1.2 Données disponibles

Dans le cadre de l'étude TESTIS, il est nécessaire de connaitre l'OCS dans un rayon maximum de 1500 mètres autour des lieux d'habitation des sujets recrutés, de leurs naissances jusqu'à leurs recrutements dans l'étude, et ce notamment durant les fenêtres critiques du développement de l'homme (petite enfance, enfance et adolescence). La détermination de l'OCS dans le temps représente aussi un intérêt majeur pour évaluer et comprendre l'évolution des territoires (*e.g.*, artificialisation) et mettre en place des politiques publiques. A titre d'exemple, Picuno et *al.* [PCS19] arguaient en 2019 que l'analyse de l'environnement rural résultant des activités humaines représente une source d'information incomparable pour estimer l'état de l'environnement. Il s'agit donc ici de déterminer les propriétés des données disponibles dans notre cadre de travail afin de justifier les choix techniques réalisés. Dans cette section, nous allons ainsi voir quelles sont les données disponibles pour obtenir des cartes d'OCS recoupant notre période d'intérêt au travers des programmes d'annotations existants, avant de nous intéresser aux images disponibles. Par souci de concision, nous exclurons ici les données que nous qualifierons de récentes, telles que le Registre Parcellaire Graphique dont les premières données ont été générées en 2002 à partir des déclarations de surfaces agricoles faites par les agriculteurs. Pour ce qui est des images (non annotées) disponibles, nous nous focaliserons sur les données visuelles acquises par un dispositif d'imagerie aérien ou satellite. Ces données sont en effet régulièrement utilisées pour générer des OCS par photo-interprétation manuelle, et ont pour avantage de permettre une vérification visuelle des résultats. De la même manière, nous exclurons les programmes d'observations ayant débuté après 1990, tels que le programme spatial Franco-Italien Pléiades, lancé en 2001 (premier satellite en orbite en 2003) ou le programme Sentinel lancé en 2007 (premier satellite en orbite en 2014).

1.2.1 Occupation du sol

Corine Land Cover (CLC)

	CLC1990	CLC2000	CLC2006	CLC2012	CLC2018
Données satellites	Landsat-5 MSS/TM	Landsat-7 ETM	SPOT-4/5 et IRS P6 LISS III	IRS P6 LISS III et RapidEye	Sentinel-2 et Landsat-8 pour combler les trous
Dates d'acquisitions	1986-1998	2000 +/- 1 an	2006 +/- 1 an	2011-2012	2017-2018
Durée de production	10 ans	4 ans	3 ans	2 ans	1.5 ans
Précision géométrique des données satellites	≤ 50 m	≤ 25 m	≤ 25 m	≤ 25 m	≤ 10 m (Sentinel-2)
Taille d'élément minimale (sortie)	25 ha	25 ha	25 ha	25 ha	25 ha
Taille minimale (sortie)	100 m	≤ 100 m	≤ 100 m	≤ 100 m	≤ 100 m

TABLEAU 1.1 – Métadonnées correspondant au progamme Corine Land Cover (CLC).

Corine Land Cover (CLC) [9] est un jeu de données à l'échelle européenne incluant actuellement 38 pays pour 5.8 millions de kilomètres carrés de surface, représentées à l'aide de 44 classes d'occupation du sol. Il est réalisé dans le cadre du programme européen Copernicus, lancé par l'Agence européenne pour l'environnement et visant à la surveillance des terres européennes. La génération des données de CLC est standardisée et se fait par photo-interprétation humaine d'images satellites. Les résultats sont obtenus au format vectoriel, incluant la notion d'objets polygonaux complexes (par opposition au format raster, où l'unité de base correspond au pixel du capteur). Le tableau 1.1 retranscrit une sélection de métadonnées concernant le programme CLC, où la ligne Données satellites référence les satellites utilisés pour générer les annotations. Nous fournissons plus d'informations quant aux programmes satellites d'observation de la terre dans la section 1.2.2. D'un point de vue temporel, la génération des données de CLC a débuté en 1985,

9. `https://land.copernicus.eu/pan-european/corine-land-cover` (accès : 2020-02-10)

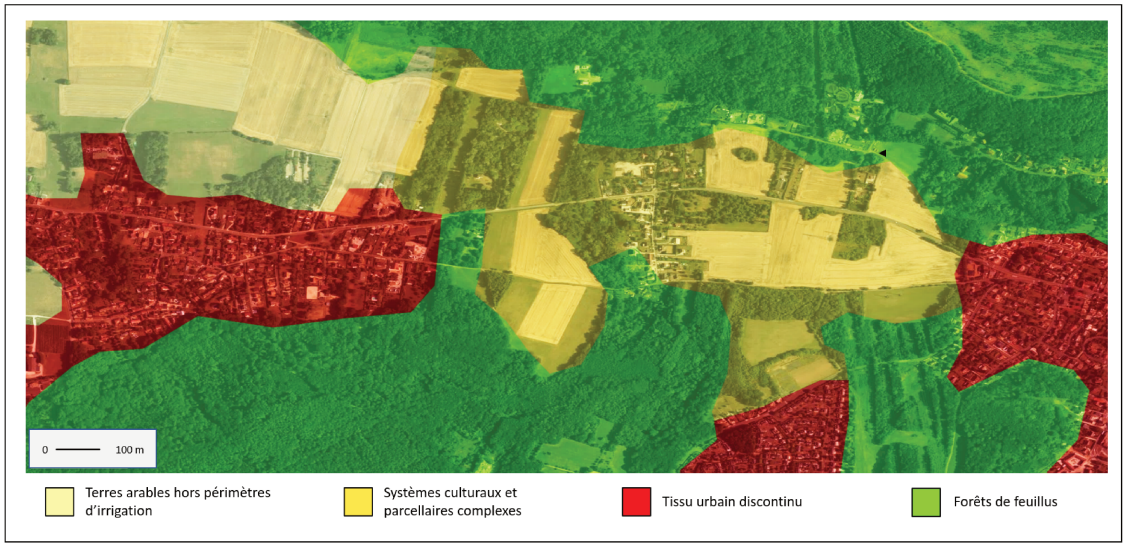


FIGURE 1.8 – Exemple d'annotations issues du jeu de données CLC 2018 en transparence avec une photographie aériennes en couleurs récentes. Données issues du géoportail [IGN20a]. On constate au centre que certaines cultures sont regroupées avec des zones forestières afin de former une zone d'au moins 25 hectares.

mais les premières données disponibles ne correspondent qu'au millésime de l'année 1990, avec une fréquence de publication des annotations relativement faible pour notre période d'intérêt (10 ans entre 1990 et 2000). Cette fréquence empêche la mise en place d'analyses précises temporellement. On constate par ailleurs que les données disponibles au sein de CLC n'incluent que des zones composées de 25 hectares ou plus, ce qui est relativement peu précis par rapport aux zones d'occupation étudiées dans le cadre de TESTIS. Concernant les éléments linéaires, tels que les routes ou les ponts, seuls sont retenus ceux qui ont une taille d'au moins 100 mètres. Un exemple d'annotations issue de CLC 2018 au niveau de la commune de Launay, près de Elancourt dans le département des Yvelines (France), est présenté dans la figure 1.8. On peut observer sur cette figure le regroupement d'objets sémantiquement différents au sein de même zones. Par exemple, des éléments urbains isolés se retrouvent annotés comme faisant partie de la forêt. L'intermittence de champs et de forêts, au centre de l'image, se retrouve annotée comme une unique zone nommée systèmes culturaux parcellaires complexes, faisant perdre l'information de localisation de chacun des éléments qui composent la zone. Pour rappel, cette information est importante dans le cadre de TESTIS afin de permettre l'intégration des vents dominants dans le calcul des scores d'expositions aux pesticides d'origine agricole. Il est cependant important de remarquer que l'ensemble des annotations fournies par CLC sont d'une résolution inférieure (25 hectares) à celle des données générées par les satellites disponibles à chaque époque (pixels de 50 mètres ou moins pour CLC1990). Ce fait permet d'envisager l'obtention de données de meilleures résolutions à partir des images exploitées pour générer les données de CLC.

HIstoric Land Dynamics Assessment (HILDA)

HILDA [FHV+14; FVCH15] est un jeu de données d'occupation du sol généré automatiquement à l'aide d'un modèle mathématique basé sur des flux de données variés et harmonisés. Il a été développé entre 2010 et 2015 par le laboratoire d'information géospatiale et de télédétection (*Laboratory of Geo-information Science and Remote Sensing*) de l'Université de Wageningen aux Pays-Bas dans le cadre du projet GHG Europe [10], visant à améliorer les capacités de prédiction des émissions de gaz à effet de serre sur le territoire européen (*Greenhouse Gases*, GHGs). Les auteurs de HILDA décrivent leur modèle comme un concept pour la reconstruction historique de l'occupation du sol et de ses changements. Il exploite en effet des données telles que des inven-

10. `http://www.europe-fluxdata.eu/ghg-europe` (accès : 2020-04-03)

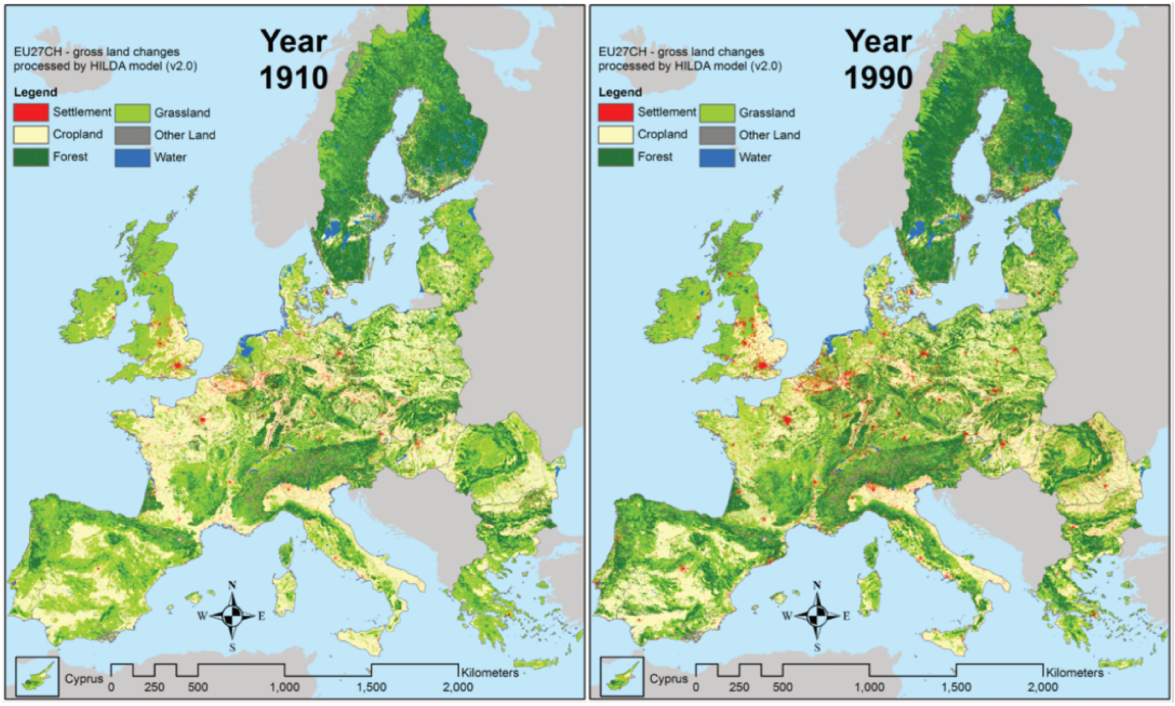


FIGURE 1.9 – Exemple d'annotations issues du jeu de données HILDA [FHV+14; FVCH15] pour les années 1910 et 1990. On constate une diminution des cultures au profit des zones forestières et urbaines. Images extraites du site web du laboratoire de géoinformation et de télédétection de Wageningen [oGiSS20].

taires nationaux, des statistiques d'occupation, des images aériennes historiques, et des données archivées. La fusion de ces informations hétérogènes permet au modèle d'estimer non seulement les tendances de changements d'occupation du sol sur de grandes périodes de temps, mais aussi d'intégrer l'aspect spatial correspondant afin de répondre à la question de la localisation de ces changements. D'un point de vue numérique, HILDA possède une résolution spatiale d'un kilomètre carré et une résolution temporelle de 10 ans entre deux cartes d'occupation du sol sur la période 1900-2010, et ce pour une grande partie du territoire européen. D'un point de vue sémantique, un total de 6 classes sont représentées, à savoir : les zones urbaines, les cultures (incluant les zones d'arboriculture), les zones forestières, les prairies, les zones aqueuses, et d'autres terrains (*e.g.*, glaciers, plages). Une illustration de HILDA est présentée sur la figure 1.9, montrant l'évolution globale du paysage au niveau européen entre 1910 et 1990. HILDA représente donc un jeu de données particulièrement pertinent pour des études environnementales liées à l'occupation du sol, et ressemble à ce dont le projet TESTIS aurait besoin. Cependant, la résolution spatiale est *a priori* trop faible par rapport aux zones de rayon 1.5 km étudiées. Par ailleurs, la résolution temporelle de 10 ans dans HILDA induirait une incertitude supplémentaire quant aux individus nés entre deux décennies. Enfin, les classes d'occupation du sol ne distinguent pas les différentes cultures qui pouvaient être présentes à un lieu et à un instant donné, et cet aspect semble nécessaire dans le cadre de TESTIS afin de pouvoir estimer les expositions environnementales aux pesticides d'origine agricole.

1.2.2 Images satellites

Chronologie abrégée des satellites

Cette section présente un bref historique des dispositifs spatiaux utilisés pour l'observation de la terre. Elle fait office d'introduction aux programmes d'observation modernes intersectant notre période d'intérêt, que sont Landsat et Système Probatoire d'Observation de la Terre (SPOT) (voir sous-sections ci-après). Par souci de clarté, on exclura donc les programmes de télécommunication et d'exploration de l'espace et de ses planètes (e.g., Pioneer, Luna, *etc.*). De la même manière,



FIGURE 1.10 – Première photographie acquise par le satellite Explorer 6 en 1959 montrant une zone ensoleillée de l'océan pacifique survolée par un nuage. Image extraite de la base publique d'images de la NASA [NAS20a].

seuls certains satellites d'observation seront mis en avant (non-exhaustivité). Les premiers satellites artificiels ont été mis en orbite dans les années 1950, avec les succès des démonstrateurs Spoutnik 1 en et Spoutnik 2 achevés en 1957 par l'Union Soviétique. Ils ont permis aux scientifiques de l'époque d'étudier l'ionosphère par l'envoi de signaux radios. Ils furent rapidement suivis en 1958 par Explorer 1, satellite conçu par les Etats-Unis (USA). Les détecteurs à radiations (compteurs Geiger) installés sur Explorer 1 permirent la découverte de la ceinture de Van Allen, une zone où les particules énergétiques chargées émises par les vents solaires sont capturées par le champ magnétique terrestre. L'année suivante, en février 1959, les Etats-Unis ont mis en orbite Vanguard-2, le premier satellite météorologique de l'histoire, qui avait pour but de mesurer l'activités solaire réfléchie et la couverture nuageuse à la surface de la terre à l'aide de caméras optiques. Celui-ci eu un succès en demi-teinte dû à une erreur de positionnement de sa caméra. Quelques mois plus tard, le satellite Explorer 6 fut mis en orbite, transmettant les premières photographies de la Terre depuis l'orbite (voir Figure 1.10).

En 1960, le satellite météorologique TIROS-1 fût mis en orbite dans un état de fonctionnement opérationnel, contrairement à Vanguard-2. Le satellite Discoverer 13, lancé lui aussi en 1960, fût le premier satellite de reconnaissance, aussi appelé satellite espion, mis en orbite à avoir permis l'observation de la terre. En 1964, le satellite Nimbus 1, constituant le début de la deuxième génération des satellites météorologiques américains, fût déployé. Équipé d'une caméra en lumière visible et d'une caméra infrarouge, il permit notamment d'observer le trou de la couche d'ozone en cours de formation. Au total, 7 autres satellites Nimbus furent lancés entre 1964 et 1978. Du côté de la France, le premier satellite mis en orbite par le Centre National d'Études Spatiales (CNES) fût Astérix en 1965, qui prit la forme d'un démonstrateur technologique. A noter que l'histoire des satellites du CNES est détaillée sur un site web interactif [11]. Outre les satellites du programme Nimbus, à visée météorologique, les premiers satellites dédiés à la télédétection et à l'observation des sols hors applications militaires - ou du moins, les plus marquants à ce jour - furent issus du programme américain Landsat, lancé en 1972, et du programme français SPOT, lancé 14 ans plus tard en 1986.

Programme Landsat

Le programme Landsat a débuté en 1972 avec le lancement de Landsat 1, le premier satellite public dédié à l'observation des terres. Il continue encore de nos jours, avec le lancement de Landsat-9 prévu pour l'an 2021. Le programme est généralement décrit par générations successives, correspondant aux technologies embarquées dans les satellites. Ces différentes générations

11. `https://wax-o.com/demo/satellites/` (accès : 2020-03-28)

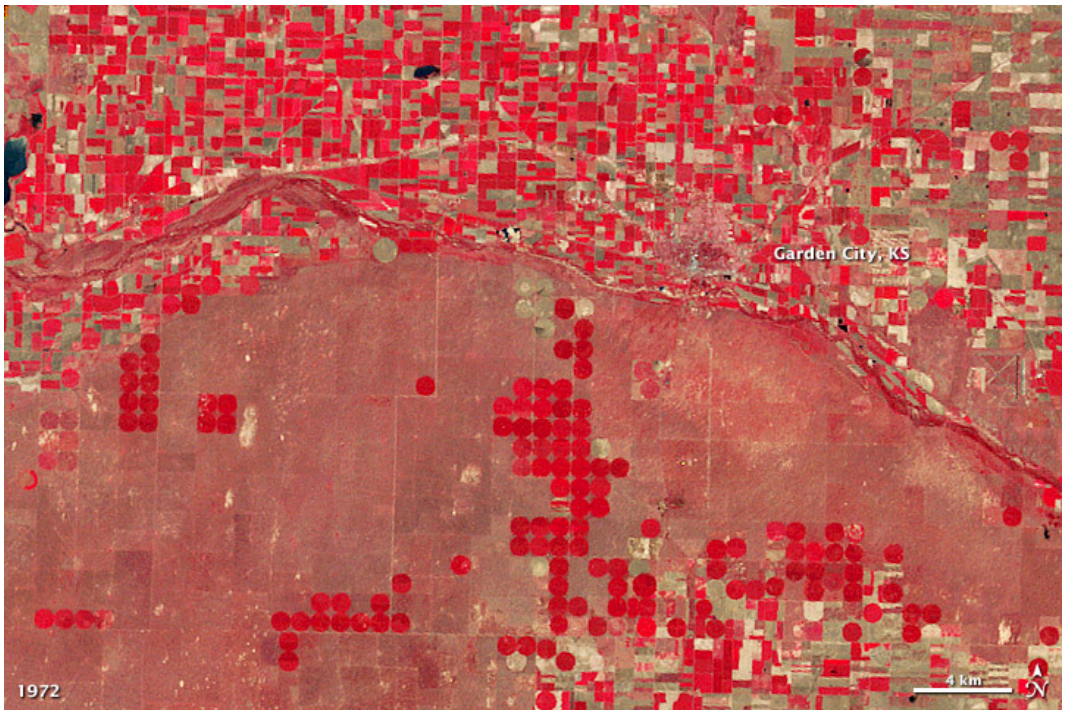


FIGURE 1.11 – Image Landsat-1 multispectrale en fausses couleurs au niveau de Garden City, Kansas, USA. Celle-ci met en avant la végétation, qui apparait en rouge. Image extraite du site web LandsatLooks de la NASA [NAS20b].

sont décrites ci-après. La première génération du programme Landsat [12] était à visée expérimentale. Elle était constituée des satellites Landsat-1 (de 1972 à 1978), Landsat-2 (de 1975 à 1982) et Landsat-3 (de 1978 à 1983), tous très similaires au niveau des dispositifs d'observation embarqués. Ils avaient pour but principal de démontrer la faisabilité de ce type d'observations depuis l'espace à l'aide de différents capteur. Ils embarquaient plusieurs instruments, dont une caméra RBV [13] (*Return Beam Vidicon*) et un capteur multispectral MSS [14] (*Multi Spectral Scanner*). Le dispositif RBV était en réalité constitué de trois caméras de télévision, chaque caméra capturant des bandes spectrales différentes (bande 1 : bleu-vert, bande 2 : jaune-rouge, bande 3 : proche infrarouge, NIR). Les trois caméras étaient alignées de façon à pouvoir mettre en correspondance les prises de vue par transformation géométrique. Le capteur MSS permettait quant à lui d'acquérir des bandes spectrales aux longueurs d'ondes spécifiques, à savoir du vert, du rouge, deux bandes de proche infrarouge. La résolution spatiale des pixels du MSS était de 79 x 57 mètres, ramenée à 60 mètres après traitement. Un exemple d'image multispectrale acquise en 1972 par le MSS de Landsat-1 est présenté sur la Figure 1.11 en fausses couleurs.

La seconde génération de satellites LandSat inclut les satellites Landsat-4 (de 1982 à 1993) et Landsat-5 (de 1985 à 2013) [15]. Cette génération est la première à être considérée comme étant en phase opérationnelle (contre expérimentale pour la précédente). Le principal changement par rapport aux modèles précédents est la disparition de la caméra RBV au profit d'un dispositif de cartographie thématique (*Thematic Mapper*, TM), en complément du capteur MSS déjà présent sur Landsat-1-3. Les bandes spectrales du TM ont un recouvrement spectral avec le MSS, auxquelles

12. https://directory.eoportal.org/web/eoportal/satellite-missions/l/landsat-1-3 (accès : 2020-04-03)
13. https://earth.esa.int/web/sppa/mission-performance/esa-3rd-party-missions/landsat-1-7/rbv/ (accès : 2020-04-03)
14. https://earth.esa.int/web/sppa/mission-performance/esa-3rd-party-missions/landsat-1-7/mss/ (accès : 2020-04-03)
15. https://directory.eoportal.org/web/eoportal/satellite-missions/l/landsat-4-5 (accès : 2020-04-03)

viennent s'ajouter des infrarouges à ondes courtes et un capteur thermique. De plus, la résolution spatiale du TM est de 30 mètres, deux fois supérieure à celle du MSS, pour l'ensemble des bandes à l'exception des données thermiques (120 mètres, ramenés à 30 mètres après traitement). Après le lancement raté de Landsat-6 en 1993 [16], la troisième génération de satellites Landsat vu le jour en 1999 avec le lancement de Landsat-7 [17], toujours en orbite. Ce satellite embarque un TM amélioré, ainsi que des capteurs panchromatiques d'une résolution deux fois supérieure (15 mètres) qui faisaient défaut aux précédents satellites du programme. Landsat 8 [18] fût quant à lui lancé en 2013, proposant un nouveau capteur augmentant le nombre de bandes spectrales disponibles pour l'observation des sols. Pour l'ensemble des satellites Landsat, la résolution temporelle d'acquisition d'images pour une même aire géographique était de 18 jours pour Landsat-1-3, et 16 jours pour les autres.

Programme SPOT

Le programme SPOT a été lancé en 1986 par le CNES. La première génération de satellites SPOT inclut les satellites SPOT-1, lancé en 1986, Spot-2, lancé en 1990, et SPOT-3, lancé en 1993. Les satellites SPOT de première génération étaient initialement prévus pour avoir une durée de vie de plusieurs centaines d'années. Ils ont cependant été désorbités en 2003 (SPOT-1) et 2009 (SPOT-2) afin de les laisser se désagréger dans l'atmosphère, mettant fin à leurs missions par la même occasion. Le satellite SPOT-3 a quant à lui arrêté de fonctionner en 1996. Ils étaient tous les trois dotés de capteurs visuels de hautes résolutions, à savoir un capteur panchromatique d'une résolution de 10 mètres permettant de couvrir le domaine visible, et un capteur multispectral sur 3 bandes permettant d'acquérir du vert, du rouge et du proche infrarouge avec une résolution de 20 mètres. A noter que cette combinaison de bandes spectrales permet d'avoir une estimation intéressante des indices de végétations tels que l'Indice de végétation par différence normalisée (*Normalized Difference Vegetation Index*, NDVI). Ces résolutions sont à comparer avec celles proposées par les satellites Landsat à la même époque (fin des années 1980, début des année 1990), qui étaient de 30 mètres au mieux. La deuxième génération est constituée du satellite SPOT-4, lancé en 1998. Aux bandes spectrales déjà présentes sur les satellites SPOT précédents viennent s'ajouter une bande dédiée aux moyens infrarouges. Les moyens infrarouges sont utiles pour détecter les nuages bas, mesurer les températures de surface pendant la nuit et pour détecter les incendies de forêt (voir cours *Suivi de l'environnement par télédétection* proposé par l'Université Virtuelle Environnement et Développement Durable (UVED) [eDDU20]. La troisième génération est uniquement constituée de SPOT-5, lancé en 2002. Il permet d'acquérir des images de 2 à 4 fois plus résolues que ses prédécesseurs, avec une résolution de 2.5 mètres ou 5 mètres pour les images panchromatiques (en fonction du mode de fonctionnement) et de 10 mètres pour les bandes multispectrales. Il embarque par ailleurs un capteur dit de Haute Résolution Stéréoscopique pour l'acquisition de couples d'images dédiés à l'estimation de la profondeur, qui représente ici l'élévation des objets au sol. La quatrième génération du programme SPOT est constituée des satellites SPOT-6 et SPOT-7, lancés respectivement en 2012 et 2014. Ces derniers permettent d'atteindre une résolution de 1.5 mètres pour les images panchromatiques et les images couleurs, et de 6 mètres pour les images multispectrales, avec une emprise au sol de 60 kilomètres par 60 kilomètres.

Accès aux données

L'accès aux données satellites a longtemps été un enjeu économique important. Depuis quelques années, un certain nombre d'images sont publiées gratuitement pour permettre leur utilisation

16. https://directory.eoportal.org/web/eoportal/satellite-missions/l/landsat-6 (accès : 2020-04-03)
17. https://directory.eoportal.org/web/eoportal/satellite-missions/l/landsat-7 (accès : 2020-04-03)
18. https://directory.eoportal.org/web/eoportal/satellite-missions/l/landsat-8 (accès : 2020-04-03)

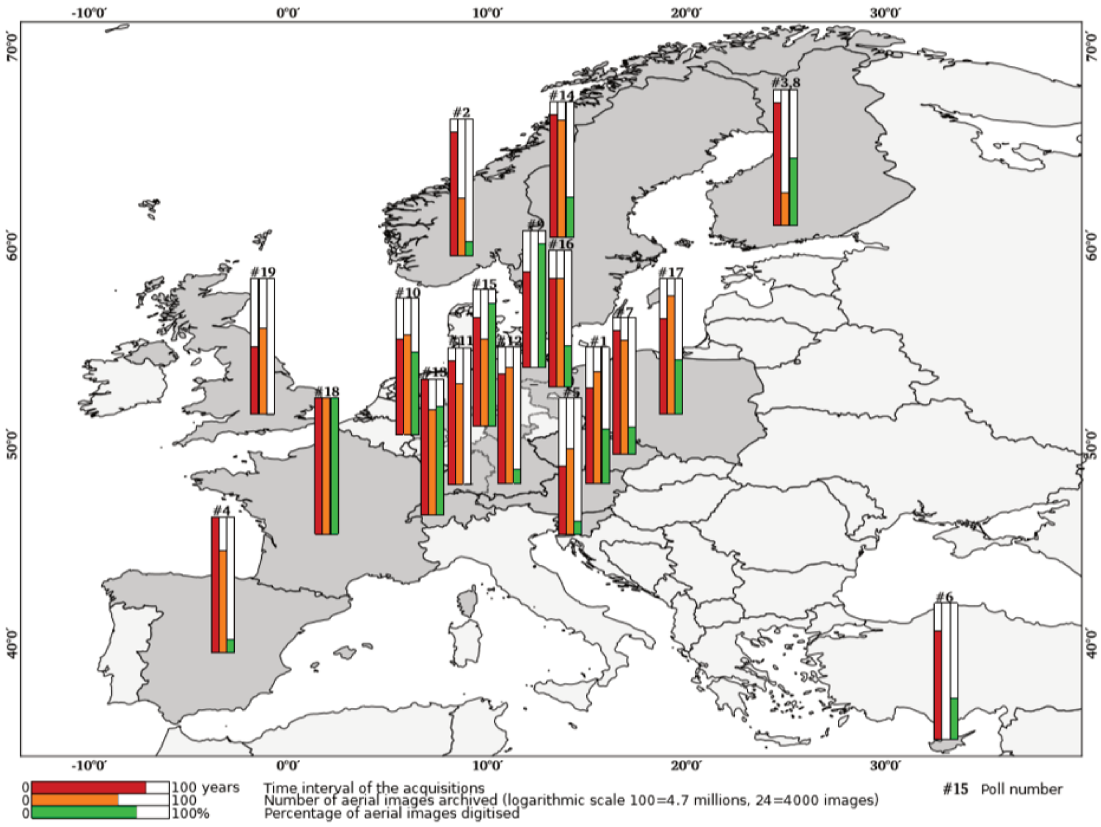


FIGURE 1.12 – Etat des lieux des images aériennes disponibles en Europe. Image issue de [GM19].

par les différentes communautés scientifiques. Pour l'accès aux données Landsat, le site web américains LandsatLook Viewer, décrit comme un prototype au 2020-03-29, permet d'accéder aux données d'archives depuis un navigateur web [19]. Les données sont proposées avec une référence géographique pour l'ensemble des modalités proposées (couleurs composites, thermique). Pour les données plus récentes, le pôle de données et de services surfaces continentales français Theia, créé en 2012 et regroupant aujourd'hui 11 institutions publiques françaises impliquées dans l'observation de la terre et les sciences de l'environnement [20], propose un portail [21] permettant d'accéder, entre autres, aux images Landsat à partir de Landsat-5, ainsi qu'aux images du programme SPOT. Il propose également l'accès à des données plus récentes, telles que celles acquises par les programmes Pléiades et Sentinel, non présentés ici.

1.2.3 Images aériennes

État des lieux en Europe

Plusieurs pays Européens possèdent des archives d'images aériennes analogiques et numériques utilisées pour administrer les territoires. Dans le cadre du projet ANR HIATUS (Historical Image Analysis for Territory evolUtion Stories), l'IGN français a réalisé un sondage [GM19] auprès de 19 organisations européennes, représentant un total de 13 pays, afin de proposer un état des lieux des données d'archives disponibles. Les résultats du sondage donnent un aperçu aussi bien quantitatif que qualitatif des données acquises et numérisées en Europe. Ils mettent en avant les stratégies utilisées par chaque organisation ayant répondu concernant les modalités d'acquisition des données (*e.g.*, type d'acquisition, but des acquisitions, consignes suivies), l'usage des données

19. `https://landsatlook.usgs.gov/` (accès : 2020-04-06)
20. `https://www.theia-land.fr/pole-theia-2/` (accès : 2020-04-06)
21. `https://theia.cnes.fr/atdistrib/rocket/#/home` (accès : 2020-04-06)

issues des campagnes d'acquisitions aériennes (*e.g.*, documentation, interprétation visuelle, génération de données topographiques), le niveau d'avancement dans la numérisation des images analogiques, ainsi que la mise à la disposition des ces données au grand public. On y apprend que, à date du sondage (2019) et parmi les organisations ayant répondu, l'IGN est l'organisation avec le plus grand nombre d'images aériennes disponibles (4.7 millions), possède des données sur les 100 dernières années ; tout comme les organisations nationales suisse et espagnole ; et était la seule organisation à avoir intégralement numérisé ses données analogiques (voir figure 1.12). Ce dernier point met en avant les difficultés rencontrées pour numériser ces données. Celles-ci ont été discutées lors de l'atelier de travail *Geoprocessing and Archiving of Historical Aerial Images* (littéralement, géotraitement et archivage des images aériennes historiques) en Juin 2019 à Paris [MGRT19]. Les difficultés principales évoquées par les participants semblaient être d'ordres logistique et économique. Les images analogiques d'archives sont en effet stockées dans des entrepôts sous conditions contrôlées pour éviter leur détérioration, avec l'utilisation de contenants tels que des boites à potentiel hydrogène nul [Wil19], ce qui limite l'accès à ces données matérielles. Il faut aussi noter que, au niveau européen, ces archives sont décentralisées dans 5.88% des cas [GM19], ce qui nécessite d'en maîtriser le transport.

La numérisation est quant à elle longue et coûteuse en ressources humaines, les images devant être scannées manuellement à l'aide d'un scanneur photogrammétrique dédié (*e.g.,* Leica DSW 700, Vexcel VX4000HT, Wehrli RM6) avant d'être géoréférencées en sein d'un SIG. Face à cette problématique, certaines organisations telles que la Collection Nationale de Photographies Aériennes (*National Collection of Aerial Photography*, NCAP) cherchent à partiellement automatiser l'étape de numérisation par la création et l'utilisation d'unités robotisées [Wil19]. Une fois les images numérisées et géoréférencées, vient alors le problème de la valorisation de ces données, pour lesquelles il faut trouver des applications permettant de financer le maintien des infrastructures mises en place.

Indépendamment de ces problématiques, les organisations ayant répondu au sondage [GM19] ont déclaré avoir majoritairement des images de hautes résolutions spatiales, avec des pixels variant de 10-20 centimètres à 1 mètre. Ces valeurs sont à opposer aux résolutions des images satellites disponibles au travers des programmes tels que Landsat (dizaines de mètres).

État des lieux en France

Les premières acquisitions d'images de la France vue d'en haut, puis mises à la disposition du grand public par la suite via le service remonterletemps [IGN20b], ont été réalisées en 1919. Pour cela, des appareils photographiques ; argentiques au départ, puis numériques par la suite ; ont été placés sur des dispositifs aériens chargés de suivre un tracé prédéfinit à vitesse et altitude données. La capture d'une prise de vue est déclenchée automatiquement à intervalle régulier. L'intervalle entre deux acquisitions est calibré de telle sorte qu'un recouvrement existe entre deux acquisitions successives afin d'assurer un suivi des acquisitions et générer des images en relief par stéréoscopie. Cette technique d'acquisition de données territoriales ayant fait ses preuves, elle a rapidement été généralisée. De nombreuses campagnes d'acquisitions aériennes incluant de multiples modalités ont ainsi été menées (couleur, infrarouge, optique, numérique). Les prises de vue aériennes continuent d'être utilisée de nos jours, et ce malgré l'apparition de satellites d'observation de la terre de plus en plus performants (voir section 1.2.2). Celles-ci ont pour avantage de permettre la génération d'images géographiquement et temporellement ciblées (*i.e.*, on peut acquérir de nouvelles images sur de nouvelles zones en fonction des besoins et des conditions atmosphériques). Les images aériennes ont par ailleurs des résolutions considérées comme étant élevées ou très élevées (inférieure à 1 mètre), et sur lesquelles peu de nuages sont présents. Ce dernier point s'explique par l'altitude relativement basse à laquelle les clichés sont obtenus par rapport aux données satellitaires, et par le fait qu'il est possible de commander la campagne de vol selon la météo (ajustement flexible). Concernant la mise à disposition de ces données, la France fait office de pays

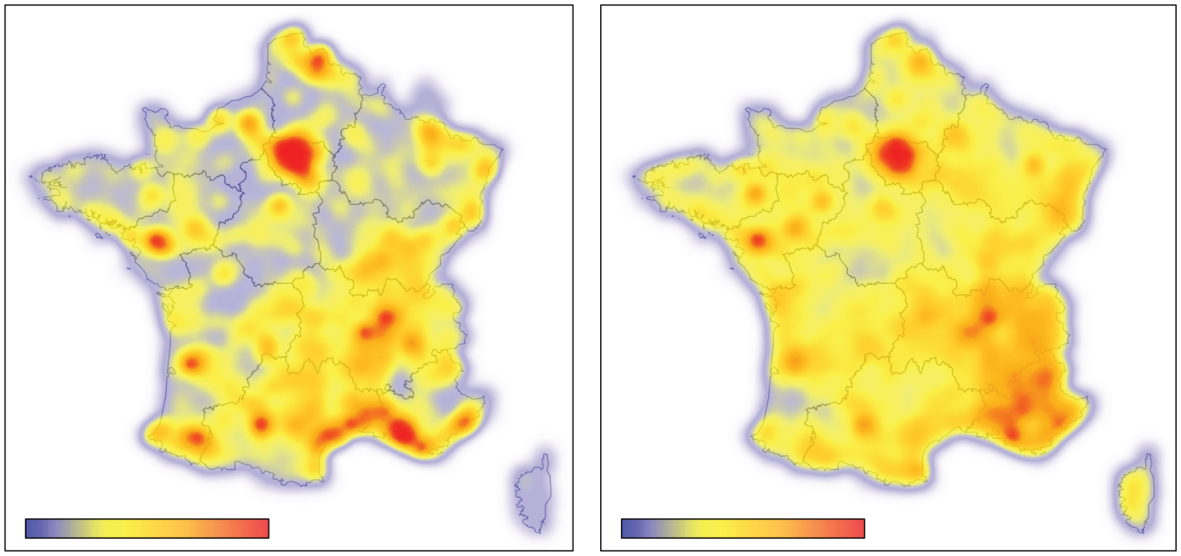


FIGURE 1.13 – Cartes de chaleur des acquisitions d'images aériennes par l'IGN (a) entre 1919 et 1970, et (b) entre 1970 et 2000.

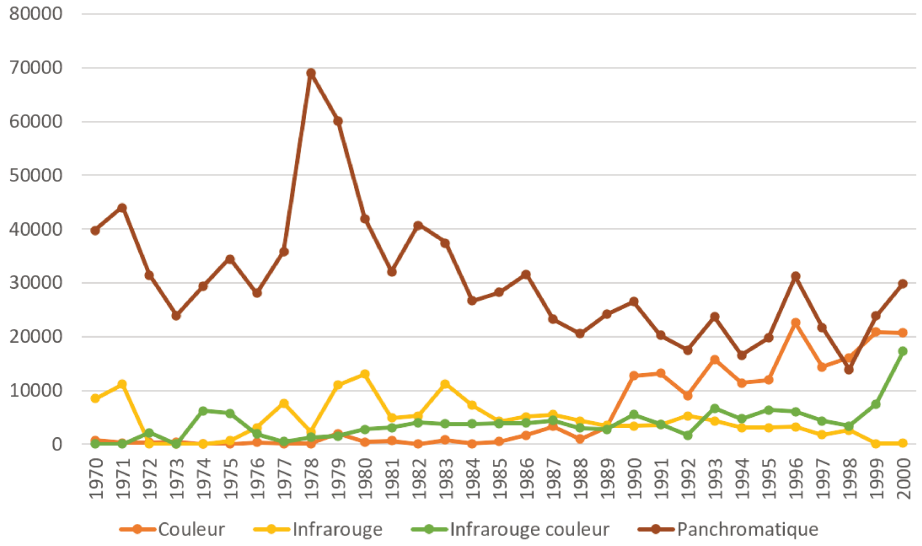


FIGURE 1.14 – Nombre relatif d'images aériennes disponibles sur remonterletemps [IGN20b] par type d'acquisition pour la période 1970-2000.

précurseur en Europe. La numérisation et la mise en ligne des données d'archives en France a été finalisée en 2016 par l'IGN.

Les métadonnées correspondant à ces prises de vues aériennes ont elles aussi été publiées en 2016 grâce au travail de Christian Quest [22]. Les métadonnées de ces images d'archives nous apprennent qu'un total de 19420 missions d'acquisitions ont été réalisées entre 1919 et 2010. A noter que cette estimation, réalisée à l'aide de la librairie geopandas [23] (version 0.6.1) exclut l'année 1941 due à des erreurs de lecture du fichier. La Figure 1.13 représente une carte de chaleur des acquisitions réalisées entre 1919 et 1970, et une carte de chaleur pour la période d'intérêt princi-

22. `https://www.data.gouv.fr/en/datasets/metadonnee-des-photos-aeriennes-anciennes-de-lign/` (accès : 2020-03-10)
23. `https://geopandas.org/`

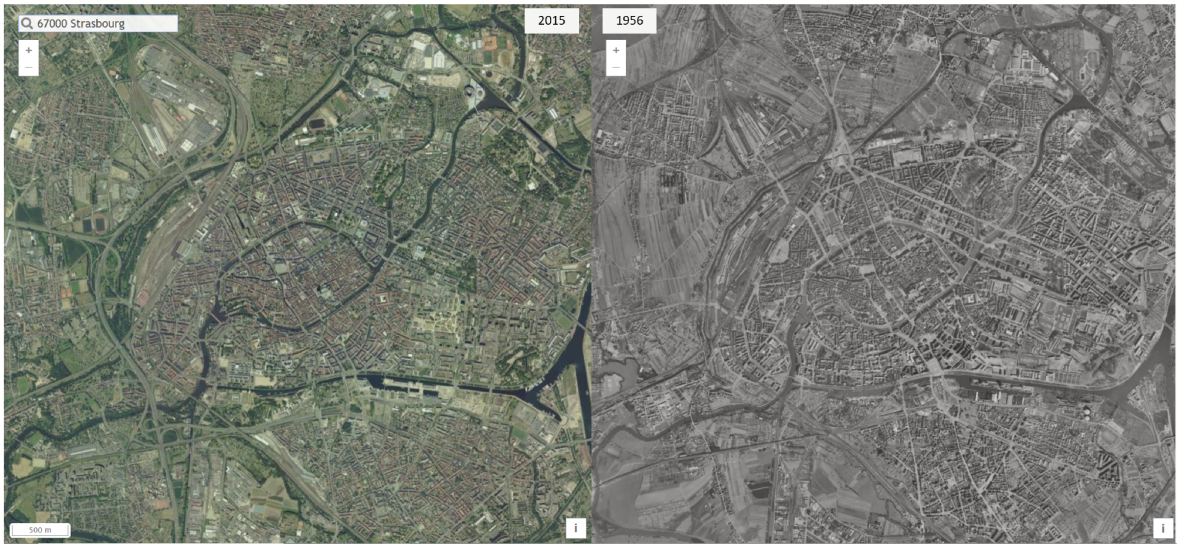


FIGURE 1.15 – Exemple d'une image aérienne historique de 1956 mis en correspondance avec une image aérienne récente de 2015 au niveau de Strasbourg, France. On constate l'urbanisation du territoire. Image extraite de remonterletemps [IGN20b].

pale dans le cadre de TESTIS (voir section 1.1.3), entre 1970 et 2000. On constate que ces cartes semblent corrélées spatialement aux lieux d'habitation des sujets inclus dans l'étude TESTIS (voir Figure 1.1), avec une proportion d'acquisitions en zones rurale plus importante après 1970. Il a cependant été constaté que la fréquence temporelle des acquisitions pour une coordonnée géographique donnée était relativement faible (entre 1 et 3 ans, fréquence irrégulière) sur cette période. Par ailleurs, les métadonnées nous apprennent qu'un total de 948 813 images panchromatiques, 184 758 images en couleurs, 139 552 images infrarouges et 122 144 images en infrarouges couleurs on été acquises sur cette période, mettant en avant la forte prédominance des acquisitions panchromatiques, et ce en particulier avant 1990 (voir Figure 1.14). Un exemple d'image aérienne historique panchromatique à côté d'une image aérienne récente issues du service remonterletemps est présenté sur la Figure 1.15, mettant en avant la diversité des représentations au cours du temps ainsi que l'évolution du paysage.

Données d'archives et applications

Outre l'héritage culturel d'intérêt public, il est à ce jour difficile d'estimer l'ensemble des applications potentielles des images aériennes d'archives. Certains organismes nationaux proposent un modèle économique basé sur la commercialisation des images, que ce soit sous forme de données numériques, de posters, ou de cartes en reliefs. On constate néanmoins certaines applications finales qui émergent grâce à la disponibilité de ces données. Parmi elles, Poli et *al.* [PSM+19] proposent de modéliser les glaciers en 3D à l'aide d'approches photogrammétriques et d'images aériennes historiques afin d'estimer et de visualiser la surface de recouvrement des glaces dans le temps. La connaissance de cette surface permet de mettre en avant les effets du réchauffement climatique sur les glaciers. Pinto et *al.* [PGBPH19] proposent d'utiliser des images aériennes historiques photo-interprétées afin de d'obtenir des informations environnementales dans le cadre d'études socio-écologiques. Les auteurs mettent en avant le fait que la photo-interprétation est un processus couteux mais fiable pour l'estimation des OCS. D'autres applications ont pour but de permettre la redécouverte des territoires ayant été modifiés, avec une approche à mi-chemin entre l'histoire et l'art, au sens où il s'agit de visualiser les territoires dans le passé. Dusanek et *al.* [DP19], à partir des travaux de Hodac et *al.* [HZ18], proposent ainsi d'identifier, de reconstruire et de visualiser des paysages du passé qualifiés de "perdus", en République Tchèque. De la même manière, Mazagol et *al.* [MNDR19] proposent de visualiser en 3D le patrimoine englouti de la Loire à partir d'images des années 1950. Ici, le patrimoine est dit englouti dû à la construction d'un barrage qui a

submergé la vallée, et il s'agit pour les auteurs de permettre la visite virtuelle de cette vallée avant la mise en place du barrage. Dans la même thématique, Kruse et *al.* [KRH19] et Ozdemir et *al.* [OR19] proposent des approches de vision par ordinateur pour inférer la position de cratères d'obus de la seconde guerre mondiale à partir d'images aériennes historiques afin de guider des équipes de déminage pour sécuriser les sols qui pourraient encore contenir des engins explosifs. Ces cratères sont considérés comme difficiles à détecter sur les images actuelles à cause des renouvellements des éléments présents au sol (*i.e.,* recouvrement par des zones urbaines, des cultures, ou autre). Gominski et *al.* [GPGBC19] explorent la possibilité de mettre en correspondance et d'associer des images actuelles et des images du passé, dont des images aériennes, afin de pouvoir les géolocaliser automatiquement. Enfin, nos travaux proposent d'estimer l'OCS à partir d'images aériennes historiques afin d'inférer les expositions environnementales aux pesticides sur une maladie avec une latence de 15 à 25 ans.

1.3 Problématique et positionnement

Nous avons vu quel était l'objectif du projet épidémiologique TESTIS, et nous avons mis en avant les données disponibles pour parvenir à estimer l'occupation du sol historique (OCS) afin d'estimer les expositions aux pesticides d'origine agricole. Cette section a pour but de clarifier le positionnement de cette thèse en informatique dans ce contexte pluridisciplinaire, et d'introduire les problématiques qui ont été abordées par nos travaux.

Nous avons vu dans la section précédente (voir section 1.2) que peu de données annotées existent sur la période d'intérêt du projet TESTIS (1970-2000). Parmi les images disponibles pour réaliser des annotations et générer des cartes d'OCS, nous rappelons que le choix réalisé par les géomaticiens pour ce projet s'est porté sur les images aériennes d'archives de l'IGN. Pour la période d'intérêt de TESTIS, la majorité de ces images ne sont disponibles qu'en niveaux de gris. Ces dernières contiennent moins d'informations spectrales que les données satellites disponibles à la même époque. Il a cependant été estimé que ces images étaient plus faciles d'accès que les données satellites du programme Landsat, l'IGN étant un organisme français qui garantit l'accès à ces données tout en proposant une interface adaptée [IGN20b]. De plus, les images aériennes possèdent une résolution bien supérieure aux premiers satellites Landsat, ce qui leur permet d'être plus aisées à interpréter par un être humain (représentation plus commune) bien que moins discriminantes spectralement. A noter que ce constat ne vaut pas pour les acquisitions satellites actuelles (*e.g.,* résolution de 1.5 mètres pour le programme SPOT).

L'analyse manuelle des images aériennes prend néanmoins beaucoup de temps, et représente un point bloquant dans le cadre de TESTIS. Une estimation grossière consisterait à considérer 1 image par adresse par sujet, soit 7623 (6.6 adresses × 1155 sujets) images à traiter pour l'ensemble de l'étude. En pratique, ce nombre est certainement plus important : il faut parfois plusieurs images pour couvrir la zone d'intérêt correspondant à une adresse, et il est parfois nécessaire d'analyser l'environnement d'une adresse à plusieurs dates différentes. Néanmoins, en supposant notre estimation grossière correcte, et en estimant le temps de traitement de chaque image à une demi-journée de travail, il faudrait l'équivalent de 3812 jours pour traiter l'ensemble des données. Le nombre de jours travaillés par an étant d'environ 220 en 2020 en France, il faudrait donc 208 homme.mois pour obtenir les résultats attendus. Face à ce constat, il semble nécessaire de proposer des solutions permettant de faciliter et d'accélérer l'étape d'annotation des images aériennes historiques, point qui constitue un véritable frein à la réalisation de TESTIS. Cependant, peu d'études se sont à ce jour intéressées à l'analyse du contenu des images aériennes historiques panchromatiques d'un point de vue vision par ordinateur, les efforts de la communauté étant principalement concentrés sur l'analyse des données actuelles et futures, porteuses d'informations temporelles et multispectrales dont la qualité et la quantité ne cessent de croître.

Dans ce cadre, nous avons consacré nos efforts au développement de méthodes originales de vision par ordinateur adaptées aux images aériennes historiques. A noter que nous n'avons pas travaillé avec des images acquises en vue oblique (*i.e.,* prise de vue non parallèle au sol). Nos travaux ont été réalisés en trois étapes.

- Dans un premier temps, nous nous sommes intéressés à la classification des différents types d'OCS à l'aide d'approches basées sur la texture et l'apprentissage profond [RCJF+19a] [RCJF+18] [RBCJT19]. On remarque en effet que les images aériennes permettent de visualiser le territoire sous forme de motifs similaires qui permettent à l'humain de distinguer différents types d'OCS (*e.g.,* forêts, zones urbaines). Nos résultats ont été intégrés au sein du logiciel Gouramic, proposant d'intégrer l'utilisateur dans la boucle pour la segmentation sémantique des images aériennes panchromatiques et présenté en annexe A [FRCJ+18; FRCJ+19].
- Dans un second temps, nous nous sommes intéressés à la colorisation automatique et à l'application de ce type d'approche aux images aériennes historiques. D'une part, nous avons cherché à combler le fossé visuel entre les acquisitions historiques panchromatiques et les acquisitions récentes en couleurs dans le but de faciliter l'annotation de ces images par les géomaticiens. D'autre part, nous souhaitions étudier l'intérêt de la colorisation comme étape intermédiaire pour la classification [RCJF+19b; RCJF+19c].
- Enfin, nous nous sommes intéressés au post-traitement des segmentations sémantiques des images aériennes historiques afin d'améliorer les résultats obtenus par les géomaticiens à l'aide du logiciel Gouramic. En particulier, nous avons étudié l'utilisation d'algorithmes de segmentation non supervisés (*clustering*) et de champs aléatoires conditionnels pour réduire les erreurs de classification et lisser spatialement les résultats obtenus [RCJF+20].

La suite de ce manuscrit découle directement de ces trois étapes. Le chapitre 2 présente les principaux éléments théoriques de la littérature sur lesquels nos travaux se sont basés. Le chapitre 3 présente nos travaux relatifs à la classification d'images aériennes historiques en différentes classes d'occupation du sol. Le chapitre 4 présente nos travaux portant sur la colorisation automatique. Le chapitre 5 s'intéresse au post-traitement des résultats obtenus par inférence pour les cartes d'occupation du sol. Le chapitre 6 conclut ce manuscrit et présente des perspectives qui nous semblent intéressantes pour la poursuite de nos travaux.

Chapitre 2

Notions de base

Ce chapitre introduit les méthodes de la littérature sur lesquelles nos travaux se sont basés. Il a pour but de fournir au lecteur un tour d'horizon des approches existantes afin de mieux situer les travaux que nous avons réalisés. Pour cela, nous traiterons d'abord des approches de traitement d'images et d'apprentissage automatique, que nous qualifierons ici de "classiques", avant de nous intéresser aux méthodes d'apprentissage "bout en bout", qui ont connu un regain de popularité ces dernières années. En particulier, nous aborderons les méthodes employées pour l'extraction de caractéristiques de textures à partir d'images numériques, avant de nous intéresser aux méthodes de sur-segmentation permettant de générer des groupes de pixels homogènes, aussi appelés segments ou objets en télédétection. Nous verrons ensuite les notions relatives aux réseaux de neurones profonds à convolutions, permettant d'optimiser simultanément les étapes d'extraction de caractéristiques et de classification. Nous présenterons également des exemples d'utilisation de ces méthodes sur des données de télédétection.

Sommaire

2.1 Extraction de caractéristiques de textures 26
2.1.1 La texture 26
2.1.2 Description de la texture 27
2.1.3 Application de la texture en télédétection 33
2.2 Sur-segmentation 34
2.2.1 Méthodes courantes 34
2.2.2 Application de la sur-segmentation en télédétection 38
2.3 Algorithmes de classification 39
2.3.1 Définitions 39
2.3.2 Algorithmes communs 40
2.4 Réseaux de neurones à convolutions 44
2.4.1 Blocs de base 46
2.4.2 Application des réseaux de neurones à convolutions en télédétection 49
2.5 Conclusion et positionnement 50

2.1 Extraction de caractéristiques de textures

Soit I une image numérique composée de pixels. On souhaite caractériser/résumer les informations contenues dans cette image. Pour cela, il est possible d'extraire des caractéristiques représentatives du contenu de I, telles que la texture et la couleur. Afin de les agréger dans une même structure de données, ces caractéristiques vont être stockées dans un vecteur de caractéristiques. Chaque valeur du vecteur de caractéristique représentera alors l'intensité du vecteur dans une direction de l'espace des caractéristiques extraites. Le fait de passer d'une source de données brute à un vecteur de caractéristiques (*i.e.*, d'extraire des caractéristiques) permet généralement de réduire significativement la complexité spatiale des images, tout en en faisant ressortir les éléments discriminants (*i.e.*, on extrait que les caractéristiques qui nous intéressent). Il s'agit, en général, d'une des premières étapes dans une chaine de traitement visant à identifier le contenu d'une image. Cette approche est représentée schématiquement sur la figure 2.1 avec des exemples de tâches à accomplir (*e.g.*, classification).

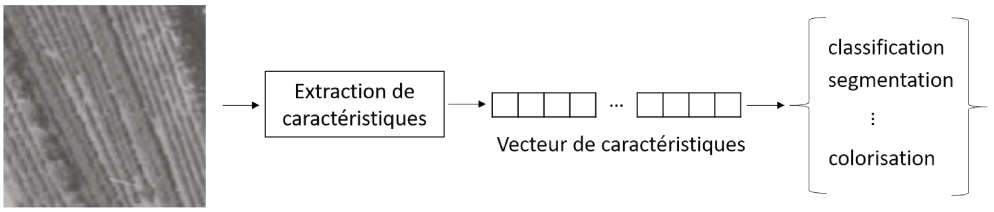


FIGURE 2.1 – Schéma générique de l'obtention d'un vecteur de caractéristiques à partir d'une image et de son utilisation pour différentes tâches. L'image de gauche a été extraite d'une image aériennes historique utilisée dans nos travaux.

2.1.1 La texture

Définition intuitive

La texture correspond aux variations d'intensités et de couleur visibles au sein des images. Ces variations forment des motifs visuels caractéristiques. Cette propriété fait de la texture une information particulièrement pertinente quand aucun *a priori* sur la forme des objets contenus dans l'image n'est connu, ou que cet *a priori* n'est pas jugé discriminant. A titre d'exemple, un cheval et un zèbre ont tous les deux des formes d'équidés qu'il serait aisé de confondre, mais les motifs représentés sur leurs pelages nous permettent de les distinguer sans difficulté [1]. Par analogies, il est également possible de comprendre la texture comme étant la surface d'un objet qu'il serait possible de reconnaître au toucher en considérant la couleur comme étant représentative de la chaleur émise par l'objet, et en considérant les variations d'intensités comme étant des variations de reliefs.

Exemples d'utilisation

De par les propriétés discriminantes de la texture pour l'œil humain, son analyse intéresse les chercheurs depuis plus d'un demi-siècle, durée symbolisée par les travaux précurseurs de Julesz [Jul62]. Depuis ces premiers travaux, les méthodes d'extraction de caractéristiques de textures développées au cours du temps ont joué un rôle prépondérant dans de nombreux domaines applicatifs, principalement pour résoudre des problèmes de classification, de segmentation, ou de synthèse d'images [LCF+19]. La figure 2.2 présente des exemples d'images de textures, mettant en avant la diversité des textures existantes qui font régulièrement l'objet de travaux de recherche.

1. A l'opposé, il est possible de confondre le pelage d'un chat avec la texture d'un tapis mœlleux, auquel cas la forme prend toute son importance.

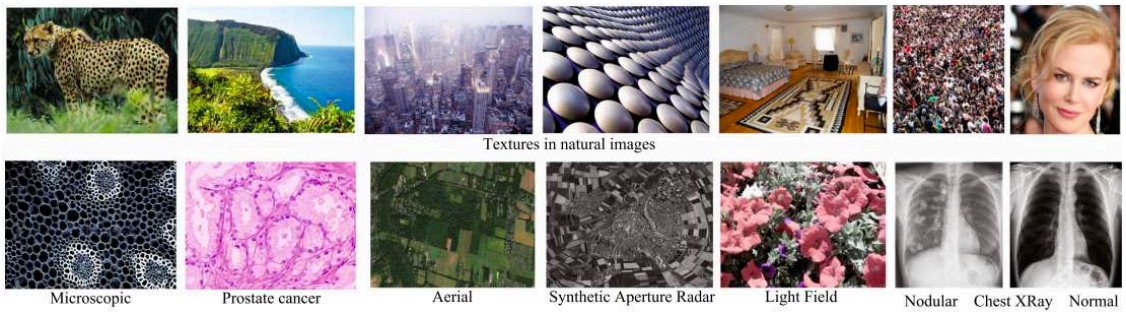


FIGURE 2.2 – Exemples de d'images texturées dans divers domaines d'applications. Image extraite de [LCF+19].

En biométrie, la texture a pu être utilisée pour reconnaître des empreintes palmaires [KZ02], des visage [AS93; RAHI13; TT10], ou encore identifier des individus par leur iris [LTYD03]. En imagerie médicale, les méthodes d'extraction de caractéristiques de textures ont montré leur intérêt pour obtenir des descriptions représentatives pour la machine [CTK15], comme pour l'être humain [LSS+17]. Dans le domaine de la reconnaissance automatique des végétaux, de nombreux travaux se sont intéressés à l'utilisation de la texture pour reconnaitre des espèces d'arbres à travers leurs écorces [PVMH14; BCT17; BAC+18]. En numismatique, la texture a pu être utilisée pour analyser les défauts visuels des pièces de monnaies pour la gradation automatique [Pan18]. En télédétection, les analyses basées sur la texture ont pu montrer leur intérêt pour l'estimation de l'occupation du sol à partir du ciel et de l'espace [ZY98; HW90; FLG15; AKvdW+18].

2.1.2 Description de la texture

La recherche en analyse de textures vise au développement de méthodes efficaces, et si possible robustes au perturbations, pour pouvoir représenter une image texturée à l'aide d'un vecteur de caractéristiques représentatif. La texture étant par définition un phénomène spatial lié aux variations d'intensité, les méthodes développées pour extraire des caractéristiques de textures s'attachent tout particulièrement à l'intégration de l'information disponible dans le voisinage d'un pixel. On parle alors de descripteurs locaux, qui, pour chaque pixel d'intérêt, génèrent des caractéristiques en se basant sur l'information portée par le voisinage du pixel. Ces caractéristiques locales sont ensuite agrégées pour représenter l'image entière à l'aide d'un unique vecteur de caractéristiques. Les agrégations les plus communes incluent l'utilisation de statistiques, d'histogrammes, de mise en commun (*pooling*), ou encore l'utilisation de textons (encodage à l'aide de groupes de caractéristiques) [LCF+19]. De façon générique, il s'agit ici de passer d'une représentation locale à une représentation globale de la texture. Dans un état de l'art étendu réalisé en 2019, Liu *et al.* [LCF+19] faisaient référence à ce type d'approche sous le terme de sac de mots (sous-entendu, visuels), par analogie avec les approches employées en traitement naturel du langage. Ici, chaque caractéristique correspondrait à un mot décrivant la texture.

Dans la suite de cette section, nous détaillons plusieurs types d'approches classiques utilisées pour l'extraction de caractéristiques de textures denses (*i.e.*, qui se basent sur l'ensemble des pixels de l'image) : matrices de cooccurrences, banques de filtres de Gabor, et motifs binaires locaux. Elles sont présentées ici car elles sont régulièrement utilisées en télédétection [WFZ+18; HCLD16] pour la classification de l'occupation du sol (voir sous-section 2.1.3). Nous porterons une attention particulière sur les méthodes basées sur les motifs binaires locaux [OPM01]. D'une part, ces méthodes ont montré leur capacité à générer rapidement des représentations discriminantes de relativement faibles dimensions [AFA+16]. D'autre part, nous les avons particulièrement étudiées dans le cadre de cette thèse. Les méthodes basées sur des réseaux de neurones profonds à convolutions sont quant à elles décrites en section 2.4, dans un contexte plus générique que celui de l'analyse de la texture.

Matrices de cooccurrences

Les travaux précurseurs de Julesz *et al.* [Jul62] suggéraient que la texture pouvait être modélisée à l'aide de statistiques représentant la cooccurrence des intensités de k paires de pixels. L'idée était ici de représenter la fréquence d'apparition de deux intensités pour caractériser les motifs présents au sein de l'image. Celle-ci fût reprise par Haralick *et al.* [HSD73; Har79] dans les années 1970, qui proposèrent alors un formalisme basé sur une matrice de cooccurrences de niveaux de gris (*Gray Level Cooccurrence Matrix*, GLCM).

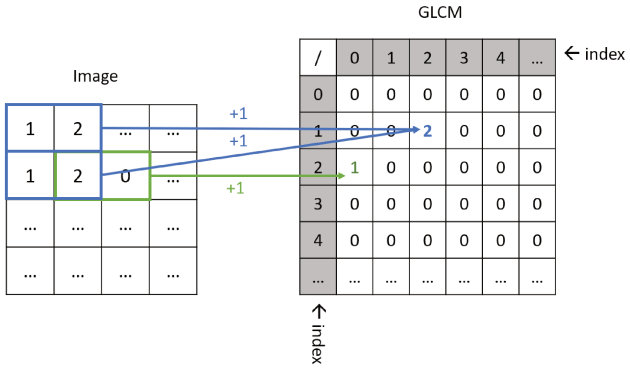


FIGURE 2.3 – Exemple de la construction d'une matrice de cooccurrences de niveaux de gris avec une distance $D_{(p_1,p_2)} = 1$ pixel et une orientation O_r horizontale.

La cooccurrence est ici à comprendre au sens où deux pixels p_1 et p_2 distants l'un de l'autre de $D_{(p_1,p_2)}$ pixels selon une orientation O_r prendront simultanément des valeurs d'intensité i_1 et i_2. La distance $D_{(p_1,p_2)}$ et l'orientation O_r définissent ici la relation d'adjacence entre les deux pixels (le voisinage). Pour représenter ce phénomène, le formalisme des GLCM définit un accumulateur à deux dimensions (un tableau), avec autant de lignes et de colonnes que de valeurs d'intensité possibles. Pour $D_{(p_1,p_2)}$ et O_r fixés, il s'agit alors de parcourir l'image, et d'ajouter une unité dans la cellule de l'accumulateur dont la ligne est définie par i_1 et la colonne définie par i_2. Dit autrement, on compte le nombre de fois où i_1 et i_2 apparaissent simultanément dans l'image selon le voisinage. Un exemple de construction d'une GLCM à partir d'une image est présenté sur la figure 2.3. Une fois la matrice de cooccurrence construite, celle-ci peut être utilisée comme base pour calculer des statistiques représentatives de la texture telles que l'énergie, l'entropie, le contraste, l'homogénéité, ou encore la corrélation. Ces statistiques sont ensuite concaténées au sein d'un même vecteur de caractéristiques.

Résumé des propriétés. Nous résumons ici les propriétés principales des matrices de cooccurrences (forces (+), faiblesses (-)) :

- (+) La méthode est relativement facile à implémenter
- (+) Les paramètres et caractéristiques calculées sont aisément compréhensibles par l'être humain
- (-) Complexité spatiale élevée des GLCM (nombre de pixels dans l'image au carré), qui augmente avec le nombre de voisinages considérés
- (-) Caractéristiques globalement moins discriminantes que celles des méthodes plus récentes (voir ci-après)

Filtres de Gabor

Parmi les filtres de textures existants, les filtres de Gabor sont probablement les plus populaires. Ce modèle, inspiré par la vision des mammifères, est particulièrement réputé pour per-

mettre de détecter des bords et des lignes à orientation et échelle variables. Chaque filtre de Gabor G est modélisé à l'aide d'une sinusoïde complexe modulée par un filtre gaussien. Dans le cas 2D, qui nous intéresse en traitement d'images, la partie réelle de ce filtre est représentée par l'équation (2.1), et la partie imaginaire par l'équation (2.2). Les paramètres a et b permettent de modifier la fréquence et l'orientation du filtre, tandis que σ^2 représente la variance de la gaussienne qui permet de faire varier l'échelle du filtre. Il est possible de définir différentes banques de filtres de Gabor en faisant varier ces paramètres [MM96; PS06].

$$G_1(x,y) = cos(ax+by) \times exp(-\frac{x^2+y^2}{2\sigma^2}) \tag{2.1}$$

$$G_2(x,y) = sin(ax+by) \times exp(-\frac{x^2+y^2}{2\sigma^2}) \tag{2.2}$$

En pratique, les filtres dans un banque de filtres de Gabor sont appliqués sur une image I à l'aide d'une convolution afin d'obtenir une image filtrée J. Une fois l'image J obtenue, des statistiques peuvent directement en être extraites, telles que la moyenne et la variance. Afin d'obtenir un représentation plus complète, il s'agit d'extraire ces statistiques à partir du résultat de chaque filtre, puis de concaténer le tout au sein d'un même vecteur de caractéristiques. A noter qu'il a été montré que, malgré la définition des filtres de Gabor à plusieurs orientations et échelles, leurs performances tendent à diminuer en présence de rotations, ou plus généralement de transformations affines [LCF+19; ZMLS07].

Résumé des propriétés. Nous résumons ici les propriétés principales des filtres de Gabor (forces (+), faiblesses (-)) :

- (+) Représentation multi-échelle
- (+) Formulation supposée robuste au bruit (filtre gaussien) et robuste aux rotations dans le plan
- (-) Nécessité d'utiliser beaucoup de filtres pour obtenir une représentation à plusieurs échelles et rotations (augmentation des temps de calculs)
- (-) Suppression des hautes fréquences (filtre gaussien) pouvant réduire la quantité de motifs détectés

Motifs Binaires Locaux

Les méthodes visant à décrire les motifs locaux à l'aide de codes binaires [OPM01] ont connu un fort engouement depuis leur apparition à la fin des années 1990 [PZ15; LCF+19]. Une taxonomie dédiée à ces méthodes a d'ailleurs été réalisée en 2017 [LFG+17], montrant la grande quantité de travaux qui leur ont été consacrés (voir tableaux 8 et 9 de [LFG+17]). Cet engouement s'explique de par la relative simplicité dans la formulation de ces approches, leurs propriétés d'invariances, leur faible complexité algorithmique et leur pouvoir discriminant pour l'analyse de textures comparé aux approches plus classiques [FÁB13], et ce notamment sur des données de télédétection [AFA+16]. Ici, nous présentons les fondamentaux liés aux méthodes basées sur les filtres de type Motifs Binaires Locaux (*Local Binary Pattern*, LBP) [OPM01], ainsi que les grandes lignes correspondant aux différentes extensions de cette approche. Nous reviendrons en détail sur les méthodes utilisées dans nos travaux et issues de cette catégorie de descripteurs dans le chapitre 3.

Principe. Les filtres basés sur les LBP sont des filtres locaux invariants aux changements d'intensité globaux. Ils permettent de calculer un code binaire local représentatif de l'information de texture en utilisant un voisinage circulaire de rayon R contenant P pixels g_p et centré sur un pixel central g_c. Il est possible de représenter un voisinage (P,R) à l'aide de coordonnées discrètes ou continues [OPM01]. Dans le premier cas, la valeur d'un pixel voisin est obtenue en considérant la valeur discrète la plus proche de la position réelle de l'élément sur le cercle de rayon R. Dans le

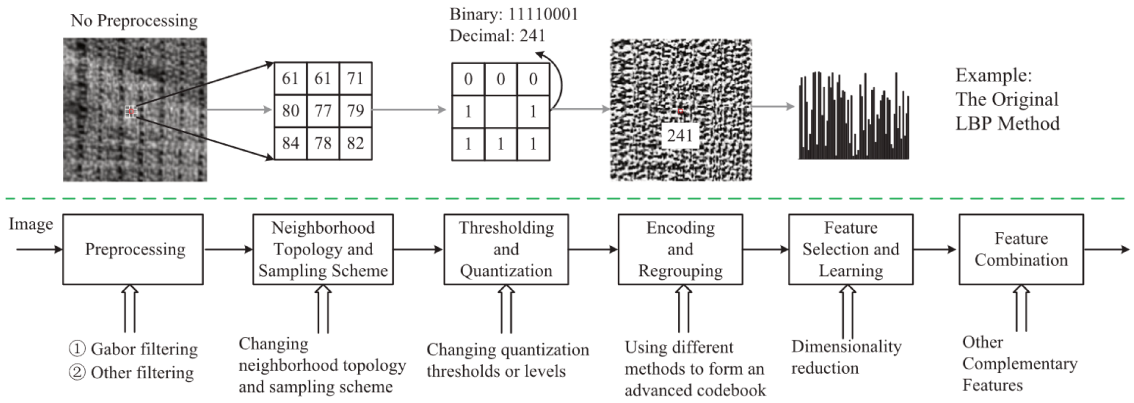


FIGURE 2.4 – Schéma représentant le pipeline générique pour l'extraction de caractéristiques à l'aide de LBP. Image extraite de [LFG+17].

second cas, la valeur d'un pixel voisin à la position réelle est obtenue par interpolation bilinéaire. Une fois le voisinage défini, il s'agit alors de déterminer les relations locales entre les intensités des pixels afin de représenter les motifs présents dans le voisinage. Pour cela, l'approche proposée par Ojala *et al.* [OPM01] consistait à estimer le signe de la différence entre le pixel central et les pixels ordonnés du voisinage. Pour chaque pixel g_p, si la différence entre g_p et g_c est positive, on concatène la valeur 1 au code binaire modélisant le voisinage, sinon la valeur 0. Ce nombre binaire est ensuite convertit en base 10 afin d'obtenir une valeur entière. Ce principe est représenté sur la figure 2.4 par l'étape seuillage et quantification (*Thresholding and Quantization*). D'un point de vue modélisation, le filtre classique de LBP [OPM01] est défini par l'équation (2.3).

$$\mathrm{LBP}_{\mathrm{P,R}} = \sum_{p=0}^{\mathrm{P}-1} s(g_p - g_c)2^p, s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \tag{2.3}$$

Le filtre LBP classique [OPM01] permet de représenter les motifs d'une image à l'aide de 2^{P} valeurs. Il est ici intéressant de constater que chaque valeur correspondra à un motif particulier au sens des différences locales qui auront été calculées, tels que des coins, des lignes, des points, ou des zones uniformes. En sortie d'un filtre de type LBP, on se retrouve donc avec une image en niveaux de gris. Afin de générer une représentation globale de cette image, l'approche standard consiste à générer un histogramme de l'image filtrée. Cet histogramme va représenter la probabilité, ou la fréquence, d'apparition d'un motif particulier. Il contiendra par défaut 2^{P} *bins* avec le LBP classique appliqué sur un unique voisinage. Il est par ailleurs possible de calculer les motifs à plusieurs échelles [OPM02]. Pour cela, il suffit de modifier le paramètre R et de concaténer les histogrammes résultants. En pratique, il est courant de faire varier R d'une unité, et de faire varier P par multiples de 8, ce qui permet de travailler sur des voisinages relativement denses (*e.g.*, $(\mathrm{P,R}) = \{(8,16,24),(1,2,3)\}$). Cependant, plus P est grand, plus le nombre de pixels g_p augmente, et plus les calculs sont longs. Il a de fait été suggérer de travailler à P constant, en faisant uniquement varier R [LYF+13] (*e.g.*, $(\mathrm{P,R}) = \{8,(1,2,3)\}$) afin de générer des représentations multi-échelles, certes moins représentatives, mais plus efficaces d'un point de vue algorithmique. A titre illustratif, différents voisinages (P,R) sont représentés sur la figure 2.5.

Mapping. En pratique, on constate que le LBP se base sur un voisinage circulaire, ce qui fait que de nombreux motifs binaires pourront être identiques à une rotation ou une permutation près. En se basant sur cette observation, plusieurs méthodes de mise en correspondance (*mapping*) ont été développées afin de réduire la taille des histogrammes générés et induire des propriétés supplémentaires [OPM02]. Parmi ces approches, les plus populaires sont l'uniformité u^2 (voir équation (2.6)), l'invariance à la rotation ri (voir équation (2.4)) et la combinaison des deux

riu^2 (voir équation (2.5)). Dans ces équations, l'opération ROR correspond à un décalage binaire circulaire réalisé sur le P-bit i fois, et $s(x)$ est définie par l'équation (2.3).

$$\mathrm{LBP}^{ri}_{\mathrm{P,R}} = min\left\{\mathrm{ROR}(\mathrm{LBP}_{\mathrm{P,R}}, i)\right\} \mid i = 0, 1, ..., \mathrm{P}-1 \tag{2.4}$$

$$\mathrm{LBP}^{riu^2}_{\mathrm{P,R}} = \begin{cases} \sum_{p=0}^{\mathrm{P}-1} s(g_p - g_c)2^p & si \quad \mathrm{U}(\mathrm{LBP}_{\mathrm{P,R}}) \leq 2 \\ \mathrm{P}+1 & sinon \end{cases} \tag{2.5}$$

avec $\mathrm{U}(\mathrm{LBP}_{\mathrm{P,R}})$ le degré d'uniformité défini par l'équation (2.6) :

$$\mathrm{U}(\mathrm{LBP}_{\mathrm{P,R}}) = |s(g_{\mathrm{P}-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{\mathrm{P}-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \tag{2.6}$$

D'un point de vue empirique, l'invariance à la rotation est définie de façon intuitive par rapport à l'aspect circulaire et symétrique du filtre. Elle permet de réduire la quantité de motifs pris en compte en considérant les motifs identiques à une rotation près comme représentant la même chose. Ojala *et al.* [OPM02] remarquaient ainsi que 36 motifs distincts et invariants à la rotation pouvaient être détectés à partir d'un voisinage $(\mathrm{P,R}) = (8,\mathrm{R})$. L'uniformité est quant à elle définie par rapport au nombre de transitions $\mathrm{U}(\mathrm{LBP}_{\mathrm{P,R}})$ qui vont apparaître sur le motif binaire. Ici, pour le *mapping* riu^2, le 2 indique que seuls les motifs binaires ayant au plus 2 transitions (de 0 à 1, ou de 1 à 0) seront considérés comme étant uniformes. Ces motifs uniformes représenteraient 90% des motifs apparaissant sur des images de textures [OPM02]. Une visualisation des 58 motifs uniformes pour un voisinage $(\mathrm{P,R}) = (8,\mathrm{R})$ avec le LBP classique est proposé sur la Figure 2.6. Dans ce formalisme, les motifs non-uniformes sont alors considérés comme ne représentant qu'un seul et même attribut, portant le nombre de motifs possibles à 59 pour ce même voisinage. A cela s'ajoute l'invariance à la rotation, permettant au final de réduire la quantité de *bins* à seulement $\mathrm{P}+2$ *bins* (9 motifs uniformes et invariants à la rotation, et 1 pour les motifs non uniformes) pour la *mapping* riu^2. Ces *mapping* ont deux avantages : (1) réduire la complexité spatiale de l'histogramme (vecteur de caractéristiques) et (2) ajouter des propriétés intéressantes aux filtres telles que l'invariance à la rotation. Cependant, il a été montré que le *mapping* peut réduire le pouvoir discriminant du filtre [OPM02], en particulier lorsque les propriétés supplémentaires qu'ils permettent d'obtenir ne sont pas particulièrement nécessaires aux données traitées.

Extensions. Les LBP proposés par Ojala *et al.* [OPM01] ont fait émerger de nombreuses méthodes. On remarquera tout d'abord que, par définition, le filtre LBP [OPM02] se cantonne à un voisinage circulaire, ne permet pas de décrire les relations entre les pixels du voisinage eux-même, et ne permet pas de décrire les relations qui existent entre deux voisinages de rayons différents. De nombreux travaux se sont donc intéressés à l'extraction de codes binaires issus de voisinages et de topologies de voisinages différents. En 2017, Liu *et al.* [LFG+17] remarquaient ainsi pas moins de 6 classes de méthodes réparties en 23 grandes catégories de topologies (voir Figure 5 de l'article

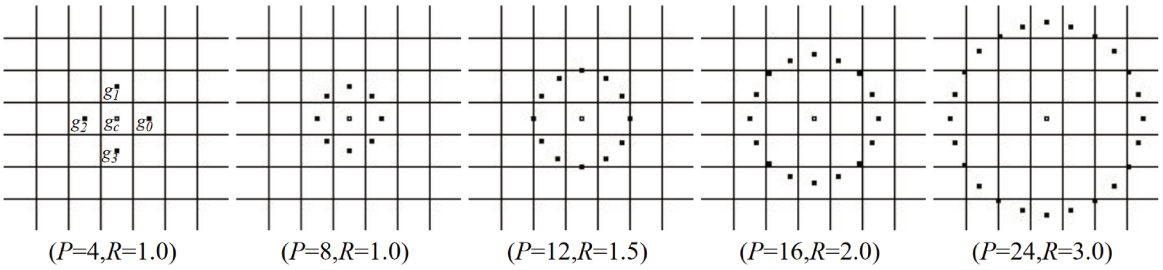


FIGURE 2.5 – Schéma représentant différents voisinages $(\mathrm{P,R})$ pour les motifs binaires de type LBP. Image extraite de [OPM02].

de Liu *et al.* [LFG+17]), que les auteurs regroupent en trois grandes catégories : information anisotropique, différences locales ou magnitudes, et micro-structures et macro-stuctures. La première catégorie correspond à l'analyse de voisinages non circulaires, tels que des lignes, des croix ou des selles. La deuxième catégorie correspond aux approches reprenant la topologie du LBP en cherchant à augmenter l'information extraite sur le voisinage en incluant la magnitude en plus du signe, en analysant les différences entre les pixels inter-voisinages de rayons différents, ou encore en étudiant les différences intra-voisinage de façons circulaires ou symétriques vis-à-vis du pixel central. La troisième catégorie de méthodes repose sur l'utilisation d'un voisinage constitué de patchs, permettant de lisser spatialement l'information représentée par chacun des voisins. Ces approches sont particulièrement intéressantes pour obtenir des représentations plus robustes au bruit (perturbations locales et aléatoire de l'intensité), auquel les filtres de type LBP tendent à être sensibles. Enfin, on remarquera qu'il est courant de générer des représentations étendues de textures en combinant, par concaténation, les histogrammes issues de plusieurs voisinages différents. Cela permet d'augmenter le pouvoir discriminant des représentations obtenues, d'une façon similaire aux représentations multi-échelles. De ce fait, chaque filtre de type LBP permettra d'obtenir des vecteurs de caractéristiques de tailles différentes. Basé sur ce principe, des extensions de ces filtres aux images en couleurs et aux vidéos ont été proposées. Pour cela, il est ou bien possible de considérer chaque canal comme une image complémentaire aux autres, ou bien de définir des voisinages inter-canaux. Enfin, le principe des LBP a inspiré des méthodes telles que SIFT [Low04] pour la description robuste à l'orientation et à l'échelle de points d'intérêts (*e.g.*, coins des objets), principalement utilisées pour des applications de mise en correspondance.

Résumé des propriétés. Nous résumons ici les propriétés principales des filtres de type LBP (forces (+), faiblesses (-)) :

- (+) Robustesse aux changement globaux d'illuminations (utilisation du signe du gradient)
- (+) Représentation locale vers globale intuitive (histogramme)

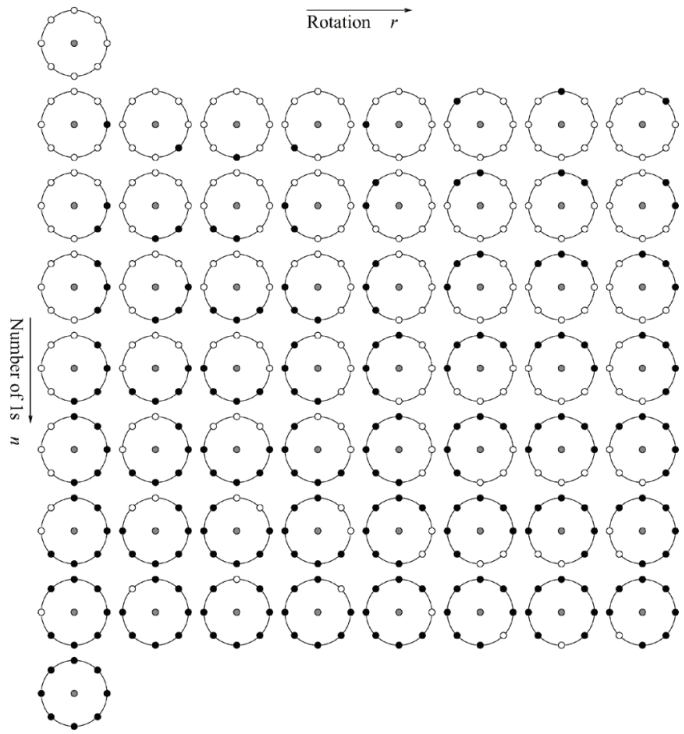


FIGURE 2.6 – Schéma représentant les 58 motifs binaires uniformes pour un degré d'uniformité égal à 2. Image extraite de [ZAMP11].

- (+) Robustesse aux rotations dans le plan (avec *mapping* de type ri ou riu^2)
- (+) Possibilité de modifier le voisinage et d'étendre la quantité de motifs détectés par concaténation d'histogrammes
- (+) Algorithmes généralement rapides, possibilité d'optimisations parallèles
- (-) Perte d'information avec le *mapping* pouvant parfois réduire l'efficacité
- (-) Histogrammes potentiellement de grandes tailles sans *mapping*
- (-) Les motifs binaires intéressants pour une tâche donnée ne sont pas *a priori* connus, et l'exhaustivité de la représentation n'est pas désirée (trop coûteuse, gain difficile à estimer)
- (-) Sensibilité au bruit (gradient local), mais possibilité d'appliquer un filtre passe bas au préalable

2.1.3 Application de la texture en télédétection

Les acquisitions réalisées en télédétection permettent d'observer la terre vue du ciel, orthogonalement à la surface (nous excluons les acquisitions en vue oblique dans le cadre de cette thèse). Les territoires apparaissent alors comme étant constitués de grandes zones texturées, dont la forme n'est pas *a priori* connue (*e.g.,* deux forêts de feuillus peuvent avoir des formes différentes). Face à ce constat, les méthodes d'analyse de la texture ont été largement utilisées en télédétection, et ce depuis de nombreuses années. A titre d'exemple, en 1974, Mauer [Mau74] étudiait déjà l'intérêt de la texture et des paramètres associés pour permettre la classification des champs de cultures à partir d'images aériennes en couleurs scannées.

Quelques cas d'utilisation dans le temps. En 1981, Irons *et al.* [IP81], proposaient d'étudier l'intérêt de statistiques locales similaires à celles extraites à l'aide des GLCM (moyenne, variance, skewness, kurtosis) pour l'analyse des images multi-spectrales de Landsat-2. Les auteurs indiquaient alors que ces représentations semblaient utiles pour la détection des hautes fréquences présentes dans l'image, mais que leur intérêt semblait limité pour la séparation des classes d'occupation du sol. En 1990, He *et al.* [HW90] proposaient l'utilisation d'unités de textures (Texture Units) pour l'analyse d'image de télédétection. Les unités de textures avaient ici une formulation très proche des filtres de motifs ternaires, basés sur les LBP, qui ont gagné en popularité 20 ans plus tard. Les résultats préliminaires obtenus par les auteurs montraient l'intérêt d'étudier ce type d'approche pour la classification d'images de télédétection de résolutions moyennes (10 m x 10m, 20 m x 20 m) en 4 classes d'occupation du sol. En 1998, Zhu *et al.* [ZY98] s'intéressaient à l'utilisation d'une banque de filtres de Gabor dans un contexte de télédétection afin de classifier l'occupation du sol en 25 catégories. En 2005, Warner *et al.* [CGDC$^+$14] comparaient l'utilisation de l'auto-corrélation avec les GLCM pour segmenter les zones cultivées correspondant à des vignes et des vergers. En 2008, Rabatel *et al.* [RDD08] proposaient une approche itérative pour détecter les vignes à partir d'une analyse des zones correspondants aux pics de hautes fréquences dans l'espace de Fourrier avec des filtres de Gabor. Encore en 2008, Caridade *et al.* [CMM08] montraient l'intérêt des statistiques GLCM pour générer automatiquement des cartes d'occupation du sol (quatre classes : eau, sol nue, arbres, prairies) à partir de 4 images en niveaux de gris du parc Peneda-Gerês au Portugal, acquises en 1958 (résolution 3m×3m, avec environ 6000×6000 pixels par image).

Plus récemment, en 2014, Champion *et al.* [CGDC$^+$14] proposaient d'exploiter les GLCM pour estimer l'age des forêts à partir d'acquisitions réalisées par un radar à synthèse d'ouverture. En 2015, Feng *et al.* [FLG15] proposaient de combiner l'information portée par les canaux RVB (Rouge-Vert-Bleu) d'une image en couleurs avec les statistiques extraites d'une GLCM afin d'améliorer l'identification de la végétation en environnement urbain à partir d'images acquises par drone. En 2016, Regniers *et al.* [RBLG16] exploraient l'utilisation d'ondelettes (*i.e.*, banque de filtres) basées sur des modèles multivariés afin de segmenter des images optiques panchromatiques de très

haute résolution en trois classes d'occupation du sol. Ils ont pu montrer que des résultats prometteurs pouvaient être obtenus sur ce type d'images, en comparaison avec l'utilisation de méthodes plus classiques telles que les GLCM. Encore en 2016, Aguilar *et al.* [AFA+16] comparaient 26 extracteurs de caractéristiques incluant plusieurs approches de type LBP et des GLCM pour l'analyse automatique d'images satellites. Les auteurs ont ainsi pu montrer que les approches de type LBP permettaient d'obtenir des taux de bonne classification plus élevés que les autres méthodes comparées, et ce pour des temps d'exécution plus faibles. Toujours en 2016, Hunag *et al.* [HCLD16] s'intéressaient à l'utilisation d'une représentation complétée des LBP combinée à un encodage à l'aide des vecteurs de Fisher [SPMV13] afin de classifier des images de télédétection. En 2018, Wang *et al.* [WFZ+18] proposaient eux aussi d'utiliser une représentation complétée des LBP, cette fois-ci pour classifier la végétation côtière à partir d'images de très hautes résolutions. En 2019, Kwak *et al.* [KP19] prenaient en compte des statistiques issues de GLCM combinées aux informations spectrales d'une série d'images (Rouge, Vert, Proche Infrarouge) acquises par drone à plusieurs dates afin d'estimer différents types de champs de cultures.

Observations. Nous remarquons ici une forte prédominance des approches de type GLCM dans les applications de la texture en télédétection, et ce malgré le fait que plusieurs études aient pu montrer l'avantage des approches de type LBP pour la classification des images texturées. Nous pouvons ici seulement supposer que cela est dû à la disponibilité de ces approches au sein des logiciels de type SIG, permettant à la communauté pluridisciplinaire de la télédétection d'utiliser ces méthodes sans avoir à les ré-implémenter. Un autre aspect important qui pourrait expliquer la popularité des GLCM est l'interprétabilité des vecteurs de caractéristiques générés (*i.e.*, exprimer empiriquement les valeurs générées). La difficulté d'interprétation des histogrammes générés par les filtres de type LBP peut en effet être un frein à leur utilisation pour certains praticiens. Par ailleurs, on constate que peu d'études se sont intéressées aux images aériennes historiques, et ce en particulier à l'aide de méthodes d'extraction de caractéristiques récentes.

2.2 Sur-segmentation

La sur-segmentation, aussi appelée segmentation non supervisée (*clustering*), consiste à partitionner une image en groupe de pixels aux propriétés homogènes afin de proposer une représentation spatiale simplifiée de la donnée. L'idée est ici de considérer qu'un pixel seul ne contient pas beaucoup d'information, et que de nombreux pixels proches les uns des autres vont posséder des informations similaires qu'il peut être intéressant de regrouper. En particulier, le fait de passer d'une représentation pixels à une représentation basée sur une sur-segmentation permet de générer ce que l'on nomme des *superpixels*, qui sont tout simplement des groupes de pixels connexes. En pratique, les superpixels sont définis comme des groupes de pixels de tailles similaires - nous utiliserons ici ce terme pour caractériser le résultat obtenu par toutes les méthodes de sur-segmentation. En télédétection, il n'est par ailleurs pas rare d'utiliser les termes segments et objets pour caractériser les superpixels [Bla10]. Ces derniers ont pu trouver des applications pour la segmentation sémantique [KHH17], le transfert de couleur [GTP17], ou encore l'analyse d'images 3D [CRN+19].

2.2.1 Méthodes courantes

Une évaluation de 28 méthodes de la littérature sur 5 jeux de données a été proposée par Stutz *et al.* [SHL18] en 2018, mettant en avant la diversité des approches qui ont été mis au point pour générer des superpixels. Les auteurs ont ainsi identifié 8 groupes de méthodes, dont celles basées sur le partage des eaux (*Watershed*), la recherche de modes, les graphs, le regroupement (*clustering*) ou encore la minimisation d'énergie. En pratique, cette évaluation vise à déterminer la qualité des sur-segmentations générées à l'aide de critères particuliers tels que :

- L'erreur de sur-segmentation (*Oversegmentation Error*, OE) : mesure le fait que plusieurs

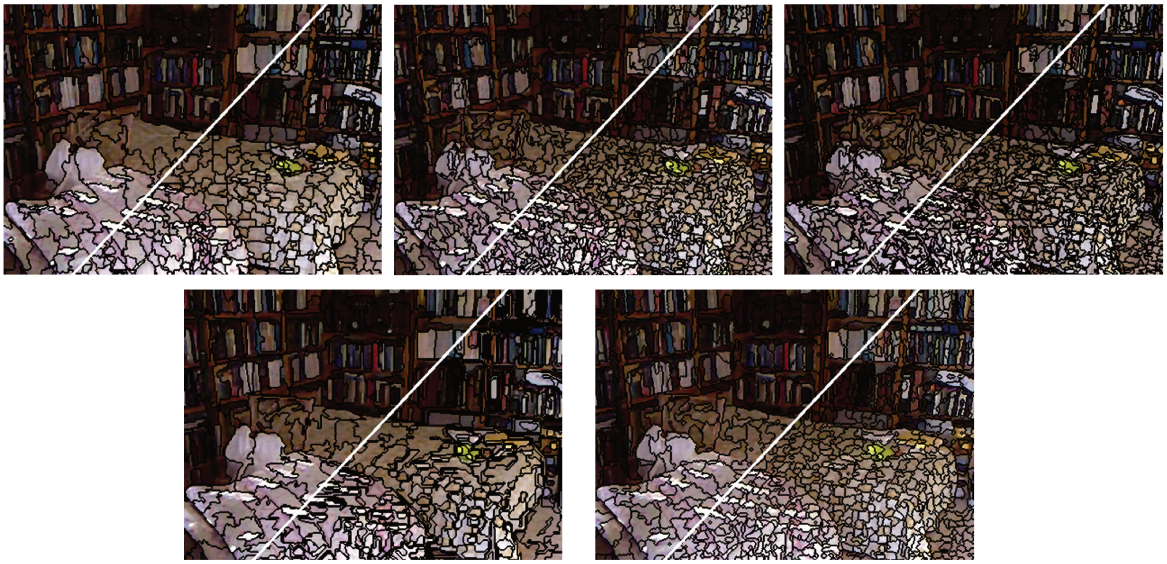


FIGURE 2.7 – Exemples de superpixels. De gauche à droite : Quick Shift [VS08], SLIC [ASS+12], ETPS [YBFU15], FH [FH04] et Watershed [BM93]. Images extraites du site web [2] lié aux travaux de Stutz *et al.* [SHL18].

groupes de pixels ont été générés au sein d'un objet d'intérêt (i.e., on aurait aimé avoir un unique groupe).

- L'erreur de sous-segmentation (*Undersegmentation Error*, UE) : mesure le fait que des groupes de pixels se superposent aux bordures des objets d'intérêts (i.e., on aurait aimé ne pas déborder).
- Le taux de bonne classification atteignable (*Achievable Segmentation Accuracy*, ASA) : mesure le taux de classification que l'on pourrait obtenir si tous les superpixels étaient classifiés correctement (nécessite une vérité terrain).

Dans la suite, nous allons décrire succinctement certaines des méthodes représentatives de la littérature. Celles-ci sont illustrées sur la Figure 2.7. Ces méthodes ont pu prouver leurs performances [SHL18] et ont déjà été utilisées sur des images aériennes ou satellites. Nous avons exploité certaines d'entre elles dans nos travaux présentés dans le chapitre 5 de ce manuscrit à des fins de post-traitement. Nous reprenons ici une partie du formalisme décrit par Mathieu *et al.* [Mat17].

Quick Shift (QS)

Quick Shift (QS) a été proposée en 2008 par Vedaldi *et al.* [VS08]. Il s'agit d'une amélioration de l'algorithme *Mean Shift* [CM02]. Cette méthode appartient à la catégorie des algorithmes par recherche de modes. Dans cet article, les auteurs proposent de représenter les N pixels d'une image à l'aide de leur couleur dans l'espace RGB et de leur position dans l'image 2D. Chaque pixel est ainsi représenté à l'aide d'un vecteur de 5 caractéristiques. Il est possible de représenter la distribution de ces vecteurs à l'aide d'une densité de probabilités. Les modes de l'image sont alors définis comme étant les maxima locaux de la densité de probabilité. Le but de QS est de réussir à trouver ces modes de façon efficace afin d'associer chaque pixel au mode dont il est le plus proche, générant ainsi des groupes de pixels ayant des positions et des couleurs proches les uns des autres. En pratique, Quick Shift va estimer la densité de probabilité F_p des pixels dans l'image à l'aide de la densité de Parzen, qui repose sur un filtrage gaussien ϕ de la différence $d(.)$ entre les vecteurs de caractéristiques dans $\mathrm{X} = \{x_1, ..., x_\mathrm{N}\}$ (voir équation (2.7)).

$$\mathrm{F}_p(x) = \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} \phi(d(x, x_i)), x_i \in \mathrm{X} \tag{2.7}$$

***Simple Linear Iterative Clustering* (SLIC)**

Simple Linear Iterative Clustering (SLIC) a été proposé par Achanta *et al.* en 2010 avant d'être revisité dans une étude comparative en 2012 [ASS+12]. Cette méthode itérative correspond à une adaptation locale de l'algorithme des k-moyennes. Le but de SLIC va être de moduler l'emprise spatiale des cellules d'une grille régulière afin qu'elles respectent un critère d'homogénéité basé sur la couleur dans l'espace LAB (3 valeurs : l,a,b) et la position dans l'image (2 valeurs : x,y), de façon similaire à l'algorithme QS. L'algorithme de SLIC proposé par [ASS+12] est décrit ci-après.

Pour une image I, on définit une grille régulière de K cellules, chaque cellule étant de taille S × S. On rappelle que chaque pixel est ici représenté à l'aide d'un vecteur de 5 valeurs (l,a,b,x,y). On initialise les K centres de masses des cellules correspondant à la valeur moyenne des pixels qui la composent : $C_k = [l_k, a_k, b_k, x_k, y_k]$. Afin de pouvoir rattacher chaque pixel à un des K centres de masses dans l'espace 5D, il est nécessaire de définir une mesure de distance. Pour cela, Achanta *et al.* [ASS+12] proposent de calculer la distance D_s d'un point à un autre à l'aide d'une combinaison linéaire de la distance euclidienne des couleurs d_{lab} et de la distance euclidienne des positions d_{xy} (voir équations (2.8), (2.9) et (2.10), où l'indice k représente un centre de masse et l'indice i un pixel). Ce choix est fait afin de pouvoir pondérer l'importance de la couleur par rapport à la position à l'aide d'un paramètre m, dit de compacité, qui se comprend intuitivement comme étant le poids relatif donné à la position des pixels.

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \tag{2.8}$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \tag{2.9}$$

$$D_s = d_{lab} + \frac{m}{S} d_{xy} \tag{2.10}$$

A l'aide de cette distance D_s, SLIC réalise à chaque itération une assignation de chaque pixel à un des K centres dans un voisinage de 2S × 2S pixels, avant de mettre à jour la position des centres. A noter qu'il est nécessaire de fixer S pour l'utilisateur. Ce paramètre permet d'obtenir des superpixels à des échelles différentes (plus S est grand, plus les superpixels seront grands).

On remarquera que de nombreuses variantes de cet algorithme ont été proposées dans la littérature, afin notamment d'améliorer la prise en compte de la texture et des contours à l'aide, par exemple, de filtres basés sur les LBP (présentés dans la section précédente) ou via l'intégration des gradients de l'image dans le calcul dans la mise à jour des groupes de pixels. Des extensions étendant la distance d_{lab} aux caractéristiques issues des couches cachées d'un réseau de neurones profond à convolutions ont également été étudiées afin d'améliorer la qualité des sursegmentation générées [JSL+18; VBt18].

***Efficient Topology Preserving Segmentation* (ETPS)**

Efficient Topology Preserving Segmentation (ETPS) [YBFU15] est un algorithme qui étend le formalisme introduit par SLIC à plusieurs échelles. Il introduit également des termes de régularisation supplémentaires pour améliorer la qualité des superpixels générés. En particulier, les termes introduits par ETPS vont pénaliser les superpixels non connectés tout en les forçant à avoir une taille finale au moins égale à un quart de leur taille initiale, et ce dans le but d'éviter que des pixels isolés forment des superpixels. Cet algorithme se plaçait en première position de l'évaluation réalisée par [SHL18] en 2018.

Tout comme SLIC, ETPS s'initialise sur une grille régulière et va chercher à assigner chaque pixel à un groupe de pixels. Cependant, tandis que SLIC se base sur une grille régulière définie uniquement lors de l'initialisation, ETPS va exploiter une grille régulière avec une échelle différente à chaque itération. Pour cela, ETPS considère initialement une grille régulière relativement

grossière, avec de grandes cellules de taille $S_1 \times S_1$. Pour chaque cellule de la grille, l'algorithme va calculer le centre de masse correspondant dans un espace 5D (tout comme SLIC). Il va alors chercher quelles sont les cellules de chaque superpixel qui sont adjacentes à une cellule d'un autre superpixel. Les auteurs nomment ces cellules les cellules frontières (*boundary blocks*). Une fois cette liste de cellules frontières obtenues, l'algorithme va alors tester la mise en commun de chaque cellule frontière avec les superpixels de ses cellules voisines sur un voisinage 4-connexe, et affecter la cellule frontière au superpixel le plus proche. A noter qu'à l'initialisation, toutes les cellules sont considérées à la fois comme des cellules frontières et comme des superpixels. Cela permet de regrouper des cellules (*i.e.*, de générer des superpixels) à l'échelle la plus grossière. Afin de raffiner la sur-segmentation obtenue à chaque itération $i > 1$, une nouvelle grille régulière de cellules de taille $S_i \times S_i$ est créée, tel que $S_i < S_{i-1}$. Cette grille régulière permet de décomposer les superpixels obtenus lors de l'itération $i-1$ en appliquant le processus basé sur les cellules frontières qui est décrit ci-dessus. On remarquera que les auteurs précisent qu'utiliser des cellules de grande taille permet à leur algorithme d'atteindre des minima locaux de meilleur qualité par rapport à leur fonction objectif (*i.e.*, distance calculée avec termes de régularisation).

Méthode de Felzenswalb et Huttenhoch (FH)

L'algorithme de segmentation proposé par Felzenszwalb et Huttenhoch [FH04] (FH) génère des régions de pixels en modélisant l'image I à l'aide d'un graphe $G =< V, E >$, avec $V = \{v_1, ..., v_N\}$ un ensemble de N sommets correspondant aux pixels de I, et E un ensemble d'arêtes reliant les sommets (*i.e.*, E définit le voisinage). La pondération des arêtes E correspond à la distance séparant deux sommets. Elle est définie comme la différence absolue entre les niveaux de gris des pixels dans l'algorithme de FH.

Les superpixels s sont ici formés en regroupant les pixels (les sommets) v en fonction d'un prédicat d'homogénéité sur les arêtes. Soit s_i un superpixel. On note alors $E_i \subset E$ l'ensemble des arêtes entre les pixels de s_i, et $E_{i,j}$ l'ensemble des arêtes reliant s_i à un autre superpixel s_j. Le prédicat défini par FH consiste alors à comparer la différence maximale entre les pixels d'un même superpixels s_i, que l'on nommera différence interne (D_{int}), avec la différence minimale entre s_i et s_j, que l'on nommera différence externe (D_{ext}). Les différences sont ici défini à l'aide des pondération des arêtes. Cette pondération correspondant à une différence absolue, les valeurs maximale et minimale peuvent être définie par les équations (2.11) pour D_{int} et (2.12) pour D_{ext}. Si $E_{i,j}$ est l'ensemble vide, alors D_{ext} prend une valeur infinie.

$$D_{int}(s_i) = max_{e_{k,l} \in E_i}(e_{k,l}) \tag{2.11}$$

$$D_{ext}(s_i, s_j) = min_{e_{k,l} \in E_{i,j}}(e_{k,l}) \tag{2.12}$$

Le prédicat P est alors donné par une fonction binaire qui indique si les superpixels s_i et s_j doivent restés distincts ou être fusionnés (voir équation (2.13)).

$$P(s_i, s_j) = \begin{cases} \text{vrai} & \text{si } D_{ext}(s_i, s_j) > min(D_{int}(s_i) + \tau_{s_i}, D_{int}(s_j) + \tau_{s_j}) \\ \text{faux} & \text{sinon} \end{cases} \tag{2.13}$$

où τ_{s_i} et τ_{s_j} sont des paramètres de la méthode.

Cette méthode est généralement appliquée sur une image ayant été filtrée à l'aide d'un filtre passe bas afin de lisser les hautes fréquences non désirées. Il est par ailleurs difficile de contrôler le nombre de superpixels qu'elle va permettre de générer (paramètre non explicite, contrairement à SLIC et ETPS).

***Watershed* (W)**

L'algorithme de *Watershed* (W) [BM93], ou de partage des eaux, consiste à considérer que les groupes de pixels dans l'image seront séparés par des gradients d'intensités. Par analogie, chaque

intensité de gradient correspondrait ici à une élévation séparant des bassins versants (zones de faibles gradients). Si l'on souhaitait représenter ces éléments en 3D, le gradient représenterait donc des éléments de relief, tandis que les bassins versants correspondraient à des crevasses. L'algorithme de partage des eaux va chercher à inonder les bassins versants en augmentant l'intensité des pixels correspondants, simulant une montée du niveau d'eau. La séparation entre deux objets correspondra alors à la position où deux bassins versants inondés se rejoignent. L'avantage principal de cette approche est de pouvoir détecter des superpixels de tailles variables en se basant sur une représentation intermédiaire, à savoir les hautes fréquences détectées dans l'image. Comme pour la méthode de FH, le nombre de superpixels généré par la méthode de partage des eaux n'est ici pas contrôlé *a priori*.

2.2.2 Application de la sur-segmentation en télédétection

Les superpixels en télédétection ont connu un fort engouement de par leur capacité à regrouper des ensembles homogènes de pixels et ainsi diminuer la quantité d'information à traiter. Une édition spéciale du journal scientifique *Remote Sensing* (Télédétection), était d'ailleurs consacrée à cette thématique en 2019 [3]. Cet intérêt s'explique de par le gain de temps que les superpixels permettent d'obtenir lors des traitements d'image aériennes et satellites qui sont généralement de très grandes tailles (*e.g.*, 12000 × 12000 pixels), mais aussi de par la possibilité qu'ils offrent d'agréger les statistiques spectrales au sein de chaque superpixel afin de les reconnaitre. En pratique, on parle beaucoup d'identification basée objets (*Object Based Image Analysis*, OBIA), où un objet est défini par un superpixel. L'avantage des approches de type OBIA est qu'elles permettent d'obtenir des résultats sémantiques qui seront spatialement lisses par rapport aux résultats obtenus au pixel-près (*i.e.*, on attribue un label à tout un superpixel d'un seul coup, plutôt qu'à un pixel à la fois). Une revue de ces méthodes était proposée par Blaschke *et al.* en 2010 [Bla10]. Les auteurs indiquaient alors que les méthodes de type OBIA devenaient de plus en plus populaires en comparaison des méthodes basées pixels. A titre d'exemple, Zhang *et al.* [ZZS+19] utilisent des superpixels à plusieurs échelles générés par un logiciel commercial [4] afin d'extraire des statistiques multispectrales pour chaque superpixel et ainsi classifier les superpixels. Un vote majoritaire entre les résultats obtenus avec les superpixels d'échelles différentes est ensuite réalisé.

De nombreuses approches combinant réseaux de neurones à convolutions (voir section 2.4) et superpixels ont également vu le jour. Postdajian *et al.* [PBMS18] proposent de classifier des imagettes centrées sur les superpixels issues de la méthode FH afin de générer des résultats spatialement cohérents tout en réalisant une classification dense de l'occupation du sol en un temps raisonnable (par superpixel plutôt que par pixel). Les auteurs indiquent que les paramètres de la méthode FH ont été selectionnés manuellement dans le cadre de leurs travaux. Ma *et al.* [MGS+19] proposent d'extraire des superpixels à l'aide de SLIC, qu'ils combinent avec un réseau de neurones à convolutions leur permettant d'extraire automatiquement des caractéristiques représentatives d'images radars et ainsi attribuer une classe d'occupation du sol à chaque superpixel. Pour cela, ils proposent de s'intéresser non seulement à chaque superpixel de l'image, mais également à une imagette englobant le superpixel et centrée sur celui-ci. Ce choix a été fait afin d'intégrer à la fois l'information spécifiquement liée au superpixel et l'information liée à son contexte (imagette). Le fait de combiner l'information portée par les superpixels avec leur environnement au travers d'imagettes et de réseaux de neurones profonds a également été étudié par Chen *et al.* [CM19]. Gharibbafghi *et al.* [GTR18] utilisent l'algorithme SLIC à plusieurs échelles afin d'extraire des bâtiments à partir de modèles numériques de terrain (représentation 3D des éléments observés au sol) générés par imagerie satellite stéréoscopique. Sherpherd *et al.* [SBD19] proposent quant à eux de comparer plusieurs algorithmes de sur-segmentation tels que FH et Quick Shift pour la génération de superpixels adaptés à l'analyse d'images satellites.

3. https://www.mdpi.com/journal/remotesensing/special_issues/Superpixel_based_Analysis_and_Classification
4. https://geospatial.trimble.com/products-and-solutions/ecognition

Observations. De nombreux travaux ont mis en avant l'avantage des approches basées superpixels afin de générer des représentations spatiales plus cohérentes qu'avec les approches basées pixels. On remarque aussi ici la volonté d'avoir accès à un contexte spatial plus étendu que les superpixels générés par les méthodes classiques afin d'améliorer les résultats obtenus. Pour chaque application (différentes images, différentes résolutions, différents objets à reconnaitre), il semble cependant y avoir une incertitude qui se dégage quant à la taille optimale des superpixels à utiliser (utilisation de plusieurs échelles, modification des paramètres à la main).

2.3 Algorithmes de classification

Dans le cadre de nos travaux, nous nous sommes particulièrement intéressés à la classification des images aériennes historiques. Cette section a pour but d'introduire la notion de classifieur (algorithme de classification) et présente les méthodes de classification supervisée les plus couramment utilisées dans la littérature.

2.3.1 Définitions

Définition intuitive

Un algorithme de classification, ou classifieur, permet d'attribuer automatiquement une classe à un objet représenté par un vecteur de caractéristiques. Les différentes classes possibles en sortie d'un classifieur dépendent de l'application visée et doivent être fixées par l'utilisateur. Afin de les représenter, il est d'usage d'avoir recours à des étiquettes correspondant à des valeurs numériques distinctes et identifiables (*e.g.*, $\{etiquette\} : classe$; $\{0\} : orange, \{1\} : pomme$). Un classifieur est donc un algorithme qui va associer des vecteurs de caractéristiques à des étiquettes.

Définition formelle

Considérons un vecteur de caractéristiques $x \in \mathbb{C}^N$ avec $N \in \mathbb{N}^*$, et Y l'ensemble des étiquettes formé par un sous-ensemble des entiers relatifs $\mathbb{Z}$ dont les éléments sont tous distincts deux à deux. Un classifieur h est alors défini comme une application injective de $\mathbb{C}^N$ dans Y qui, à tout x de dimension n à valeur dans $\mathbb{C}^N$, associe une étiquette y de Y. Cette définition est succinctement représentée par l'équation (2.14) pour un nombre d'étiquettes k arbitraire.

$$y = h(x), x \in \mathbb{C}^N, y \in Y = \{y_1, ..., y_k\} \tag{2.14}$$

Il est important de remarquer que les vecteurs x sont pris dans $\mathbb{C}^N$ afin de nous assurer de l'existence d'un produit scalaire entre les vecteurs de notre ensemble de départ, condition nécessaire à l'apprentissage de certains classifieurs tels que les machines à vecteurs de support.

Ensembles d'entraînement, de validation et de test

Afin de pouvoir classifier un vecteur de caractéristiques, les classifieurs ont besoin d'être exposés à un ensemble connu de paires caractéristiques-étiquettes (cas surpervisé). Cet ensemble est généralement nommé ensemble d'entraînement, que l'on notera X_{train}. Le terme entraînement est ici lié au fait que certains algorithmes vont optimiser leurs paramètres, au sens mathématique, par rapport à X_{train} afin de réaliser la tâche de classification. Cet ensemble est généralement couplé à un ensemble de validation, X_{val}, qui permet de sélectionner les hyperparamètres (*i.e.*, paramètres difficilement optimisables mathématiquement) de l'algorithme par recherche exhaustive sur grille de paramètres. L'ensemble de validation est constitué de données qui ne sont pas présentes dans X_{train}. En pratique, il est possible de se passer de X_{val} en ayant ou bien recours à des approches par validation croisée, ou bien en ne cherchant pas à optimiser les hyperparamètres, ou encore en cherchant à évaluer les algorithmes dans un scénario optimiste (*i.e.*, algorithmes "au meilleur de leur forme") lorsque que peu de données existent. Enfin, l'ensemble des données de

test X_{test} va servir à évaluer l'algorithme. On va ici fournir uniquement le vecteur de caractéristiques au classifieur, et comparer le résultat généré par rapport à l'étiquette réelle. Les données de X_{test} ne sont pas présentes ni dans X_{train} ni dans X_{val}. On distingue ici X_{val} et X_{test} afin de pouvoir évaluer plusieurs algorithmes sur une base commune (X_{test}) qui n'a jamais été utilisée ni pour optimiser le modèle, ni pour en sélectionner les hyperparamètres.

2.3.2 Algorithmes communs

Dans cette section, nous décrivons les algorithmes de la littérature les plus utilisés.

K plus proches voisins

La méthode des K plus proches voisins (*k-nearest* neighbors, KNN) est une des méthodes les plus classiques utilisée en apprentissage automatique pour la classification. Cet algorithme se décompose comme suit :

- Soit $x_{test} \in X_{test}$. Calculer la distance de x_{test} à tous les éléments de X_{train}. La distance utilisée peut varier en fonction des vecteurs de caractéristiques utilisés (*e.g.*, distance euclidienne pour valeurs continues, distance de Hamming pour valeurs binaires).
- Ordonner les éléments de X_{train} par ordre décroissant de distance avec x_{test}.
- Ne garder que les K éléments de X_{train} ayant la plus petite distance avec x_{test}. Le K est ici défini par l'utilisateur.
- Faire un vote majoritaire entre les étiquettes des K éléments retenus. L'étiquette majoritaire y_m est ici retenue comme étant l'étiquette prédite par le classifieur. Dit autrement, l'algorithme associe l'étiquette y_m à x_{test}.

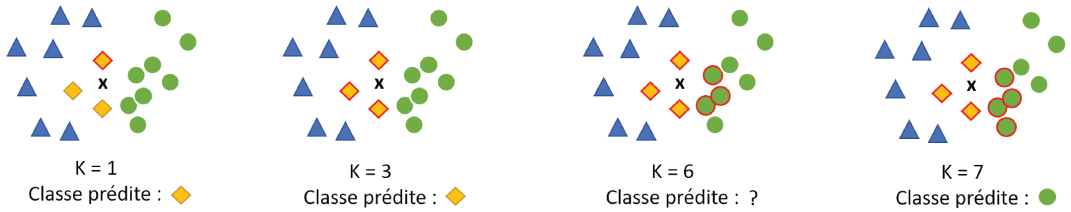


FIGURE 2.8 – Schéma illustrant le principe des K plus proches voisins avec 3 classes représentées par des formes géométriques colorées dans un espace 2D. Les formes aux contours rouges sont les K plus proches voisins de la croix selon la distance euclidienne dans le plan.

Les avantages de cet algorithme sont qu'il est relativement facile à implémenter et qu'il permet d'obtenir des résultats de classification automatique sans nécessiter d'entraînement. En revanche, il est nécessaire de fixer l'hyperparamètre K. Celui-ci est souvent fixé à l'aide d'un nombre impair à cause de l'étape de vote majoritaire (voir Figure 2.8). Il peut également être obtenu par validation croisée. Par ailleurs, plus l'ensemble d'entraînement X_{train} contient d'éléments, plus l'algorithme s'exécutera lentement (recherche exhaustive). Afin de résoudre ce problème, des approches optimisées à l'aide d'arbres de recherche (*k-d trees*) ont été proposées [O+13]. Enfin, le KNN tend à être sensible au problème de déséquilibre de classes (*i.e.*, lorsqu'une classe est très majoritairement représentée par rapport à une autre). Des approches pondérant la distance calculée en fonction du nombre d'éléments par classe ont été proposées pour lutter contre ce problème [DP13].

Support Vector Machine

Une machine à vecteurs de support (*Support Vector Machine*, SVM) [CV95] est un classifieur initialement introduit pour la classification binaire, avant d'être étendu au cas de la classification multi-classes ($k > 2$) en se ramenant à un cas binaire [DK05; CV95]. Le but d'une SVM est de déterminer un hyperplan séparateur optimal, dit de marge maximale, entre les points que l'on

souhaite classifier. La marge est ici définie comme étant la distance entre l'hyperplan et les points qui en sont les plus proches. Les points les plus proches de l'hyperplan sont nommés les vecteurs de supports (chaque point correspondant à un vecteur de caractéristiques). Dans le cas linéaire, nous pouvons considérer un classifieur h (un hyperplan) comme une fonction qui va pondérer un vecteur de N caractéristiques x à l'aide d'un vecteur de poids $w = \{w_1, ..., w_N\}$ afin d'en estimer l'étiquette y (voir équation (2.15)).

$$h(x) = w^T x + w_0 \tag{2.15}$$

Il peut être montré que la fonction h optimale s'obtient en minimisant $\frac{1}{2}||w||^2$ sous contraintes $y_k(w^T x_k + w_0) \geq 1$, avec (x_k, y_k) l'ensemble des paires caractéristiques-étiquettes de $X_{train} = \{(x_1, y_1), ..., (x_p, y_p)\}$, telles que $y_k \in \{-1, 1\}$. Ce problème peut être résolu à l'aide des multiplicateurs de Lagrange (voir [CV95]), dont la solution duale met en avant que seul un sous-ensemble de points est nécessaire pour obtenir une solution (les vecteurs de support), et que l'hyperplan solution dépend uniquement du produit scalaire entre le vecteur d'entrée x et les vecteurs de support x_k (voir équation (2.16), où α_k^* est un multiplicateur de Lagrange optimal).

$$h(x) = \sum_{k=1}^{p} \alpha_k^* y_k (x \cdot x_k) + w_0 \tag{2.16}$$

Ce dernier point permet d'utiliser l'astuce du noyau (*kernel-trick*), qui consiste à projeter les vecteurs de caractéristiques non linéairement séparables dans un espace de redescription où ils sont linéairement séparables. Pour cela, on utilise un noyau $K(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j)$, ce qui donne la solution décrite pas l'équation (2.17).

$$h(x) = \sum_{k=1}^{p} \alpha_k^* y_k K(x, x_k) + w_0 \tag{2.17}$$

En pratique, les noyaux les plus régulièrement utilisés sont le noyau polynomial (2.18) et le noyau gaussien (aussi appelé fonction de base radiale, RBF) (2.19). A noter que le choix du noyau RBF tend à fonctionner correctement dans la majorité des cas (*i.e.*, il est à privilégier quand aucun *a priori* n'est connu sur la structure des données), mais qu'il est nécessaire de fixer le paramètre γ à l'aide d'un jeu de validation.

$$K(x_i, x_j) = (x_i^T \cdot x_j)^d, d \in \mathbb{N} \tag{2.18}$$

$$K(x_i, x_j) = e^{-\gamma||x_i - x_j||^2}, \quad avec \quad \gamma > 0 \tag{2.19}$$

Par ailleurs, il est courant de ne pas pouvoir trouver d'hyperplan séparant linéairement les points, et ce même dans l'espace de redescription. Les SVM sont donc généralement optimisés à l'aide d'une *marge souple* [CV95], prenant la forme d'un terme de régularisation permettant une tolérance à l'erreur. Ce terme de régularisation est pondéré par un paramètre $C > 0$, qui va permettre de réaliser un compromis entre les erreurs commises et la largeur de la marge. En pratique, ce terme de régularisation permet d'éviter le sur-apprentissage. Il est généralement déterminé à l'aide d'un ensemble de validation.

Forêts aléatoires d'arbres décisionnels

Les arbres décisionnels sont des classifieurs qui vont être optimisés afin de générer une décision basée sur des règles logiques successives. Un arbre est ici constitué d'un ensemble de nœuds, chaque nœud étant responsable de séparer l'ensemble des données qu'il prend en entrée en deux groupes à l'aide d'un seuil sur les caractéristiques des données. De nombreuses approches ont été proposées dans la littérature pour construire des arbres de décisions, telles que ID3 (*Iterative Dichotomiser* 3) [Qui86] ou encore CART (*Classification And Regression Trees*) [BFOS84]. Ici, nous allons nous intéresser tout particulièrement à l'algorithme CART, à la base des forêts aléatoires.

L'algorithme CART est basé sur des règles logiques binaires, il fonctionne avec des valeurs continues et va permettre d'optimiser le choix des caractéristiques et les seuils logiques à chaque

séparation selon un critère d'homogénéité / d'impureté. En pratique, lorsqu'à un nœud S donné, la caractéristique j est sélectionnée pour la séparation avec un seuil a_j, cette séparation génère deux sous-nœuds S_g (gauche) et S_d (droit). On défini alors l'homogénéité de la séparation par $\delta I(S) = I(S) - E[I(S_{gd})]$, qui est une mesure de la différence entre l'impureté du nœud I(S) et de l'espérance (moyenne statistique) des impuretés des sous-nœuds $E[I(S_{gd})] = p_g I(S_g) + p_d I(S_d)$, avec $p_{g/d} = \#S_{g/d}/\#S$. L'optimisation consiste ici à maximiser $\delta I(S)$ sur X_{train} pour chaque nœud afin de déterminer la caractéristique qui va être choisie pour la séparation, ainsi que le seuil qui va être utilisé. En classification, les mesures d'impureté régulièrement utilisées sont l'index de Gini $G_{gn}(S)$ (voir équation (2.20)) et l'entropie H(S) (voir équation (2.21)), toutes deux définies à partir de la probabilité p_i qu'un élément de S se retrouve dans une des k classes de $Y = \{y_1, ..., y_k\}$ après la séparation du nœud (*i.e.*, $p_i \approx \frac{|y_i|}{|S|}$). Dans un contexte de régression (*i.e.*, génération de données continues à la place d'étiquettes), l'erreur quadratique moyenne est généralement utilisée.

$$G_{gn}(S) = \sum_{i=1}^{k} p_i(1 - p_i) \tag{2.20}$$

$$H(S) = -\sum_{i=1}^{k} p_i log(p_i) \tag{2.21}$$

En pratique, les règles de décisions des arbres peuvent être biaisées vis à vis des données observées dans X_{train}, rendant l'algorithme instable. Face à ce problème, les forêts aléatoires d'arbres décisionnels (Random Forest, RF) ont été proposées par Léo Breiman en 2001 [Bre01]. L'idée est ici d'apprendre plusieurs arbres entraînés sur des sous-ensembles de données de X_{train} tirés aléatoirement, et d'effectuer un vote majoritaire des classes prédites par l'ensemble des arbres (*bagging*). Ce processus est schématiquement représenté sur la Figure 2.9. De plus, à chaque fois que l'on va chercher à séparer un nœud, RF va tirer au hasard une partie des caractéristiques afin de réaliser la séparation, permettant d'introduire une robustesse supplémentaire dans les décisions. Les paramètres principaux de cet algorithme sont ici le nombre d'arbres à considérer, le nombre de caractéristiques à considérer à chaque nœud (minimum, maximum) et le critère d'impureté. Ils sont généralement déterminés à l'aide d'un ensemble de validation, ou par validation croisée.

Perceptrons multicouche (MLP)

Un perceptron multicouche (MLP) est un réseau de neurones organisé sous forme de couches de neurones, où les données en sortie d'une couche sont les entrées de la suivante. Chaque couche contient un certain nombre de neurones non liés entre eux, et chaque neurone est modélisé par un perceptron [Ros58].

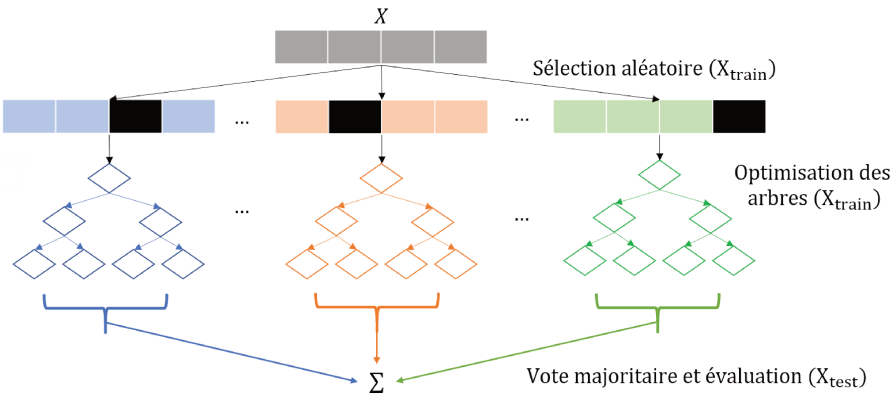


FIGURE 2.9 – Schéma illustrant le principe des forêts aléatoires d'arbres décisionnels. Chaque couleur représente une arbre décisionnel. Les cases noires représentent des sous-ensembles de données de X qui ne sont pas pris en compte pour entraîner l'arbre.

Le perceptron s'inspire des neurones biologiques, qui vont pondérer, biaiser et intégrer les signaux qu'ils reçoivent avant d'appliquer une fonction d'activation $g(.)$ sur un signal (*e.g.*, un vecteur de caractéristiques). Les fonctions d'activations ont ici pour but de produire des décisions non-linéaires en sortie de chaque neurone afin de simuler un effet de seuil. Elles sont nécessairement continues, dérivables et de dérivées continues afin de permettre le calcul du gradient et l'optimisation du MLP. Les fonctions d'activation les plus courantes sont la tangente hyperbolique ($f(v) = tanh(z)$) et la sigmoide ($f(v) = (1+e^{-z})^{-1}$).

En pratique, un MLP va avoir une couche d'entrée avec un nombre de neurones correspondant à la taille du vecteur de caractéristiques. Sa couche de sortie aura quant à elle un nombre de neurones k, égal au nombre de classes que l'on cherche à reconnaitre. Chaque neurone de la couche de sortie va générer une probabilité que la donnée traitée appartienne à la classe correspondant à l'index du neurone (*i.e.*, un neurone par étiquette). On considère généralement l'index qui correspond à la probabilité la plus élevée comme étant celui de la classe prédite (*top-1 classification*), mais il est également possible de considérer un résultat positif si sa probabilité est parmi le k plus élevées (*top-k classification*). Outre ces deux couches, le MLP pourra avoir un nombre non pré-déterminé de couches dites cachées, c'est à dire entre la couche d'entrée et la couche de sortie. Plus un réseau de neurones aura de couches cachées, plus il sera dit profond. Un exemple d'un tel réseau de neurones est schématisé sur la Figure 2.10.

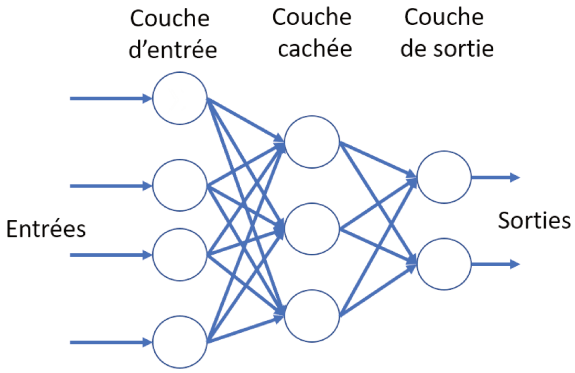


FIGURE 2.10 – Schéma illustrant le principe d'un réseau de neurones.

Plus formellement. Soit l le numéro de la couche neuronale considérée, alors le signal se propage des k neurones de la couche $(l-1)$ au neurone j de la couche (l) via l'équation (2.22), où a_k indique la sortie générée par le neurone k de $(l-1)$, w_{jk} le poids entre les neurones j et k, et $g(.)$ est la fonction d'activation. Les indices (l) et $(l-1)$ indiquent la couche à laquelle appartient l'élément.

$$a_j^{(l)} = g^{(l)}(z_j^{(l)}) = g^{(l)}(\sum_k w_{jk}^{(l)} a_k^{(l-1)} + b_j^{(l)}) \tag{2.22}$$

Les poids w et les biais b représentent les paramètres du réseau. Ils sont appris sur un ensemble d'entraînement X_{train} à l'aide de l'algorithme de rétropropagation du gradient [RHW86; Wer90]. Dans sa forme la plus simple, ce dernier correspond à une descente stochastique du gradient (SGD) à l'aide de dérivées composées. Pour cela, on définit l'erreur réalisée par le réseau en calculant la différence entre la sortie estimée $\hat{y}_j$ par les j neurones de la couche de sortie et la vérité terrain correspondante y_j. L'erreur est ici calculée à l'aide d'une fonction de coût $\mathscr{L}(\hat{y}_j, y_j)$, à définir en fonction des applications. L'algorithme de rétropropagation du gradient permet alors de calculer la dérivée partielle, les gradients, de la fonction de coût par rapport à chacun des paramètres du réseau, tels que défini par les équations (2.23), (2.24), (2.25), et (2.26), où $g^{(l)\prime}(.)$ est la fonction dérivée de la fonction d'activation de la couche (l), $\mathscr{L}'(.)$ est la fonction dérivée de la fonction de coût, et l'indice $sortie$ indique la dernière couche du réseau.

$$\delta_j^{sortie} = g^{sortie\prime}(z_j^{sortie})\mathscr{L}'(\hat{y}_j, y_j) \tag{2.23}$$

$$\delta_j^{(l)} = g^{(l)\prime}(z_j^{(l)})\sum_{jk} w_{jk}^{(l+1)}\delta_j^{(l+1)} \tag{2.24}$$

$$\frac{\partial \mathscr{L}}{\partial w_{jk}^{(l)}} = a_j^{l-1}\delta_j^{(l)} \tag{2.25}$$

$$\frac{\partial \mathscr{L}}{\partial b_j^{(l)}} = \delta_j^{(l)} \tag{2.26}$$

Ces dérivées partielles permettent de mettre à jour les paramètres du réseau à l'aide des équations (2.27) et (2.28), où λ est le taux d'apprentissage (*learning rate*, LR), dont la valeur est généralement faible pour éviter les variations trop fortes des paramètres. Celui-ci défini la vitesse de mise à jour des paramètres lors de la descente de gradient.

$$w_{jk}^{(l)} = w_{jk}^{(l)} - \lambda\frac{\partial \mathscr{L}}{\partial w_{jk}^{(l)}} \tag{2.27}$$

$$b_j^{(l)} = b_j^{(l)} - \lambda\frac{\partial \mathscr{L}}{\partial b_j^{(l)}} \tag{2.28}$$

Des algorithmes de mise à jour des paramètres plus évolués, tenant notamment compte de l'intensité du gradient et des variations passées (*momentum*), ont par ailleurs été proposés dans la littérature (*e.g.*, RMSPROP [HSS12], ADAM [KB14]). Ces algorithmes permettent d'optimiser les réseaux de neurones plus rapidement et plus efficacement en régularisant les variations des paramètres.

En pratique, l'entraînement d'un réseau de neurones se fait à l'aide d'un sous ensemble de données d'entraînement à chaque itération. Ce sous ensemble est nommé *batch*, et la taille du *batch* correspond au nombre d'échantillons utilisés par itération. L'utilisation d'un *batch* permet de cumuler l'erreur sur plusieurs échantillons de données avant de calculer et de rétropropager le gradient. Cela permet d'accélérer l'entraînement via la parallélisation des algorithmes.

Les hyperparamètres principaux du MLP concernent le choix du taux d'apprentissage, le choix de la fonction de coût, le choix des fonctions d'activation, le choix de l'algorithme de mise à jour des paramètres, le nombre de couches cachées et le nombre de neurones par couche ; qui définissent la profondeur du réseau ainsi que le nombre de paramètres à optimiser. En pratique, il est difficile d'estimer *a priori* ces hyperparamètres et une étape de validation peut être nécessaire. La règle générale veut cependant que plus il y a de neurones, plus il y a de paramètres, donc plus les représentations qu'un réseau de neurones pourra apprendre seront complexes. En revanche, plus il aura de paramètres, plus il y aura besoin d'un grand nombre de données pour que l'optimisation converge vers une solution mathématiquement optimisée. De la même manière, plus la quantité de paramètres est importante, plus l'entraînement des réseaux de neurones est lent.

2.4 Réseaux de neurones à convolutions

Les réseaux de neurones à convolutions (*Convolutional Neural Network*, CNN) sont apparus dès la fin des années 1990 [LBBH98], mais ne sont vraiment devenus populaires qu'à partir de 2012 avec le succès sans précédent de ces approches pour la classification à grande échelle [KSH12] sur le jeu de données ImageNet (ILSVRC 2012) [RDS+15].

Ils permettent d'étendre la notion de neurones aux filtres de convolutions, usuellement utilisés pour filtrer les images. Le but recherché est ici double : (1) permettre le partage des poids d'un réseau de neurones pour l'ensemble des pixels d'une image afin de limiter le nombre de paramètres à optimiser et (2) apprendre des filtres de convolutions optimisés pour la tâche à accomplir afin d'extraire automatiquement des caractéristiques.

Réseaux de neurones à convolutions

A l'instar des réseaux de neurones classiques (*e.g.*, MLP), les réseaux de neurones à convolutions vont être constitués de couches, appelées couches de convolutions. La plupart des réseaux actuels sont tous profonds (on parle alors de *Deep Convolutional Neural Network*, DCNN). Cependant, à la place d'un modèle uniquement basé sur le perceptron, certaines de ces couches vont être constituées d'opérateurs issus du traitement du signal et de l'image, tels que le filtre de convolutions (la convolution étant ici un neurone, remplaçant le perceptron) et le *pooling* (opération de sous-échantillonnage non linéaire). En particulier, les filtres de convolutions vont permettre de filtrer l'image afin d'en extraire des caractéristiques, qui pourront ensuite servir d'entrées à des couches dites entièrement connectées (*Fully Connected*, FC) représentées par un MLP (cas de la classification). Le grand avantage de cette approche est qu'elle permet d'optimiser simultanément l'algorithme de classification et les filtres d'extraction de caractéristiques (voir Figure 2.11).

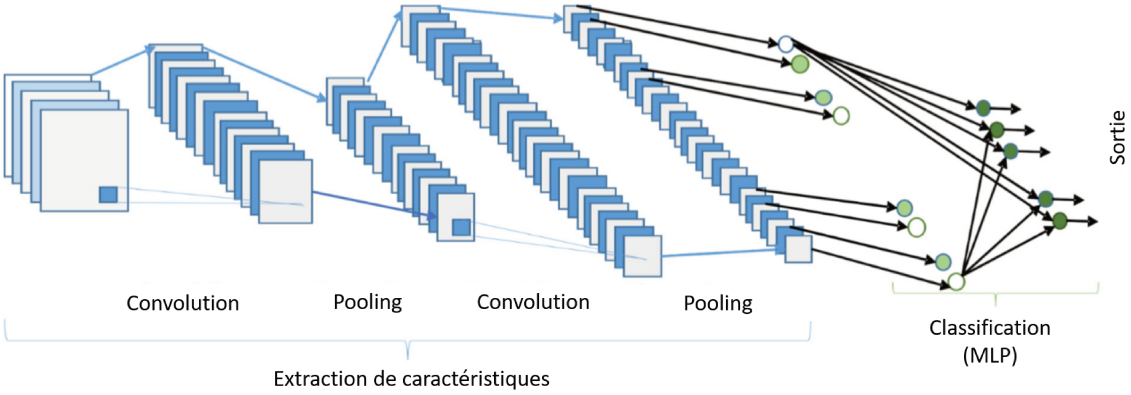


FIGURE 2.11 – Schéma illustrant le principe d'un réseau de neurones à convolutions pour la classification. Image adaptée de [ATY+19].

Réseaux de neurones entièrement convolutifs

L'utilisation de réseaux de neurones entièrement convolutifs (FCN) a été également explorée [LSD15]. Pour cela, il suffit de supprimer les couches entièrement connectées et de les remplacer par des filtres convolutionnels. Les FCN sont particulièrement populaires pour la génération de données [RMC15; YGZS17] et la segmentation sémantique [BKC17; ALSL16; RFB15]. En particulier, l'utilisation d'architectures encodeur-décodeur tend à être privilégiée. Celle-ci permet d'encoder une image à l'aide de couches convolutives et d'opérations de *pooling*, avant de la décoder à l'aide de convolutions transposées (ces opérations sont décrites ci-après). Une illustration d'un réseau de neurones de type encodeur-décodeur est présenté sur la figure 2.12.

Dans la suite de cette section, nous allons présenter les blocs de base régulièrement utilisés avec les réseaux de neurones à convolutions, avant de brièvement présenter les applications de ces méthodes pour l'analyse d'images aériennes et satellites. A noter qu'une revue de la littérature générique des réseaux de neurones profonds et de leurs architectures a été réalisée en 2019 par Alom *et al.* [ATY+19], mettant en avant l'engouement général pour ces approches.

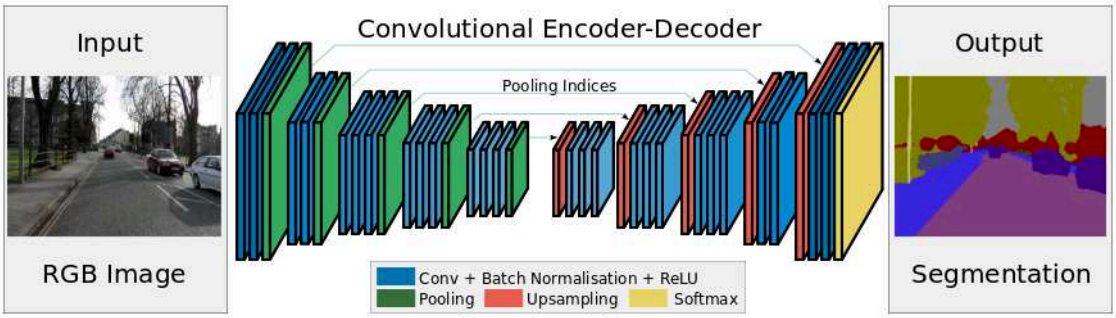


FIGURE 2.12 – Schéma illustrant le principe d'un réseau de neurones entièrement convolutif (FCN) de type encodeur-décodeur pour la segmentation sémantique. Il s'agit ici du réseau SegNet [BKC17].

2.4.1 Blocs de base

Nous décrivons ici certains des blocs de base pouvant être utilisés au sein d'un réseau de neurones à convolutions.

Filtres de convolutions

Les filtres de convolutions sont l'essence même des réseaux de neurones à convolutions. Ce sont eux qui vont permettre de filtrer la donnée. En pratique, ils vont pondérer et sommer les pixels d'une image au travers d'une opération de convolution (opération bilinéaire, associative, commutative). Par définition, ces filtres sont donc invariants à la position des pixels dans l'image. Dit autrement, les poids (*i.e.*, les paramètres) des neurones convolutifs sont partagés par tous les pixels de l'image.

La dimension de ces filtres est définie en nombre de pixels par la hauteur H, la largeur W, et la profondeur C. Ces dimensions définissent le nombre de paramètres du filtre qui peuvent être optimisés par descente de gradient ($W \times H \times C$). La profondeur correspond ici au nombre de canaux de l'image qui vont être vus par le filtre. Ces filtres prennent généralement une taille impaire de pixels afin que le résultat soit centré sur le pixel central du filtre. Les filtres de taille supérieure à $1 \times 1 \times C$ ($H \times W \times C$) vont dépasser de l'image lorsqu'ils vont traiter des pixels aux bords de celle-ci. Une solution courante est d'ajouter des pixels aux bords de l'image. Cette opération se nomme le *padding*, et va généralement de pair avec la convolution. Les pixels ajoutés par *padding* peuvent être de plusieurs types, tels que des zéros, du bruit blanc, ou une copie des pixels aux bords de l'image. Elle permet d'obtenir une image filtrée de la même taille que l'image en entrée. Ce constat n'est cependant valable que lorsque la fenêtre glissante représentant le filtre de convolutions va parcourir les pixels de l'image avec un pas de 1 (*stride* égal à 1). Le fait de faire varier ce pas permet de sous-échantillonner l'image sans avoir recours à des opérations supplémentaires (*e.g.*, un pas de 2 divisera les dimensions H et W de l'image par deux). Un exemple de filtre de convolutions de taille $3 \times 3 \times 1$ appliqué sur une image mono-canal est présenté sur la figure 2.13.

Il existe par ailleurs plusieurs types de convolutions, que nous détaillons ci-dessous. Certaines, parmi les plus courantes, sont également représentées sur la figure 2.14.

- **Convolution classique**. Il s'agit ici du filtre de convolutions standard de taille $H \times W \times C$. L'application d'une convolution classique avec *padding* sur une image de taille $I_h \times I_w \times I_c$ va générer une image filtrée de taille $I_h \times I_w \times 1$ (agrégation de l'information spectrale contenue par les différents canaux). Il est de fait nécessaire d'utiliser plusieurs filtres de convolutions différents pour générer des images filtrées différentes. Ces images filtrées sont aussi appelées cartes de caractéristiques (*features map*). A titre d'exemple, l'application de n_f filtres de convolutions va permettre de générer n_f cartes de caractéristiques, qui seront concaténées dans la dimension des canaux (image de taille $I_h \times I_w \times n_f$).

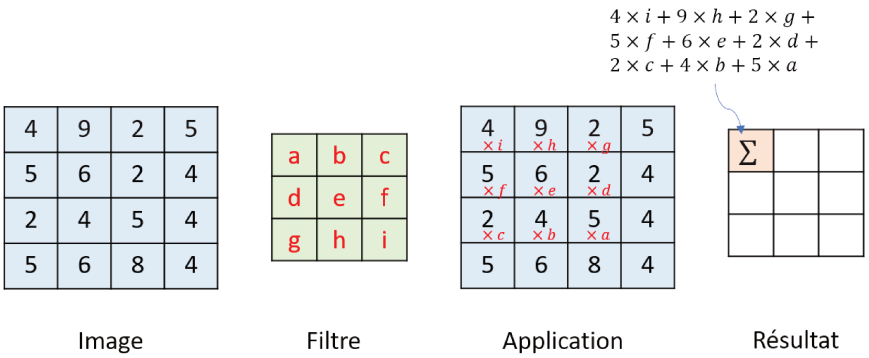


FIGURE 2.13 – Schéma illustrant le principe de la convolution classique sans *padding*. La rotation du filtre est ici effectuée à titre d'illustration afin de respecter les propriétés de l'opération de convolution. Celle-ci n'est généralement pas appliquée dans les réseaux de neurones à convolutions. On confond alors convolution et corrélation.

- **Convolution** 1 × 1. Les convolutions 1 × 1 sont des filtres de convolutions classiques dont le seul but est de réaliser une combinaison linaire des canaux de l'image d'entrée. Ils sont de taille 1 × 1 × C.
- **Convolution dilatée**. Les convolutions dilatées vont être des filtres de convolutions avec des trous. Le but est ici de pouvoir considérer des filtres de même taille que les convolutions classiques H × W × C tout en observant un voisinage de rayon plus grand (nommé paramètre de dilatation) [CPSA17].
- **Convolution transposée**. Les convolutions classiques vont permettre de générer des images filtrées sans modifier leur taille (avec *padding*). Les convolutions transposées vont quant à elle chercher à filtrer une image en augmentant sa taille. Pour cela, l'image d'entrée est d'abord aplatie en un vecteur, et le filtre de convolutions appliqué par fenêtre glissante est représenté sous la forme d'une matrice éparse que l'on va transposer et multiplier à l'image aplatie. Les filtres de convolutions transposées sont également de taille H × W × C [LSD15].
- **Convolution séparable**. Les filtres de convolutions classiques sont symétriques par rapport à un pixel central. Ils ont de fait une forme rectangulaire. Les convolutions séparables se basent sur l'observation que certains de ces filtres peuvent être obtenu par convolution d'un filtre horizontal (W × 1 × C) et d'un filtre vertical (1 × H × C). Par propriété d'associativité de la convolution, il est possible d'appliquer ces deux filtres l'un après l'autre sur une image afin d'obtenir le même résultat qu'un filtre de convolutions classique. Leur utilisation permet de réduire la quantité de paramètres [MG12].
- **Convolution séparable en profondeur**. On va ici décomposer la donnée d'entrée en plusieurs groupes de canaux, et appliquer des filtres convolutifs différents sur chaque groupe. On génère ainsi 1 carte de caractéristique par groupe ($C_{out} = C_{in}/\#groupes$). On va ensuite appliquer des convolutions 1 × 1 pour générer plusieurs cartes de caractéristiques. Le nombre de canaux par groupe est défini par un hyperparamètre. Cette approche permet de réduire significativement le nombre de paramètres utilisés [HZC+17].

Fonctions d'activation

Tout comme avec les MLP, une fois la pondération et l'agrégation des valeurs réalisées (ici, par des filtres de convolutions), il est courant d'appliquer une fonction dite d'activation. Les plus courantes sont :

- Unité linéaire rectifiée (ReLU) : $g(z_i) = max(0, z_i)$
- Tangente hyperbolique : $g(z_i) = tanh(z_i)$
- Sigmoïde : $g(z_i) = (1 + e^{-z_i})^{-1}$

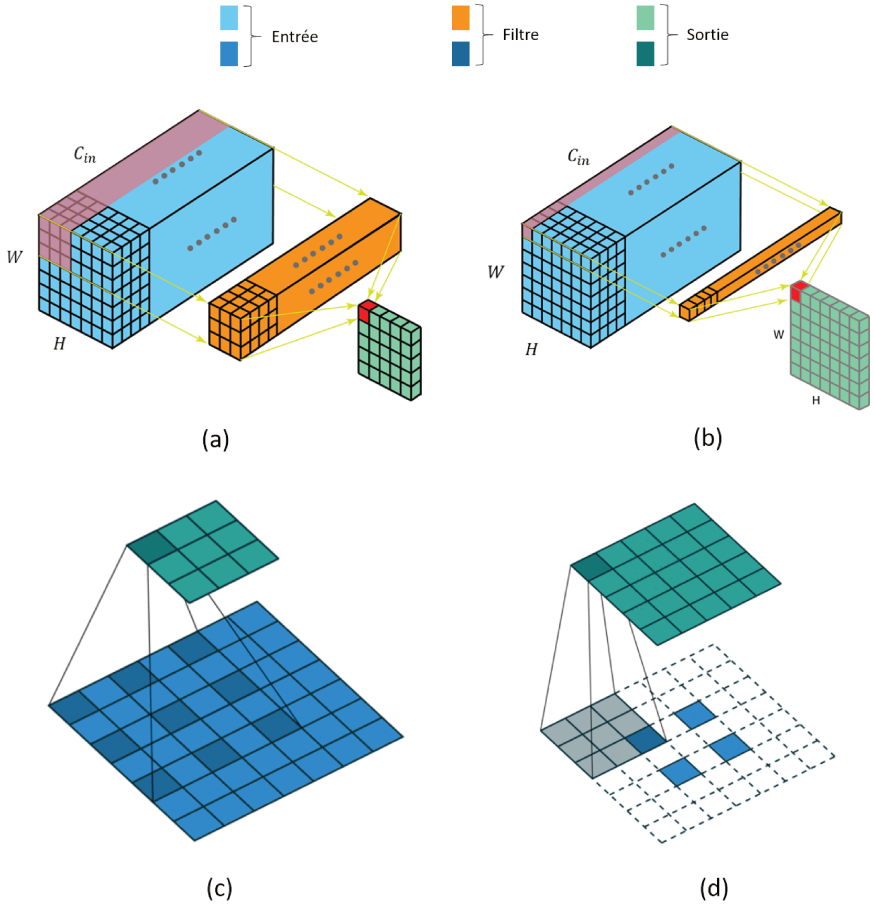


FIGURE 2.14 – Schéma illustrant plusieurs filtres de convolutions 2D. (a) Convolution classique. (b) Convolution 1×1. (c) Convolution dilatée. (d) Convolution transposée. Images extraites de l'article de blog *A comprehensive introduction to different types of convolutions in deep learning*[5].

- Softmax : $g(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$

Ces fonctions vont être appliquées à chaque pixel des cartes de caractéristiques. En pratique, le choix d'une telle fonction est guidé par le besoin d'avoir des données bornées (cas de la tangente hyperbolique et de la sigmoïde), et par la complexité du calcul de la dérivée de ces fonctions. La fonction *softmax* est en générale utilisée sur la couche de sortie d'un réseau classifieur afin que la somme des probabilités générées soit égale à 1. Il est possible d'utiliser plusieurs fonctions d'activation au sein d'un même réseau (*e.g.*, ReLU pour toutes les couches, et Softmax pour la couche de sortie).

Pooling

Afin de réduire l'empreinte mémoire et de réduire progressivement la taille des cartes de caractéristiques en vue de générer un vecteur, des opérations de sous-échantillonnage de type *pooling* sont généralement appliquées (notion de couche de *pooling*). Pour cela, nous définissons une fenêtre carrée de taille $S \times S$. Le pas d'application de cette fenêtre est en général égal à S pour l'opération de *pooling* (dit autrement, un même pixel ne sera vu qu'une seule fois par l'opérateur). Chaque opérateur de *pooling* va permettre de conserver une seule valeur par pas, permettant ainsi de réduire la taille de l'image. A titre d'exemple, un opérateur de taille 2×2 avec un pas de 2 va permettre de diviser par 2 la taille de son entrée. Cette opération est appliquée de façon indépendante sur chaque canal.

5. https ://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215 (accès : 2020-06-15)

Les opérateurs les plus courants sont les suivants. Ils sont illustrés sur la figure 2.15.

- ***Max-pooling.*** Dans un voisinage donné, ne conserve que la valeur maximale.
- ***Average-pooling.*** Dans un voisinage donné, calcule la moyenne.
- ***Min-pooling.*** Dans un voisinage donné, ne conserve que la valeur minimale.

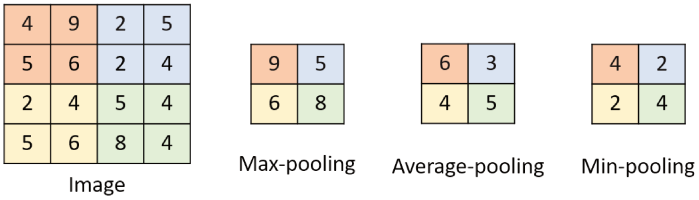


FIGURE 2.15 – Schéma illustrant le principe du *pooling* avec une taille de 2 pixels et un pas de 2 pixels.

En pratique, le *Max-pooling* tend à être privilégié du fait de sa rapidité et car il permet de conserver les éléments avec les intensités les plus élevées, souvent perçus comme étant les plus saillants.

Normalisation

Afin d'améliorer l'apprentissage des paramètres d'un réseau de neurones à convolutions, l'utilisation de normalisation est régulièrement utilisée. On distinguera ici deux types principaux de normalisation, à savoir la normalisation par *batch* et la normalisation par instance, mais d'autres approches telles que la normalisation par couche [BKH16] existent.

- **Normalisation par *batch* [IS15].** L'idée est ici de remplacer la moyenne et la variance d'un *batch* à l'aide de deux paramètres à apprendre (γ, β) afin de régulariser et d'accélérer l'entraînement des réseaux de neurones profonds à convolutions [IS15]. Soit un *batch* $\mathrm{B} = \{x_{1\ldots m}\}$ de m échantillons. On calcule la moyenne $\mu_\mathrm{B} = \frac{1}{m}\sum_{i=1}^{m} x_i$ et la variance $\sigma_\mathrm{B}^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\mathrm{B})^2$ du *batch*. On normalise chaque échantillon du *batch* $\hat{x}_i = \frac{x_i - \mu_\mathrm{B}}{\sqrt{\sigma_\mathrm{B}^2 + \epsilon}}, 1 >> \epsilon > 0$. Enfin, on multiplie le résultat centré réduit par γ (*i.e.*, on modifie la variance), et on ajoute β (*i.e.*, on modifie la moyenne) : $\gamma\hat{x}_i + \beta$. En pratique, la normalisation par *batch* est appliquée sur les cartes de caractéristiques extraites par les couches convolutives, éventuellement après l'application d'une fonction d'activation.
- **Normalisation par instance [UVL16].** Il s'agit ici de normaliser chaque canal de chaque carte de caractéristiques de façon indépendante. Le but est ici de rendre le réseau à convolutions indépendant du contraste des images d'origine. Cette approche a montré son intérêt pour le transfert de style et la génération d'images, permettant d'obtenir des résultats plus vraisemblables que la normalisation par *batch*.

2.4.2 Application des réseaux de neurones à convolutions en télédétection

Nous allons ici aborder les applications principales des réseaux de neurones à convolutions utilisés pour les images aériennes et satellites.

Dans une méta-analyse de la littérature réalisée en 2019 par Ma *et al.* [MGS+19] mettaient en avant une quantité exponentiellement croissante de travaux contenant les mots clefs "deep learning" (apprentissage profond) et "remote sensing" (télédétection), avec une forte prédominance des approches basée sur les réseaux de neurones à convolutions. Les auteurs distinguent plusieurs applications principales, à savoir la reconnaissance des classes d'occupation du sol (Land Use Land Cover classification, LULC classification), la détection d'objets, la reconnaissance de scènes, la fusion d'informations ou encore la segmentation. En 2016, Masi *et al.* [MCVS16] proposaient

ainsi de fusionner des images multispectrales de basse résolution avec des images panchromatiques de haute résolution (pansharpening) à l'aide d'un CNN. Pour cela, Masi *et al.* ont utilisé un réseau de neurones entièrement convolutif prévu pour la super-résolution (*i.e.*, l'agrandissement des images en minimisant les pertes) qu'ils ont conditionné à la fois sur les images panchromatiques et sur les images multispectrales. Audebert *et al.* [ALSL16] proposaient l'utilisation d'un réseau entièrement convolutif suivant une architecture encodeur-decodeur pour segmenter des images multispectrale acquises en environnement urbain. Pour cela, les auteurs proposaient une approche multi-échelle en combinant les sorties générées par trois convolutions transposées de tailles différentes. Encore en 2016, Maggiori *et al.* [MTCA16] utilisaient un réseau entièrement convolutif pour détecter des bâtiments à partir d'images aériennes de haute résolution ($\geq 1 \quad m$). La même année, Chen *et al.* [CJL+16] proposaient l'utilisation de réseaux de neurones à convolutions pour extraire et classifier des caractéristiques à partir d'images hyperspectrales. En 2017, Wang *et al.* [WLH+17] étudiaient la possibilité d'affiner les poids de réseaux de neurones à convolutions pré-entraînés pour classifier des images aériennes [YN10] et satellites [PNDS15]. En 2018, Kellenberger *et al.* [KMT18] s'intéressaient à l'utilisation des réseaux de neurones à convolutions pour la détection de mammifères à partir d'images acquises par drone. Lin *et al.* [LFW+17] s'intéressaient quant à eux à l'apprentissage non supervisé de caractéristiques pour classifier des images de télédétection. Ils ont pour cela utilisé un réseau de neurones adversaire afin d'entraîner un réseau de neurones discriminant à reconnaitre de vraies images aériennes d'images aériennes générées. Les paramètres du réseau discriminant sont ensuite fixés, et celui-ci est utilisé pour extraire des vecteurs de caractéristiques afin de décrire des images aériennes et de les classifier en plusieurs classes d'occupation du sol. En 2018 encore, Maltezos *et al.* [MPD+18] s'intéressaient à l'utilisation des réseaux de neurones à convolutions pour détecter les ombres et les bâtiments à partir d'images aériennes.

2.5 Conclusion et positionnement

Nous avons introduit plusieurs blocs de base de la littérature pour l'extraction de caractéristiques, la classification, la sur-segmentation et l'apprentissage automatique d'approches "bout en bout". Cependant, ces approches n'ont que très peu été appliquées sur des images aériennes historiques panchromatiques. Dans le cadre de nos travaux, nous avons dans un premier temps cherché à générer des cartes d'occupation du sol en nous basant sur la classification de textures et les réseaux de neurones profonds à convolutions. Notre but était d'obtenir rapidement des résultats vraisemblables. Pour cela, nous avons notamment réalisé une étude comparative des approches existantes de type LBP et de DCNN, auxquelles nous avons pu proposer deux nouvelles variantes de filtres de type LBP (chapitre 3). Dans un second temps, nous avons cherché à exploiter et à développer des réseaux de neurones entièrement convolutifs pour la colorisation d'images aériennes historiques, et ce dans le but de proposer une visualisation alternative de ces données et d'améliorer les résultats obtenus par classification (chapitre 4). Pour cela, nous nous sommes tout particulièrement intéressés aux approches non-supervisées, permettant d'optimiser les DCNN à l'aide de données dont la vérité terrain (la couleur des images historiques dans notre cas) n'est pas connue. Enfin, nous avons cherché à améliorer les cartes d'occupation du sol obtenues par classification dans un contexte de post-traitement à l'aide de sur-segmentations que nous avons intégrées au sein d'un champs aléatoire conditionnel (chapitre 5). En particulier, nous avons proposé l'utilisation d'une représentation intermédiaire pour la génération de superpixels à l'aide d'un DCNN optimisé pour l'estimation de bords sémantiquement intéressants.

Chapitre 3

Classification de textures

Ce chapitre présente les travaux que nous avons réalisés concernant la classification automatique de textures, principalement appliquée aux images aériennes historiques. Pour réaliser ces travaux, nous avons dans un premier temps constitué un jeu de données annotées, que nous avons nommé HistAerial. Ce jeu de données nous a permis de comparer l'utilisation de plusieurs méthodes d'extraction de caractéristiques et de classification, ainsi que des réseaux de neurones à convolutions. Nous avons ensuite étendu nos travaux à l'analyse d'images couleur extraites d'écorces d'arbres dans le cadre d'une collaboration avec une autre doctorante. Le but était ici de vérifier la possibilité de combiner des caractéristiques issues d'images en niveaux de gris avec des caractéristiques de couleur afin d'améliorer le pouvoir discriminant des représentations obtenues. Ces expériences nous ont par la suite menées à réaliser des travaux sur la colorisation automatique, présentés dans le chapitre suivant.

Sommaire

3.1 Introduction 52
3.2 HistAerial, un nouveau jeu de données 53
3.2.1 Images sources 53
3.2.2 Propriétés des images sources 53
3.2.3 Génération du jeu de données 54
3.3 Algorithmes évalués sur HistAerial 58
3.3.1 Algorithmes d'extraction de caractéristiques de la littérature 58
3.3.2 Proposition de nouveaux filtres pour la texture 63
3.3.3 Classifieurs utilisés avec les descripteurs de textures 65
3.3.4 Réseaux de neurones profonds à convolutions évalués 65
3.4 Résultats et discussions 67
3.4.1 Mise en place des expériences 67
3.4.2 Comparaison globale 69
3.4.3 Importance du contexte spatial 73
3.4.4 Conclusion partielle 75
3.5 Extension aux images en couleurs : cas des écorces d'arbres 75
3.5.1 Jeux de données 76
3.5.2 Méthodes 77
3.5.3 Expériences et résultats 79
3.5.4 Conclusion partielle 80
3.6 Conclusion 81

3.1 Introduction

La reconstruction de l'occupation du sol est une tâche particulièrement populaire en télédétection. De nombreux travaux se sont ainsi intéressés à segmenter et à classifier automatiquement les sols observés par des dispositifs aériens et satellites. A titre d'exemple, Kussul et al. [KLSS17] ont proposé de classifier les champs cultivés à l'aide de réseaux de neurones profonds. Albert et al. [AKG17] ont utilisé ce même type d'outil pour analyser l'environnement urbain à partir d'images satellites. Slimene et al. [SCB+17] ont mis au point une approche d'apprentissage actif pour segmenter les parcelles cultivées à l'aide d'indices de végétations extraits d'images satellites multi-spectrales. L'extraction de caractéristiques de textures pour analyser les sols a également été beaucoup étudiée (voir chapitre 2). En 2018, Wegner et al. [WTYM18] ont ainsi présenté un aperçu des algorithmes de vision par ordinateur utilisés dans la littérature pour analyser les images aériennes et satellites de très hautes résolutions.

Cependant, il semblerait que très peu de travaux se soient à ce jour intéressés à l'utilisation des images aériennes historiques panchromatiques. Ce point peut s'expliquer par la mise à disposition relativement tardive de ces données face à l'abondance d'images multi-spectrales actuelles (voir chapitre 1). Néanmoins, les données historiques tendent à gagner de l'importance dans le cadre d'études rétrospectives appliquées à l'environnement, la santé, ou l'urbanisme. C'est en particulier le cas des études épidémiologiques s'intéressant aux maladies dont le développement peut être long. Dans notre cadre de travail, l'étude TESTIS [BPB+14] vise (entre autres) à utiliser l'occupation du sol historique pour estimer un score d'exposition aux pesticides d'origine agricole afin de comprendre les déterminants du développement du cancer du testicule. D'autres travaux proches de TESTIS, tels que ceux de Brouwer et al. [BHvdM+17], proposent de raffiner les occupations du sol historiques (annotées manuellement, 3 classes) à l'aide d'occupations du sol actuelles (9 classes) et d'un modèle statistique, et ce dans le but d'analyser l'impact de l'exposition aux pesticides sur la maladie de Parkinson. Dans notre cas, nous supposons que l'occupation du sol historique n'est généralement pas connue, car coûteuse à obtenir en termes de temps (voir chapitre 1). Par ailleurs, le territoire français vu du ciel est particulièrement difficile à analyser : les zones naturelles comme artificielles ne sont pas symétriques et n'ont pas de formes représentatives ou répétitives comparées à d'autres territoires, tels que ceux analysés par Yan et al. [YR14] aux États-Unis. De plus, durant la période d'intérêt de l'étude TESTIS (après 1970), la France a connu des changements démographiques qui ont modifié les paysages urbains et ruraux, ce qui augmente la variabilité des représentations potentielles. Tout l'enjeu est alors de déterminer l'efficacité des algorithmes de vision par ordinateur existants pour reconnaitre l'occupation du sol à partir de ces données, et d'explorer des alternatives éventuellement plus performantes. A titre illustratif, la figure 3.1 présente deux exemples d'images aériennes historiques et leurs occupations du sol dans un rayon de 1.5 km (cas de l'étude TESTIS). Les occupations du sol ont ici été annotées manuellement par des géomaticiens. Elles représentent le résultat que l'on souhaiterait, dans l'absolu, obtenir.

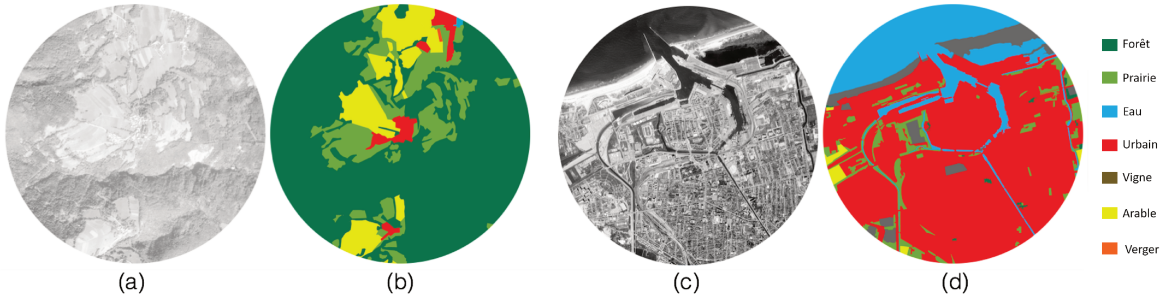


FIGURE 3.1 – Deux exemples d'images aériennes historiques acquises en France (a)(c) et leurs occupations du sol annotées manuellement par des géomaticiens (b)(d).

Pour cela, nous nous intéressons ici à la classification d'images aériennes historiques comme substitut pour la génération de cartes d'occupation du sol. Le choix de nous intéresser à la classification a été fait afin de pouvoir comparer sur une base commune des chaines de traitements classiques basées sur l'extraction de caractéristiques de textures et des réseaux de neurones à convolutions (voir chapitre 2). Afin de réaliser cette étude, nous avons dans un premier temps construit un jeu de données que nous avons nommé HistAerial, composé de plusieurs millions d'imagettes annotées. Nous présentons ce jeu de données dans la section 3.2. Nous avons ensuite sélectionné et comparé des algorithmes d'extraction de caractéristiques et de classification (filtres de type LBP + classifieur, réseaux de neurones à convolutions). Nos travaux ont été intégrés dans le logiciel Gouramic, présenté en Annexe A. Enfin, nous avons collaboré avec une autre doctorante, travaillant sur la reconnaissance des végétaux, pour étendre nos résultats à des images texturées en couleurs (écorces d'arbres), et ce dans le but de déterminer si la couleur pouvait permettre d'améliorer les résultats obtenus en classification.

3.2 HistAerial, un nouveau jeu de données

Cette section présente le jeu de données HistAerial, que nous avons créé en collaboration avec les géomaticiens du département Cancer et Environnement du Centre Léon Bérard afin d'évaluer des chaines de traitements pour la classification des images aériennes historiques. Ce jeu de données a été mis gratuitement et publiquement à la disposition de la communauté afin d'encourager les efforts de développements pour l'analyse automatique de ce type d'images (voir `http://eidolon.univ-lyon2.fr/~remi1/HistAerialDataset/`).

3.2.1 Images sources

HistAerial a été conçu à partir d'images aériennes historiques panchromatiques acquises en France entre les années 1970 et 1990. Elles ont été téléchargées via le service remonterletemps de l'IGN [IGN20]. Ces images sont disponibles sans annotations de l'occupation du sol. De par la faible quantité de données disponibles en infrarouge et en couleurs dans les années 1970 et 1980, seules des images panchromatiques ont été ici utilisées (voir chapitre 1). Une fois les images téléchargées, elles ont été géoréférencées manuellement par les géomaticiens du Centre Léon Bérard via le logiciel ArcGis. Pour cela, 7 points d'ancrage en moyenne ont été utilisés pour projeter les images dans un repère géographique (Lambert 93). Le Lambert 93 est le système de projection en vigueur en France. Le choix des images à télécharger a ici été fait afin d'obtenir des zones de rayon 1.5 km à partir de l'adresse des sujets recrutés pour une précédente étude au Centre Léon Bérard (TESTEPERA, en région Rhônes-Alpes, France), les sujets de l'étude TESTIS n'ayant pas tous été recrutés et géocodés lorsque nous avons démarré nos travaux. A ces données se sont ajoutées des images d'autres zones géographiques en France qui ont été sélectionnées afin de combler la présence relativement faible de certains types d'occupations du sol sur les premières images. Au total, 81 images annotées ont été utilisées (ordre de grandeur de la taille des images : 6000×6000 pixels). Les détails liés à l'annotation de ces images sont présentés en Section 3.2.3.

3.2.2 Propriétés des images sources

Avant de détailler la construction du jeu de données HistAerial à proprement parler, nous allons d'abord nous intéresser aux propriétés des images aériennes historiques utilisées. Ces propriétés ayant déjà été partiellement introduites dans le chapitre 1, il s'agit ici de mettre en avant les difficultés que ces images représentent d'un point de vue traitement d'image.

Les images utilisées pour constituer HistAerial ont les propriétés suivantes :

- Elles sont monochromatiques.

- Elles sont de hautes résolutions. Il a été estimé sur un sous échantillon de 25 images que la résolution des images utilisées varie de 0.17 à 1.4 mètres, pour une résolution moyenne estimée à 0.5 mètres.
- Elles ont été acquises durant les périodes estivales, lorsque le soleil était haut dans le ciel et que peu de nuages étaient présents. Ces conditions permettent de limiter l'apparition d'ombres portées par les bâtiments et les arbres, et de limiter également l'apparition de nuages sur les clichés photographiques.
- Elles sont géolocalisées suite au géoréférencement réalisé. Cela signifie qu'il est potentiellement possible d'exploiter des méta-données géographiques en plus des images. Ce point constitue une perspective potentielle à nos travaux.
- Pour une coordonnée géographique donnée (un sujet), plusieurs images aériennes peuvent avoir été acquises dans le temps. Dans le cadre de nos travaux, les géomaticiens ont estimé qu'il y avait peu de chance d'obtenir plus d'une zone d'intérêt par an (une zone d'intérêt pouvant être constituée de plusieurs images).
- La qualité exacte des images est supposée inconnue et variable. Ce point est dû au fait que les systèmes d'acquisition et de numérisation (non connus pour nous) ont pu évoluer avec le temps, et que les conditions d'acquisitions extérieures sont incontrôlables (*e.g.*, présence de vent, de poussière, *etc.*).
- Les images ont été acquises dans un passé lointain, ce qui empêche l'acquisition de nouvelles données pour les périodes étudiées.

Ces propriétés induisent une variabilité intra-classe élevée (*i.e.*, une même classe d'occupation du sol peut être représentée à l'aide d'images très différentes), ainsi qu'une variabilité inter-classe faible (*i.e.*, des images de classes d'occupation du sol différentes se ressemblent). Cette remarque est valable à la fois dans l'espace et dans le temps (*e.g.*, les cultures et les prairies n'ont pas une représentation statique, et ces représentations varient d'une région à une autre). Il n'est par ailleurs pas possible de se baser sur des informations telles que l'index NDVI ou les distributions multispectrales [HLZ14] pour distinguer les différentes classes d'occupation du sol car seul le canal panchromatique est disponible. Enfin, l'écart temporel important entre deux images pour une localisation géographique donnée associé aux modifications du territoire dans le temps compliquent l'utilisation de séries temporelles pour produire des résultats plus robustes, tels que ceux obtenus par Kussul et al. [KLSY16].

3.2.3 Génération du jeu de données

Annotations

Les annotations manuelles ont été réalisées à l'échelle de la parcelle de terrain à l'aide de 7 classes d'occupation du sol, à savoir : Verger, Terres Arable (abrégé Arable par la suite), Prairie, Vigne, Urbain, Forêt, Eau. Ces annotations ont été réalisées densément (*i.e.*, tous les pixels de l'image ont été annotés), avec l'ensemble des classes disponibles, pour 56 des 81 images aériennes. Les autres annotations ont été réalisées de façon ciblée (*i.e.*, seules certaines parcelles ont été annotées) afin de combler le manque en données pour certaines classes. Ainsi, 15 images aériennes ont été partiellement annotées avec uniquement la classe Verger, et 10 images ont été partiellement annotées avec uniquement la classe Vigne. Les annotations partielles ont été réalisées car les classes Vergers et Vignes était sous-représentées dans les 56 premières images. Par ailleurs, compte tenu de la résolution moyenne des images (0.5 mètres) et la taille des zones étudiées (rayon 1.5 km), les annotations denses correspondent à des zones composées de très nombreux pixels ($\approx 6000 \times 6000$ pixels, voir Figure 3.1).

Extraction d'imagettes pour la classification

Nous nous sommes ensuite inspirés des approches basées sur l'utilisation d'imagettes (*i.e.*, sous-image) proposées par Gonzalo *et al.* [GGLM16] et Porebski et al. [PVMH14] pour la classification. A partir des images aériennes annotées, nous avons extrait des imagettes carrées de trois tailles arbitraires (*i.e.*, 25 pixels × 25 pixels ; 50 pixels × 50 pixels ; 100 pixels × 100 pixels). Aucune imagette de taille supérieure à 100 pixels × 100 pixels n'a été extraite afin de conserver une quantité suffisante de données pour l'utilisation d'algorithmes d'apprentissage profond. Comme toutes les imagettes ont été extraites de la même base d'images initiales, les imagettes de tailles différentes représentent les mêmes zones géographiques avec un contexte spatial plus ou moins étendu : plus l'imagette est de grande taille, plus elle intègre des informations liées à son contexte spatial. En pratique, nous avons uniquement considéré des imagettes sans recouvrement pour une taille d'imagette donnée (*i.e.*, le pas entre deux imagettes est égal à la taille de l'imagette). Seules les imagettes correspondant à une seule et unique classe ont été retenues (*i.e.*, tous les pixels de l'imagette ont la même étiquette sur l'annotation manuelle) afin de limiter l'introduction d'informations contradictoires dans l'évaluation des chaines de traitements. Les imagettes ainsi obtenues ont été sauvegardées selon leur taille et leur classe (voir figure 3.2). Nous rappelons que ce jeu de données complet a été nommé HistAerial (voir tableau 3.1).

On constate que le nombre d'imagettes par taille et par classe n'est pas équilibré dans HistAerial. Ce fait est limitant pour l'évaluation d'algorithmes différents dans un cadre de classification. Les méthodes permettant de réduire l'effet d'un déséquilibre de classe, lorsqu'elles existent, ne sont en effet pas les mêmes pour tous les algorithmes. Afin de palier ce problème, nous avons choisi de créer deux sous-ensembles du jeu de données HistAerial par échantillonnage aléatoire (voir tableau 3.2 et tableau 3.3). Nous avons par la suite considéré ces deux jeux de données indépendamment l'un de l'autre.

Le premier sous-ensemble de données (voir tableau 3.2) contient le même nombre d'images pour chaque taille et pour chaque classe, de telle sorte que les disproportions en termes de quantité de données en fonction de la taille n'ont pas d'effet direct sur la comparaison des filtres et des classifieurs (*i.e.*, on travaille à quantité de données fixée). Ce sous-ensemble est dit équilibré en taille (*size-balanced*). On remarque cependant que celui-ci ne permet de tenir compte de la variabilité des représentations qui sont présentes entre les imagettes d'une même taille. A titre d'exemple, échantillonner aléatoirement 6000 imagettes à partir 43 000 imagettes (100 pixels × 100 pixels, classe urbain) devrait induire une variabilité plus faible qu'un échantillonnage de 6000 imagettes à partir de 891 000 imagettes (25 pixels × 25 pixels, classe urbain). Le second sous-ensemble

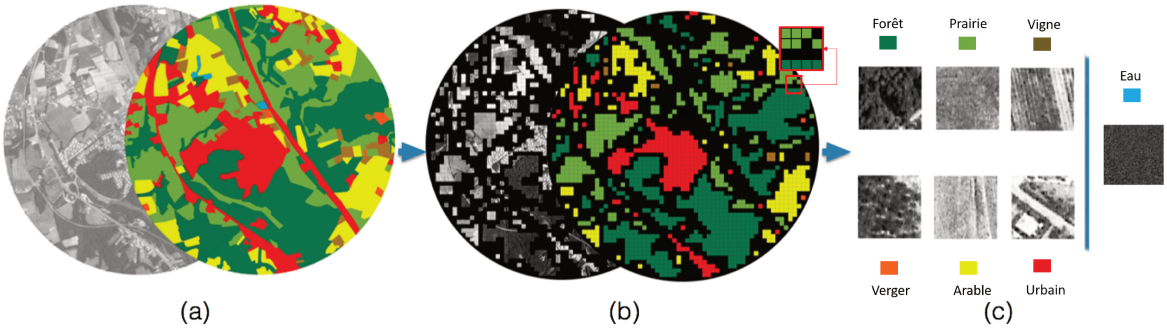


FIGURE 3.2 – Processus d'extraction d'imagettes pour la création de HistAerial. Les imagettes sur ce schéma sont de taille 100 pixels × 100 pixels. Seules les imagettes représentant une seule et unique classe sont considérées. Il n'y a pas de recouvrement entre les imagettes de même taille. (a) Image et occupation du sol manuelle / vérité terrain. (b) Visualisation des imagettes extraites, correspondant à une seule classe. Les carrés noirs représentent ici les imagettes exclues du jeu de données. (c) Exemples d'imagettes extraites. L'imagette de la classe Eau provient d'une image aérienne différente de celle présentée sur cette figure.

(voir tableau 3.3) a été créé afin de tenir compte de cette observation. Pour chaque taille, des imagettes ont été échantillonnées en se basant sur le nombre le plus faible d'imagettes par classe. Ce processus permet d'avoir, de façon approximative, la même proportion d'imagettes pour chaque taille disponible tout en ayant le même nombre d'imagettes par classe (*class-balanced*). Ces deux sous-ensembles de données ont été échantillonnés une fois pour toute, de telle sorte que les expériences détaillées dans la Section 3.4 ont toutes été réalisées sur les même données. Des exemples d'imagettes sont présentés sur la figure 3.3 pour chacune des classes et chacune des tailles considérées dans HistAerial.

TABLEAU 3.1 – Le jeu de données HistAerial complet.

	Nombre d'imagettes par taille (en pixels)		
Classe	25 × 25	50 × 50	100 × 100
Verger	319 804	76 866	17 888
Arable	631 015	145 097	30 754
Prairie	348 349	71 334	11 984
Vigne	174 288	40 528	8 889
Urbain	891 500	204 746	43 254
Forêt	443 760	95 945	18 554
Eau	121 294	28 173	6 207
Total	2 930 010	662 689	137 530

TABLEAU 3.2 – Le sous ensemble équilibré en taille du jeu de données HistAerial.

	Nombre d'imagettes par taille (en pixels)		
Classe	25 × 25	50 × 50	100 × 100
Verger	6 000	6 000	6 000
Arable	6 000	6 000	6 000
Prairie	6 000	6 000	6 000
Vigne	6 000	6 000	6 000
Urbain	6 000	6 000	6 000
Forêt	6 000	6 000	6 000
Eau	6 000	6 000	6 000
Total	42 000	42 000	42 000

TABLEAU 3.3 – Le sous ensemble équilibré en classe du jeu de données HistAerial.

	Nombre d'imagettes par taille (en pixels)		
Classe	25 × 25	50 × 50	100 × 100
Verger	120 000	28 000	6 000
Arable	120 000	28 000	6 000
Prairie	120 000	28 000	6 000
Vigne	120 000	28 000	6 000
Urbain	120 000	28 000	6 000
Forêt	120 000	28 000	6 000
Eau	120 000	28 000	6 000
Total	840 000	196 000	42 000

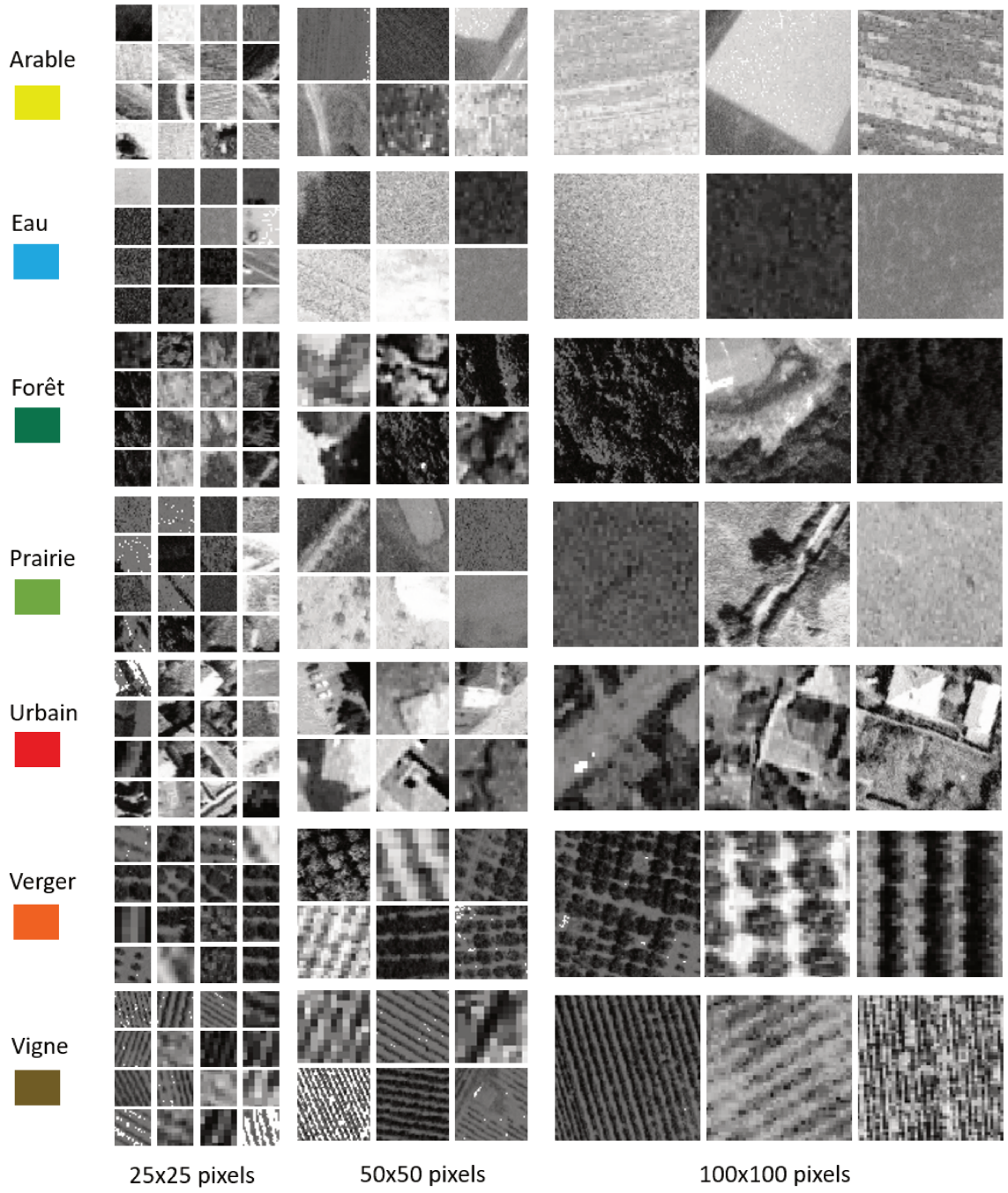


FIGURE 3.3 – Exemples d'imagettes de différentes tailles dans HistAerial. Ces imagettes mettent en avant les variabilités inter- et intra-classe présentes dans le jeu de données HistAerial.

Avantages et inconvénients des imagettes dans HistAerial

- (+) Toutes les imagettes sont carrées et de même taille. Toute opération de redimensionnement linéaire sur le jeu de données devrait conserver l'aspect relatif des images entre elles. Ce point est particulièrement intéressant pour l'analyse des textures à l'aide des réseaux de neurones à convolutions ayant des couches entièrement connectées : celles-ci possèdent un nombre de neurones fixe, ce qui nécessite des images de taille fixe à l'entrée du réseau. Les imagettes sont adaptées à la comparaison d'algorithmes d'extraction de caractéristiques et de classification (avec ou sans réseaux de neurones à convolutions).
- (+) Les imagettes dans HistAerial ont été obtenues sans recouvrement. Elles peuvent être

considérées comme étant des instances spatialement indépendantes, de façon similaire au jeu de données d'écorces d'arbres proposé par Porebski *et al.* [PVMH14]. Elles sont de fait aptes à capturer les variabilités inter-classes et intra-classes sans incorporer de corrélation explicite entre deux imagettes (voir figure 3.3). Il est cependant à noter que des imagettes issues d'une même parcelles auront plus de chances de se ressembler que des imagettes de parcelles différentes (*i.e.*, même capteur, même résolution, même structure de la parcelle).

- (+) Les imagettes permettent de réaliser une étude à plusieurs échelles. Seule la taille des imagettes nécessite d'être modifiée lors de leur extraction afin d'acquérir des données de tailles différentes. Il est également possible de relaxer les contraintes liées à la présence de plusieurs classes au sein d'une même imagette.
- (-) Le nombre d'imagettes par classe dépend d'image aériennes annotées de tailles fixes. Elles ne permettent pas de représenter chaque classe disponible de façon équilibrée. Les sous-ensembles de HistAerial visent à résoudre ce problème pour comparer différents algorithmes.
- (-) Par choix, chaque imagette ne représente qu'une seule et unique classe, ce qui empêche toute acquisition d'imagettes à la frontière entre deux parcelles / éléments sémantiques. Ainsi, plus les imagettes sont grandes, moins la quantité d'imagettes extraite est importante (voir tableau 3.1). L'utilisation d'imagettes avec recouvrement permettrait de résoudre ce problème. Cependant, à l'image de Porebski *et al.* [PVMH14], nous avons préféré éviter toute redondance spatiale entre les imagettes d'une même taille et provenant d'une même image.
- (-) La forme carrée des imagettes a été choisie de façon arbitraire afin de comparer différents algorithmes. Elle ne tient pas compte de la nature hiérarchique des classes d'occupation du sol qui possèdent des caractéristiques à plusieurs échelles sémantiques (*i.e.*, notion d'objets, tels que des parcelles / des superpixels). La représentation de chacune des classes dans HistAerial est, de fait, moins précise que si nous avions accès à la géométrie des parcelles auxquels ils appartiennent. Ce problème est abordé dans un cadre de post-traitement dans le chapitre 5 de ce manuscrit.

3.3 Algorithmes évalués sur HistAerial

Cette section présente les algorithmes évalués sur HistAerial. Le but est ici de trouver des approches performantes pour la classification des images aériennes historiques, tout en ayant le volonté de limiter les temps d'exécution et la taille des vecteurs de caractéristiques extraits. Cette volonté est guidée par le besoin de pouvoir appliquer ces algorithmes en un temps raisonnable sur des machines non dédiées aux calculs scientifiques, telles que les ordinateurs utilisés par les praticiens / géomaticiens.

3.3.1 Algorithmes d'extraction de caractéristiques de la littérature

Les algorithmes d'extraction de caractéristiques "artisanaux" (*handcrafted*) qui ont été étudiés dans nos travaux ont été principalement introduits pour la classification de textures. L'utilisation de filtres de textures a déjà été appréciée dans des travaux antérieurs sur des images aériennes et satellites (voir chapitre 2). Nous rappelons en effet que les images aériennes représentent des zones à grande échelle constituées d'objets spatialement proches observés à partir d'un point d'observation élevé et généralement perpendiculaire au sol. De ce point de vue, la surface de la terre est représentée avec des motifs structurels spécifiques et presque répétitifs, qui correspondent implicitement à la définition des textures inhomogènes en vision par ordinateur. Sur la base de ces observations, nous avons comparé plusieurs filtres de textures artisanaux de la littérature basés sur les motifs binaires locaux (LBP) [OPM00] sur le jeu de données HistAerial. Ils sont présentés ci-dessous pour des images en niveaux de gris. D'autres filtres plus classiques tels que la matrice de cooccurrence de niveau de gris (GLCM) [HSD73] et les filtres de Gabor [Mar11] n'ont

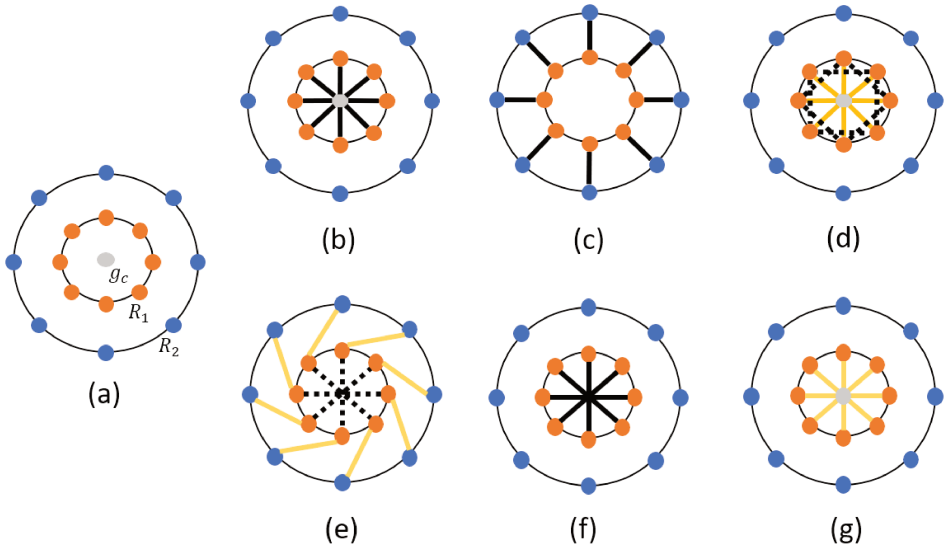


FIGURE 3.4 – Exemples de filtres et de voisinages utilisés avec les filtres de type LBP. (a) voisinage générique avec deux rayons, (b) différence centre-voisins (LBP, CLBP, LTP), (c) différence radiale (ELBP_RD, MRELBP_RD), (d) TPLBP, (e) FPLBP, (f) CSLBP, (g) XCSLBP. Les traits noirs représentent les différences signées. Les traits colorés représentent l'utilisation de métriques intermédiaires. Les traits noirs en points tillés indiquent que la différence se fait entre les éléments positionnés à chaque extrémité du trait à partir de métriques intermédiaires.

pas été inclus dans nos travaux. Des études antérieures ont déjà évalué l'efficacité de ces filtres sur des images de télédétection [AFA+16] [HLZ14], ainsi que sur des jeux de données de textures plus classiques [FÁB13], mettant en avant la supériorité des filtres de type LBP pour la classification. La figure 3.4 met en avant différents types de filtres LBP, basés sur différents voisinages.

***Local Binary Pattern* (LBP) [OPM00]**

Le filtre LBP de base [OPM00] a été introduit dans le chapitre précédent. Nous rappelons ici l'équation principale (3.1) de ce filtre sans réintroduire la notion de *mapping* (voir chapitre 2). Nous rappelons également que ce type de filtre est défini sur un voisinage (P,R), avec P le nombre de pixels voisins g_p au pixel central du voisinage g_c (voir figures 3.4 (a) et (b)). Les P pixels voisins sont situés sur le cercle de rayon R. Dans l'ensemble de ce chapitre, le voisinage est considéré comme étant continu (*i.e.*, les valeurs des pixels g_p sont obtenues par interpolation bilinéaire).

$$\mathrm{LBP}_{\mathrm{P,R}} = \sum_{p=0}^{\mathrm{P}-1} s(g_p - g_c)2^p, s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \tag{3.1}$$

***Variance Local Binary Pattern* (VAR-LBP) [OPM01]**

Le filtre VAR-LBP [OPM01] consiste en un filtre LBP combiné avec les informations de contraste locales représentées par la variance (VAR) du voisinage circulaire (P,R) (voir équation (3.2)). Étant donné que le filtre LBP est invariant en niveaux de gris, il n'intègre pas les informations de contraste. Le filtre VAR et le filtre LBP sont considérés comme étant complémentaires.

$$\mathrm{VAR}_{\mathrm{P,R}} = \frac{1}{\mathrm{P}} \sum_{p=0}^{\mathrm{P}-1} (g_p - \mu_l)^2 \tag{3.2}$$

Dans l'équation (3.2), $s(x) = 1$ si $x > 0$, $s(x) = 0$ sinon, et μ_l est la moyenne des pixels du voisinage définie par l'équation (3.3).

$$\mu_l = \frac{1}{\mathrm{P}} \sum_{p=0}^{\mathrm{P}-1} g_p \tag{3.3}$$

Une fois la variance locale calculée pour chaque pixel du voisinage circulaire, un histogramme de 128 *bins* représentant la variance globale de l'image filtrée est calculé. Cet histogramme de variance est concaténé à l'histogramme LBP [OPM00]. Le filtre VAR-LBP génère ainsi un histogramme de 2^{P} + 128 *bins* en supposant que aucun *mapping* n'est appliqué.

***Center Symmetric Local Binary Pattern* (CSLBP) [HPS06]**

le filtre CSLBP [HPS06] tient compte uniquement de l'information portée par les pixels du voisinage g_p. Il utilise la symétrie du voisinage pour calculer le signe de la différence entre les pixels opposés par g_c (symétrie centrale, voir (voir figure 3.4 (f)). Cette opération est représentée par l'équation (3.4). La valeur du pixel central n'est pas utilisée ici. Le filtre CSLBP produit un code binaire de $\frac{\mathrm{P}}{2}$ bits par voisinage, résultant en un histogramme $2^{\frac{\mathrm{P}}{2}}$ *bins*.

$$\mathrm{CSLBP}_{\mathrm{P,R}} = \sum_{p=0}^{\frac{\mathrm{P}}{2}-1} s(g_p - g_{p+\frac{\mathrm{P}}{2}})2^p \tag{3.4}$$

avec $s(x)$ défini par l'équation (3.5).

$$s(x) = \begin{cases} 1, x \geq \tau \\ 0, sinon \end{cases} \tag{3.5}$$

où τ est une valeur faible (*e.g.* $\tau = 0.01$).

***Extended Center Symmetric Local Binary Pattern* (XCSLBP) [SBF15]**

Le filtre XCSLBP [SBF15] a été introduit comme une amélioration du filtre CSLBP dans un contexte de soustraction d'arrière-plan. Il a été conçu pour être plus robuste au bruit que le CSLBP tout en conservant un pouvoir discriminant équivalent. Il utilise des métriques intermédiaires $g_1(p,c)$ et $g_2(p,c)$ pour calculer le code binaire à partir des pixels du voisinage opposés par symétrie centrale en tenant comme de la valeur du pixel central g_c (voir équations (3.6) et (3.7), et figure 3.4 (g)). On remarquera ici que ces métriques intermédiaires intègrent la valeur de g_c, qui est absente du filtre CSLBP.

$$\mathrm{XCSLBP}_{\mathrm{P,R}} = \sum_{p=0}^{\frac{\mathrm{P}}{2}-1} s(g_1(p,c) + g_2(p,c))2^p, \tag{3.6}$$

avec $s(x)$ défini par l'équation (3.5) et

$$\begin{cases} g_1(p,c) = g_p - g_{p+\frac{\mathrm{P}}{2}} + g_c \\ g_2(p,c) = (g_p - g_c) \times (g_{p+\frac{\mathrm{P}}{2}} - g_c) \end{cases} \tag{3.7}$$

Tout comme le CSLBP, le XCSLBP résulte en un histogramme de $2^{\frac{\mathrm{P}}{2}}$ *bins*.

***Three Patch Local Binary Pattern* (TPLBP) [WHT08]**

Pour définir le TPLBP [WHT08], on considère un patch C comme étant représenté par une fenêtre de w pixels × w pixels centrée sur un pixel du voisinage (P,R). Le code binaire du filtre TPLBP est obtenu en calculant la différence entre deux distances euclidiennes, elles même calculées entre le patch central C_c et deux de ses patchs voisins C_p et $\mathrm{C}_{p+\alpha}$ (voir équation (3.8)). Ces patchs sont présents sur le même rayon R et radialement espacés d'un angle de valeur α. Dans [WHT08], α est égal à 2, résultant en un angle de 90 degrés entre C_p et $\mathrm{C}_{p+\alpha}$ lorsque le nombre de voisins P est égal à 8. Le filtre TPLBP appliqué sur une image en niveaux de gris génère un histogramme de 2^{P} *bins* si aucun *mapping* n'est utilisé. Dans nos travaux, nos avons utilisé utilisé $w = 1$ (*i.e.*, un

patch est un pixel) et $\alpha = 2$ afin de limiter le nombre de paramètres de la méthode (illustration sur la figure 3.4 (d)).

$$\mathrm{TPLBP}_{\mathrm{P,R},w,\alpha} = \sum_{p=0}^{\mathrm{P}-1} s(d(\mathrm{C}_p, \mathrm{C}_c) - d(\mathrm{C}_{p+\alpha}, \mathrm{C}_c))2^p \tag{3.8}$$

avec $s(.)$ définie par l'équation (3.5) et $d(.)$ la distance euclidienne.

***Four Patch Local Binary Pattern* (FPLBP) [WHT08]**

Le filtre FPLBP [WHT08] calcule la différence entre deux distances euclidiennes obtenues de manière symétrique par rapport au pixel central en comparant deux patchs espacés radialement avec un angle de α degrés et présents sur deux rayons différents R_1 et R_2 (voir équation (3.9)). La valeur de α est généralement choisie comme étant égale à $\frac{180}{\mathrm{P}}$. La différence entre les distances euclidiennes est comparée à zéro pour produire un code binaire de $\frac{\mathrm{P}}{2}$ bits, résultant en un histogramme de $2^{\frac{\mathrm{P}}{2}}$ *bins*. Il est illustré sur la figure 3.4 (e).

$$\mathrm{FPLBP}_{\mathrm{P,R_1,R_2},w,\alpha} = \sum_{p=0}^{\frac{\mathrm{P}}{2}-1} s(d(\mathrm{C}_{\mathrm{R}_1,p}, \mathrm{C}_{\mathrm{R}_2,p+\alpha}) - d(\mathrm{C}_{\mathrm{R}_1,p+\frac{\mathrm{P}}{2}}, \mathrm{C}_{\mathrm{R}_2,p+\frac{\mathrm{P}}{2}+\alpha}))2^p \tag{3.9}$$

***Completed Local Binary Pattern* (CLBP) [GZZ10]**

Le filtre CLBP [GZZ10] combine trois filtres complémentaires de type LBP, tous définis avec le même voisinage (P,R) que le filtre LBP d'origine (voir figure 3.4 (b)). Le premier est le filtre LBP classique, renommé CLBP_S (voir équation (3.1)). Les autres filtres sont CLBP_M et CLBP_C, où M représente la magnitude et C le niveau de gris central correspondant à la valeur de g_c. L'amplitude correspond à la valeur absolue de la différence entre g_c et un pixel voisin g_p. Elle représente une information complémentaire au signe qui est par définition indépendant de l'intensité. Celle-ci est encodée à l'aide d'un code binaire défini par l'équation (3.10), où m_p et τ_m sont respectivement la magnitude de la différence entre g_p et g_c et la moyenne de toutes les magnitudes dans l'image. La fonction $s(.)$ est ici définie comme étant la fonction signe classique.

$$\mathrm{CLBP_M}_{\mathrm{P,R}} = \sum_{p=0}^{\mathrm{P}-1} s(m_p - \tau_m)2^p \tag{3.10}$$

Le code binaire du CLBP_C est quant à lui obtenu en comparant g_c avec le niveau de gris moyen μ de l'image entière [GZZ10] (voir équation (3.11)). La fonction $s(.)$ est là aussi définie comme étant la fonction signe classique.

$$\mathrm{CLBP_C}_{\mathrm{P,R}} = s(g_c - \mu) \tag{3.11}$$

Le filtre CLBP permet d'obtenir un histogramme concaténé de $2^{\mathrm{P}+1} + 2$ *bins* lorsque aucun *mapping* n'est utilisé.

***Local Ternary Patterns* (LTP) [TT10]**

Le filtre LTP [TT10] est une extension du filtre LBP (même voisnage, voir figure 3.4 (b)). Il génère un code ternaire au lieu d'un code binaire. Les valeurs ternaires sont obtenues en appliquant deux seuils opposés $(\tau, -\tau)$ et choisis arbitrairements. Afin de simplifier sa représentation et de le rendre moins coûteux en calculs, le code LTP peut être séparé en deux codes LBP : un pour la partie positive et un pour la partie négative [TT10]. La partie positive est obtenue en mettant toutes les valeurs positives à 1 et les autres à 0 tandis que la partie négative est obtenue en mettant toutes les valeurs négatives à 1 et les autres à 0. En fin de compte, le filtre LTP génère soit un histogramme de 3^{P} bins, soit deux histogrammes de 2^{P} *bins* pouvant être concaténés pour former un histogramme unique de $2^{\mathrm{P}+1}$ *bins*. La génération d'un code ternaire est montrée par une fonction signe telle que définie par l'équation (3.12).

$$s(x) = \begin{cases} +1, x \geq +\tau \\ 0, |x| < +\tau \\ -1, x \leq -\tau \end{cases} \tag{3.12}$$

***Robust Local Ternary Patterns* (RLTP) [WSFW15]**

Le filtre RLTP [WSFW15] est défini comme étant une version robuste au bruit du filtre LTP. Pour chaque voisinage (P,R) incluant le pixel central g_c, la valeur moyenne du voisinage μ_c est calculée (voir équation (3.13)).

$$\mu_c = \frac{1}{P+1}(g_c + \sum_{p=0}^{P-1} g_p) \tag{3.13}$$

Les seuils positifs et négatifs sont alors définis comme des fractions de μ_c (voir équations (3.14) et (3.15)). La constante α dans l'équation (3.14) est égale à 1 par défaut. Cette valeur n'a pas été modifiée dans nos travaux. Il a cependant été montré que régler la valeur α par recherche exhaustive pouvait permettre d'obtenir des caractéristiques plus robustes aux variations d'illuminations avec le filtre RLTP [WSFW15].

$$\tau_c = \alpha \times \mu_c \tag{3.14}$$

$$s(x) = \begin{cases} +1, x \geq +\tau_c \\ 0, |x| < +\tau_c \\ -1, x \leq -\tau_c \end{cases} \tag{3.15}$$

Le filtre RLTP génère des histogrammes de même dimension que le filtre LTP.

***Soft Concave-Convex Orthogonal Combination of Robust Local Ternary Patterns* (SCCOCRLTP) [WSFW15]**

Le filtre SCCOCRLTP [WSFW15] est basé sur le filtre RLTP. Il propose d'augmenter le nombre de motifs discriminants tout en réduisant leur empreinte mémoire grâce aux concepts de combinaison orthogonale [ZBC13] et de discrimination concave-convexe [SFYW14]. L'idée derrière la combinaison orthogonale est qu'une concaténation de K histogrammes obtenus à partir de K filtres de type LBP orthogonaux entre eux sur un voisinage (P,R) devrait permettre de représenter la même information qu'un histogramme unique obtenu à partir d'un filtre LBP complet, tout en étant plus compacte (*i.e.*, $K \times 2^{P/K}$ *bins* vs 2^P bins). Une illustration de ce principe est présenté sur la figure 3.5. La discrimination concave-convexe d'un voisinage LBP est quant à elle basée sur une comparaison entre la moyenne locale du voisinage LBP avec la moyenne globale de l'image entière (voir (3.16)). Dans l'équation (3.16), μ_c est la moyenne locale définie par l'équation (3.13), μ est la moyenne globale et β est une petite valeur égale à 0 par défaut.

$$g_c \quad est \quad \begin{cases} concave, & si \quad \mu_c < (1-\beta)\mu \\ convexe, & si \quad \mu_c \geq (1+\beta)\mu \end{cases} \tag{3.16}$$

***Extended Local Binary Pattern* (ELBP) [LZL+12]**

Le filtre ELBP est une combinaison de trois filtres de type LBP nommés respectivement ELBP_CI, ELBP_NI et ELBP_RD. Le filtre ELBP_CI représente l'intensité du pixel central g_c. Cette intensité est comparée à la valeur moyenne de l'image entière μ pour obtenir un code binaire a 1 bit (voir équation (3.17)), soit un histogramme de 2 *bins*. Il correspond au filtre CLBP_C utilisé par le CLBP.

$$\text{ELBP_CI}_{P,R} = s(g_c - \mu) \tag{3.17}$$

Le filtre ELBP_NI représente les intensités des P pixels g_p, voisins de g_c, d'une manière robuste au bruit additif gaussien [LZL+12]. De façon similaire à l'approche employée par le RLTP, pour chaque voisinage (P,R), la moyenne locale $\mu_{l,R}$ de l'intensité des P pixels voisins est calculée (voir

équation (3.18)). Cette moyenne locale est ensuite comparée à chaque pixel voisin g_p pour générer un code binaire (voir équation (3.19)).

$$\mu_{l,\mathrm{R}} = \frac{1}{\mathrm{P}} \sum_{p=0}^{\mathrm{P}-1} g_{p,\mathrm{R}} \tag{3.18}$$

$$\mathrm{ELBP_NI}_{\mathrm{P,R}} = \sum_{p=0}^{\mathrm{P}-1} s(g_{p,\mathrm{R}} - \mu_{l,\mathrm{R}}) 2^p \tag{3.19}$$

Le filtre ELBP_RD représente la différence radiale entre deux pixels voisins à la même position angulaire p, mais localisés sur deux rayons R_1 et R_2 différents tels que $\mathrm{R}_1 < \mathrm{R}_2$. Le signe de la différence entre g_{p,R_1} et g_{p,R_2} est utilisé pour créer le code binaire. Il est schématisé sur la figure 3.4 (c).

$$\mathrm{ELBP_RD}_{\mathrm{P,R}} = \sum_{p=0}^{\mathrm{P}-1} s(g_{p,\mathrm{R}_2} - g_{p,\mathrm{R}_1}) 2^p \tag{3.20}$$

Étant donné que le filtre ELBP_RD et le filtre ELBP_NI résultent tous deux en un histogramme de 2^{P} *bins* et que le filtre ELBP_CI donne un histogramme de 2 bins, le filtre ELBP se traduit par un histogramme de $2^{\mathrm{P}+1}+2$ *bins* sans utiliser de *mapping*.

Median Robust Extended Local Binary Pattern (MRELBP) [LLF+16]

Le filtre MRELBP [LLF+16] a été présenté comme une mise à jour du filtre ELBP dédié à la classification des textures bruitées. Les auteurs proposent ici d'appliquer un filtre passe-bas $\psi(x)$, centré sur le pixel x, avant le calcul des caractéristiques décrites par le filtre ELBP. Le choix d'un filtre médian a ici été fait à travers une comparaison qualitative avec les filtres gaussiens et moyens [LLF+16]. Le filtre MRELBP fonctionne particulièrement bien sur les jeux de données de textures bruitées. Dans nos travaux, nous avons suivit Liu et al. [LLF+16] en fixant la taille du filtre médian appliqué au pixel central g_c à 3 pixels par 3 pixels. La taille du filtre médian pour les P pixels voisins g_p de différents rayons (R_1, R_2, R_3), avec $\mathrm{R}_3 > \mathrm{R}_2 > \mathrm{R}_1$, a été fixée à (3, 3, 5).

3.3.2 Proposition de nouveaux filtres pour la texture

La plupart des filtres présentés ci-dessus génèrent des vecteurs de caractéristiques de hautes dimensions (*i.e.* 2^{P} bins) pour un seul ensemble de paramètres défini par le voisinage (P,R). Nous rappelons par ailleurs que l'utilisation de *mapping* dans un contexte inadéquat peut diminuer le pouvoir discriminant de ces filtres (voir chapitre 2). Leur utilisation peut donc résulter soit en des résultats moins précis, soit en des étapes d'apprentissage (optimisation d'un classifieur) et de classification plus lentes dû au nombre de caractéristiques à traiter. Cette deuxième situation n'est *a priori* pas souhaitée dans le cadre d'un apprentissage en ligne (*online*) à partir de données fournies par un utilisateur (*e.g.*, les traces dans le logiciel Gouramic, voir Annexe A). Cela est particulièrement vrai sur des ordinateurs avec des capacités de calculs limitées (par exemple, sans GPU) comme ceux utilisés par les praticiens. Afin de trouver un compromis approprié entre le pouvoir discriminant et la taille du vecteur de caractéristiques, tout en proposant des approches complémentaires à l'état de l'art actuel, nous avons proposé deux nouveaux filtres permettant d'obtenir des vecteurs de caractéristiques de faibles dimensions.

Rotated-CorneR Local Binary Pattern (R-CRLBP)

Le filtre R-CRLBP est un nouveau filtre que nous avons introduit dans le cadre de nos travaux. Il a été inspiré par le filtre *Binary Gradient Contours* (BGC) [FÁB13] et par la combinaison orthogonale [ZBC13]. Il considère le signe des différences successives entre les pixels voisins présents sur un même rayon R. La différence successive entre deux pixels voisins consécutifs est définie par la relation suivante : $(g_p - g_{p-1})$. Cette opération peut être opposée à la différence symétrique centrale utilisée dans le filtre CSLBP [HPS06] et à la différence classique centre-voisins utilisée dans

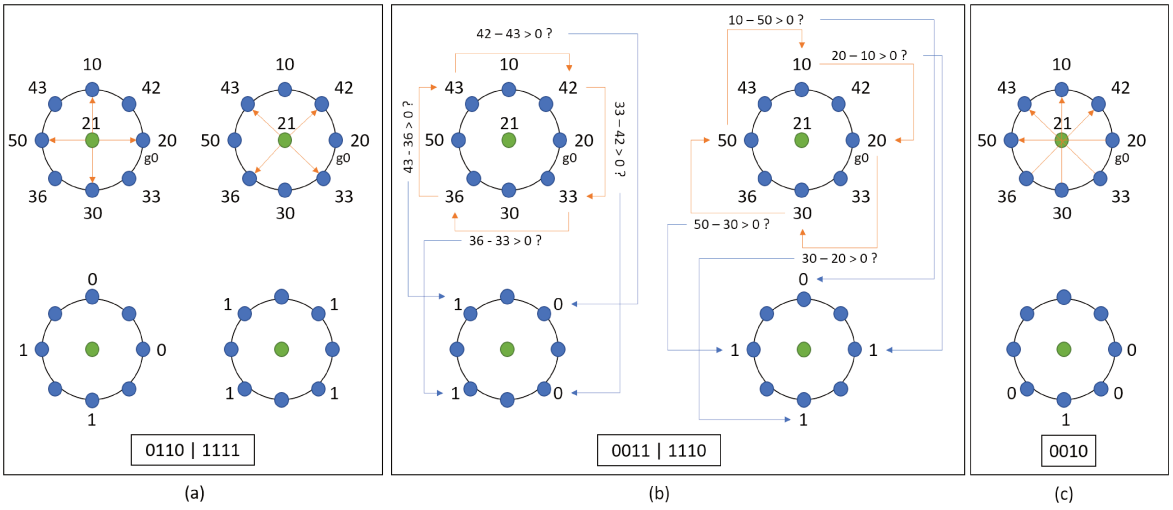


FIGURE 3.5 – Différents types de filtres LBP et leurs codes binaires obtenus sur un même voisinage (P, R). Ces filtres sont complémentaires afin de classifier des motifs qui seraient potentiellement confondus avec un seul code binaire. (a) Le filtre LBP originel utilisant une différence centre-voisins ($g_c - g_p$) et la combinaison orthogonale, (b) le filtre R-CRLBP utilisant une différence successive ($g_p - g_{p-1}$) avec deux rotations, (c) le filtre CSLBP avec une différence entre pixels symétriques par g_c. Les flèches orange indiquent des soustractions. Les flèches bleues indiquent la position du bit correspondant. Les codes binaires sont générés en traversant le voisinage dans le sens des aiguilles d'une montre en partant de g_0.

le filtre LBP [OPM01] : ici, les pixels voisins ne sont ni comparés symétriquement ni comparés au pixel central. D'un point de vue du motif de textures, le filtre CRLBP (notez l'absence du préfixe *Rotated*, "R-") va chercher à représenter les motifs du *gradient circulaire* d'un voisinage (P, R). Pour P = 8 et R fixés, on a va échantillonner les 4 voisins formant des angles de +/− 45 degrés avec les axes horizontaux et verticaux du plan, et présents sur le cercle de même rayon R. Le signe de la différence successive des voisins échantillonnés est alors utilisé pour générer le code binaire local (voir figure 3.5). Le résultat de ce filtre est ensuite stocké dans un histogramme de 2^4 *bins*. Le pixel central n'est pas utilisé dans le filtre CRLBP. Inspiré par la combinaison orthogonale [ZBC13], on remarque ici que des rotations peuvent être appliquées au centre du filtre CRLBP afin de considérer les autres pixels voisins présents sur le cercle de rayon R. Grâce à ces rotations, un total de $\frac{P}{4}$ histogrammes peuvent être obtenus en supposant que le nombre de pixels voisins P est un multiple de 8 (cas usuel avec les filtres de type LBP). Chacune des rotations CRLBP est calculée avec un quaternion unique de voisins qui n'est pas pris en compte dans les autres rotations (voir la figure 3.5). On parle alors de R-CRLBP (ajout du préfixe *Rotated*). Notez que pour P = 8, le filtre R-CRLBP est équivalent à l'un des motifs utilisés dans le filtre BGC [FÁB13]. La concaténation des histogrammes obtenus permet de produire un histogramme unique composé de $\frac{P}{4} \times 2^4$ *bins*. Le R-CRLBP est défini par l'équation (3.21), où nous supposons que le nombre de pixels voisins P est un multiple de 8.

$$\begin{aligned} \mathrm{R-CRLBP}_{\mathrm{P,R}} &= \sum_{p=\alpha, p+=\frac{P}{4}}^{\alpha+\frac{3\times P}{4}} s(g_p - g_{p-(\frac{P}{4})})2^i \\ avec \quad i &= \frac{p-\alpha}{\frac{P}{4}} \quad et \quad \alpha = (0, 1, .., \frac{P}{4} - 1) \end{aligned} \tag{3.21}$$

Light Combination of Local Binary Patterns (LCoLBP)

LCoLBP est une combinaison empirique de filtres de type LBP similaire aux CLBP [GZZ10], SCCORLTP [WSFW15] et ELBP [LZL+12], que nous avons également introduite dans le cadre de nos travaux. Son développement a été motivé par la volonté de fournir une représentation complétée des motifs de textures grâce à un vecteur de caractéristiques discriminant de faible dimension. En pratique, le résultat du filtre LCoLBP consiste en une concaténation de quatre histogrammes obte-

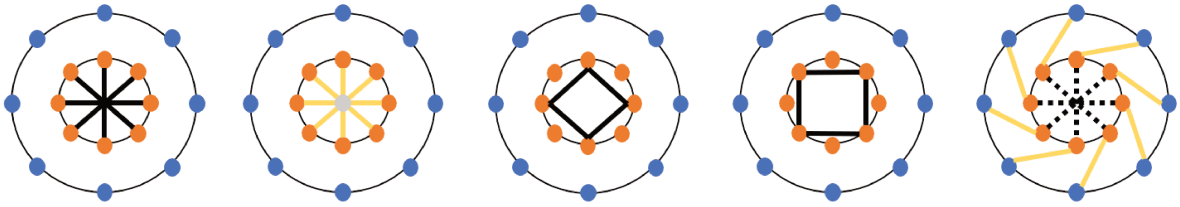


FIGURE 3.6 – Représentation schématique des filtres utilisés dans le LCoLBP. De gauche à droite : CSLBP [HPS06], XCSLBP [SBF15], R-CRLBP$_1$ (rotation numéro 1), R-CRLBP$_2$ (rotation numéro 2), FPLBP [WHT08]. Les points colorés représentent les pixels considérés sur deux rayons différents. Les traits noirs représentent les différences signées. Les traits colorés représentent l'utilisation de métriques intermédiaires. Les traits noirs en points tillés indiquent que la différence se fait entre les éléments positionnés à chaque extrémité du trait à partir de métriques intermédiaires (cas du FPLBP). On visualise ici la complémentarité des motifs recherchés sur un voisinage (P, R).

nus avec des filtres de type LBP, à savoir les FPLBP, XCSLBP, CSLBP et R-CRLBP, présentés ci-dessus. Ces filtres ont la particularité de mettre en évidence l'utilisation des pixels voisins d'un voisinage (P, R). Seul XCSLBP utilise le pixel central. Ces filtres calculent des caractéristiques basées sur différentes topologies d'une manière qui les rend complémentaires (voir figure 3.6). En particulier, pour un voisinage (P, R) donné, les histogrammes CSLBP et XCSLBP représentent respectivement les motifs de gradient internes au voisinage et leur variante robuste au bruit. L'histogramme obtenu avec le FPLBP représente les motifs de gradient externes au voisinage. L'histogramme du R-CRLBP représente quant à lui des motifs circulaires, en périphérie du voisinage. De plus, chacun de ces filtres génère un histogramme de faible dimension, tous contenant un nombre de *bins* équivalent (égal pour P = 8) en considérant les rotations du R-CRLBP comme étant indépendantes. La concaténation de ces histogrammes pour un voisinage (P, R) donné permet d'obtenir histogramme final de $\frac{P}{4} \times 2^4 + 3 \times 2^{P/2}$ *bins*. Dans l'équation (3.22), la fonction *concat(.)* représente une concaténation d'histogramme 1D, tandis que la fonction *histomap(.)* représente le calcul de l'histogramme appliqué sur chacun des éléments d'une liste. L'histogramme obtenu avec le LCoLBP est représenté sur la figure 3.7 avec P = 8 pour trois rayons différents R = {1, 2, 3}.

$$histo(\text{LCOLBP}) = concat(histomap([\text{FPLBP}, \text{R}-\text{CRLBP}, \text{XCSLBP}, \text{CSLBP}])) \tag{3.22}$$

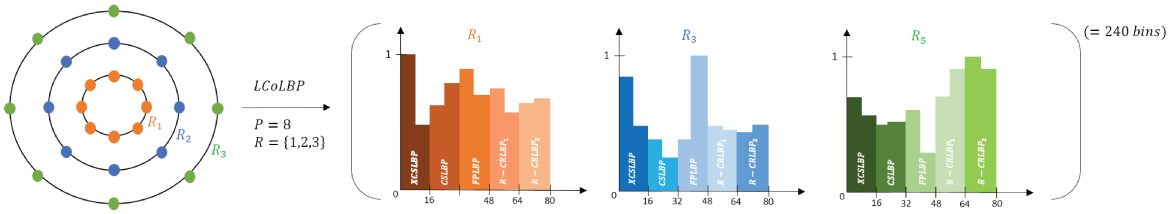


FIGURE 3.7 – Schéma de la génération d'un histogramme de textures avec le LCoLBP appliqué sur un voisinage (P, R) = (8, {1, 2, 3}).

3.3.3 Classifieurs utilisés avec les descripteurs de textures

Les classifieurs utilisés ici ont été présentés dans le chapitre 2. Ils incluent les K plus proches voisins (KNN), les machines à vecteurs de support multi-classes (SVM), les perceptrons multi-couches (MLP) et les forêts aléatoires d'arbres décisionnels (RFOREST).

3.3.4 Réseaux de neurones profonds à convolutions évalués

Dans cette étude comparative menée sur HistAerial, nous avons également intégré des réseaux de neurones profonds à convolutions "bout en bout" (DCNN). Ces méthodes ont déjà été appli-

quées avec succès sur des images satellite [MTCA16]. Les DCNN ont, de manière générale, tendance à surpasser les extracteurs de caractéristiques classiques dans les tâches de classification. Des banques de filtres issus des réseaux de neurones à convolutions ont également pu démonter leur efficacité pour la segmentation (classification au pixel près) d'objets texturés (*i.e.*, un élément visuel dans une scène dont la texture est discriminante) [CMV15]. Néanmoins, des études théoriques [BKD+16] et expérimentales [LFG+17] ont pu montrer que les DCNN peuvent ne pas être aussi performants qu'attendu sur des images de textures (*i.e.*, gains faibles par rapport aux méthodes plus classiques). Ils seraient en revanche aptes à générer des caractéristiques complémentaires aux descripteurs de textures artisanaux [QZS+16]. Il n'y a cependant pas eu, à notre connaissance, d'études antérieure sur l'efficacité des DCNN pour la classification des images aériennes historiques. Nous avons de fait choisit d'évaluer les performances de méthodes existantes sur HistAerial. Les DCNN présentés dans cette section ont été sélectionnés sur la base d'études antérieures notables, avec l'idée que les architectures les moins profondes (*i.e.*, avec moins de couches, et moins de filtres) devraient être capables de reproduire au moins les performances des filtres de textures présentés dans les sections précédentes.

LeNet [LBBH98]

Le modèle LeNet [LBBH98] est un pionnier parmi les réseaux de neurones profonds à convolutions. Il a d'abord été appliqué pour la classification de chiffres manuscrits via le jeu de données MNIST [LBBH98]. Il a permis d'introduire les concepts de base des couches de convolutions, des couches de *pooling* et des couches entièrement connectées présentées dans le chapitre 2. Dans LeNet, chaque couche de convolutions est suivie d'une couche de *pooling* moyen. Lorsqu'il est utilisé comme un extracteur de caractéristiques (*i.e.*, lorsque l'on retire les couches entièrement connectées), LeNet permet de générer un vecteur de 500 caractéristiques.

AlexNet [KSH12]

AlexNet [KSH12] est le premier modèle publié publiquement a avoir obtenu un taux d'erreur en classification (top-5) inférieur à 20% sur le jeu de données ImageNet [KSH12; RDS+15], constitué de plus de 1000 classes différentes. AlexNet étend l'architecture de LeNet en rajoutant plusieurs couches de convolutions afin d'extraire des caractéristiques plus profondes, et remplace le *pooling* moyen par un *pooling* max (plus rapide). Pour nos travaux, nous avons utilisé la version d'AlexNet proposée par la librairie Caffe. Cette implémentation exploite la technique du *dropout* afin d'inhiber aléatoirement des neurones durant l'entraînement et ainsi minimiser le sur-apprentissage. Lorsqu'utilisé en tant qu'extracteur de caractéristiques, AlexNet permet de générer un vecteur de 4096 caractéristiques.

VGG-16 [SZ14]

VGG-16 [SZ14] empile plusieurs couches de convolutions avec des filtres de petite taille 3×3 pixels, par rapport aux filtres d'AlexNet qui diminuent en taille à mesure que l'on ajoute des couches (*e.g.*, 11×11, 5×5, *etc.*). L'architecture de VGG-16 se base sur l'idée qu'en empilant plusieurs petits filtres, on peut obtenir la même précision qu'avec des filtres moins nombreux mais plus larges. Ce point permet de réduire le nombre de paramètres dans le réseau. De plus, VGG-16 applique un *pooling* uniquement après deux ou trois convolutions, tandis que LeNet et AlexNet appliquent cette opération après chaque couche de convolutions. Au final, VGG-16 contient plus de couches qu'AlexNet, ce qui devrait l'aider à apprendre une représentation plus significative des données. Comme AlexNet, l'implémentation utilisée pour VGG-16 utilise des couches de *dropout*. Il génère également un vecteur de 4096 caractéristiques lorsqu'il est utilisé comme extracteur de caractéristiques.

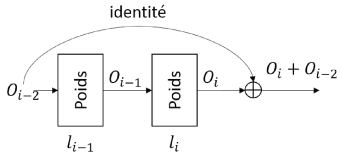


FIGURE 3.8 – Schéma simplifié d'un block résiduel.

ResNet-18 [HZRS16]

ResNet-18 [HZRS16] est un réseau résiduel dont la profondeur est comparable à VGG-16. ResNet-18 est constitué de blocs résiduels. Un bloc résiduel (voir figure 3.8) pourrait être défini comme suit. Étant donné les couches successives l_{i-2}, l_{i-1} et l_i, un bloc résiduel intègre la sortie O_{i-2} (après ReLU) de l_{i-2} et la sortie O_i (avant ReLU) de l_i tel que $O_i := O_i + O_{i-2}$. L'idée derrière cette formulation est qu'un réseau profond devrait toujours avoir la possibilité de fonctionner au moins aussi bien que des réseaux moins profonds en apprenant à ignorer les couches intermédiaires qui pourraient apprendre l'identité. Cette propriété permet généralement d'apprendre des modèles plus profonds en propageant le gradient via les connexions ignorées (connexion entre O_i et les poids de l_{i-2}). Un effet secondaire de ces connexions est que les caractéristiques moins profondes, souvent assimilées à de la texture, ont plus de chances d'être préservées par le réseau. En tant qu'extracteur de caractéristiques, ResNet-18 génère un vecteur de 512 caractéristiques, quantité inférieure aux autres DCNN de tailles comparables (LeNet exclu).

SqueezeNet [IHM+16]

SqueezeNet [IHM+16] est un réseau entièrement convolutif. Il remplace les couches entièrement connectées par un vecteur de probabilité issu de filtres de convolutions. Il a initialement été présenté comme étant une architecture compacte capable d'obtenir des résultats similaires à AlexNet. Il utilise largement les convolutions 1×1 pour réduire le nombre de canaux dans les cartes de caractéristiques intermédiaires, ce qui conduit à un modèle globalement plus rapide. Cependant, selon les auteurs de [IHM+16], le vecteur de caractéristiques de sortie recommandé [1] contient 86528 caractéristiques, quantité bien supérieure aux autres réseaux lorsqu'utilisés en tant qu'extracteurs de caractéristiques.

3.4 Résultats et discussions

Les résultats obtenus pour le problème de la classification top-1 (*i.e.*, pourcentage de prédiction correcte ne tenant compte que de la prédiction avec la probabilité la plus élevée) ont été calculés pour les deux sous-ensembles du jeu de données HistAerial en utilisant les algorithmes présentés dans la section précédente. Pour ces deux sous-ensembles, les données ont été divisées aléatoirement en ensemble d'entraînement, de validation et de test, comme indiqué sur les figures 3.9 et 3.10. Les expériences réalisées et les résultats obtenus sont discutés ci-dessous.

3.4.1 Mise en place des expériences

Paramètres des filtres de textures

Les filtres artisanaux présentés dans les sections précédentes ont été implémentés en C++ avec la version 3.2 de la bibliothèque OpenCV [Bra00]. Dans cette étude, les filtres basés sur le LBP ont été mis en place en considérant un voisinage circulaire continu (voir chapitre 2). Guidés par les considérations sur la complexité des calculs présentées dans [LYF+13], les valeurs pour le rayon R

1. https://github.com/DeepScale/SqueezeNet/issues/13 (accès : (09/2018)

et le nombre de voisins P ont été définies telles que R = (1,2,3) et P = (8) respectivement. Les histogrammes obtenus avec les trois combinaisons (P,R) ont été concaténés pour produire un histogramme 1D. Le *mapping* riu^2 a été appliqué de façon systématique pour les filtres et sous-filtres produisant un histogramme de plus de 2^P *bins* afin de les rendre moins coûteux d'un point de vue algorithmique. Le *mapping* riu^2 n'a pas été appliqué sur l'histogramme final résultant d'une combinaison de filtres LBP s'il était déjà appliqué sur les sous-filtres utilisés dans cette combinaison. L'utilisation du *mapping* riu^2 est *a priori* cohérente pour les données de HistAerial car les images aériennes ont été acquises à des années différentes et dans des conditions incontrôlables impliquant éventuellement des rotations entre les images. Comme inconvénient, il peut cependant en résulter des caractéristiques moins discriminantes. Aucun *mapping* n'a été appliqué avec les autres filtres (ceux générant des histogrammes de moins de 2^P bins). Le filtre LBP d'origine a néanmoins été évalué avec et sans le *mapping* riu^2 afin de vérifier l'efficacité du *mapping* sur les images aériennes historiques. Aucun pré-traitement n'a été appliqué sur les images avant l'extraction des caractéristiques. Les vecteurs de caractéristiques (*i.e.*, histogrammes) ont été normalisés avant l'étape de classification.

Hyperparamètres des classifieurs

Les classifieurs présentés dans la section 3.3.3 ont été entraînés pour chaque filtre sur le sous-ensemble équilibré en taille du jeu de données HistAerial (voir tableau 3.2). Seuls les meilleures pipelines de traitement (*i.e.* filtre puis application d'un classifieur sur le vecteur de caractéristiques résultant) ont été appliqués sur le sous-ensemble équilibré par classe (voir tableau 3.3). Les hyperparamètres des classifieurs ont été automatiquement obtenus à l'aide d'une recherche sur grille des paramètres pour chaque filtre à l'aide du jeu de validation. Les étapes d'entraînement et de test ont toutes deux été effectuées à l'aide de la bibliothèque Scikit-Learn [PVG+11] (version v.0.19.1) en Python. Pour le KNN, K a été choisi dans la plage (1,..,19) avec un pas de 2 entre deux K, et la distance euclidienne a été utilisée. Le SVM a été entraîné en utilisant le noyau RBF (voir chapitre 2). Les autres paramètres du SVM ont été automatiquement sélectionnés pendant la phase d'entraînement à l'aide du jeu de validation, avec les valeurs de C dans la gamme de (1,10,100,1000) et de γ dans la gamme de (0,01, 0.001, 0.0001, 0.00001). Pour le classifieur MLP, le nombre de couches cachées a été automatiquement choisi dans la plage (1, 2, 3). Le nombre de neurones pour la première couche cachée a été sélectionné comme le maximum entre le nombre de classes N_C et 0,75 % de la taille S_v du vecteur de caractéristiques. Pour la deuxième couche, il a été choisi comme le maximum entre N_C et 0,5% de S_v. Pour la troisième couche cachée, il a été choisi comme le maximum entre N_C et 0,25 % de S_v. Le choix d'utiliser un pourcentage décroissant de S_v a été fait pour éviter le coût de calcul élevé qu'une validation croisée exigerait sur les 3 tailles de chaque sous-ensemble de HistAerial, et ce pour chacun des filtres considérés. L'algorithme d'optimisation utilisé avec le MLP était la descente de gradient stochastique (SGD). Son meilleur taux d'apprentissage a été automatiquement sélectionné entre (0.01, 0.001, 0.0001). Les

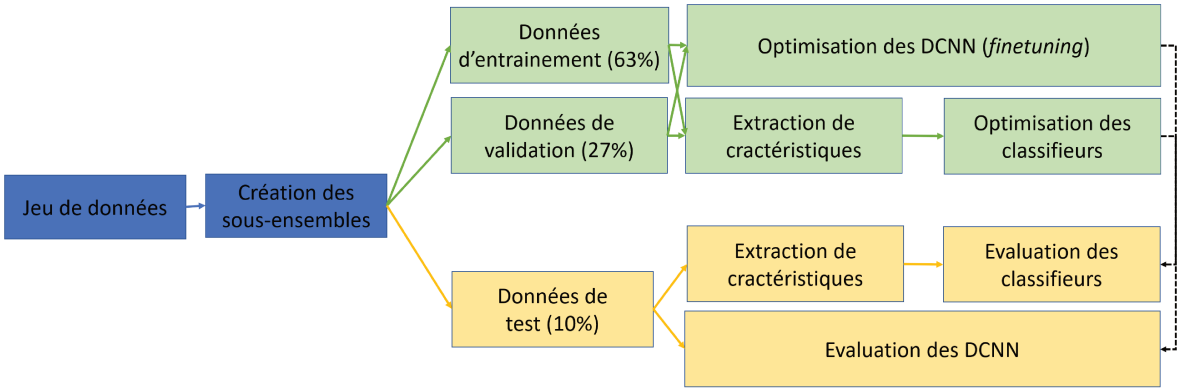


FIGURE 3.9 – Processus générique d'évaluation sur le jeu de données HistAerial.

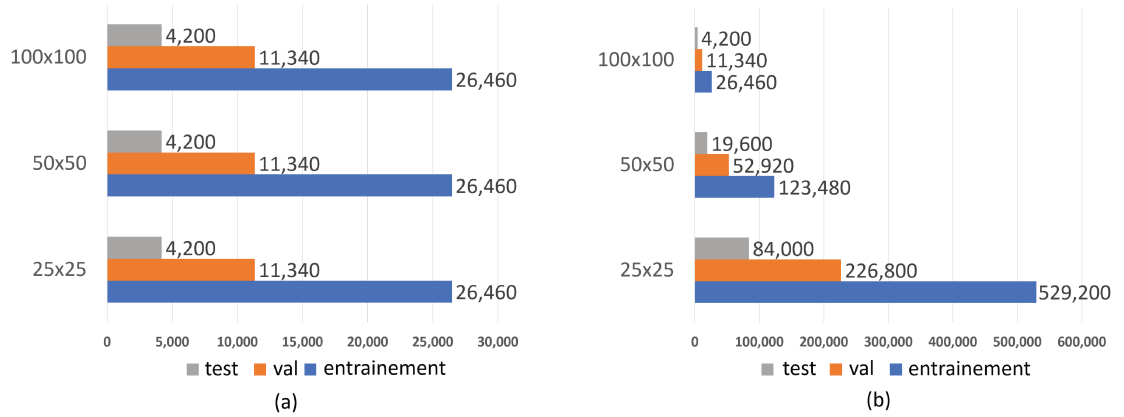


FIGURE 3.10 – Composition des jeux d'entraînement, de validation et de test pour (a) le sous ensemble équilibré en taille et (b) le sous ensemble équilibré en classe du jeu de données HistAerial. Les nombres indiquent le nombre d'imagettes. Chaque jeu contient le même nombre d'imagettes par classe. Ces imagettes ont été échantillonnées aléatoirement à partir de HistAerial.

paramètres de la forêt aléatoire ont été définis à 100 pour le nombre d'arbres, 2 pour le nombre minimal d'échantillons requis au niveau d'un nœud, $\sqrt{Ns}$, avec Ns comme nombre d'échantillons, pour le nombre maximal d'échantillons à considérer pour diviser un nœud, et dans la plage de (5, 10, $\sqrt{Ns}$) pour le nombre minimal d'échantillons requis pour diviser un nœud. Le critère de qualité de partage a été choisi entre l'impureté de Gini et le gain d'informations comme décrit dans le chapitre 2. Toutes les expériences effectuées avec ces classifieurs ont été réalisées sur une machine utilisant un processeur Intel i7 cadencé à 1,7 GHz avec 16 Go de mémoire disponible.

Hyperparamètres des réseaux de neurones profonds

Les DCNN présentés dans la section 3.3.4 ont été pré-entraînés sur le jeu de données MNIST (LeNet) [LBBH98] ou sur ImageNet (AlexNet, VGG-16, ResNet-18, SqueezeNet) [RDS+15]. Ils ont ensuite été raffinés sur une version redimensionnée bilinéairement des sous-ensembles du jeu de données HistAerial (voir tableau 3.2 et tableau 3.3) pendant 40 *epochs* (*i.e.*, le réseau a été optimisé sur le jeu de données d'entraînement complet 40 fois de suite). Aucune amélioration significative n'a été observée après 40 épochs d'entraînement. Les opérations de redimensionnement ont été effectuées pour rendre les tailles des imagettes cohérentes par rapport aux entrées des DCNN. Des images à trois canaux (*i.e.*, équivalent Rouge-Vert-Bleu, RVB) ont été obtenues pour les modèles basés sur ImageNet en empilant les mêmes valeurs de niveaux de gris sur chacun des trois canaux. Comme exposé dans la section 3.2, l'opération de redimensionnement est supposée ne pas modifier la représentation *relative* des imagettes de HistAerial car ce sont des imagettes carrées. Les algorithmes de descente de gradient stochastique (SGD) et de propagation moyenne quadratique de l'erreur (RMSPROP) ont été explorés comme algorithmes d'optimisation pour la partie d'apprentissage. La valeur du taux d'apprentissage initial a également été étudiée dans la plage de (0.01 , 0.001, 0.0001, 0.00001) pour déterminer le meilleur taux d'apprentissage initial, et ce pour chaque modèle et expérience. La décroissance du taux d'apprentissage durant l'entraînement a été fixée à 0.1 et appliquée toutes les 10 *epochs* afin d'éviter le sur-apprentissage. L'entraînement et les tests des DCNN ont été effectués à l'aide de la bibliothèque Caffe [JSD+14] via l'application DIGITS dans sa version 4 [Nvi19] à l'aide de trois GPU NVIDIA GeForce GTX 1080 Ti.

3.4.2 Comparaison globale

Les filtres de textures ont d'abord été comparés sur l'ensemble de données Outex TC_10_-000 [OMP+02] pour évaluer les implémentations utilisées. Ce jeu de données est constitué d'images de textures acquises en laboratoire, avec des rotations entre chaque image. Un classifieur KNN

TABLEAU 3.4 – Meilleurs résultats obtenus sur le sous ensemble équilibré en taille du jeu de données HistAerial pour des imagettes de 25 pixels × 25 pixels. Les valeurs manquantes correspondent à des arrêts prématurés de l'entraînement des DCNN (optimisation divergente).

Filtres de textures et classifieurs									
Filtre	**Paramètre** (P,R)	***mapping***	**Nombre de caractéristiques**	**Classifieur - *Accuracy* (%)**					**Rang**
				KNN	**SVM**	**RFOREST**	**MLP**	**Best**	
LBP	(8,{1,2,3})	riu^2	30	65.5	62.6	67.5	64.3	67.5	9
LBP	(8,{1,2,3})	*none*	768	63.2	66.7	66.1	63.7	66.7	11
VAR-LBP	(8,{1,2,3})	riu^2	414	54.3	67.9	69.6	65.0	69.6	8
CSLBP	(8,{1,2,3})	*none*	48	50.4	49.3	60.8	53.1	60.8	15
XCSLBP	(8,{1,2,3})	*none*	48	62.5	59.1	65.9	59.0	65.9	12
TPLBP	(8,{1,2,3})	riu^2	30	61.6	56.7	62.3	59.6	62.3	14
FPLBP	(8,{1,2,3})	*none*	48	58.4	58.4	59.8	59.9	59.9	17
CLBP	(8,{1,2,3})	riu^2	66	69.4	69.0	**72.1**	68.9	**72.1**	4
LTP	(8,{1,2,3})	riu^2	60	66.9	65.9	69.1	69.2	69.2	7
RLTP	(8,{1,2,3})	riu^2	60	60.5	53.4	63.8	54.1	63.8	13
SCCOCRLTP	(8,{1,2,3})	*none*	384	52.2	54.5	54.5	50.2	54.5	20
ELBP	(8,{1,2,3})	riu^2	66	56.3	45.9	57.2	40.0	57.2	19
MRELBP	(8,{1,2,3})	riu^2	66	49.4	49.2	57.4	49.4	57.4	18
R-CRLBP	(8,{1,2,3})	*none*	96	63.0	65.6	65.8	66.9	66.9	10
LCOLBP	(8,{1,2,3})	*none*	240	68.6	71.0	71.2	**72.9**	**72.9**	3

Réseaux de neurones profonds à convolutions									
Modèle	**Algorithme d'optimisation**	***Epochs***	**Nombre de caractéristiques**	***Accuracy* (%) par taux d'apprentissage**					**Rang**
				0.01	**0.001**	**0.0001**	**0.00001**	**Best**	
LeNet	RMSPROP	40	500	60.0	55.3	60.2	51.7	60.2	16
AlexNet	SGD	40	4096	73.0	**73.6**	68.6	59.1	**73.6**	1
VGG-16	SGD	40	4096	—	70.3	69.9	65.8	70.3	6
SqueezeNet	RMSPROP	40	86528	—	72.6	73.1	65.2	73.1	2
ResNet-18	SGD	40	512	71.6	66.71	42.9	32.8	71.6	5

avec la distance chi2 (adaptée aux histogrammes) et K = 1 a été utilisé. Des résultats comparables à ceux de la littérature ont pu être observés, indiquant que nos implémentations semblaient cohérentes. A titre d'exemple, le filtre MRELBP combiné au *mapping* riu^2, considéré comme une référence sur cet ensemble de données [LFG+17], a permis d'obtenir un taux de bonne classification moyen de 97.6% avec P = 8 et R = (1,2,3). En comparaison, le filtre LCoLBP a permis d'obtenir un score de seulement 51.7% avec les mêmes paramètres. Le score obtenu avec le filtre LCoLBP peut être expliqué par sa définition non invariante à la rotation, tandis que l'ensemble de données Outex TC_10_000 représente des images de textures orientées pour lesquelles l'utilisation du *mapping* riu^2 est particulièrement justifiée.

Ensuite, les méthodes ont été comparées sur le sous-ensemble équilibré en taille du jeu de données HistAerial (voir tableau 3.2). La métrique utilisée est le taux de bonne classification (*accuracy*) en pourcentage. Les meilleurs résultats obtenus pour ces comparaisons sont visibles sur les tableaux 3.4, 3.5 et 3.6. Pour les imagettes de 25 pixels × 25 pixels, le filtre LCoLBP a permis d'obtenir le score le plus élevé entre les filtres de textures, avec un taux de bonne classification de 72.9% en utilisant un MLP. Le filtre CLBP appliqué avec le *mapping* riu^2 combiné a une forêt aléatoire d'arbres décisionnels s'est classé deuxième parmi les filtres de textures, avec un taux de bonne classification de 72.1%. En comparaison, AlexNet a permis d'atteindre le score le plus élevé (73.6%) avec un taux d'apprentissage initial de 0.001, une décroissance du taux d'apprentissage de 0.1 appliquée toutes les 10 époques et un l'algorithme d'optimisation SGD. Il a généré un vecteur caractéristique de 4096 valeurs, à comparer aux 240 *bins* du LCoLBP et aux 66 *bins* du CLBP. Pour la même taille de vecteur de caractéristiques, VGG-16 a permis d'atteindre un taux de bonne classification de seulement 70.3% avec un taux d'apprentissage initial de 0.0001 et l'algorithme d'optimisation SGD. Toutes les combinaisons filtre-classifieur et DCNN ont obtenu des taux de classification plus élevés avec les imagettes de plus grandes tailles. En particulier, le filtre LCoLBP combiné a une forêt aléatoire d'arbres décisionnels s'est classé premier, au-dessus des

TABLEAU 3.5 – Meilleurs résultats obtenus sur le sous ensemble équilibré en taille du jeu de données HistAerial pour des imagettes de 50 pixels × 50 pixels. Les valeurs manquantes correspondent à des arrêts prématurés de l'entraînement des DCNN (optimisation divergente).

Filtres de textures et classifieurs									
Filtre	**Paramètre** (P,R)	***mapping***	**Nombre de caractéristiques**	**Classifieur - *Accuracy* (%)**					**Rang**
				KNN	**SVM**	**RFOREST**	**MLP**	**Best**	
LBP	(8,{1,2,3})	riu^2	30	78.9	72.1	79.0	75.8	79.0	10
LBP	(8,{1,2,3})	*none*	768	80.5	77.9	78.9	78.5	80.5	6
VAR-LBP	(8,{1,2,3})	riu^2	414	67.1	77.6	80.3	78.1	80.3	8
CSLBP	(8,{1,2,3})	*none*	48	63.4	56.2	68.6	63.5	68.6	19
XCSLBP	(8,{1,2,3})	*none*	48	76.3	70.6	78.3	70.9	78.3	12
TPLBP	(8,{1,2,3})	riu^2	30	68.9	65.7	73.6	70.1	73.6	17
FPLBP	(8,{1,2,3})	*none*	48	72.8	70.5	74.0	71.9	74.0	16
CLBP	(8,{1,2,3})	riu^2	66	79.5	77.8	**80.9**	77.1	**80.9**	5
LTP	(8,{1,2,3})	riu^2	60	79.1	76.1	80.4	79.0	80.4	7
RLTP	(8,{1,2,3})	riu^2	60	74.4	64.2	76.6	70.8	76.6	15
SCCOCRLTP	(8,{1,2,3})	*none*	384	76.3	68.3	76.8	66.8	76.8	14
ELBP	(8,{1,2,3})	riu^2	66	69.1	73.7	77.9	75.0	77.9	13
MRELBP	(8,{1,2,3})	riu^2	66	65.7	61.5	71.8	65.4	71.8	18
R-CRLBP	(8,{1,2,3})	*none*	96	76.1	74.7	78.8	77.2	78.8	11
LCOLBP	(8,{1,2,3})	*none*	240	80.4	80.6	**82.9**	81.6	**82.9**	1

Réseaux de neurones profonds à convolutions									
Modèle	**Algorithme d'optimisation**	***Epochs***	**Nombre de caractéristiques**	***Accuracy* (%) par taux d'apprentissage**					**Rang**
				0.01	**0.001**	**0.0001**	**0.00001**	**Best**	
LeNet	RMSPROP	40	500	68.3	61.8	65.8	56.6	68.3	20
AlexNet	SGD	40	4096	82.0	**82.5**	78.4	68.7	**82.5**	2
VGG-16	SGD	40	4096	—	79.0	80.0	77.7	80.0	9
SqueezeNet	RMSPROP	40	86528	—	79.2	82.4	75.5	82.4	4
ResNet-18	SGD	40	512	82.4	74.5	60.7	37.4	82.4	3

DCNN, sur les imagettes de 50 pixels × 50 pixels avec un taux de bonne classification de 82.9%. Le meilleur DCNN a permis d'obtenir un taux de bonne classification de 82.5 % sur ces données. Le filtre LCoLBP s'est classé deuxième sur les imagettes de 100 pixels × 100 pixels avec un score de 89.3%. AlexNet s'est classé premier sur ces données avec un score de 90.4%.

Pour l'ensemble des méthodes, nous nous sommes également intéressés au temps nécessaire pour l'extraction des caractéristiques. Pour cela, nous nous sommes placés dans des conditions équivalentes à celle d'un praticien, et nous avons utilisé un ordinateur avec un processeur cadencé à 1.7 GhZ sans carte graphique. Nous avons utilisé les implémentations optimisées de OpenCV 3.4 pour les DCNN, et nos propres implémentations pour les filtres de textures. De plus, nous avons considéré uniquement les imagettes de 100 × 100 pixels, ces dernières étant les plus longues à traiter pour les filtres de textures. Pour les DCNN utilisés, l'imagette est redimensionnée pour correspondre à la taille attendue à l'entrée de chaque réseau. Le temps d'exécution pour l'extraction de caractéristiques à l'aide de l'un de ces DCNN est donc constant quelque soit la taille de l'imagette considérée dans HistAerial. Les résultats sont reportés dans le tableau 3.6. On y observe que, dans ces conditions, le filtre LCoLBP est approximativement 33 fois plus rapide que AlexNet pour l'extraction de caractéristiques, mais moins rapide que le CLBP. Concernant les étapes de classification, à classifieur constant, un vecteur de caractéristiques plus petit nécessitera moins d'opérations, donnant ici un avantage aux filtres de textures. Ces résultats sont cependant à nuancer : les temps d'exécution varient linéairement avec la taille de l'image pour les filtres artisanaux utilisés ici (ils seraient moins rapides sur des images plus grandes).

Du point de vue des performances de classification globales, les filtres de textures semblent donc permettre d'obtenir des résultats similaires aux DCNN sur le sous-ensemble équilibré en taille de HistAerial, tout en étant moins gourmands en calculs aux étapes d'extraction et de classification des caractéristiques. En particulier, le filtre LCoLBP proposé a atteint des résultats au niveau de l'état de l'art lorsque combiné avec une forêt aléatoire d'arbres décisionnels.

TABLEAU 3.6 – Meilleurs résultats obtenus sur le sous ensemble équilibré en taille du jeu de données HistAerial pour des imagettes de 100 pixels × 100 pixels. Les valeurs manquantes correspondent à des arrêts prématurés de l'entraînement des DCNN (optimisation divergente). Les temps moyens d'extraction de caractéristiques obtenus sur un CPU cadencé à 1.7 Ghz avant classification sont donnés en millisecondes.

Filtres de textures et classifieurs										
Filtre	**Pramètres** (P, R)	***mapping***	**Nombre de Caractéristiques**	**Classifieur - *Accuracy***					**Rang**	**Temps moyen d'extraction (ms)**
				KNN	**SVM**	**RFOREST**	**MLP**	**Best**		
LBP	(8,{1,2,3})	riu^2	30	87.4	81.1	87.3	83.0	87.4	9	1.047
LBP	(8,{1,2,3})	*none*	768	89.1	85.6	86.8	84.2	89.1	5	0.964
VAR-LBP	(8,{1,2,3})	riu^2	414	73.6	80.8	84.5	81.9	84.5	17	1.800
CSLBP	(8,{1,2,3})	*none*	48	75.7	63.2	80.3	72.8	80.3	18	0.624
XCSLBP	(8,{1,2,3})	*none*	48	84.4	78.2	86.0	77.5	86.0	11	0.8124
TPLBP	(8,{1,2,3})	riu^2	30	72.5	71.0	80.1	73.7	80.1	19	1.310
FPLBP	(8,{1,2,3})	*none*	48	84.7	79.7	85.2	81.3	85.2	15	1.023
CLBP	(8,{1,2,3})	riu^2	66	85.8	85.4	88.1	84.9	88.1	6	2.701
LTP	(8,{1,2,3})	riu^2	60	87.6	83.6	88.0	83.5	88.0	7	3.891
RLTP	(8,{1,2,3})	riu^2	60	83.6	69.3	85.3	78.1	85.3	14	2.338
SCCOCRLTP	(8,{1,2,3})	*none*	384	84.6	73.7	85.5	67.0	85.5	13	8.589
ELBP	(8,{1,2,3})	riu^2	66	73.9	81.8	84.8	80.5	84.8	16	3.180
MRELBP	(8,{1,2,3})	riu^2	66	74.8	82.2	85.9	79.6	85.9	12	3.528
R-CRLBP	(8,{1,2,3})	*none*	96	85.6	82.2	86.7	84.6	86.7	10	1.053
LCOLBP	(8,{1,2,3})	*none*	240	88.4	86.8	**89.3**	85.8	**89.3**	2	3.491

Réseaux de neurones profonds à convolutions										
Modèle	**Algorithme d'optimisation**	***Epochs***	**Nombre de caractéristiques**	***Accuracy* (%) par taux d'apprentissage**					**Rang**	**Temps moyen d'extraction (ms)**
				0.01	**0.001**	**0.0001**	**0.00001**	**Best**		
LeNet	RMSPROP	40	500	72.3	69.2	72.1	64.4	72.3	20	0.675
AlexNet	RMSPROP	40	4096	—	86.9	**90.4**	89.7	**90.4**	1	99.610
VGG-16	RMSPROP	40	4096	—	—	87.8	**89.1**	**89.1**	4	1256.500
SqueezeNet	RMSPROP	40	86528	—	86.0	89.2	84.3	89.2	3	60.772
ResNet-18	SGD	40	512	87.8	82.9	72.0	45.7	87.8	8	144.633

Le filtre MRELBP ne semble pas permettre d'obtenir des taux de classification au niveau des autres algorithmes sur l'ensemble de données HistAerial par rapport à l'ensemble de données Outex TC_10_000. Cela pourrait s'expliquer par l'effet de lissage du filtre médian appliqué avec le filtre MRELBP. Ce filtre non-linéaire réduit le nombre possible de motifs que la méthode peut extraire, ce qui peut conduire à des représentations moins discriminantes. Son impact négatif est particulièrement visible sur les petites imagettes, qui sont susceptibles de contenir moins de hautes fréquences que les plus grandes. Cette hypothèse est renforcée par les résultats obtenus avec le filtre LTP et sa version robuste au bruit, le filtre RLTP. Elle n'est pas vérifiée pour le XCSLBP comparé au CSLBP. Cela peut s'expliquer par l'absence de filtre passe-bas explicite dans la formulation du XCSLBP afin d'être plus robuste au bruit.

Le *mapping* riu^2 appliqué sur le filtre LBP n'a quant à lui apporté aucune amélioration ni perte significative sur le sous-ensemble équilibré en taille de HistAerial. Son utilisation semble donc être indiquée sur ce jeu de données afin de réduire le coût de calcul des filtres de type LBP de la littérature.

Par ailleurs, comme indiqué dans le paragraphe précédent, les résultats obtenus par VGG-16, ResNet-18 et SqueezeNet sur le sous-ensemble équilibré en taille de HistAerial sont inférieurs aux résultats obtenus par AlexNet, et ce pour chaque taille. Ces résultats relatifs sont inattendus, les réseaux comparés à AlexNet étant plus profonds et donc plus à même d'extraire des caractéristiques représentatives. Ils nécessiteraient probablement d'autres expériences pour être étudiés de manière approfondie, ce qui n'est pas le but de nos travaux. Cependant, sur la base des travaux des auteurs de [UVL18], nous pouvons faire l'hypothèse que les DCNN se comporteraient naturellement comme des filtres passe-bas résultant en une efficacité réduite sur les données de textures. Par conséquent, un réseau plus profond générerait des cartes de caractéristiques plus lisses que des réseaux moins profonds, ce qui entraînerait une baisse des performances sur les jeux de données de textures.

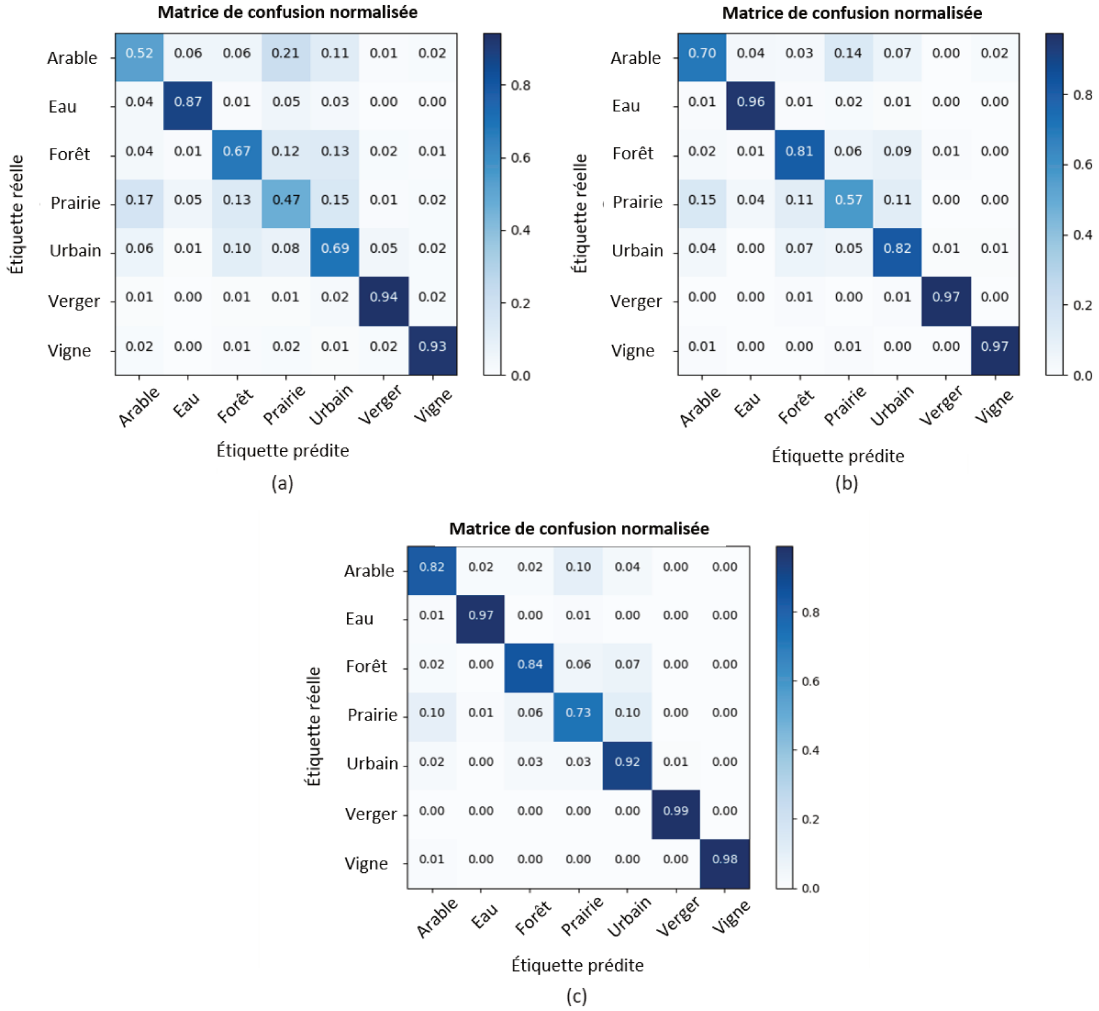


FIGURE 3.11 – Matrices de confusion normalisées pour le filtre LCoLBP sur le jeu de données équilibré en taille. a) 25 pixels × 25 pixels ; b) 50 pixels × 50 pixels ; c) 100 pixels × 100 pixels.

Enfin, on a pu observer sur les figures 3.11 et 3.12 que le filtre LCoLBP et AlexNet ont permis d'obtenir des résultats différents par classe. AlexNet semble avoir optimisé la représentation des classes Arable, Forêt, Eau et Urbain, tandis que le filtre LCoLBP a fourni des taux de bonnes classification plus élevés pour les imagettes de Vignes et de Vergers. Ces résultats donnent un indice quant aux caractéristique apprises par le DCNN par rapport au LCoLBP. Ils mettent en avant le fait que ces méthodes génèrent des représentations qui seraient éventuellement complémentaires aux filtres binaires. Ces observations concordent avec les résultats obtenus par Qi et al. [QZS+16], abordés dans la section 3.3.4. Cependant, il semble que les deux représentations (*i.e.*, DCNN et textures) ont des difficultés à différencier les classe Prairies et Arable. Ce point peut s'expliquer par la similitude (*i.e.*, faible variabilité inter-classe) des textures représentées par ces deux types de sols en l'absence de couleur discriminante sur l'ensemble de données HistAerial. De manière générale, durant les périodes ensoleillées (*e.g.*, au printemps), une prairie est souvent représentée avec des couleurs vertes et un champ cultivé (terre arable) avec des variations de bruns, de vert et de jaune sur les images RVB.

3.4.3 Importance du contexte spatial

Les résultats obtenus sur le sous-ensemble équilibré en taille du jeu de données HistAerial (tableaux 3.4, 3.5, 3.6) fournissent déjà des informations sur les performances de chaque méthode sur différentes tailles d'imagettes. Cependant, comme décrit dans la section 3.2.3, le sous-ensemble

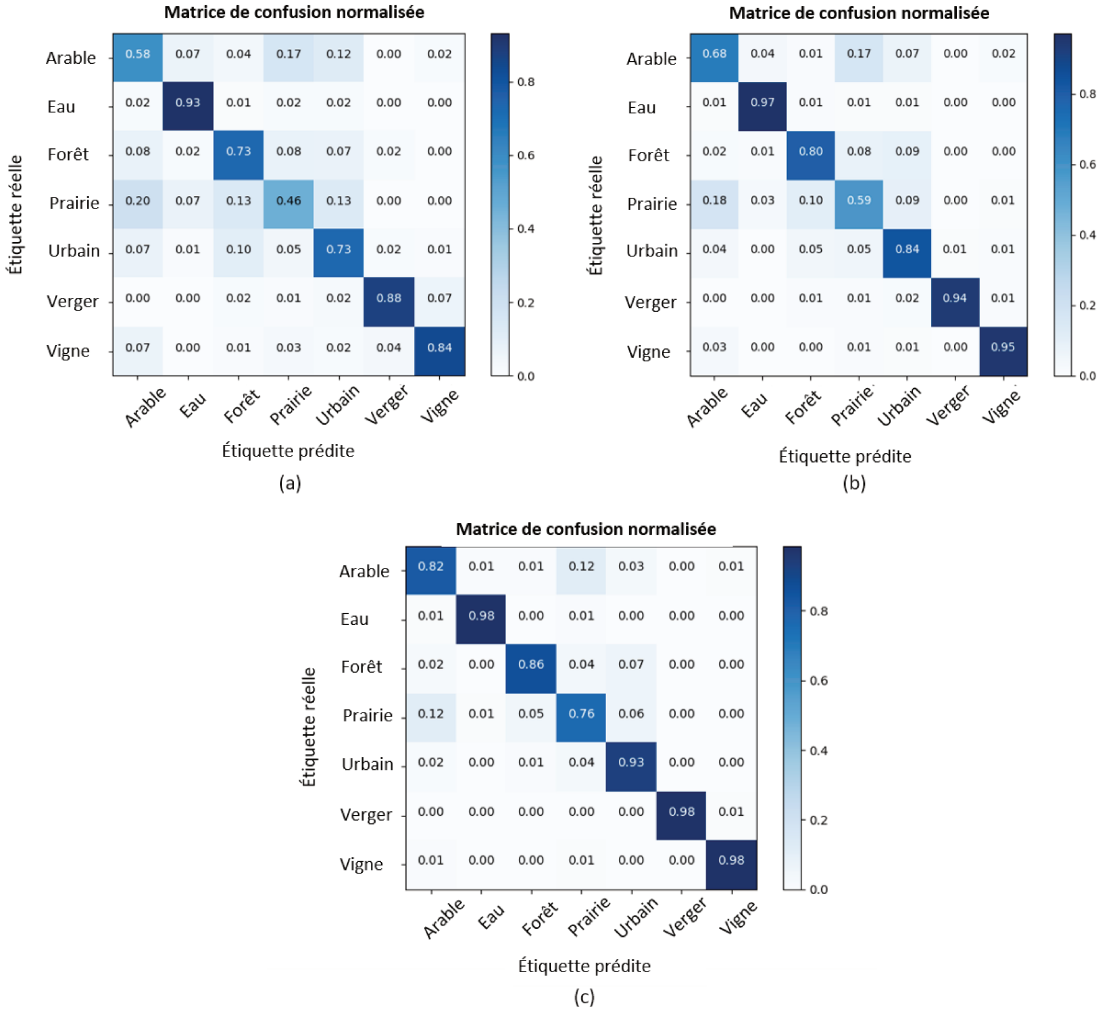


FIGURE 3.12 – Matrices de confusion normalisées pour AlexNet sur le jeu de données équilibré en taille. a) 25 pixels × 25 pixels ; b) 50 pixels × 50 pixels ; c) 100 pixels × 100 pixels.

équilibré en taille ne permet pas à lui seul de donner une indication sur l'importance du contexte spatial pour la classification des images aériennes historiques contenues au sein de HistAerial. Une autre expérience a donc été menée sur le sous-ensemble équilibré en classe. Seules les méthodes qui ont permis d'obtenir les scores les plus élevés sur le sous-ensemble équilibré en taille ont été évaluées sur le sous-ensemble équilibré en classe, à savoir LCoLBP combiné avec une forêt aléatoire d'arbres décisionnels et AlexNet. Leurs résultats sont présentés sur le tableau 3.7.

On observe des taux de classification similaires sur le sous-ensemble à équilibré en classe et le sous-ensemble équilibré taille. Le filtre LCoLBP et AlexNet, respectivement, ont permis d'obtenir des scores de classification de 75.0% et 73.4% sur des imagettes de 25 pixels × 25 pixels (120 000 imagettes par classe), et 84.1% et 85.6% sur les imagettes de 50 pixels × 50 pixels (28 000 imagettes par classe). Ces résultats montrent qu'une variabilité représentative de l'ensemble de données HistAerial a déjà été capturée dans le sous-ensemble équilibré en taille.

Deuxièmement, on remarque que le filtre LCoLBP a permis d'obtenir des résultats étonnamment meilleurs qu'AlexNet sur les imagettes de 25 pixels × 25 pixels. Nous nous attendions à que ce sous-ensemble favorise le réseau de neurones profond en raison de la grande quantité de données disponibles, censée permettre une optimisation plus efficace des poids du réseau. AlexNet n'a pas donné les résultats escomptés. Nous pouvons seulement supposer que cette observation résulte de l'utilisation de filtres de convolutions sur des données représentant un petit contexte

TABLEAU 3.7 – Meilleurs résultats obtenus sur le sous ensemble de HistAerial équilibré en classe (*i.e.* même proportion d'imagettes par taille).

Meilleurs résultats obtenus pour chaque méthode						
Filtre	**Paramètres** (P, R)	***mapping***	**Nombre de caractéristiques**	***Accuracy* (%) par taille d'imagette (pixels)**		
				25 × 25	50 × 50	100 × 100
LCoLBP + Random Forest	(8,{1,2,3})	*none*	240	**75.0**	84.1	89.3
AlexNet + SGD	learning rate : 0.001	*	4096	73.4	**85.6**	*
AlexNet + RMSPROP	learning rate : 0.0001	*	4096	*	*	**90.4**

spatial, bien que cette hypothèse soit en désaccord avec les résultats observés sur le tableau 3.4. Cette architecture a obtenu des scores légèrement plus élevés sur les autres tailles d'imagettes que le filtre LCoLBP, avec des gain 1.5% sur les imagettes de 50 pixels × 50 pixels et 1.1% sur les imagettes de 100 pixels × 100 pixels. Ces résultats sont conformes à l'hypothèse présentée par Basu et al. [BKD+16] : les réseaux de neurones à convolutions semblent ne pas être aussi performants sur les données de textures (non spatialisées) que sur des images plus classiques (représentant des entités dans leur contexte, des objets).

On constate enfin que le contexte spatial semble fournir une amélioration significative (+15% de taux de bonne classification entre les plus petites et les plus grandes imagettes). Ce point est en accord avec ce qui était observé sur le sous-ensemble équilibré en taille.

3.4.4 Conclusion partielle

Dans ces travaux, un nouveau de jeu de données a été proposé pour l'analyse d'images aériennes historiques panchromatiques. Il est composé de plusieurs millions d'imagettes annotées à trois niveaux d'échelle spatiale. Une comparaison des méthodes d'extraction de caractéristiques et de classification de la littérature a été réalisée sur ce jeu de données. Deux nouveaux filtres ont également été proposés. Parmi eux, le LCoLBP combiné à une forêt aléatoire d'arbres décisionnels a permis d'obtenir des résultats similaires (légèrement inférieurs) aux réseaux de neurones profonds à convolution, et ce pour un vecteur de caractéristiques 17 fois plus petit et un temps d'exécution bien inférieur. De manière générale, nous n'avons pas décelé de contre-indications à l'utilisation des filtres basés sur la texture. Ces derniers semblent être particulièrement adaptés pour des applications sur des ordinateurs peu puissants. On notera néanmoins que les DCNN tendent à obtenir des taux de classification plus élevés, et ce quelle que soit l'architecture utilisée. La principale limitation des DCNN dans notre cadre de travail est liée aux ressources matérielles qu'ils nécessitent, les rendant peu praticables sans carte graphique pour des applications interactives (*e.g.*, Gouramic, voir Annexe A). Ils semblent cependant indiqués pour des applications hors ligne (*i.e.*, l'utilisateur n'attendant pas devant l'écran).

3.5 Extension aux images en couleurs : cas des écorces d'arbres

Nous avons vu dans les sections précédentes que la texture est un facteur discriminant viable pour l'analyse automatique d'images aériennes historiques. Nous avons cependant fait la remarque que l'absence d'informations sur la couleur pouvait avoir un impact sur les taux atteignables de bonne classification. Afin de vérifier cette hypothèse avant de nous lancer dans des travaux sur la colorisation automatique d'images aériennes historiques (voir chapitre 4), nous avons collaboré une autre doctorante du LIRIS travaillant sur la classification d'écorces d'arbres dans un environnement mobile (*i.e.*, identifier un arbre par son écorce sur *smartphone*). D'un point de vue application, les images d'écorces d'arbres représentent des éléments sur lesquels les filtres de textures ont tendance à être particulièrement efficaces.

Ainsi, en 2004, Wan *et al.* [WDH+04] ont proposé de comparer plusieurs approches statistiques pour reconnaitre des textures, dont les GLCM (voir chapitre 2). Afin de conserver l'information

portée par la couleur, Wan *et al.* ont appliqué leur approche sur chaque canal couleur (espace RVB) avant de concaténer les caractéristiques obtenues. Huang *et al.* [HHD+06] ont pour leur part exploré l'utilisation d'une banque de filtres de Gabor (voir chapitre 2) pour la classification d'écorces. Bakic *et al.* [BMOL+13] ont quant à eux exploré l'utilisation de plusieurs espaces couleur (RVB, HSV) pour représenter les écorces d'arbres. Bertrand *et al.* [BCT17] ont cherché à combiner des caractéristiques orientées, obtenues à l'aide de filtres de Gabor, avec une représentation éparse de la texture représentée à l'aide du détecteur de contours de Canny et d'un échantillonnage linéaire éparse en deux dimensions. L'information de couleur a ici été ajoutée par les auteurs en concaténant l'histogramme de teinte (espace couleur HSV) aux caractéristiques précédentes. Parmi les approches basées sur les filtres des type LBP, Boudra *et al.* [BYB18] ont proposé un descripteur de textures nommé *Statistical Macro Binary Pattern* (SMBP). SMBP encode l'information entre différentes "macro-structures" à l'aide d'une représentation statistique de chaque échelle. Porebski *et al.* [PVMH14] ont quant à eux appliqué des filtres de type LBP sur plusieurs espaces couleur en cherchant à concaténer de manière optimale les histogrammes obtenus (*e.g.*, concaténation des histogrammes RVB et HSV). Les auteurs ont réussi à obtenir des taux de classification supérieurs à l'état de l'art, au prix de vecteurs de caractéristiques de très hautes dimensions.

Ici, nous nous sommes intéressés au cas particulier de la reconnaissance des écorces dans un environnement contraint (sur mobile). Nous avons de fait cherché à minimiser les ressources nécessaires, avec un focus particulier sur la mémoire utilisée (taille des vecteurs de caractéristiques).

3.5.1 Jeux de données



FIGURE 3.13 – Exemples d'images d'écorces d'arbres du jeu de données Bark-101 [RBCJT19].

Nous avons travaillé sur plusieurs jeux de données de la littérature auxquels nous avons ajouté Bark-101 (voir `http://eidolon.univ-lyon2.fr/~remi1/Bark-101/`). Bark-101 est un nouveau jeu de données créé par Sarah Bertrand dans le cadre de sa thèse et présenté dans le cadre de nos travaux joints [RBCJT19]. Les caractéristiques des jeux de données que nous avons utilisés sont résumés sur le tableau 3.8. On remarquera que la plupart de ces jeux de données sont constitués d'une faible quantité d'images, pour un faible nombre de classes. Le jeu de données Bark-101 propose quant à lui une quantité de données relativement faible, mais un nombre de classes conséquent (101 classes). Il a été conçu à partir des images du défi PlantCLEF. Ces images ont été acquises en milieu naturel dans des conditions non controllées afin de permettre le développement d'algorithmes de reconnaissance des végétaux. Ici, seules les images correspondant à des troncs d'arbres ont été utilisées pour créer Bark-101. Ces dernieres ont été manuellement segmentées afin de supprimer l'information contenue dans le fond et ne conserver que les écorces. Afin de simuler des conditions réelles d'utilisation, nous avons choisi de suivre Wendel *et al.* [WSG11] en n'imposant pas de contraintes sur la taille des images segmentées. De par la méthode d'acquisition des images originelles, Bark-101 possède une forte variabilité intra-classe. De plus, le grand nombre de classes dans ce jeu de données induit une variabilité inter-classe relativement faible (plus on augmente le nombre d'espèces, plus les chances d'avoir des images similaires entre classes différentes sont élevées).

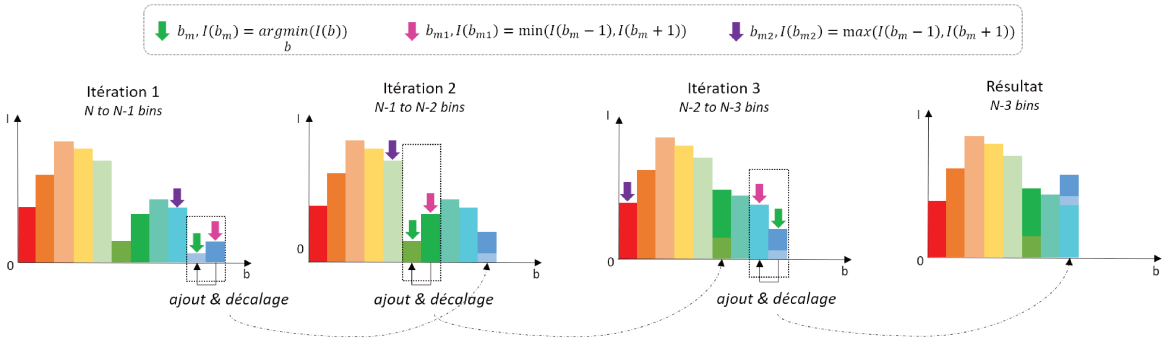


FIGURE 3.14 – Schéma de l'algorithme de réduction de l'histogramme de teintes appliqué pour 3 itérations. Le nombre de *bins* dans l'histogramme est réduit de 1 après chaque itération. La réduction est réalisée à partir du bin b d'intensité la plus faible.

3.5.2 Méthodes

Afin de classifier automatiquement les écorces d'arbres, nous nous sommes donc attachés à combiner les informations de couleur représentées par l'histogramme de teintes dans l'espace HSV à celles extraites à partir des filtres de type LBP. Cependant, la combinaison de ces deux informations peut générer des vecteurs de caractéristiques de grandes dimensions. Nous avons donc cherché à réduire la taille des vecteurs de caractéristiques de chacune des informations.

Réduction de l'histogramme de couleur

L'approche proposée par Sarah Bertrand afin de réduire la taille de l'histogramme couleur se base sur l'observation que certaines couleurs, telles que le bleu ou le violet, n'ont pas une contribution significative à la signature des images d'écorces (couleurs sous-représentées). Cependant, supprimer complètement l'information portée par ces couleurs pourrait résulter en des histogrammes moins discriminants, rendant par la même occasion le processus de classification des écorces moins performant. Afin de réduire la taille de l'histogramme de teinte, l'approche proposée ici consiste à fusionner les effectifs des *bins* couleur les moins représentés via un processus itératif.

Soit X un jeu de données constitué de k images en couleurs, séparées en un jeu d'entraînement X_{train} et un jeu de test X_{test}. On commence par calculer l'histogramme de teinte de chaque image de X_{train}, que l'on accumule (somme) bin à bin dans un histogramme sommée H_s. L'histogramme sommé H_s est ici supposé représenter l'*a priori* sur la teinte du jeu de données. Afin de réduire le nombre de *bins* que possède cet histogramme, nous allons ajouter itérativement l'effectif du bin b ayant l'effectif le plus faible de H_s, à l'un de ses *bins* voisins $b+1$ ou $b-1$. Le *bin* voisin sélectionné est celui qui a le plus faible effectif parmi les deux. Une fois cette opération réalisée, l'histogramme est décalé vers la gauche afin de réduire sa dimension. Ce processus itératif est arrêté lorsque le nombre de *bins* désiré, fixé par l'utilisateur, est atteint. Il a été fixé à 30 *bins*, par validation croisée, dans le cadre des expériences menées sur les écorces [RBCJT19]. L'ordre et la position des opérations d'ajout et de décalage sont stockés dans une table de correspondances M.

TABLEAU 3.8 – Caractéristiques de différents jeux de données d'écorces d'arbres considérés.

Caractéristiques	**BarkTex** [Lak98]	**NewBarkTex** [PVMH14]	**Trunk12** [Š14]	**AFF** [WSG11]	**Bark-101** [RBCJT19]
Nombre de classes	6	6	12	11	101
Nombre d'images	408	1632	393	1082	2587
Nombre d'images par classe	68	272	30-45	16-213	2-138
Taille des images	256x384	64x64	1000x1334	1000x(478-1812)	(69-800)x(112-804)
Différentes illuminations	✓	✓	✗	✓	✓
Différentes échelles	✓	✓	✗	✓	✓
Perturbations (ombres, lichen)	✗	✗	✗	✓	✓
Séparation entraînement / test	✗	50/50	✗	✗	50/50

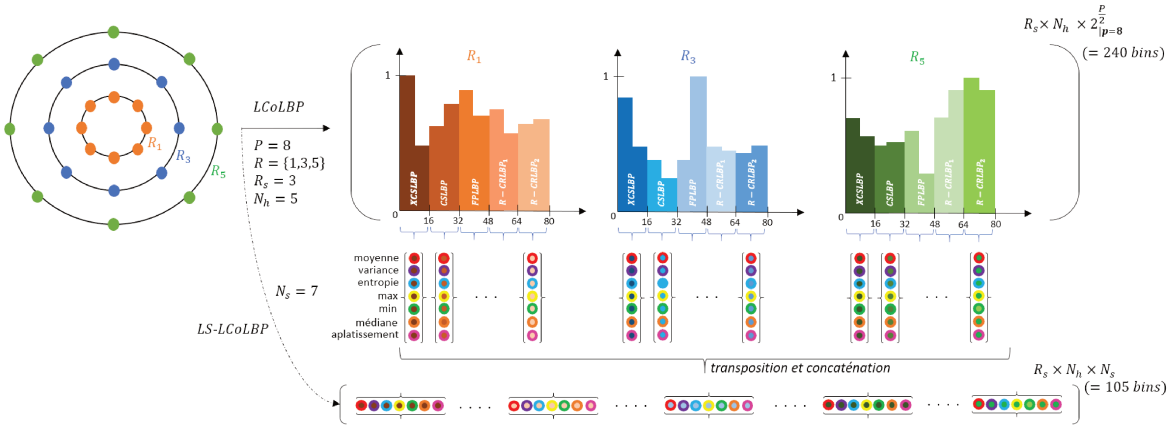


FIGURE 3.15 – Schéma représentant l'extraction de $N_s = 7$ statistiques tardives à partir de l'histogramme du filtre LCoLBP calculé sur un voisinage composé de 3 rayons ($R_s = 3$) avec 8 pixels voisins P par rayon. Les points colorés représentent les statistiques (couleur extérieure du point) obtenues pour les $N_h = 5$ sous-histogrammes du LCoLBP (couleur intérieure du point).

Cette table de correspondances M, associée aux opérations décrites ci-dessus, est ensuite utilisée sur chaque histogramme de teinte du jeu de test X_{test} afin de calculer un histogramme réduit pour chaque image. Les opérations d'ajout et de décalage sont illustrées sur la figure 3.14.

Réduction de l'histogramme de LBP

Afin de réduire la taille des histogrammes issus de filtres de type LBP, nous avons étudié l'intérêt de l'extraction de statistiques à partir de leurs histogrammes. Nous avons nommé les caractéristiques ainsi extraites les "statistiques tardives" (*late statistics*, LS), par opposition aux statistiques utilisées par Boudra *et al.* [BYB18] qui sont obtenues avant la génération des codes binaires. On remarquera que les statistiques tardives sont, par construction, similaires à celles générées par les GLCM (voir chapitre 2).

Soit H_t un histogramme obtenu à l'aide d'un filtre de type LBP, lui même constitué d'une concaténation ordonnée de N_h sous-histogrammes $\{h_1, h_2, ..., h_{N_h}\}$ de tailles connues. Pour chaque sous histogramme $h_i, i \in \{1, ..., N_h\}$, on calcule N_s statistiques tardives. Les statistiques ainsi obtenues sont concaténées dans le même ordre que les sous-histogrammes afin de constituer un vecteur de caractéristiques de dimension réduite. En supposant un unique histogramme H_t par voisinage (P,R), ce processus permet d'obtenir $R_s \times N_s \times N_h$ caractéristiques avec R_s le nombre de rayons. Un exemple avec $N_s = 7$ statistiques et $R_s = 3$ rayons est présenté sur la figure 3.15 pour le filtre LCoLBP. On considère ici chaque rotation du R-CRLBP comme étant indépendante.

Les statistiques tardives ont plusieurs avantages. D'une part, elles ne nécessitent pas un nouvel échantillonnage des motifs binaires, contrairement à l'approche proposée par [BYB18], car elles se basent uniquement sur des histogrammes déjà générés. D'autre part, elles ne nécessitent pas la réimplémentation des descripteurs de textures. Enfin, ces statistiques sont supposées agir comme un algorithme de normalisation spatiale, au sens où chaque sous histogramme sera résumé par un nombre N_s fixé de statistiques, quelque soit le nombre de *bins* qu'il contient et quelque soit le voisinage (P,R) sur lequel il a été généré. Cette propriété est particulièrement intéressante pour générer des vecteurs de caractéristiques contenant une quantité équilibrée de caractéristiques pour des motifs de textures différents. Cependant, les statistiques vont significativement réduire (résumer) l'information portée par l'histogramme. A l'image des *mapping* classiquement utilisés avec les filtres de type LBP, cette approche est donc susceptible de diminuer le pouvoir discriminant des filtres utilisés.

3.5.3 Expériences et résultats

Nous avons réalisé une étude comparative avec et sans réduction d'histogrammes sur les jeux de données d'écorces présentés précédemment. Pour cela, nous avons considéré le filtres LCoLBP et le filtre CLBP avec *mapping* riu^2 afin d'étudier l'intérêt des statistiques tardives avec et sans *mapping*.

Stratégies d'évaluation

On distingue ici deux stratégies d'évaluation en fonction des jeux de données et des approches utilisées dans la littérature.

- Évaluation standard : on entraîne un classifieur sur le jeu de données d'entraînement, et on utilise le jeu de test pour évaluer la performance de l'approche. Cette approche est appliquée sur NewBarkTex et Bark-101, tous deux proposant une séparation claire du jeu de données. Aucun ensemble de validation n'est ici inclu. Les paramètres des classifieurs sont de fait optimisés, si nécessaire, par validation croisée sur le jeu d'entraînement.
- Évaluation en *leave-one-out* : il s'agit ici d'une approche particulièrement utilisée sur les petits ensembles de données. Soit S un ensemble de N échantillons. On réalise alors N itérations. A chaque itération, $i \in \{1, ..., N\}$, l'échantillon $s(i) \in S$ est réservé pour le test, et tous les autres échantillons $S - \{s(i)\}$ sont utilisés pour l'entraînement. Si $s(i)$ est correctement classifié, le résultat de l'itération i est positif, sinon il est négatif. Le taux de bonne classification (*accuracy*) est obtenu en moyennant les résultats obtenus pour toutes les itérations. Cette approche a été appliquée sur les jeux de données BarkTex, Trunk12 and AFF, en accord avec les travaux réalisés par Boudra *et al.* [BYB18].

Pour les deux types d'évaluation, la métrique utilisée est le taux de bonne classification en top-1. Le classifieurs KNN K = 1 avec la distance L1 a été utilisé comme référence, celui-ci étant le plus utilisé dans le contexte de la classifcation d'écorces. Pour la stratégie d'évaluation standard, nous avons également utilisé un SVM multi-classes avec un noyau RBF, en accord avec Porebski *et al.* [PHVH18]. Les paramètres du SVM ont été obtenus par validation croisée pour chaque filtre et chaque jeu de données indépendamment.

Choix des statistiques tardives

Nous avons considéré 7 statistiques dans nos expériences : la moyenne, la variance, l'entropie, le minimum, le maximum, la valeur médiane et l'aplatissement (*kurtosis*). Afin de déterminer les meilleures combinaisons de statistique pour chacun des filtres, nous avons mené une étude par ablation sur le jeu de données BarkTex. Les résultats de cette étude sont présentés sur le tableau 3.9. On y observe qu'ajouter naïvement des statistiques (7 premières lignes) peut réduire les taux de bonne classification. Ainsi le choix des statistiques tardives doit être fait de façon judicieuse pour chacun des filtres, ce qui représente une faiblesse pour la méthode. En nous basant sur ces résultats, le nombre de statistiques N_s a été fixé à 6 pour le LCoLBP et à 4 pour le CLBP. Nous n'avons pas calculé de statistiques pour le sous-histogramme obtenu à l'aide du CLBP_C, ce dernier ne contenant que deux *bins*. Ainsi, les statistiques tardives du LCoLBP (LS − LC*o*LBP) génèrent des vecteurs de $R_s \times 5 \times 6$ caractéristiques, où 5 est le nombre de sous filtres et 6 le nombre de statistiques. Les statistiques tardives du CLBP génèrent des vecteurs de $R_s \times (2 + 2 \times 4)$ caractéristiques.

Résultats

Les résultats que nous avons obtenus sont reportés sur les tableaux 3.10 et 3.11. Les taux de bonne classification d'études précédentes ont été reportés et indiqués à l'aide d'une étoile (*). Pour les jeux de données AFF, Trunk12 et BarkTex, nous avons reporté les résultat obtenus avec

les méthodes *MSLBP** et *SLBP** de [BYB18]. Pour le jeu de données NewBarkTex, nous avons reporté les résultats obtenus par les méthodes de *Wang17** [JW17], de *Sandid16** [SD16], et de *Porebski18** [PHVH18]. Nous avons également considéré les résultats des méthodes proposées par [BCT17], que nous avons renommées *GWs* et *GWs/H180* (concaténation avec l'histogramme de teintes complet). Tous les résultats non reportés correspondent à nos propres implémentations en C++ pour les filtres de type LBP et les histogrammes de couleur, et Python avec la librairie Scikit-learn pour le calcul des statistiques tardives et l'utilisation des classifieurs. Nous discutons les résultats obtenus ci-après.

Apport de la couleur. Nous pouvons observer que les filtres de textures permettent d'obtenir des taux de classification plus élevés lorsqu'ils sont combinés avec les histogrammes de couleur réduits (H30) ou non (H180). Pour rappel, les histogrammes de couleur sont obtenus à partir du canal de teinte dans l'espace couleur HSV. Lorsqu'il est utilisé seul, H180 permet d'obtenir des taux de classification supérieurs à ceux obtenus avec H30 de 3.3% en moyenne sur AFF, Trunk12 et BarkTex. Cependant, lorsqu'ils sont combinés aux histogrammes issus des filtres de type LBP, leurs contributions apparaissent équivalentes. Ces résultats montrent l'intérêt de l'algorithme de réduction d'histogramme présenté précédemment. De plus, ils confirment que la couleur est un indice visuel non négligeable *a priori* pour la classification d'images texturées, et en particulier pour la classification d'écorces. Cette observation est en accord avec les travaux de [JW17].

Apport des statistiques tardives. Les statistiques tardives permettent de diminuer la taille des vecteurs des caractéristiques d'un facteur 2.7 pour le filtre LCoLBP et 2.2 pour le CLBP, avec une diminution des taux de bonne classification de seulement 5.5% en moyenne. Ce chiffre est néanmoins à nuancer en fonction des jeux de données et des stratégies d'évaluation utilisées. Les statistiques tardives semblent ainsi particulièrement efficaces dans le cadre d'un stratégie de type *leave-one-out* (voir tableau 3.10). Elles semblent cependant moins intéressantes dans le cadre d'une stratégie standard (voir tableau 3.11). A noter que ces résultats peuvent partiellement s'expliquer par la faible quantité de données d'entraînement disponible comparé à l'approche *leave-one-out*.

3.5.4 Conclusion partielle

Nous avons évalué l'intérêt de combiner des filtres de textures avec la couleur, représentées ici par des histogrammes de teintes. Les résultats obtenus sur 5 jeux de données d'écorces d'arbres nous ont montré une complémentarité entre ces deux types d'informations. Par ailleurs, nous avons évalué deux approches permettant respectivement de réduire la taille des vecteurs de caractéristiques obtenus par les filtres de LBP et la taille des histogrammes de teintes. Nous avons ainsi pu mettre en avant l'intérêt de ces algorithmes pour réduire la quantité d'information nécessaire pour classifier des images d'écorces. Cependant, ces statistiques réduisent les taux de

TABLEAU 3.9 – Étude par ablation des statistiques tardives appliquées aux filtres LCoLBP et CLBP sur le jeu de données BarkTex.

Late Statistics							Accuracy (%)	
moyenne	variance	entropie	minimum	maximum	médiane	aplatissement	*LS-LCOLBP*	*LS-CLBP*
✓	–	–	–	–	–	–	81.9	71.8
✓	✓	–	–	–	–	–	82.8	59.6
✓	✓	✓	–	–	–	–	78.4	64.7
✓	✓	✓	✓	–	–	–	82.8	63.2
✓	✓	✓	✓	✓	–	–	83.1	69.4
✓	✓	✓	✓	✓	✓	–	86.3	72.1
✓	✓	✓	✓	✓	✓	✓	89.5	62.8
✓	✓	–	✓	✓	✓	✓	88.2	60.1
✓	–	✓	✓	✓	✓	✓	**89.5**	62.5
✓	–	–	✓	✓	✓	✓	88.2	59.6
✓	–	–	✓	✓	✓	–	88.2	**75.3**

TABLEAU 3.10 – Résultats obtenus avec un 1-NN sur les jeux de données BarkTex, AFF et Trunk12. En bleu : Résultats les plus élevés reportés dans la littérature. En vert : résultats les plus élevés dans cette comparaison. En rouge : résultats les plus élevés avec les statistiques tardives.

Filtre	Nombre de caractéristiques	*Accuracy* / Jeu de données (%)		
		AFF	Trunk12	BarkTex
*MSLBP**	2 816	63.3	63.3	**86.8**
*SMBP**	10 240	**71.7**	**71.0**	84.3
H30	30	50.5	64.4	55.4
H180	180	55.6	69.0	61.3
LCoLBP	240	75.3	77.1	92.1
LCoLBP / H30	270	**80.7**	**84.2**	**92.4**
LCoLBP / H180	420	80.7	84.2	91.7
CLBP	66	68.1	70.0	78.7
CLBP / H30	96	72.9	77.4	83.8
CLBP / H180	246	73.5	78.1	84.3
GWs	121	48.2	39.9	56.1
GWs / H30	151	64.7	74.3	66.2
GWs / H180	301	66.5	76.1	69.6
LS-LCoLBP	90	69.4	74.6	89.5
LS-LCoLBP / H30	120	**76.9**	**80.7**	90.2
LS-LCoLBP / H180	270	76.9	80.7	**91.2**
LS-CLBP	30	59.1	70.0	75.3
LS-CLBP / H30	60	65.4	77.4	78.2
LS-CLBP / H180	210	67.9	78.1	79.4

TABLEAU 3.11 – Résultats obtenus sur les jeux de données NewBarkTex et Bark-101.

		Accuracy / Jeu de données (%)			
		NewBarkTex		Bark-101	
Filtre	Nombre de caractéristiques	KNN	SVM	KNN	SVM
*Porebski18**	10 752	–	**92.6**	–	–
*Wang17**	267	84.3	–	–	–
*Sandid16**	3 072	–	82.1	–	–
H30	30	48.0	50.6	19.1	20.4
H180	180	48.5	53.6	22.2	20.9
LCoLBP	240	78.8	**89.3**	34.2	**41.9**
LCoLBP / H30	270	–	–	–	**44.0**
LS-LCoLBP	90	66.5	79.4	28.3	30.1
LS-LCoLBP / H30	120	71.9	82.0	27.6	**32.1**
LS-LCoLBP / H180	270	72.3	**82.2**	27.8	31.0
GWs / H30	151	60.4	74.1	28.2	**31.7**
GWs / H180	301	54.1	63.6	31.8	**32.2**

classification obtenus, et ce particulièrement dans un contexte où relativement peu de données d'entraînement sont disponibles par rapport aux données évaluées. Face à cette observation, nous ne les avons pas appliquées sur les histogrammes issus de HistAerial, les gains obtenus en termes de mémoire ne compensant pas la perte de précision (*accuracy*) dans un environnement moins contraint que les applications mobiles.

3.6 Conclusion

Résumé des travaux réalisés. Nous avons présenté HistAerial, un jeu de données contenant plusieurs millions d'imagettes à plusieurs échelles pour 7 classes d'occupation du sol. Au travers de ce jeu de données, nous nous sommes intéressés à la classification des images aériennes historiques panchromatiques à l'aide de filtres de textures, de classifieurs classiques et de réseaux de neurones profonds à convolutions. Ces travaux comparatifs nous ont permit de montrer l'intérêt des filtres de textures pour cette tâche. Les caractéristiques extraites par ces derniers permettent d'obtenir des résultats équivalents aux réseaux de neurones profonds sur HistAerial, et ce pour

des temps de traitement et des besoins en mémoire (taille des vecteurs de caractéristiques) moins importants. Nous avons par la suite étendu nos travaux à la classification d'images en couleurs en concaténant les caractéristiques extraites par les filtres de textures et des histogrammes de teintes. Nous avons pu montrer la complémentarité de ces deux types d'informations. Nous avons également proposé une approche pour réduire la taille des vecteurs de caractéristiques, avec des résultats que nous qualifierons de contrastés : la taille des histogrammes est effectivement réduite de moitié (cas du LCoLBP), mais des pertes plus ou moins conséquentes de taux de bonne classification (*accuracy*) ont pu être observées en fonction des jeux de données (*e.g.*, 7.1% sur NewBarkTex).

Vision critique sur les travaux réalisés. Le jeu de données HistAerial que nous avons proposé est principalement localisé sur la région Rhône-Alpes. Malgré la quantité de données qu'il contient, il n'est probablement pas représentatif du cas général associé au territoire français. De plus, il ne représente que le cas où une seule et unique classe est supposée présente sur les imagettes (aux erreurs d'annotations près). Il y a ici la nécessité de collecter des données sur l'ensemble du territoire. Pour cela, nous avons développé le logiciel Gouramic, présenté en Annexe A, qui permet non seulement d'obtenir des cartes d'occupation du sol de façon interactive, mais également de sauvegarder les annotations partielles fournies par l'utilisateur. L'application de ce logiciel dans le cadre de TESTIS permet la génération de carte d'occupation du sol et de données annotées manuellement. Concernant les méthodes proposées, le choix de la combinaison des filtres utilisés dans le LCoLBP a été réalisée de façon empirique, en se basant sur les types de motifs représentés. D'autres filtres, et d'autres combinaisons de filtres de type LBP mériteraient d'être étudiées (*e.g.*, ajout de l'histogramme du LBP avec *mapping* riu^2). De plus, bien que le LCoLBP soit relativement performant sur le jeu de données HistAerial, sa formulation non-invariante à la rotation est peut-être moins intéressante que celle des filtres existants associés au *mapping* riu^2 dans le cas général (*e.g.*, le CLBP). En pratique, nous ne pouvons ici que recommander l'évaluation des méthodes sur les jeux de données d'intérêts. Le R-CRLBP permet quant à lui d'obtenir des résultats au niveau de nombreux algorithmes de la littérature, mais il est moins performant que les meilleures méthodes existantes. Celui-ci permet principalement de compléter la représentation du LCoLBP. Enfin, nous n'avons pas cherché à étudier l'intérêt d'un voisinage différent de $(P,R) = (8,\{1,2,3\})$ dans ces travaux afin de limiter l'espace des paramètres et le coût algorithmique associé à un $P > 8$. Il pourrait néanmoins être intéressant de faire varier P et R afin d'obtenir des résultats plus approfondis. Par ailleurs, les statistiques extraites des histogrammes générés par des filtres de type LBP ne permettent pas d'obtenir des taux de classifications au niveau de ceux provenant de l'utilisation de filtres seuls. Le gain en termes de complexité spatiale qu'ils permettent d'obtenir reste quant à lui limité. Nous ne recommandons pas leur usage dans des environnements où les contraintes matérielles ne seraient pas fortes (*e.g.*, les ordinateurs ont moins de contraintes que les mobiles).

Chapitre 4

Colorisation automatique

Ce chapitre présente les travaux réalisés portant sur la colorisation automatique des images aériennes historiques. Notre but était ici double : (1) proposer une visualisation alternative des images historiques aux géomaticiens afin de les aider dans le processus d'annotation, et (2) évaluer l'intérêt des couleurs générées pour la classification. Nous nous sommes particulièrement intéressés à l'utilisation de réseaux de neurones profonds à convolutions non-supervisés. Le choix d'une approche non-supervisée a été fait afin de pouvoir optimiser les réseaux de neurones d'une part à l'aide des images historiques, uniquement disponibles en niveaux de gris, et d'autre part en utilisant des images récentes en couleurs. Nous avons également étendu nos travaux à d'autres types d'images afin d'évaluer une nouvelle méthode de colorisation que nous avons proposée.

Sommaire

4.1 Introduction 84
4.2 Travaux connexes et notions spécifiques 85
4.2.1 Réseaux de neurones adversaires génératifs (GAN) 85
4.2.2 Réseaux de neurones cycliques 85
4.2.3 Approches pour la colorisation 86
4.3 Vers une colorisation automatique des images aériennes historiques 88
4.3.1 Col-Cycle 89
4.3.2 Reconstruction des images colorisées 91
4.3.3 Données et entraînement 93
4.3.4 Résultats et discussions 93
4.3.5 Application à la classification 94
4.3.6 Conclusion partielle 96
4.4 Vers une amélioration de la colorisation 97
4.4.1 Blocs de base 97
4.4.2 SpyncoGan 100
4.4.3 Mise en place des expériences 104
4.4.4 Résultats et discussions 105
4.4.5 Application à la classification 110
4.4.6 Conclusion partielle 112
4.5 Conclusion 114
4.6 Visualisations supplémentaires 114

4.1 Introduction

La colorisation automatique consiste à générer des images en couleurs à partir d'images panchromatiques (voir figure 4.1). Le développement d'algorithmes de colorisation est un problème considéré comme étant mal posé (*ill-posed*) du fait de la non injectivité de la transformation recherchée. En effet, les intensités des pixels sur les images en niveaux de gris représentent une moyenne pondérée des couleurs. Il existe de fait une multitude de mélanges de couleur possibles pour un niveau de gris donné. Pour les images numériques, si nous considérons un espace couleur cible de type RVB, c'est à dire que chaque pixel couleur est représenté par 3 valeurs différentes, la colorisation consiste alors à estimer 3 valeurs à partir d'une seule : l'intensité du pixel en niveaux de gris. En pratique, il s'agit de réaliser cette estimation pour tous les pixels de l'image en se basant sur des caractéristiques représentatives de son contenu, de telle sorte que l'image colorisée puisse être considérée comme étant réaliste. Ce critère étant subjectif, les algorithmes de colorisation sont généralement évalués ou bien à l'aide d'un questionnaire portant sur les images générées, ou bien à l'aide de métriques quantitatives usuellement utilisées pour la génération de données continues (régression). Des méthodes d'évaluations basées sur l'estimation d'un score de qualité à l'aide de réseaux de neurones profonds à convolutions ont également vu le jour. Celles-ci peuvent cependant souffrir d'un biais d'apprentissage lié aux jeux de données sur lesquels les paramètres des réseaux ont été optimisés.

De nombreux travaux se sont ainsi intéressés à la problématique de la colorisation, que ce soit à l'aide d'approches guidées par l'utilisateur (*i.e.*, l'utilisateur indique la couleur attendue pour certains pixels), d'approches automatiques [ISSI16; LMS16], ou d'approches hybrides [ZZI+17]. Parmi les approches existantes, l'utilisation de réseaux de neurones entièrement convolutifs est devenue particulièrement populaire pour la génération d'images colorisées. Ces approches se basent en général sur la disponibilité d'images en couleurs réelles appariées à leur équivalent en niveaux de gris. Dans le cas qui nous intéresse, les images aériennes historiques ne sont disponibles qu'en niveaux de gris, et elles ont été acquises à des dates et résolutions variées. Afin de tenir compte de ce problème, nous avons choisi d'explorer l'utilisation de méthodes non-supervisées développées pour la translation d'image à image et l'adaptation de domaines (*e.g.*, donner un rendu photo-réaliste à des images de jeux vidéo). Pour la colorisation, tâche qui nous intéresse ici, ces approches permettent d'optimiser un DCNN de type encodeur-décodeur à l'aide d'images non appariées en exploitant à la fois des images historiques disponibles en niveaux de gris, et des images plus récentes disponibles en couleurs. Cela permet de tenir compte implicitement de la variabilité des représentations disponibles au sein des données que l'on cherche à coloriser (échelles, lieux, dates, capteurs) lors de l'optimisation du réseau : les paramètres du réseau sont directement entraînés à partir de ces données.

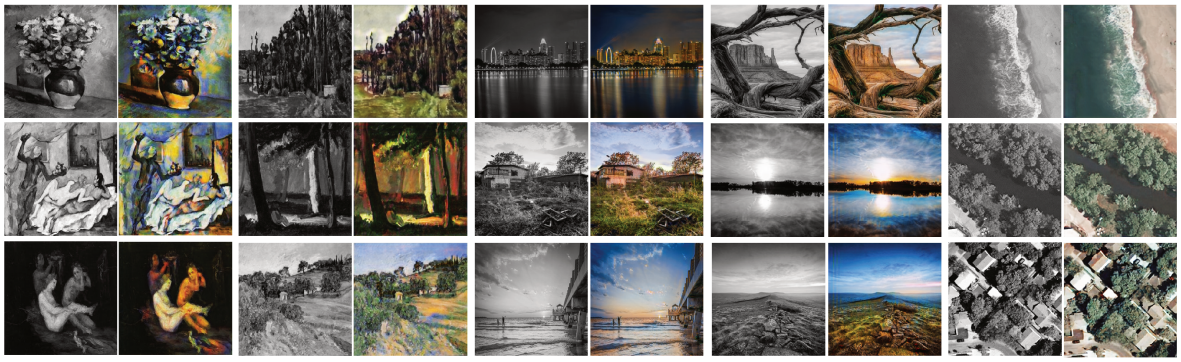


FIGURE 4.1 – Exemples de peintures, de paysages et d'images aériennes colorisées avec SpyncoGan.

4.2 Travaux connexes et notions spécifiques

Cette section présente les travaux connexes aux méthodes que nous avons étudiées pour la colorisation. Nous allons ici introduire les notions de réseaux de neurones adversaires génératifs et de réseaux de neurones cycliques. Nous présenterons ensuite des travaux récents sur la colorisation d'images.

4.2.1 Réseaux de neurones adversaires génératifs (GAN)

Les réseaux de neurones adversaires génératifs (*GAN*) [GPAM$^+$14], ainsi que leurs variantes à convolutions (DCGAN) [RMC15], sont composés de deux éléments :

- un réseau de neurones générateur G, qui va chercher à convertir un signal (vecteur) z, échantillonné aléatoirement à partir d'une distribution connue (*e.g.*, distribution gaussienne), en une donnée cible réaliste G(z) par rapport à un *ensemble* de données réelles (une distribution).
- un réseau de neurones discriminateur D, qui va avoir pour tâche de différencier les données réelles et les données générées artificiellement par G (les fausses données).

Une fonction de coût est calculée à partir de la sortie du discriminateur. Elle permet de contraindre le générateur afin qu'il génère des images de plus en plus réalistes ; aptes à tromper le discriminateur ; et de contraindre le discriminateur à être de plus en plus performant. Pour cela, on attribue une étiquette positive (= 1) à chaque donnée réelle x et une étiquette nulle (= 0) à chaque donnée générée G(z). La fonction de coût va comparer ces étiquettes aux prédictions réalisées par D (D(x) avec 1, et D(G(z)) avec 0). La fonction de coût peut alors s'exprimer à l'aide de l'équation (4.1) [GPAM$^+$14], où G cherche à maximiser l'erreur de classification commise par D, et D à la minimiser. Cette fonction de coût est calculée comme étant la moyenne statistique (espérance $\mathbb{E}[.]$) sur un $batch$ de données.

$$\mathscr{L}_{\text{GAN}} = \min_{\text{G}} \max_{\text{D}} [\mathbb{E}[log(\text{D}(x))] + \mathbb{E}[log(1 - \text{D}(\text{G}(z)))]] \tag{4.1}$$

Le but final est ici d'atteindre un état proche de l'équilibre de Nash, où le générateur et le discriminateur obtiendraient tous deux des résultats satisfaisants. A noter que l'utilisation de la fonction de coût en sortie du discriminateur est communément nommée fonction de coût adversaire (*adversarial loss*, ou GAN *loss*). Cette fonction de coût peut être utilisée pour contraindre des réseaux de neurones de type encodeur-décodeur, les caractéristiques encodées remplaçant alors le signal échantillonné aléatoirement z. Par souci de clarté, nous exclurons les fonctions *min* et *max* dans les notations des fonctions de coût par la suite.

4.2.2 Réseaux de neurones cycliques

Les réseaux de neurones cycliques [IZZE17; ZPIE17] ont fortement contribué à populariser les méthodes de translation d'image à image. Ils ont initialement été développés pour convertir des images entre deux espaces de représentations (deux domaines), A et B, à l'aide d'un réseau de neurones générateur de type encodeur-décodeur pour chaque translation (*i.e.*, un réseau G pour réaliser la translation de A vers B, et un réseau F pour réaliser la translation de B vers A).

Parmi les méthodes les plus populaires, le réseau de neurones Pix2Pix, proposé par Isola *et al.* [IZZE17], requiert l'existence de données appariées (*i.e.*, correspondance 1 : 1 entre une image du domaine A et une image du domaine B) pour contraindre de façon supervisée la génération d'images réalistes dans chacun des domaines. Afin de considérer un cas plus général, l'utilisation de réseaux de neurones cycliques non-supervisés tels que CycleGan [ZPIE17] et MartaGan [LFW$^+$17] ont été proposés en exploitant une fonction de coût adversaire permettant de contraindre l'optimisation de chaque générateur. On parle alors de réseaux de neurones adversaires cycliques. En

particulier, CycleGan a mis en avant l'utilisation d'un critère lié à la consistance cyclique (*cycle-consistency*), qui consiste à contraindre l'optimisation des réseaux sous l'hypothèse qu'une image $I_A \in A$ convertie du domaine A vers B puis de nouveau vers A devrait être égale à elle-même (et vice versa pour une image $I_B \in B$). Ce principe est illustré par la figure 4.2. Afin d'améliorer la qualité des résultats générés par ces réseaux, Liu *et al.* [LGCL18] ont proposé l'utilisation d'architectures imbriquées, impliquant plusieurs générateurs par translation. Ma *et al.* [MFWCM18] se sont quant à eux intéressés à l'amélioration de la translation d'images représentants des instances d'objets (*i.e.*, un objet dans son contexte) en se basant sur le mécanisme d'attention (*i.e.*, apprendre une carte de caractéristiques complémentaires par multiplication afin de donner plus d'importance à certains éléments dans la scène). De façon similaire aux réseaux de neurones cycliques, l'utilisation de réseaux inter-domaines [LBK17; GGvdWB18] visant à apprendre un espace latent (encodage) commun à deux domaines ; ou plus [YXW18; CCK+18] ; a également été étudiée. Dans nos travaux, nous nous sommes particulièrement intéressés aux réseaux de neurones cycliques non-supervisés [ZPIE17] pour la colorisation. Cette approche présente l'avantage de ne pas nécessiter d'images appariées et ne contraint pas la génération d'un encodage identique pour les images de chaque domaine, ce qui lui permet d'être particulièrement générique. De plus, de nombreuses implémentations sont disponibles pour ce type de méthode dans les *frameworks* d'apprentissage profond les plus utilisés à ce jour (*e.g.*, Pytorch, Tensorflow). Ce dernier point a facilité la mise en place de nos travaux par rapport à l'existant. Nous revenons plus en détail sur ce type de réseau dans la section 4.3.

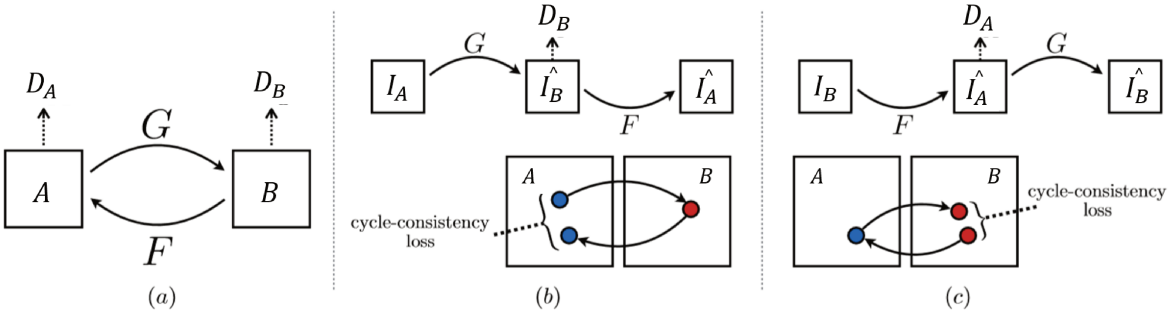


FIGURE 4.2 – Illustration du principe d'un réseau de neurones cyclique non-supervisé basé sur deux GAN (a) exploitant la consistance cyclique (b)(c). Schéma adapté de [ZPIE17].

4.2.3 Approches pour la colorisation

La colorisation est une forme particulière de translation d'image à image, visant à convertir une image en niveaux de gris en une image en couleurs (*i.e.*, on souhaite halluciner les canaux couleur à partir de la texture). Les plus récentes avancées en colorisation ont été réalisées à l'aide de réseaux de neurones profonds entièrement convolutifs.

Approches supervisées

Zhang *et al.* [ZIE16] ont proposé de raffiner (*finetuning*) un réseau de neurones entièrement convolutif afin de générer les canaux couleur AB (de l'espace couleur LAB) à partir de l'image d'intensité (luminance, L) représentée en niveaux de gris. Pour cela, les auteurs ont cherché à quantifier l'espace couleur LAB afin de traiter le problème de la colorisation comme une tâche de classification au pixel près, où chaque classe correspondrait à une valeur de l'espace LAB quantifié. Cette approche basée sur un espace couleur quantifié a été aussi utilisée par Larsson *et al.* [LMS16]. Larsson *et al.* ont également choisi de concaténer les caractéristiques extraites par différentes couches d'un DCNN (VGG-16 pré-entraîné sur ImageNet) pour générer des canaux de teintes et de saturation à partir d'une image en niveaux de gris. Iizuka *et al.* [ISSI16] ont quant à eux présenté une méthode combinant des caractéristiques extraites à plusieurs échelles à partir d'un réseau de

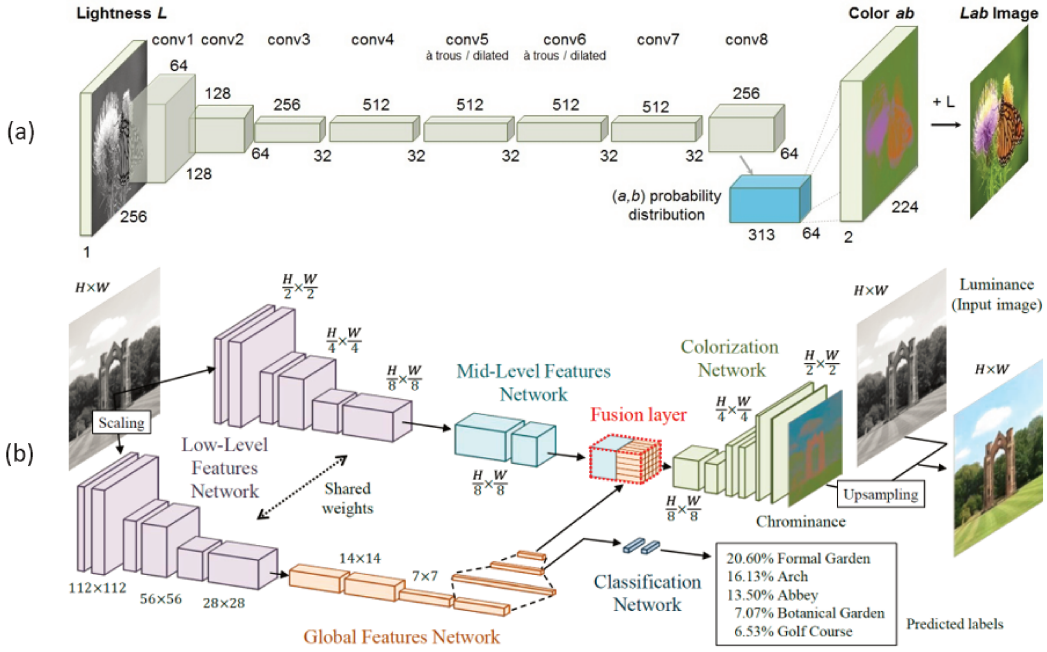


FIGURE 4.3 – Illustration des méthodes de colorisation proposées par (a) Zhang *et al.* [ZIE16] et (b) Iizuka *et al.* [ISSI16].

neurones de type encodeur-décodeur. Ils utilisaient l'encodeur pour générer des caractéristiques pour la colorisation et la classification. Les auteurs indiquent dans leurs travaux que l'intégration jointe de caractéristiques liées à la fois à la classification et à la colorisation permet de guider l'optimisation du réseau en tenant compte de la sémantique contenue dans les images. Cette idée a ensuite été reprise par Vitoria *et al.* [VRB20], où les auteurs ajoutèrent un discriminateur à un réseau quasiment identique à celui de Iizuka *et al.* afin d'exploiter la fonction de coût adversaire, et ainsi améliorer la génération d'images colorisées. Nazeri *et al.* [NNE18] ont quant à eux proposé une approche exploitant la fonction de coût adversaire pour générer des images en couleurs, régularisée par la comparaison de l'image générée et de celle réellement en couleurs. Les méthodes de Zhang *et al.* [ZIE16] et de Iizuka *et al.* [ISSI16], probablement les plus marquantes de ces dernières années, sont illustrées sur la figure 4.3.

Approches hybrides

Nous définissons ici les approches hybrides comme étant des méthodes exploitant des réseaux de neurones profonds combinés à des informations fournies par un utilisateur afin de guider le processus de colorisation. Sangkloy *et al.* [SLF+17] se sont intéressés à l'utilisation d'un réseau de neurones de type encodeur-décodeur combiné à une fonction de coût adversaire et à des traces utilisateurs pour coloriser automatiquement des dessins et des images. Zhang *et al.* [ZZI+17] ont proposé d'intégrer aléatoirement des informations locales (traces utilisateurs) et globales (histogramme couleur) au sein d'un réseau de type encodeur-décodeur afin de guider la colorisation. L'intégration aléatoire de ces informations permet d'envisager plusieurs stratégies de colorisations une fois le réseau entraîné, à savoir : réseau seul, réseau avec informations locales, réseau avec informations globales (histogramme extrait d'une autre image), réseaux avec informations locales et globales. He *et al.* [HDL+18] ont étudié l'intégration d'informations globales issues d'une image de référence en couleurs au travers d'un réseau de neurones supplémentaire. Pour cela, les auteurs ont cherché à calculer des cartes de similarité entre l'image à coloriser et l'image de référence. Cette information est ensuite intégrée en entrée d'un réseau encodeur-décodeur en concaténant la luminance de l'image à coloriser avec les cartes de similarité et les canaux couleur de l'image de référence.

Approches non-supervisées

Les approches présentées ci-dessus sont principalement supervisées, au sens où chaque image en niveau de gris est appariée à une image en couleurs. Cela permet d'introduire des contraintes fortes quant aux couleurs que l'on souhaite générer, mais rend difficile l'intégration d'images uniquement disponibles en niveaux de gris. Malgré tout, il semblerait que les approches supervisées donnent des résultats satisfaisants sur des images historiques très proches des jeux de données sur lesquelles elles ont été entraînées (voir [ZIE16; ISSI16]). Des chercheurs se sont néanmoins intéressés à l'utilisation d'approches non-supervisées pour la colorisation afin de pouvoir entraîner des réseaux de neurones à convolutions directement à partir d'images non appariées. C'est par exemple le cas de Cao *et al.* [CZZY17], qui ont proposé de guider l'entraînement d'un réseau générateur de type encodeur-décodeur à l'aide d'un réseau discriminateur. En plus d'une GAN *loss*, les auteurs ont suggéré de contraindre la génération d'images colorisées spatialement réalistes en convertissant les images générées vers une représentation en niveaux de gris et en les comparant aux images initiales. Dans nos travaux, nous avons repris cette contrainte en la formulant dans un réseau de neurones cyclique (voir section 4.4).

4.3 Vers une colorisation automatique des images aériennes historiques

Nous présentons ici nos travaux sur la colorisation automatique des images aériennes historiques. Notre but était d'évaluer l'intérêt des réseaux de neurones cycliques pour coloriser de façon non-supervisée ces données. Il s'agit, à notre connaissance, des premiers travaux réalisés sur ce sujet. Notre approche est décrite dans cette section et résumée sur la figure 4.4 : (1) création d'un jeu de données constitué d'imagettes extraites d'images aériennes historiques et d'images aériennes récentes, (2) entraînement d'un réseau de neurones générateur adversaire cyclique, (3) colorisation à l'aide du réseau entraîné, et (4) reconstruction des images aériennes et remplacement de la texture générée par le réseau (flèches sur la figure 4.4). Sur cette figure, les images en niveaux de gris et les images en couleurs n'ont pas été appariées (acquisitions à différentes dates et coordonnées géographiques). On remarquera également que nous avons choisi de travailler par imagettes de 1024 × 1024 pixels afin de tenir compte des contraintes matérielles liées à l'utilisation de DCNN (mémoire disponible lors de l'inférence). Nous détaillons les données que nous avons utilisées en section 4.3.3.

Outre les données, la méthode de colorisation d'images aériennes historiques développée est formée de deux composants principaux. Le premier est Col-Cycle, un réseau de neurones à convo-

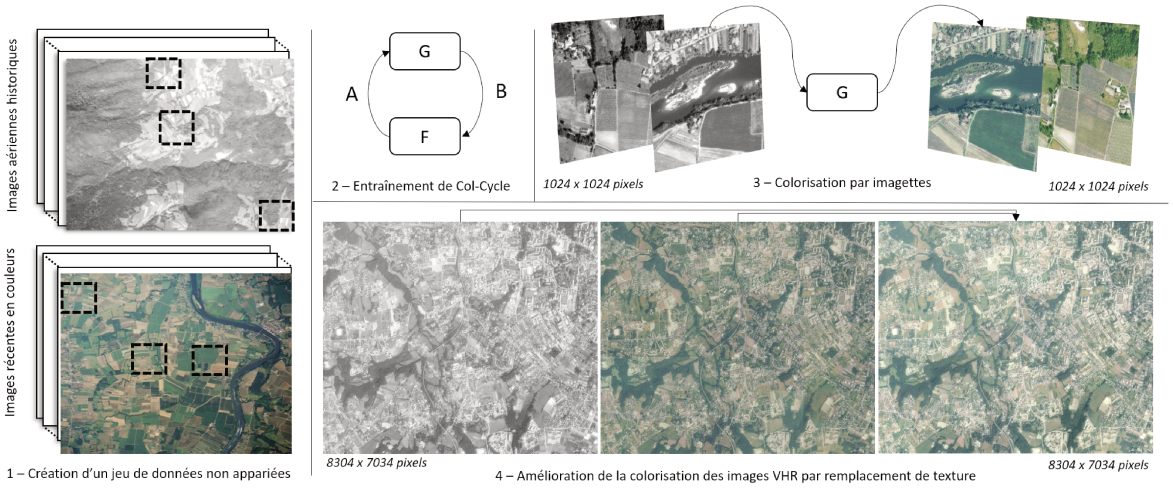


FIGURE 4.4 – Méthode de colorisation non-supervisée d'images aériennes historiques proposée.

lutions basé sur des représentations cycliques consistantes, similaire à CycleGan [ZPIE17]. Le second est une approche simple mais efficace de remplacement de textures, que nous avons utilisée afin d'améliorer la visualisation des images de très hautes résolutions constituées d'une mosaïque d'imagettes colorisées indépendamment les unes des autres. Nous décrivons ces deux composants ci-après.

4.3.1 Col-Cycle

Description du réseau

Col-Cycle est un réseau de neurones entièrement convolutif. Il est directement inspiré de CycleGan [ZPIE17]. Comme ce dernier, Col-Cycle est basé sur deux GAN qui vont chercher à collaborer afin de générer des images réalistes entre deux domaines (niveaux de gris et couleur). Les deux GAN ont la même architecture (hyperparamètres identiques), mais leurs paramètres (poids) ne sont pas partagés.

Soit A le domaine des images en niveaux de gris, et B le domaine des images en couleurs. Dans notre cas, les images en couleurs sont représentées dans l'espace couleur RVB afin d'outrepasser l'absence de relations linéaires entre les canaux couleur et l'intensité dans l'espace LAB, classiquement utilisé en colorisation supervisée. Nous supposons en effet qu'il est plus aisé d'apprendre une translation linéaire de B vers A qu'une translation non linéaire. Nous définissons les deux GAN chargés de réaliser la translation de A vers B et la translation de B vers A de la manière suivante : $GAN_{A \rightarrow B} = \{G, D_B\}$ et $GAN_{B \rightarrow A} = \{F, D_A\}$. Ici, G et F sont les réseaux générateurs, et D_A et D_B sont les réseaux discriminateurs associés aux images des domaines A et B respectivement.

L'architecture des réseaux générateurs G et F de Col-Cycle prend une forme encodeur-décodeur (voir figure 4.5). Ils possèdent une couche de convolutions dite d'entrée, qui va être chargée de transformer l'image en une représentation intermédiaire. Celle-ci préserve la taille de l'image. Cette couche d'entrée est suivie de deux couches de sous-échantillonnages, qui vont permettre d'encoder l'information en augmentant le nombre de caractéristiques. Viennent ensuite 3 couches résiduelles, telles que définies dans le chapitre précédent (voir ResNet). Les couches résiduelles vont encoder l'information sous-échantillonnée en préservant les informations nécessaires pour le décodage. Elles vont également permettre de rétropropager le gradient plus en amont dans le réseau, ce qui devrait *a priori* améliorer l'optimisation de l'encodeur. Les couches résiduelles sont suivies de deux couches de sur-échantillonnage, et d'une couche de sortie qui va convertir la représentation profonde en une image du domaine cible. Dans notre cas, la couche d'entrée est composée de 64 filtres de 7×7 pixels. Elle génère ainsi une représentation intermédiaire de 64 canaux, 1 par filtre. La couche de sortie génère des images avec 3 canaux, en utilisant des filtres de 7×7 pixels. Toutes les autres couches sont constituées de filtres de 3×3 pixels. Tous les filtres sont appliqués avec du zéro *padding* (*i.e.*, ajout de pixels à 0 au bord de l'image) afin de préserver la taille des images qu'ils prennent en entrée. Le sous-échantillonnage est ici réalisé à l'aide de la valeur du pas (*stride value* égale à 2) des filtres de convolutions. Les couches de sur-échantillonnage vont réaliser l'opération opposée des couches de sous-échantillonnage. En pratique, nous avons choisi d'utiliser un sur-échantillonnage classique (*e.g.*, interpolation bilinéaire) suivi de filtres de convolutions à la place de convolutions transposées afin de limiter les artefacts visuels de type damier (*checkerboard artifacts*) [ODO16]. Des opérations de normalisation par instance (instance norm, IN) sont par ailleurs utilisées afin d'améliorer la qualité des images générées [UVL16].

Le discriminateur est un réseau de neurones entièrement convolutif dont l'architecture est décrite sur la Figure 4.5. Seule particularité de ce réseau : un *pooling* global (*i.e.*, dont la taille est égale à l'image qu'il prend en entrée) est appliqué sur la dernière couche de caractéristiques afin de pouvoir obtenir une valeur unique indiquant si l'image est réelle ou fausse.

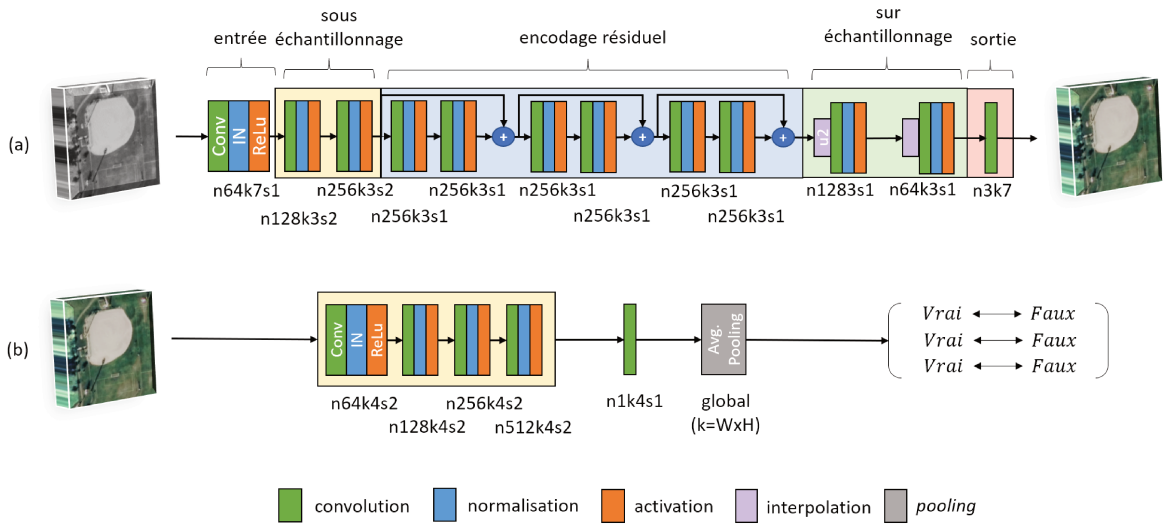


FIGURE 4.5 – Architecture des générateurs et des discriminateurs utilisés dans Col-Cycle. (a) Générateurs F et G. (b) Discriminateurs D_A et D_B. Les poids de chaque réseau ne sont pas partagés. Le paramètre n indique le nombre de filtres, k la taille du filtre, et s la valeur de pas (*stride*).

Fonctions de coûts

Les fonctions de coûts utilisées ici sont similaires à celles utilisées par [ZPIE17].

GAN *losses*. Le générateur et le discriminateur de chaque GAN sont entraînés à l'aide de la GAN *loss* (voir équation (4.1)), que nous reformulons ici sous forme quadratique. Cette reformulation est inspirée des travaux de [MLX+17]. L'équation (4.2) est associée au $GAN_{A\rightarrow B}$, et l'équation (4.3) est associée au $GAN_{B\rightarrow A}$. Cette formulation, dite *minimax*, reprend celle de de l'équation 4.1.

$$\mathscr{L}_{GAN_{A\rightarrow B}} = \mathbb{E}_{\mathbb{B}}[\|D_B(I_B)\|_2^2] + \mathbb{E}_{\mathbb{A}}[\|1 - D_B(G(I_A))\|_2^2] \tag{4.2}$$

$$\mathscr{L}_{GAN_{B\rightarrow A}} = \mathbb{E}_{\mathbb{A}}[\|D_A(I_A)\|_2^2] + \mathbb{E}_{\mathbb{B}}[\|1 - D_A(F(I_B)\|_2^2] \tag{4.3}$$

En pratique, la GAN *loss* que nous avons re-définie ci dessus est implémentée en cherchant à minimiser des fonctions de coûts différentes pour le générateur et le discriminateur [MLX+17]. Pour la minimisation, on considère les images réelles étiquetées par un 1, et les images générée étiquetées par un 0. L'optimal pour le générateur G est alors obtenu en minimisant l'équation (4.4), et celui pour le discriminateur D_B en minimisant l'équation (4.5). Ces équations sont aussi valables pour $GAN_{B\rightarrow A}$ en intervertissant les indices A et B, et en remplaçant G par F. A noter que l'objectif pour le discriminateur est alors un minimum, et non un maximum comme représenté par la fonction *minimax*. Par abus de langage, on pourrait ici parler d'une fonction de coût "duale" pour le discriminateur.

$$\mathscr{L}_G = \mathbb{E}_{\mathbb{A}}[\|D_B(G(I_A)) - 1\|_2^2] \tag{4.4}$$

$$\mathscr{L}_{D_B} = \mathbb{E}_{\mathbb{B}}[\|D_B(I_B) - 1\|_2^2] + \mathbb{E}_{\mathbb{A}}[\|D_B(G(I_A)\|_2^2] \tag{4.5}$$

Cycle-consistency loss. Nous exploitons également la fonction de coût liée à la consistance cyclique afin de contraindre le réseau à générer des images dont la représentation dans le domaine cible se rapproche des images réelles [ZPIE17]. Les réseaux générateurs sont en effet capable de générer plusieurs images différentes qui permettront de satisfaire les fonctions de coût de type GAN. Afin d'ajouter une contrainte supplémentaire qui va guider les représentations que l'on veut obtenir, nous allons considérer le fait que l'image translatée du domaine A vers B puis de nouveau vers A devrait être égale à elle-même. Cela permet d'utiliser directement les images de chaque domaine afin de guider l'entraînement de chacun des réseaux générateurs, et ce sans avoir d'images appariées. On remarquera ici que les fonctions G(F(.)) et F(G(.)) forment alors des auto-encodeurs.

Pour cela, la fonction de coût liée à la consistance cyclique est définie à l'aide de la norme L1 (voir équation (4.6)). Les auteurs de [ZPIE17] ont également essayé d'utiliser la norme L2 pour cette fonction de coût, sans observer d'amélioration particulière.

$$\mathscr{L}_{cycle} = \mathbb{E}_{\mathbb{B}}[||G(F(I_B)) - I_B||_1] \quad + \quad \mathbb{E}_{\mathbb{A}}[||F(G(I_A)) - I_A||_1] \tag{4.6}$$

Identity loss. Enfin, nous utilisons la fonction de coût identité afin d'aider à préserver les informations liées au domaine cible. Pour cela, les réseaux G et F doivent générer des images proches de la réalité lorsqu'ils translatent des images du domaine cible, vers le même domaine cible (*e.g.*, avec J ∈ B, on cherche à obtenir G(J) = J). Cette fonction de coût est définie par l'équation (4.7). En pratique, cette fonction permet de réduire les cas où une seule couleur prédominante serait prédite.

$$\mathscr{L}_{identity} = \mathbb{E}_{\mathbb{B}}[||G(I_B) - I_B||_1] \quad + \quad \mathbb{E}_{\mathbb{A}}[||F(I_A) - I_A||_1] \tag{4.7}$$

La fonction de coût totale est alors définie comme une somme des fonctions de coût ci-dessus (voir equation (4.8).

$$\mathscr{L} = \mathscr{L}_{GAN_{A\rightarrow B}} + \mathscr{L}_{GAN_{B\rightarrow A}} + \mathscr{L}_{cycle} + \mathscr{L}_{identity} \tag{4.8}$$

Différences avec CycleGan

Les différences entre Col-Cycle et CycleGan sont ici mineures d'un point de vue conceptuel. Elles résident principalement dans la quantité de couches résiduelles utilisées. Nous avons fait le choix d'en utiliser 3, contre 9 pour CycleGan. Cela nous permet de réduire significativement la quantité de paramètres à optimiser, mais également de réduire le nombre de cartes de caractéristiques intermédiaires stockées en mémoire sur les cartes graphiques lors de l'inférence. De fait, nous avons pu travailler avec des imagettes de tailles relativement grandes par rapport au réseau originel (1024 × 1024 contre 256 × 256). Travailler avec des images plus grandes lors de l'inférence est utile pour diminuer l'effet mosaïque observé lors de la "reconstruction" des images aériennes historiques colorisées (voir section suivante).

4.3.2 Reconstruction des images colorisées

La seconde étape de notre approche vise à reconstituer les images de très hautes résolutions (VHR) à l'aide des imagettes colorisées avec Col-Cycle. Des exemples de résultats sont proposés ici : `http://eidolon.univ-lyon2.fr/~remi1/Col-Cycle-Res/`.

Reconstitution et effet mosaïque

Nous commençons par extraire toutes les imagettes de taille 1024 × 1024 pixels sans recouvrement à partir des images VHR, et nous stockons les coordonnées correspondantes. Nous utilisons ensuite Col-Cycle pour coloriser chacune de ces imagettes indépendamment les unes des autres. Enfin, nous reconstituons l'image VHR en concaténant spatialement les imagettes colorisées à l'aide de leurs coordonnées initiales. Ce processus est particulièrement simple à implémenter et efficace d'un point de vue computationnel (le fait de ne pas avoir de recouvrement entre les imagettes évite la redondance lors des traitements).

Cependant, nous pouvons observer sur la figure 4.6 que les imagettes colorisées puis concaténées semblent produire, par endroits, des discontinuités locales faisant ressortir la structure des imagettes dans l'image. Nous nommons cet effet non désiré "effet mosaïque", par analogie avec les mosaïques d'images. En télédétection, de telles mosaïques apparaissent régulièrement, à plus grande échelle, lors de la visualisation d'images aériennes et satellites acquises à des dates différentes.

Les convolutions étant par définition invariantes à la translation spatiale dans l'image, nous pouvons supposer que cet effet mosaïque est lié à l'utilisation d'opérations de normalisation par

instance dans le réseau. Celles-ci vont modifier l'échelle des valeurs possibles pour chaque imagette par rapport à elle-même, sans tenir compte des imagettes qui lui sont adjacentes. Retirer les opérations de normalisation n'est cependant pas désiré : leur ajout dans la littérature avait été suggéré afin d'améliorer l'optimisation des réseaux de neurones profonds à convolutions (voir chapitre 2). Nous remarquons néanmoins que l'effet mosaïque que nous observons peut-être induit par les couleurs générées, mais aussi par la luminosité des pixels (*i.e.*, la texture en niveaux de gris) qui est elle-même modifiée par le réseau (l'espace couleur cible étant ici le RVB). En effet, malgré la contrainte introduite par la fonction de coût cyclique, il est possible que le réseau ait inventé des structures qui n'existent pas dans les images initiales en niveaux de gris.

Remplacement de textures

Afin de trouver une solution efficace au problème posé par l'effet mosaïque, nous proposons de séparer les composants liés à la texture et à la couleur des images RVB générées avec Col-Cycle. Pour cela, nous convertissons les images VHR colorisées dans l'espace LAB. Observons que les canaux représentant la couleur dans l'espace LAB (canaux A et B) ont une représentation spatiale plus lisse (moins de hautes fréquences) que la texture (canal de luminance L), rendant la texture plus prompte à représenter des hautes fréquences non désirées. Cette observation est à la base d'algorithmes d'encodage tels que le JPEG, où les canaux couleur sont sous-échantillonnés par rapport à l'information de luminance. Dans notre cas, nous proposons de remplacer le canal de luminance généré par Col-Cycle à l'aide de l'image VHR initiale en niveaux de gris (*i.e.*, $\mathrm{L} := \mathrm{I}_{\mathrm{A}}^{\mathrm{VHR}}$), avant de reformer l'image LAB en concaténant les canaux correspondants. L'image LAB est ensuite reconvertie dans l'espace RVB à des fins de visualisation. Par abus de langage, nous avons nommé cette approche "remplacement de textures" (*texture replacement*). Cette idée est inspirée par les travaux de colorisation supervisée qui visent à générer directement les canaux AB de l'espace LAB, en considérant l'image en niveaux de gris I_{A} comme étant la luminance.

On observe sur la figure 4.6 que le remplacement de textures permet effectivement d'améliorer

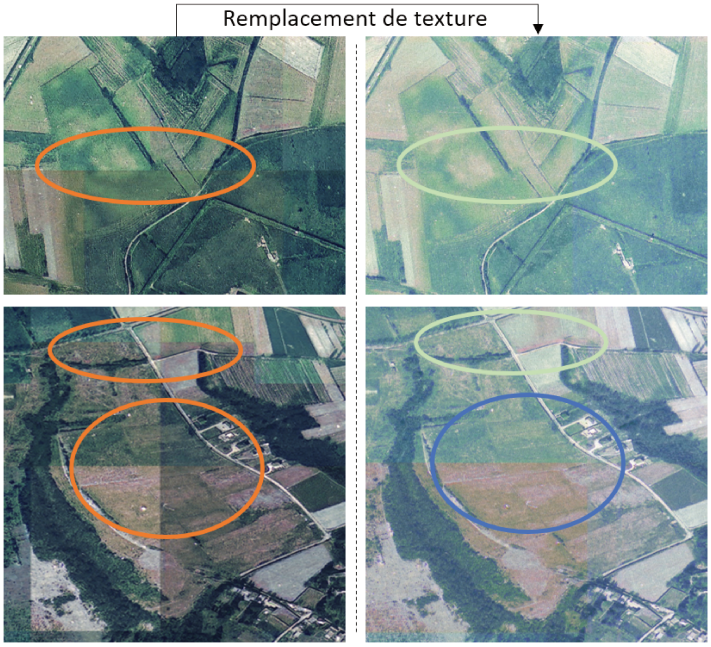


FIGURE 4.6 – Résultats visuels mettant en avant l'effet mosaïque (ellipsoïdes oranges). Avant (à gauche) et après (à droite) le remplacement de textures. Les ellipsoïdes vertes indiquent les corrections réalisées. Les ellipsoïdes bleues mettent en avant la contribution des composants couleur à l'effet mosaïque.

la qualité de la visualisation des images VHR générées en réduisant l'effet mosaïque (ellipsoïde vertes sur la figure 4.6). Cependant, cette approche seule ne permet pas de supprimer l'intégralité des incohérences de couleur visibles (ellipsoïde bleues sur la figure 4.6). L'utilisation d'algorithmes de transferts de couleur (*e.g.*, mise en correspondance d'histogrammes) inter-imagettes est une piste possible à l'amélioration des résultats obtenus, mais il faudrait alors choisir quelle serait l'imagette source dont nous souhaiterions préserver la couleur.

4.3.3 Données et entraînement

Données

Nous avons travaillé avec 10 images aériennes en couleurs de très hautes résolutions acquises en France, et avec les 81 images aériennes présentent au sein de HistAerial (voir chapitre 3). Ces images ont été découpées en imagettes de 1024 × 1024 pixels pour un total de 1702 imagettes en couleurs (recouvrement de 50% entre deux imagettes) et de 572 imagettes en niveaux de gris (pas de recouvrement). Des exemples d'imagettes utilisées pour l'entraînement de Col-Cycle sont présentées sur la figure 4.7. On constate la diversité des représentations de couleur possibles.

Entraînement de Col-Cycle

A l'aide de ces données, nous avons entraîné Col-Cycle durant 200 *epochs* via la librairie Pytorch (version 0.4) et deux cartes graphiques NVidia GTX 1080 Ti. Le taux d'apprentissage a été fixé à 0.0002, avec une taille de *batch* de 2 (1 image par GPU avec un découpage aléatoire d'aires de 512 × 512 pixels avec retournements verticaux et horizontaux aléatoires pour "augmenter" les données lors de l'entraînement - ces augmentations n'ont pas lieu lors de l'inférence). Après 100 *epochs* d'entraînement, nous diminuons linéairement le taux d'apprentissage vers zéro afin d'aider à la convergence du réseau. Le remplacement de textures n'est pas utilisé lors de l'entraînement, mais uniquement lors de l'inférence.

4.3.4 Résultats et discussions

Nous avons mis en place deux expériences pour évaluer l'intérêt de la colorisation pour l'analyse des images aériennes historiques.

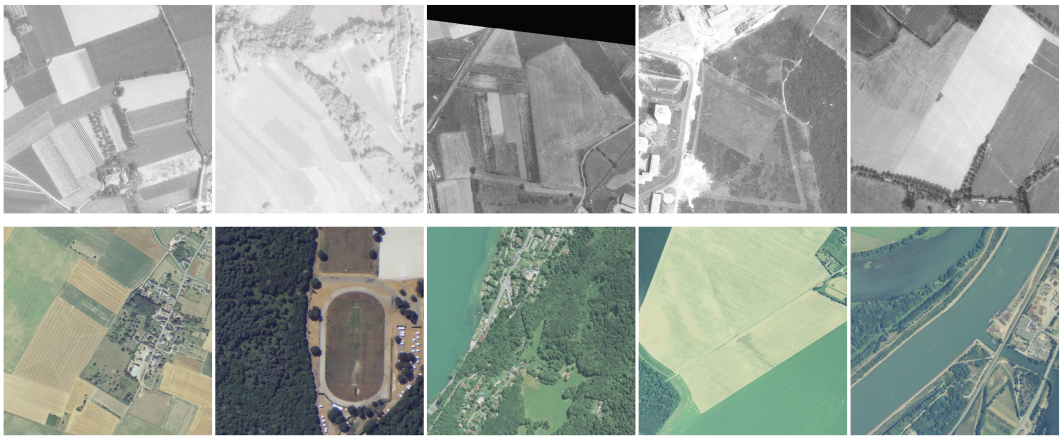


FIGURE 4.7 – Exemples d'imagettes utilisées pour l'entraînement de Col-Cycle.

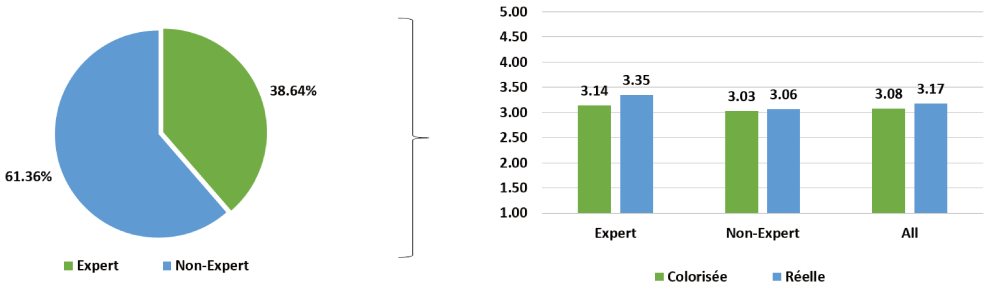


FIGURE 4.8 – Résultats de l'évaluation par note moyenne d'opinions visant à déterminer la qualité de la couleur des images générées. Plus la valeur est élevée, meilleure est la qualité.

Évaluation par note moyenne d'opinions

Nous avons réalisé une évaluation par note moyenne d'opinions afin d'évaluer la qualité des imagettes colorisées après 60 *epochs* d'entraînement de Col-Cycle. Les études d'opinions sont régulièrement utilisées lorsque l'évaluation d'une quantité est subjective. Dans notre cas, nous cherchons à faire évaluer par des êtres humains la qualité de la couleur d'images générées. Le choix de prendre les images générées après 60 *epochs* a été fait de façon arbitraire.

Afin de réaliser cette évaluation, nous avons élaboré un questionnaire et demandé à des personnes anonymes d'y répondre sur la base du volontariat (répondants). Le questionnaire avait pour but de donner un score subjectif à la qualité de la couleur de 50 imagettes de 1024×1024 pixels sélectionnées aléatoirement. Parmi ces imagettes, 15 étaient des images réelles, et 35 étaient des images colorisées. Les répondants avaient connaissance de ce fait, mais ne savaient pas quelles étaient les images réelles et les images colorisées à l'aide de Col-Cycle. Pour chaque image, chaque répondant devait fournir un score de qualité entre 1 et 5, la note la plus élevée étant la meilleure. Il n'était pas demandé aux répondants de classer les images selon leur type (image réelle ou colorisée). Les répondants devaient également indiquer s'ils avaient de l'expérience quant à l'utilisation d'images aériennes ou satellites dans leur activité professionnelle. Les répondants avec une telle expérience ont été catégorisés en tant que "experts" (*i.e.*, individus ayant l'habitude des données visualisées), et les autres en tant que "non-experts".

Un total de 44 répondants (à date du 06/2020) a rempli notre questionnaire, dont 38.64% d'experts et 61.36% de non-experts. Les notes moyennes d'opinions obtenues sont résumées sur la Figure 4.8. D'un point de vue global, les images colorisées ont obtenus des notes presque égales aux images réellement en couleurs, que ce soit pour les experts ou les non-experts. Une légère différence en faveur des images en couleurs est cependant visible (0.1 point). Au final, nous avons conclu de ce questionnaire que les images colorisées proposées semblent relativement réalistes.

Nous pouvons envisager l'utilisation de Col-Cycle afin de proposer une visualisation alternative aux images aériennes historiques panchromatiques. Cette visualisation devrait permettre de simplifier la tâche d'annotation dans le cadre de TESTIS. Les images en couleurs sont en effet considérées comme étant plus faciles à interpréter que les images en niveaux de gris. Cette approche devrait également être utile afin de proposer des visualisations intéressantes du territoire dans le passé, dans un but d'archivage et de compréhension de l'environnement.

4.3.5 Application à la classification

Afin d'évaluer l'apport de la colorisation pour la classification des images aériennes historiques, nous avons appliqué Col-Cycle avec remplacement de textures à l'ensemble des images

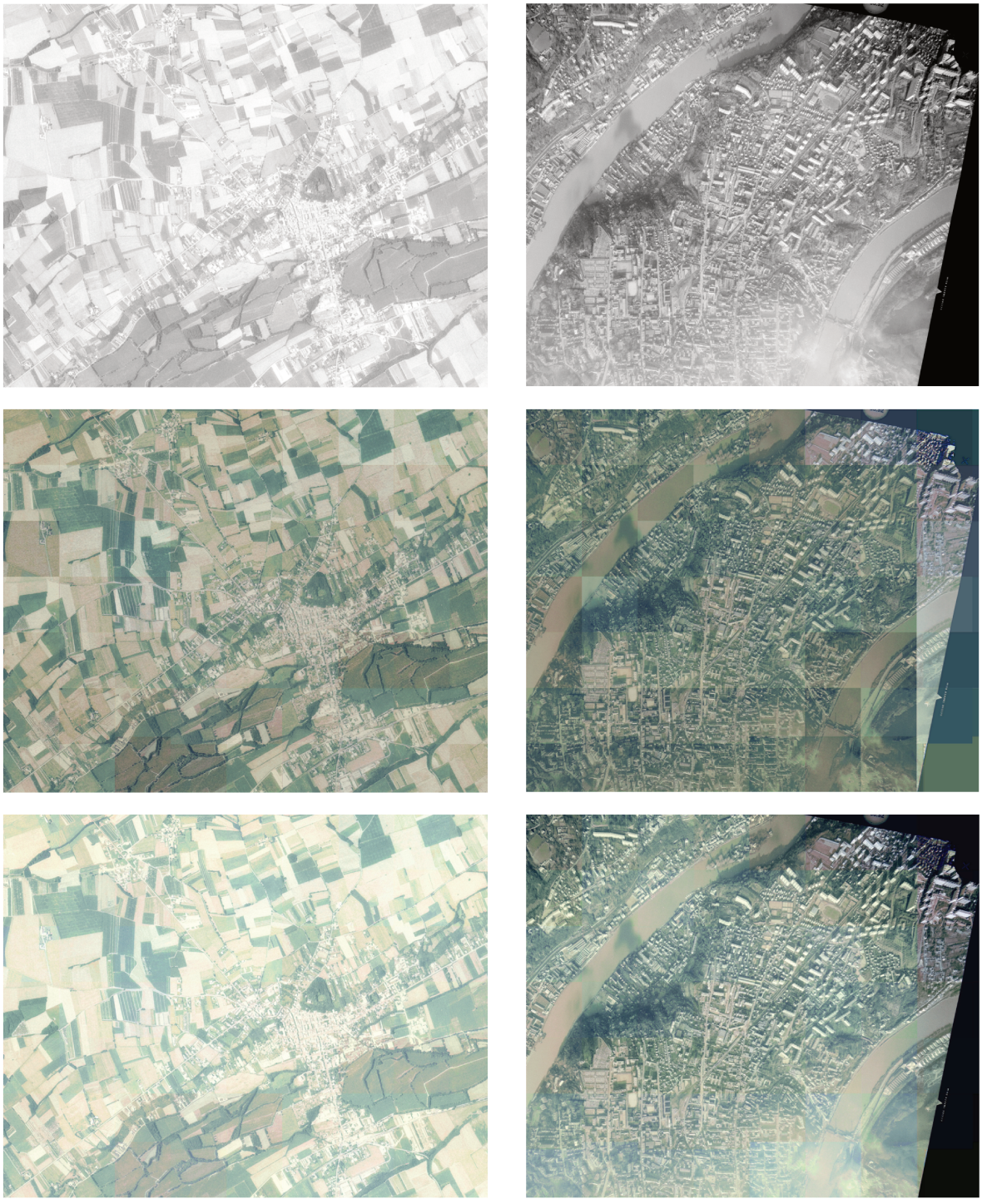


FIGURE 4.9 – Exemples d'images aériennes VHR colorisée avec Col-Cycle. En haut, images panchromatiques. Au milieu, résultats avant remplacement de textures. En bas, résultats après remplacement de textures. La bande noire à droite correspond aux bordure des images aériennes historiques scannées. Tailles originelles ≈ 8800 × 7000 pixels et 8300 × 7000 pixels. Nous invitons le lecteur à zoomer sur la version électronique pour mieux percevoir les différences.

aériennes VHR de HistAerial. Des exemples de telles colorisations sont présentés sur la figure 4.9. Nous avons ensuite régénéré le sous-ensemble de données composé d'imagettes de 100 × 100 pixels pour les 7 classes d'occupation du sol, résultant une version colorisée de ces imagettes. Pour chaque imagette considérée dans l'espace couleur LAB, nous avons extrait les caractéristiques de

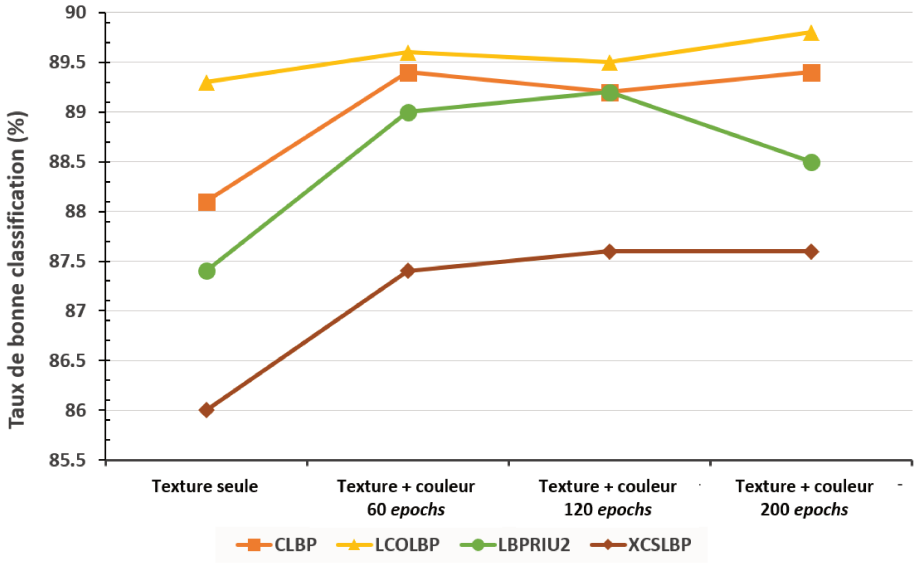


FIGURE 4.10 – Taux de bonne classification sur le jeu de données HistAerial (7 classes d'occupation du sol) à l'aide de filtres de textures et de statistiques couleur. Les résultats sont présentés pour des images colorisées avec Col-Cycle considéré à différentes *epochs*.

textures à l'aide de filtres de textures sur le canal de luminance (L, qui est ici égal à l'image en niveaux de gris initiale), et nous avons extrait des statistiques représentatives de la couleur à partir des canaux AB. Les filtres de textures considérés ici sont le LCoLBP, le CLBP, le XCSLBP et le LBP avec *mapping* riu^2, tels que décrits dans le chapitre 3. Pour les statistiques couleur, nous avons calculé la moyenne et la déviation standard des valeurs des canaux A et B séparément. Nous avons également calculé les histogrammes de ces canaux, à partir desquels nous avons calculé 4 statistiques (aplatissement, asymétrie, variance, maximum). Au total, nous avons donc ajouté 12 caractéristiques de couleur aux caractéristiques de texture. A l'aide de ces vecteurs de caractéristiques, nous avons entraîné une forêt aléatoire d'arbres décisionnels dont les paramètres ont été obtenus en suivant le processus décrit dans le chapitre 3. Le choix d'utiliser des filtres de textures combiné à des statistiques liées à la couleur nous permet de déterminer explicitement l'apport des couleurs générées par rapport aux caractéristiques extraites des filtres de type LBP.

Les résultats obtenus sont présentés sur la figure 4.10 en considérant les images colorisées à différentes *epochs* de l'entraînement de Col-Cycle. Nous pouvons observer que l'ajout de caractéristiques extraites des canaux couleur générés tend à améliorer les taux de bonne classification sur HistAerial pour l'ensemble des filtres de textures considérés, avec un gain de 1.3% en moyenne. Nous observons également qu'entraîner le réseau plus longtemps ne génère pas nécessairement de gain très important en classification. Ce phénomène peut s'expliquer par le manque de contrôle que nous avons sur l'entraînement de réseau de neurones de type GAN. Par opposition aux réseaux de neurones classiques, utilisés pour la classification par exemple, il est difficile de déterminer si (et quand) les paramètres du réseau ont atteint un état optimal ou non. Ce point s'explique également par l'absence de contraintes explicites quant à la classification, que nous aurions pu inclure afin de guider le réseau vers la génération d'un domaine couleur adapté à cette tâche. Ici, nous avons plutôt cherché à évaluer l'intérêt implicite de la colorisation pour la classification des images aériennes historiques.

4.3.6 Conclusion partielle

Nous avons présenté une approche non-supervisée pour la colorisation automatique d'images aériennes historiques de très hautes résolutions. Les colorisations générées après 60 *epochs* d'en-

traînement ont été jugées correctes (note moyenne d'opinions) par deux groupes d'humains composés de personnes ayant déjà travaillé avec des données aériennes et satellites, et des personnes inexpérimentées. Cela nous permet d'envisager l'utilisation de la colorisation pour aider les géomaticiens à annoter les images aériennes historiques, les images en couleurs étant communément jugées plus faciles à interpréter que les images panchromatiques. Nous avons également montré que les couleurs générées à différentes *epochs* permettaient d'améliorer légèrement les taux de bonne classification par rapport à l'utilisation de la texture seule, ce qui laisse envisager l'intégration d'une étape de colorisation préalable à la classification. Nous avons cependant remarqué l'apparition d'un effet mosaïque sur les images VHR colorisées que nous avons supposé lié à l'utilisation d'opérations de normalisations au sein du réseau. Afin de lutter contre cet effet indésirable, nous avons proposé une approche simple et efficace dite de remplacement de textures. Cependant, certaines aberrations liées à la couleur persistent, et un traitement automatique *a posteriori* ne nous parait pas trivial. Il serait par exemple possible de considérer des imagettes avec recouvrement durant l'entraînement, et de contraindre la représentation des histogrammes couleur calculés sur les zones qui se superposent.

4.4 Vers une amélioration de la colorisation

Encouragés par nos travaux sur la colorisation non-supervisée des images aériennes historiques, nous nous sommes intéressés au développement d'un nouveau réseau de neurones profond à convolutions afin d'obtenir des colorisations encore plus proches des images réelles. Pour cela, nous avons proposé une architecture dite pseudo-cyclique, basée sur des *a priori* empiriques. Ces *a priori* ont été explicitement introduits sous la forme d'une translation "artisanale" (*Handcrafted Translation*, H_t). Nous avons également utilisé une pyramide spatiale de sortie (*Output Spatial Pyramids*, OSP) afin de contraindre la génération de caractéristiques à plusieurs échelles. Nous avons observé expérimentalement que la *Handcrafted Translation* pouvait être une solution viable pour supprimer l'un des deux GAN utilisés par les approches cycliques telles que Col-Cycle ou CycleGan (voir figure 4.11). Nous présentons ces éléments dans les paragraphes suivants.

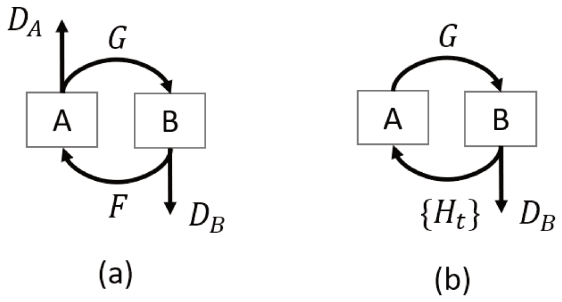


FIGURE 4.11 – Schéma d'un réseau de neurones cyclique [ZPIE17] (a) et d'un réseau de neurones pseudo-cyclique. G et F sont des réseaux générateurs, D_A et D_B des réseaux discriminateurs, et $\{H_t\}$ représente une *Handcrafted Translation* du domaine B vers A. On remarquera l'absence de D_A sur la sous-figure (b).

4.4.1 Blocs de base

Handcrafted Translation

Le terme de *Handcrafted Translation* est ici utilisé dans le contexte de translation d'image à image, auquel appartiennent les méthodes que nous utilisons. Par opposition aux translations apprises à l'aide d'un réseau de neurones à convolutions, les *Handcrafted Translation* sont définies à l'aide d'un *a priori* que nous avons sur le problème à résoudre (*i.e.*, ce sont des filtres "classiques").

Dans un contexte de colorisation, nous rappelons que notre but est d'optimiser un réseau générateur G qui va générer une image en couleurs $I_B \in B = \mathbb{R}^{3\times W\times H}$ à partir d'une image en niveaux de gris $I_A \in A = \mathbb{R}^{1\times W\times H}$. Nous définissons alors la *Handcrafted Translation* H_t comme étant une fonction capable d'effectuer la translation inverse, de B vers A. Afin d'étudier l'intérêt d'une H_t pour contraindre la colorisation d'images à l'aide d'un réseau générateur, et non pas l'inverse, nous avons souhaité utiliser une fonction H_t aussi simple que possible. Nous avons pour cela utilisé l'une des premières représentations de l'intensité en niveaux de gris : la somme pondérée des canaux RVB. Pour un pixel $x^{i,j}$ positionné sur la i^{th} ligne et la j^{th} colonne d'une image numérique $I \in \mathbb{R}^{3\times W\times H}$, l'opération correspondant à H_t est alors exprimée à l'aide de l'équation (4.9). Dans cette équation, les poids ont été fixés afin de mimer la vision biologique humaine, plus sensible aux teintes vertes, que rouges, que bleu. A noter que H_t est alors définie comme la fonction $Gray$ utilisée par Cao *et al.* [CZZY17].

$$x^{i,j}_{gris} = 0.299 \times x^{i,j}_{R} + 0.587 \times x^{i,j}_{V} + 0.114 \times x^{i,j}_{B} \tag{4.9}$$

Comme cette fonction représente une somme pondérée des canaux RVB avec des poids constants, elle peut être facilement implémentée à l'aide d'une convolution 1×1 dont les poids sont fixés. L'utilisation d'une convolution 1×1 présente comme avantage de préserver la majorité des structures spatiales présentent au sein des images en entrée, telles les formes, les contours ou les textures. Elle permet également l'intégration de cette fonction au sein des librairies d'apprentissage profond déjà en place, ce qui permet de rétropropager le gradient via H_t (*cycle-consistency*). Par conséquent, formuler la translation H_t à l'aide d'une convolution 1×1 permet de directement contraindre les propriétés spatiales des images couleur générées par G. En pratique, H_t a pour but de remplacer $GAN_{B\rightarrow A} = (F, D_A)$: F est directement remplacé par H_t, et D_A ne devient plus nécessaire étant donné que H_t est une fonction déterministe (*i.e.*, si $G(I_A)$ est correctement colorisée, H_t donnera un résultat proche de celui espéré). Le terme pseudo-cyclique se comprend ici par le remplacement de l'un des deux GAN par une transformation fixée (le cycle existe bien, mais seule la moitié de celui-ci est apprise).

L'opération H_t définie à l'aide d'un filtre 1×1 est par ailleurs à opposer aux translations apprises à l'aide de filtres de convolutions, qui ne permettent pas de garantir la préservation des propriétés spatiales et des hautes fréquences des images traitées. En effet, d'une part, les convolutions spatiales tendent à lisser les images [UVL18]. D'autre part, les réseaux générateurs cycliques (*e.g.*, G et F avec Col-Cycle et CycleGan) peuvent apprendre à satisfaire un critère d'optimisation sans chercher à préserver les structures spatiales entre les domaines concernés par la translation, hallucinant alors des structures qui n'existent pas [IZZE17]. Le fait de pouvoir modifier les structures spatiales est une propriété particulièrement intéressante pour des applications telles que le débruitage [XXC12], la segmentation sémantique [BKC17], ou la modification d'objets [RRVB17], mais elle n'est pas désirée lorsque l'on souhaite que l'image dans le domaine cible partage ses hautes fréquences avec l'image dans le domaine source (cas de la colorisation).

Enfin, l'utilisation de H_t, telle que définie ci-dessus contraint la génération d'images dans l'espace couleur RVB, et ce malgré le fait que plusieurs études ont montré l'intérêt des espaces couleur LAB et HCL afin de découpler luminance, teinte et intensité [LMS16; ISSI16]. Nous rappelons que le choix de travailler avec l'espace RVB a été fait afin de nous assurer de l'existence d'une translation linéaire entre l'espace couleur cible (RVB) et celui des intensités (niveaux de gris), ce qui n'est pas possible avec l'espace LAB (non-linéarité entre AB et L).

Output Spatial Pyramids

Afin de tenter d'améliorer les résultats que l'on peut espérer obtenir en colorisation à l'aide d'un réseau générateur, nous nous sommes également intéressés à l'utilisation de représentations multi-échelles. Ces représentations sont également nommées pyramides spatiales dans la littérature. Elles consistent soit à considérer un ensemble d'images représentant le même contenu mais

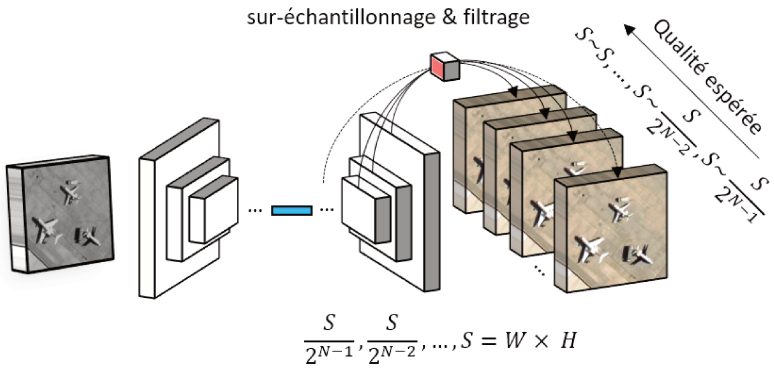


FIGURE 4.12 – Schéma d'une pyramide spatiale de sortie (OSP). S indique ici l'échelle de l'image d'entrée (W × H).Le sur-échantillonnage est réalisé à l'aide d'une fonction d'interpolation classique. Le filtrage correspond à la transformation des caractéristiques profondes en une image dans les domaine cible. Elle est réalisée à l'aide d'une unique couche de convolutions.

dont on aurait tronqué les hautes fréquences de façon itérative (filtre passe-bas, ou masque binaire dans l'espace de Fourrier), soit à considérer un ensemble d'images représentant le même contenu mais redimensionnées les unes par rapport aux autres. L'utilisation de plusieurs échelles d'images a depuis longtemps trouvé des applications allant de la détection d'objets à la mise en correspondance de caractéristiques [Low04; ZYS09]. Plus récemment, le terme de pyramide a été introduit dans le cadre de l'utilisation des réseaux de neurones profonds, définissant l'utilisation jointe de cartes de caractéristiques obtenues entre deux couches de sous-échantillonnage ou de sur-échantillonnage. La fonction de coût perceptuelle [GEB16] a ainsi été proposée afin de contraindre l'entraînement de réseaux de neurones générateurs par la comparaison des cartes de caractéristiques de deux images extraites à partir de plusieurs couches d'un réseau de neurones pré-entraîné. En parallèle, les pyramides spatiales de caractéristiques (*Feature Spatial Pyramids*, FSP) ont été développées afin de combiner les prédictions réalisées sur des cartes de caractéristiques à plusieurs échelles [LDG$^+$17]. Pour cela, Lin *et al.* [LDG$^+$17] proposaient l'utilisation de couches de convolutions supplémentaires afin de passer de l'espace latent (espace des caractéristiques profondes) à l'espace de sortie. Les hypercolonnes [HAGM15] ont quant à elles été proposées pour représenter une image en concaténant dans la dimension des canaux (*i.e.*, dimension des caractéristiques) les cartes de caractéristiques issues de plusieurs couches différentes et redimenssionnées à la même taille. Elles ont ensuite été utilisées avec succès dans un contexte de colorisation [LMS16].

Dans le cadre de nos travaux, nous avons étudié l'utilisation de ce que nous avons nommé "pyramides spatiales de sorties" (*Output Spatial Pyramids*, OSP). Celles-ci se rapprochent des FSP et des hypercolonnes, au sens où elles permettent de contraindre l'optimisation d'un réseau de neurones en utilisant des caractéristiques à plusieurs échelles. Cependant, les OSP ne nécessitent pas de concaténer les caractéristiques profondes (contrairement aux hypercolonnes), et ne se reposent pas sur des convolutions intermédiaires pour projeter les caractéristiques profondes vers l'espace de sortie (domaine cible). Elles sont ici formulées en supposant que toutes les cartes de caractéristiques profondes du décodeur auront le même nombre de canaux, et ce afin de pouvoir projeter toutes les caractéristiques, individuellement, vers l'espace de sortie à l'aide d'une unique couche de convolutions (voir figure 4.12).

Nous rappelons que G est un réseau de neurones entièrement convolutif de type encodeur-décodeur qui va translater les images du domaine A vers le domaine B. Notons S l'échelle d'une image $I_A \in A$, telle que S = W × H, avec W la largeur de l'image (*width*) et H sa hauteur (*height*). Nous définissons la sortie finale de G telle que $O_{d_1} = G(I_A)$, dont l'échelle est égale à S (*i.e.*, I_A et

O_{d_1} font la même taille). D'un point de vue notation, O_{d_1} est générée par la couche l^{d_1} de G (qui est la couche de sortie ici). Elle est supposée être représentée dans le domaine B. L'optimisation usuelle de G se fait alors en calculant une ou plusieurs fonctions de coût en se basant uniquement sur la sortie finale O_{d1}. Cependant, nous remarquons qu'atteindre un état optimal pour l'ensemble des couches internes du décodeur $\{l^{d_2}, ..., l^{d_N}\}$ est complexe lorsqu'une grande quantité de paramètres est impliquée. Afin de faciliter l'entraînement de G, et ainsi améliorer la génération d'images en couleurs réalistes, nous avons intégré les cartes de caractéristiques / sorties intermédiaires du décodeur $\{O_{d_2}, ..., O_{d_N}\}$ dans le calcul de la fonction de coût (voir section 4.4.2). Pour cela, on remarque que deux sorties successives O_{d_i} et O_{d_j}, avec $i \in \{1, .., N-1\}$, $j = i+1$, diffèrent d'un facteur d'échelle. On suppose que ce facteur d'échelle est égal à deux (cas classique). Afin de pouvoir calculer une fonction de coût identique pour chaque O_{d_i}, celles-ci sont redimensionnées afin d'avoir la même échelle spatiale que I_A avant d'être translatées dans l'espace de sortie. Pour toutes les sorties, cela permet de n'avoir qu'un seul discriminateur par opposition aux travaux de [WLZ+18; GGY+18], et de pouvoir se baser sur les images $I_A \in A$ à pleine résolution pour le calcul des fonctions de coût cycliques (pas de perte d'information liée à un sous-échantillonnage de I_A). Cette opération de sur-échantillonnage peut être exprimée à l'aide de l'équation (4.10), où $up(.)$ transforme une image d'échelle $\frac{S}{2}$ en une image d'échelle S (*e.g.*, interpolation, super-résolution). La notation $up^k(.)$ indique la composition de la fonction $up(.)$ avec elle-même k fois (on sur-échantillonne k fois l'image).

$$O_{d_i}^{W \times H} = up^{i-1}(O_{d_i}), i \in \{1, ..., N\} \tag{4.10}$$

Une fois que l'opération de ré-échantillonnage est appliquée sur les sorties du décodeur, il est nécessaire de les projeter dans le domaine cible B afin de pouvoir calculer les fonctions de coût et rétropropager le gradient associé à chacune des sorties. Pour cela, il est possible d'utiliser une couche de convolutions par sortie [HAGM15]. Cependant, cette approche ne permet pas de nous assurer que les représentations profondes obtenues à partir des différentes sorties seront similaires les unes des autres (elles auront été filtrées par des filtres *a priori* différents). Par extension, nous ne pouvons pas nous assurer que les cartes de caractéristiques intermédiaires vont s'optimiser vers le même objectif simplement en observant les sorties projetées dans le domaine B (*i.e*, deux sorties peuvent être réalistes sans que les caractéristiques se ressemblent). Dans le but de contraindre la génération de cartes de caractéristiques intermédiaires plausibles pour la colorisation à plusieurs échelles, nous avons proposé l'utilisation d'une unique couche de sortie dont les poids sont partagés pour toutes les $O_{d_i}^{W \times H}$. Cette idée est en partie empruntée des travaux de [LDG+17], mais à la place d'utiliser des convolutions 1×1 pour gérer des cartes de caractéristiques avec un nombre de caractéristiques différent, nous avons proposé de garder le nombre de caractéristiques n constant dans tout le décodeur. Ce choix nous permet de nous assurer que les caractéristiques intermédiaires sont des représentations adaptées, proches les unes des autres, et ce pour une tâche donnée (la colorisation dans notre cas). Il nous est également possible de les visualiser à travers une couche de sortie, celle-ci faisant alors office de "lentille d'observation". D'un point de vue intuitif, si les caractéristiques intermédiaires (*e.g.*, O_{d_2}) permettent d'obtenir des images parfaitement générées, du point de vue du discriminateur, les couches de convolutions suivantes n'auront qu'à améliorer la résolution des cartes de caractéristiques (*i.e.*, les dernières couches du décodeur n'ont plus besoin d'apprendre à extraire des caractéristiques pour la colorisation *et* le sur-échantillonnage, mais uniquement pour le sur-échantillonnage).

4.4.2 SpyncoGan

Pour évaluer l'intérêt des composants présentés précédemment, nous avons introduit SpyncoGan (Spynco pour *Spatial PYramids and haNdcrafted translation COmbined*). SpyncoGan est un réseau pseudo-cyclique qui se base sur Col-Cycle, mais qui intègre la *Handcrafted Translation* et la OSP. Son architecture est présentée sur la Figure 4.13.

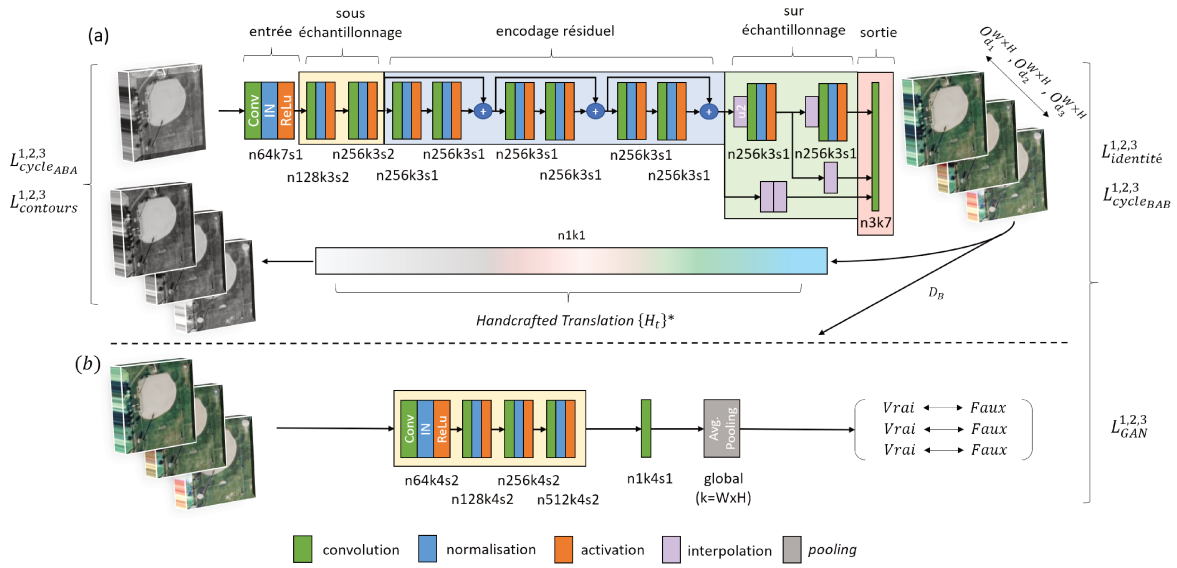


FIGURE 4.13 – Schéma de SpyncoGan avec N = 3 sortie dans l'OSP et une *Handcrafted Translation* H_t entre le domaine des images en couleurs RVB et celui des images en niveaux de gris. (a) Générateur avec H_t. (b) Discriminateur. Le paramètre n indique le nombre de filtres, k la taille du filtre, et s la valeur de pas (*stride*).

Architecture de SpyncoGan

Comme Col-Cycle, SpyncoGan est composé de couches de convolutions et de normalisations par instance. L'utilisation de padding est systématique. Le sous-échantillonnage est réalisé à l'aide de la valeur de pas des filtres de convolutions. Le sur-échantillonnage est réalisé à l'aide d'une interpolation avant l'application d'un filtre de convolutions afin d'éviter certains artefacts visuels [ODO16].

Contrairement à Col-Cycle, SpyncoGan se base sur les OSP. Afin de réduire le nombre d'hyperparamètres à étudier, nous avons fixé le nombre N de sorties de l'OSP à N = 3. Nous utilisons un nombre constant de filtres de convolutions ($n = 256$) dans le décodeur afin de générer des cartes de caractéristiques ayant le même nombre de canaux. Ce nombre de filtres est identique à celui des couches résiduelles. Cela nous permet projeter les cartes de caractéristiques dans le domaine cible B à l'aide d'une unique couche de convolutions, dont les poids sont utilisés / partagés entre toutes les sorties de l'OSP. SpyncoGan n'utilise qu'un seul réseau générateur G et un seul réseau discriminateur D_B, le second GAN étant remplacé par H_t. De plus, SpyncoGan utilise des convolutions séparables afin de réduire la quantité de paramètres à optimiser [MG12] dans l'encodeur, le décodeur et le discriminateur, pour un total de ≈ 7.063 millions de paramètres à optimiser, dont ≈ 4.978 millions de paramètres pour le générateur et ≈ 2.085 millions. A titre comparatif, chaque générateur de Col-Cycle possède ≈ 4.126 millions de paramètres (moins de filtres dans le décodeur) - mais Col-Cycle a deux générateurs. Les discriminateurs sont ici identiques pour les deux réseaux, si ce n'est pour l'usage des convolutions séparables avec SpyncoGan. La *Handcrafted Translation* H_t est quant à elle composée de 3 paramètres fixés, à savoir les poids associés à chacun des canaux RVB.

Notons une nouvelle fois que nous utilisons un nombre de filtres n fixé dans le décodeur, ce qui est une nécessité pour l'utilisation d'une unique couche de convolutions de sortie avec l'OSP. Par construction, cette approche requiert plus de mémoire pour l'entraînement du générateur que l'utilisation d'un décodeur classique, où le nombre de filtres est inversement proportionnel à l'échelle (*e.g.*, Col-Cycle). En particulier, les cartes de caractéristiques intermédiaires possèdent des volumes beaucoup plus importants que celles utilisées avec Col-Cycle, et pour lesquelles de

multiples gradients seront calculés (fonctions de coût liées à toutes les sorties de l'OSP, voir sous-section suivante). Malgré le fait que l'utilisation de convolutions séparables permette de réduire substantiellement le nombre de paramètres du réseau, et que H_t ait une empreinte mémoire faible, ce point est une contrainte pratique forte quant à l'entraînement de réseaux très profonds basés sur les OSP (nécessité d'avoir beaucoup de mémoire disponible).

Fonctions de coût

Cette section définit les fonctions de coût utilisées pour optimiser SpyncoGan à l'aide des sortie de l'OSP.

Notations. Nous rappelons que $I_A \in A$ and $I_B \in B$ sont deux images d'échelle $S = W \times H$. L'OSP nous permet d'obtenir N sorties $O_{d_i}^{W \times H}$ telles que définies par l'équation (4.11). Par souci de simplicité, nous confondons ici les sorties avant et après projection dans le domaine cible B. Après redimensionnement, ces sorties ont toutes la même échelle, égale à celle de I_A. Celles-ci vont permettre d'optimiser G en les intégrant dans le calcul des fonctions de coût.

$$\{O_{d_1}^{W \times H}, ..., O_{d_N}^{W \times H}\} = \{G_1(I_A^{W \times H}), ..., G_N(I_A^{W \times H})\} = G(I_A^{W \times H}) \tag{4.11}$$

avec $\forall i \in \{1, ..., N\}, O_{d_i}^{W \times H} \in B$, et G_i représentant la sortie i de l'OSP de G (*i.e.*, $O_{d_i}^{W \times H}$).

Par souci de concision, nous omettrons l'indice $W \times H$ par la suite, et nous utiliserons la notation $G_i(.)$ à la place de $O_{d_i}^{W \times H}$ lorsque nous jugerons que cela facilite la compréhension.

A l'aide de ces notations, nous pouvons alors redéfinir les fonctions de coût utilisées avec Col-Cycle. Pour cela, nous avons recours à une somme pondérée des coûts calculés pour chaque sortie de l'OSP (voir équations (4.12), (4.18), (4.15)). La pondération associée à chaque sortie (indice i) est gérée par de nouveaux paramètres $\alpha_i, \beta_i, \gamma_i$ et ζ_i, dont les valeurs sont explicitées dans la sous-section suivante.

GAN *loss*. La fonction de coût (formulation *minimax*) liée au GAN peut-être re-définie par l'équation (4.12) en se basant sur un objectif quadratique, inspiré par les travaux de [MLX+17].

$$\mathscr{L}_{\text{GAN}_{A \rightarrow B}}^{1,...,N}(G, D_B) = \sum_{i=1}^{N} \gamma_i \mathbb{E}_A[\|1 - D_B(G_i(I_A))\|_2^2] + \mathbb{E}_B[\|D_B(I_B)\|_2^2] \tag{4.12}$$

En pratique, lors de l'implémentation, on reformule cette contrainte d'une façon similaire aux fonctions de coût utilisées pour Col-Cycle. Ces fonctions, (4.13) et (4.14), sont toutes deux à minimiser. On remarquera que l'équation (4.14) est une formulation "duale" du problème posé par la fonction *minimax*, au sens où l'on chercherait à maximiser pour D_B plutôt que de minimiser. A noter que dans nos expériences, nous nous sommes restreints à $G_1(I_A)$ pour entraîner le discriminateur à l'aide de l'équation (4.14) afin d'éviter que son entraînement ne soit biaisé par les sorties intermédiaires de l'OSP, plus à même de contenir des artefacts visuels.

$$\mathscr{L}_G = \sum_{i=1}^{N} \gamma_i \mathbb{E}_A[\|D_B(G_i(I_A)) - 1\|_2^2] \tag{4.13}$$

$$\mathscr{L}_{D_B} = \mathbb{E}_B[\|D_B(I_B) - 1\|_2^2] + \sum_{i=1}^{N} \gamma_i \mathbb{E}_A[\|D_B(G_i(I_A)\|_2^2] \tag{4.14}$$

Par ailleurs, comme nous utilisons une *Handcrafted Translation* pour réaliser la translation du domaine B vers le domaine A, cette équation n'est définie que pour $\text{GAN}_{A \rightarrow B}$: si les images générées par G sont capables de tromper le discriminateur D_B, nous supposons *a priori* que H_t arrivera à translater correctement l'image générée vers le domaine A.

Cycle-consistency loss. Similairement, la fonction de coût cyclique est redéfinie à l'aide de l'équation (4.15), somme des équations (4.16) et (4.17) (décomposition réalisée par souci de clarté). Elle permet d'ajouter une contrainte cyclique pour chacune des sorties de l'OSP.

$$\mathscr{L}_{cycle}^{1,\dots,\mathrm{N}}(\mathrm{G}) = \mathscr{L}_{cycle_{\mathrm{BAB}}}^{1,\dots,\mathrm{N}}(\mathrm{G}) + \mathscr{L}_{cycle_{\mathrm{ABA}}}^{1,\dots,\mathrm{N}}(\mathrm{G}) \tag{4.15}$$

$$\mathscr{L}_{cycle_{\mathrm{BAB}}}^{1,\dots,\mathrm{N}}(\mathrm{G}) = \sum_{i=1}^{\mathrm{N}} \beta_i \mathbb{E}_{\mathrm{B}}[\|(\mathrm{G}_i(\mathrm{H}_t(\mathrm{I}_{\mathrm{B}})) - \mathrm{I}_{\mathrm{B}}]\|_1] \tag{4.16}$$

$$\mathscr{L}_{cycle_{\mathrm{ABA}}}^{1,\dots,\mathrm{N}}(\mathrm{G}) = \sum_{i=1}^{\mathrm{N}} \beta_i \mathbb{E}_{\mathrm{A}}[\|\mathrm{H}_t(\mathrm{G}_i(\mathrm{I}_{\mathrm{A}})) - \mathrm{I}_{\mathrm{A}}\|_1] \tag{4.17}$$

Identity loss. La fonction de coût identité est redéfinie à l'aide de l'équation (4.18). Son utilité reste identique à celle utilisée avec Col-Cycle, si ce n'est qu'elle va ici apporter une contrainte supplémentaire sur les couches cachées du générateur. Par définition de H_t, elle n'a pas lieu d'être définie pour la translation de B vers A.

$$\mathscr{L}_{identity}^{1,\dots,\mathrm{N}}(\mathrm{G}) = \sum_{i=1}^{\mathrm{N}} \alpha_i \mathbb{E}_{\mathrm{B}}[\|\mathrm{G}_i(\mathrm{I}_{\mathrm{B}}) - \mathrm{I}_{\mathrm{B}}\|_1] \tag{4.18}$$

Contours loss. Enfin, nous ajoutons ici une fonction de coût supplémentaire visant à contraindre la génération d'une image colorisée $\mathrm{G}_i(\mathrm{I}_{\mathrm{A}})$ dont les hautes fréquences seraient proches de celles de l'image en niveaux de gris I_{A}. Pour cela, nous nous basons sur l'existence d'une relation spatiale directe entre les images des deux domaines telle que permise par la *Handcrafted Translation*, et nous comparons les hautes fréquences de I_{A} et $\mathrm{H}_t(\mathrm{G}_i(\mathrm{I}_{\mathrm{A}}))$ vues par un filtre de Sobel $\mathrm{S}_k(.)$ (voir équation (4.19)). Le filtre de Sobel est aisément applicable avec des convolutions, et sa définition symétrique permet de donner plus d'importance au pixel qui se trouve au centre du filtre. Ce dernier point permet d'obtenir des hautes fréquences mieux localisées qu'avec une fonction de coût liée à un gradient local par exemple (*i.e.*, comparaison directe des pixels adjacents).

$$\mathscr{L}_{contours}^{1,\dots,\mathrm{N}}(\mathrm{G}) = \sum_{i=1}^{\mathrm{N}} \zeta_i \mathbb{E}_{\mathrm{A}}[\|\mathrm{S}_k(\mathrm{H}_t(\mathrm{G}_i(\mathrm{I}_{\mathrm{A}}))) - \mathrm{S}_k(\mathrm{I}_{\mathrm{A}})\|_1] \tag{4.19}$$

La fonction de coût totale est alors définie comme étant la somme des fonctions de coût précédentes (voir équation (4.20)).

$$\mathscr{L}^{1,\dots,\mathrm{N}} = \mathscr{L}_{\mathrm{GAN}_{\mathrm{A}\rightarrow\mathrm{B}}}^{1,\dots,\mathrm{N}} + \mathscr{L}_{cycle}^{1,\dots,\mathrm{N}} + \mathscr{L}_{identity}^{1,\dots,\mathrm{N}} + \mathscr{L}_{contours}^{1,\dots,\mathrm{N}} \tag{4.20}$$

Choix des paramètres des fonctions de coût

Les paramètres des fonctions de coût de SpyncoGan (α_i, β_i, γ_i, ζ_i) ont été fixés empiriquement afin de donner plus d'importance à la sortie finale du réseau. Ce choix a été fait afin de contrebalancer la contribution multiple des cartes de caractéristique intermédiaires pour lesquelles le gradient est rétropropagé plusieurs fois à cause de l'OSP. Ils ont été fixés en considérant $\mathrm{N} = 3$, comme décrit précédemment. En pratique : $\alpha_{i\in\{1,2,3\}} = \{5,3,2\}$, $\beta_{i\in\{1,2,3\}} = \{10,6,4\}$, $\gamma_{i\in\{1,2,3\}} = \{1,1,1\}$ et $\zeta_{i\in\{1,2,3\}} = \{1,0,0\}$, avec i l'indice de la sortie de l'OSP (pour rappel, ici, plus i est petit, plus on se rapproche de la couche de sortie). On remarquera qu'à cause de la valeur de ζ_i, seule la sortie finale permet ici de contraindre les hautes fréquences. Ce choix a été fait afin de ne pas tenir compte explicitement des hautes fréquences liées aux sorties intermédiaires, celles-ci ayant plus de chance de contenir des artefacts visuels (moins de filtres de convolutions appliqués, mais plus d'opérations de redimensionnements).

4.4.3 Mise en place des expériences

Nous avons cherché à évaluer l'intérêt de la *Handcrafted Translation* comme remplacement d'un des deux GAN pour la colorisation non-supervisée. Nous nous sommes également intéressés à l'intérêt des contraintes imposées via les différentes sorties de l'OSP. Nos expériences ont été réalisées à l'aide de 2 cartes graphiques NVIDIA GeForce GTX 1080 Ti et des librairies Pytorch, Scikit, Caffe et OpenCV.

Jeux de données

Afin d'évaluer la qualité des colorisations générées, nous nous sommes placés dans un cadre plus générique que celui du traitement des images aériennes historiques. Nous souhaitions avoir accès à un panel d'images ayant des vérités terrains en couleurs et représentant des scènes différentes afin de pouvoir calculer des métriques de similarité. Nous avons pour cela adapté des jeux de données classiquement utilisés pour la classification, à savoir : Cifar-10 [Kri09] et UCMerced Land Use [YN10]. Le jeu de données Cifar-10 est constitué de 60 000 images en couleurs regroupées en 10 classes communes (*e.g.*, avion, chat, bateau, *etc.*). Chaque image a une résolution très faible (*thumbnails*, 32×32 pixels). UCMerced Land Use contient quant à lui des images couleur d'occupation du sol regroupées en 24 classes (*e.g.*, zone résidentielle, plage, rivière). Ces images ont une résolution proche des images aériennes historiques, et font chacune 256×256 pixels. En complément, nous avons aussi utilisé des images de peintures de Cézanne (580 images) et des images de paysages (*Landscape*, 7038 images) qui avaient déjà été utilisées dans un contexte de transfert de domaine (pas de classes, images de 256×256 pixels).

Ces jeux de données étant composés d'images en couleurs, nous les convertissons tout d'abord en niveaux de gris afin d'avoir deux ensembles d'images. Cependant, nos approches étant non-supervisées, ces images ne sont pas appariées explicitement durant l'entraînement afin de simuler un entraînement non-supervisé (*i.e.*, les images de A et B sont échantillonnées aléatoirement, sans mise en correspondance). En pratique, l'entraînement est réalisé en utilisant un sous-ensemble d'entraînement, et l'évaluation est réalisée avec un sous-ensemble de test. Pour les jeux de données Cifar-10, Cézanne et *Landscape*, nous utilisons les sous-ensembles proposés par les auteurs. Pour UCMerced Land Use, aucun sous-ensemble par défaut n'est proposé. Nous avons, de fait, échantillonné aléatoirement 80% des images pour l'entraînement, et 20% pour l'évaluation.

Métriques

L'évaluation de la qualité de la colorisation est effectuée toutes les 10 *epochs* pour quantifier l'évolution des métriques durant l'entraînement. Nous calculons l'erreur quadratique moyenne (MSE) et le score de similarité structurelle (SSIM) entres les images colorisées et les images en couleurs réelles (mesures calculées pour chaque canal couleur, puis moyennées par le nombre de canaux). La MSE permet de déterminer grossièrement la différence entre deux images (plus la valeur est basse, plus les deux images sont proches). Cette métrique est couramment utilisée pour évaluer les résultats des algorithmes de régression, et sa variante monotone (racine MSE, RMSE) a déjà été appliquée pour évaluer les algorithmes de colorisation [LMS16]. La SSIM indique la qualité d'une image par rapport à une autre (plus sa valeur est élevée, plus les images comparées sont proches), en mettant l'accent sur les différences structurelles. Ces deux mesures fournissent un aperçu de la qualité de la colorisation lorsque des images en couleurs réelles sont disponibles (cas de nos ensembles de données). Le choix d'utiliser ces mesures quantitatives de la qualité de la colorisation nous permet de comparer plusieurs approches sans à avoir recours à un questionnaire (long et complexe à mettre en place lorsque plusieurs méthodes et plusieurs jeux de données sont utilisés).

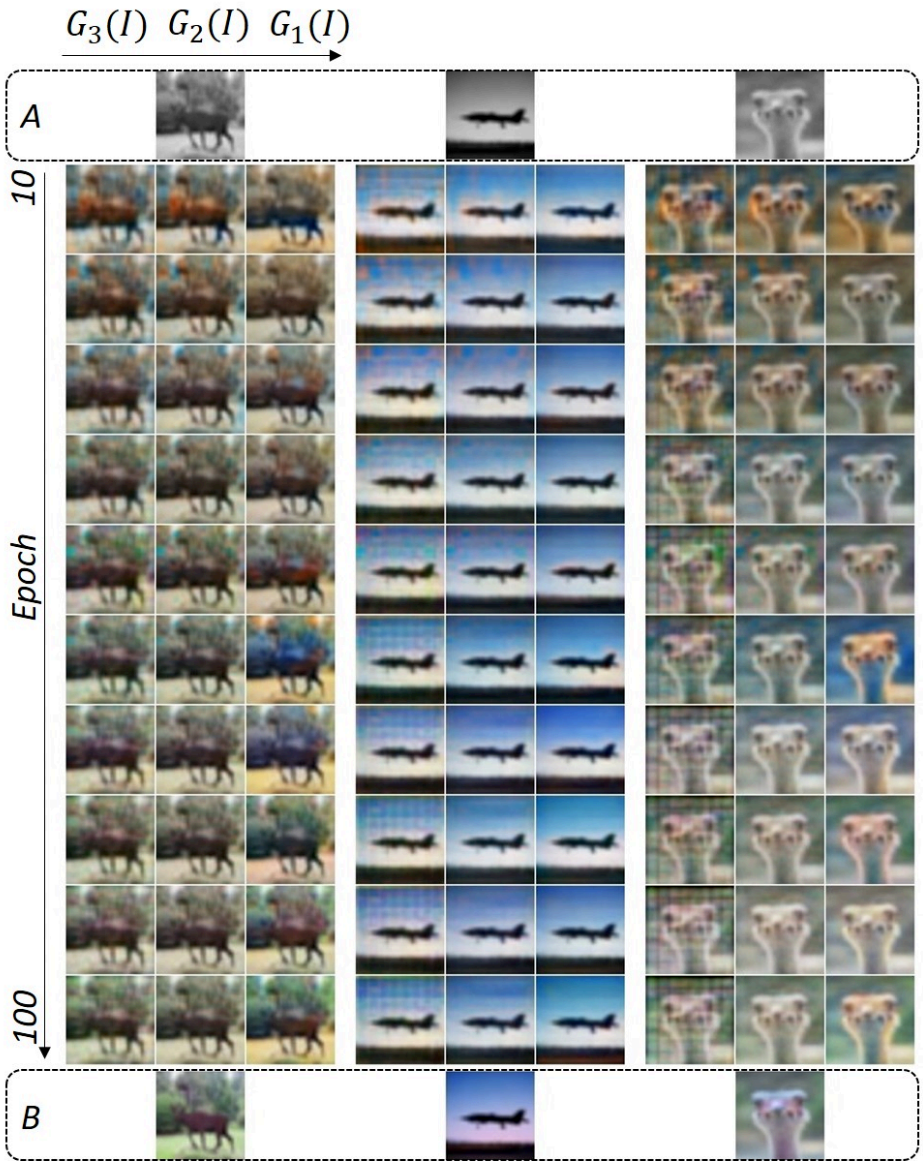


FIGURE 4.14 – Résultats qualitatifs obtenus durant l'entraînement de SpyncoGan sur le jeu de données Cifar-10.

4.4.4 Résultats et discussions

Résultats qualitatifs

La figure 4.1 présente des exemples de résultats qualitatifs obtenus sur des peintures de Cézanne, des photos de paysages et des images aériennes de UCMerced Land Use. Visuellement, nous trouvons que ces résultats semblent plutôt réalistes pour une approche non-supervisée. Plus de résultats sont disponibles dans la section 4.6. Ces résultats incluent des cas limites où le réseau n'a pas réussi à coloriser correctement les images en fonction des *epochs*. Néanmoins, l'ensemble des représentations obtenues semblent indiquer que l'utilisation de H_t à la place de l'un des deux GAN utilisé par certains réseaux cycliques est une piste viable dans un contexte de colorisation.

La figure 4.14 montre les résultats qualitatifs obtenus avec SpyncoGan sur trois échantillons d'images de Cifar-10. De haut en bas, les images sont représentées en niveaux de gris (domaine A), en fausses couleurs (images colorisées) et en couleurs réelles (domaine B). Les images colorisées sur différentes lignes ont été générées à différentes *epochs* de l'entraînement de SpyncoGan

TABLEAU 4.1 – Résultat de l'ablation des sorties. Métriques calculées toutes les 10 *epochs* (entraînement de 50 *epochs*) puis moyennées (*Avg.*).

Données	Fonction de coût	*Avg.* MSE ↓	*Avg.* SSIM (%) ↑
Cézanne	$\mathscr{L}^1$	92.9	82
Cézanne	$\mathscr{L}^{1,2,3}$	91.5	82
Landscape	$\mathscr{L}^1$	85.7	83
Landscape	$\mathscr{L}^{1,2,3}$	85.1	83
UCMerced Land Use	$\mathscr{L}^1$	85.5	86
UCMerced Land Use	$\mathscr{L}^{1,2,3}$	83.1	85
Cifar-10	$\mathscr{L}^1$	87.2	89
Cifar-10	$\mathscr{L}^{1,2,3}$	86.8	89

(de 10 à 100 avec un pas de 10). De gauche à droite, les résultats obtenus sont présentés pour les différentes sorties de l'OSP à savoir $G_3(I_A)$, $G_2(I_A)$ et $G_1(I_A)$. D'un point de vue global, nous observons des artefacts en damier sur les images $G_3(I_A)$ (image la plus à gauche) qui semblent perdurer durant l'entraînement. Ils semblent néanmoins avoir été filtrés par les couches plus profondes, ce qui est le comportement attendu pour notre réseau. Cependant, puisque $G_3(I_A)$ a été directement obtenu à partir des couches résiduelles après ré-échantillonnage *et* convolution spatiale ; dont les poids sont partagés entre toutes les sorties ; nous pensons que les couches résiduelles n'ont pas pu apprendre une représentation suffisante pour éliminer les artefacts causés par le ré-échantillonnage, ou ont causé les artefacts eux-mêmes. On constate également que les représentations obtenues avec $G_2(.)$ et $G_3(.)$ semblent parfois très proches, ce qui met en avant les contraintes imposées par l'OSP.

Par ailleurs, nous observons sur cette figure la diversité des représentations possibles au cours de l'entraînement. Cette diversité met en avant l'exploration de l'espace des paramètres pour générer des images en couleurs vraisemblables. Ce point est particulièrement intéressant dans un contexte de colorisation car il permet de créer des représentations couleur variées uniquement en chargeant les poids du réseau optimisés à une *epoch* différente, et ce sans modifier son architecture.

Évaluation quantitative par ablations

Ablation des sorties. Nous avons commencé par étudier l'intérêt de l'OSP en considérant N = 1 sortie pour l'entraînement de SpyncoGan. Nous avons nommé cette étude "ablation des sorties". Pour cela, nous considérons l'architecture de SpyncoGan présentée en section 4.4.2, ce qui signifie que les trois sorties de l'OSP sont disponibles mais que seule $G_1(I)$ est utilisée dans le calcul des fonctions de coût durant l'entraînement (N = 1). Dans un but comparatif, on distingue donc les fonctions de coût avec N = 1 et N = 3 (SpyncoGan sans ablation), à savoir $\mathscr{L}^1$ et $\mathscr{L}^{1,2,3}$. Le tableau 4.1 présente les scores de MSE et de SSIM calculés toutes les 10 *epochs* puis moyennés entre les *epochs* 10 et 50. Nous observons qu'utiliser toutes les sorties (N = 3) pour l'optimisation de SpyncoGan permet d'obtenir une diminution moyenne de 1.2 points pour la MSE, mais une diminution de 0.25% de la SSIM. Il semblerait qu'utiliser plus de contraintes pour l'optimisation des couches cachées à travers l'OSP permette d'obtenir des couleurs plus consistantes par rapport aux images réelles (MSE plus faible), mais tende à diminuer le réalisme relatif des structures générées. Le gain apporté par l'inclusion des sorties de l'OSP dans le calcul des fonctions de coût semble donc limité sur ces données.

Ablation de la fonction de coût. Pour les deux cas de figures $\mathscr{L}^1$ et $\mathscr{L}^{1,2,3}$, nous avons également étudié l'intérêt de la fonction de coût liée aux hautes fréquences $\mathscr{L}^{1,...,N}_{contours}$. Les résultats ob-

TABLEAU 4.2 – Résultats de l'ablation de la fonction de coût liée aux hautes fréquences sur les peintures de Cézanne. Métriques calculées toutes les 10 *epochs* (entraînement de 50 *epochs*) puis moyennées (Avg.).

Fonction de coût	Ablation	*Avg.* MSE ↓	*Avg.* SSIM (%) ↑
$\mathscr{L}^1$	$\mathscr{L}^1_{contours}$	92.6	79
$\mathscr{L}^1$	/	92.9	82
$\mathscr{L}^{1,2,3}$	$\mathscr{L}^{1,2,3}_{contours}$	92.0	77
$\mathscr{L}^{1,2,3}$	/	91.5	82

tenus sont présentés sur le tableau 4.2 pour les peintures de Cézanne. La colonne Ablation sur ce tableau indique si la fonction de coût liée aux hautes fréquences a été retirée (nom de la fonction) ou pas (symbole "/"). Malgré la faible contribution de $\mathscr{L}^{1,...,N}_{contours}$ dans le calcul de la fonction de coût totale, représentée par une petite valeur de ζ_i, on observe que ne pas utiliser cette fonction réduit significativement l'indice de similarité structurel. Comme attendu, le fait de contraindre la génération de contours réalistes à travers H_t permet de préserver les hautes fréquences, et ce même sans OSP.

Visualization des sorties intermédiaires de l'OSP. Malgré des différences faibles en termes de scores MSE et SSIM pour $G_1(I_A)$ avec et sans ablation des sorties, nous nous sommes demandé à quoi ressemblaient les caractéristiques profondes des sorties intermédiaires vues par la couche de sortie, et ce avec et sans ablation des sorties. La figure 4.15 permet de visualiser les sortie $G_2(I_A)$ et $G_3(I_A)$ pour des peintures de Cézanne au cours de l'entraînement. On distingue les visualisations obtenues par SpyncoGan entraîné avec $\mathscr{L}^1$ et avec $\mathscr{L}^{1,2,3}$. On peut ainsi observer que les sorties intermédiaires de SpyncoGan entraîné avec $\mathscr{L}^1$ sont beaucoup moins réalistes que celles obtenues lorsque SpyncoGan est entraîné avec $\mathscr{L}^{1,2,3}$. De plus, ces visualisations nous indiquent que les caractéristiques profondes extraites de la couche l^{d_3} et celles extraites de l^{d_2} sont très différentes les unes des autres lorsque SpyncoGan est entraîné avec $\mathscr{L}^1$. Néanmoins, ces visualisations mettent en avant la capacité des réseaux générateurs à préserver les structures spatiales, et ce même sans avoir recours à des contraintes imposées directement sur les cartes de caractéristiques profondes / intermédiaires (contrainte que nous imposons à l'aide de l'OSP).

Qualité de la colorisation par rapport à Col-Cycle

Nous avons ensuite comparé la qualité relative des colorisations générées avec SpyncoGan et Col-Cycle durant l'entraînement de chacun des réseaux, sans remplacement de textures (*i.e.*, on évalue la sortie "brute"). Les deux réseaux ont été entraînés durant 100 *epochs* (mais seulement 50 pour les photographies de paysages) avec un taux d'apprentissage de 0.0002 et une diminution linéaire du taux d'apprentissage vers 0 appliqué après que la moitié des *epochs* aient été réalisées. Les poids des deux réseaux ont été initialisés aléatoirement. L'algorithme d'optimisation utilisé pour les générateurs comme pour les discriminateurs a été fixé sur la méthode ADAM [KB14], avec les paramètres par défaut ($\beta_1 = 0.9, \beta_2 = 0.999$). La taille du *batch* a été fixée pour chaque réseau de façon à pouvoir traiter un maximum d'images en parallèle sur les cartes graphiques à notre disposition durant l'entraînement. Les résultats obtenus toutes les 10 *epochs* à l'aide des métriques MSE et SSIM sont présentés sur la figure 4.16.

On observe que les sorties de SpyncoGan permettent systématiquement d'obtenir des score MSE plus faibles que ceux de Col-Cycle, et des scores SSIM plus élevés. Cette observation se traduit par le fait que les colorisations générées par SpyncoGan sont plus réalistes au sens des métriques utilisées. En particulier, SpyncoGan semble permettre d'obtenir des résultats dont les couleurs sont plus proches des vérités terrain, et préserve mieux les structures visibles dans les images. Ces deux points s'expliquent par l'utilisation d'une fonction de coût dédiée aux hautes fréquences,

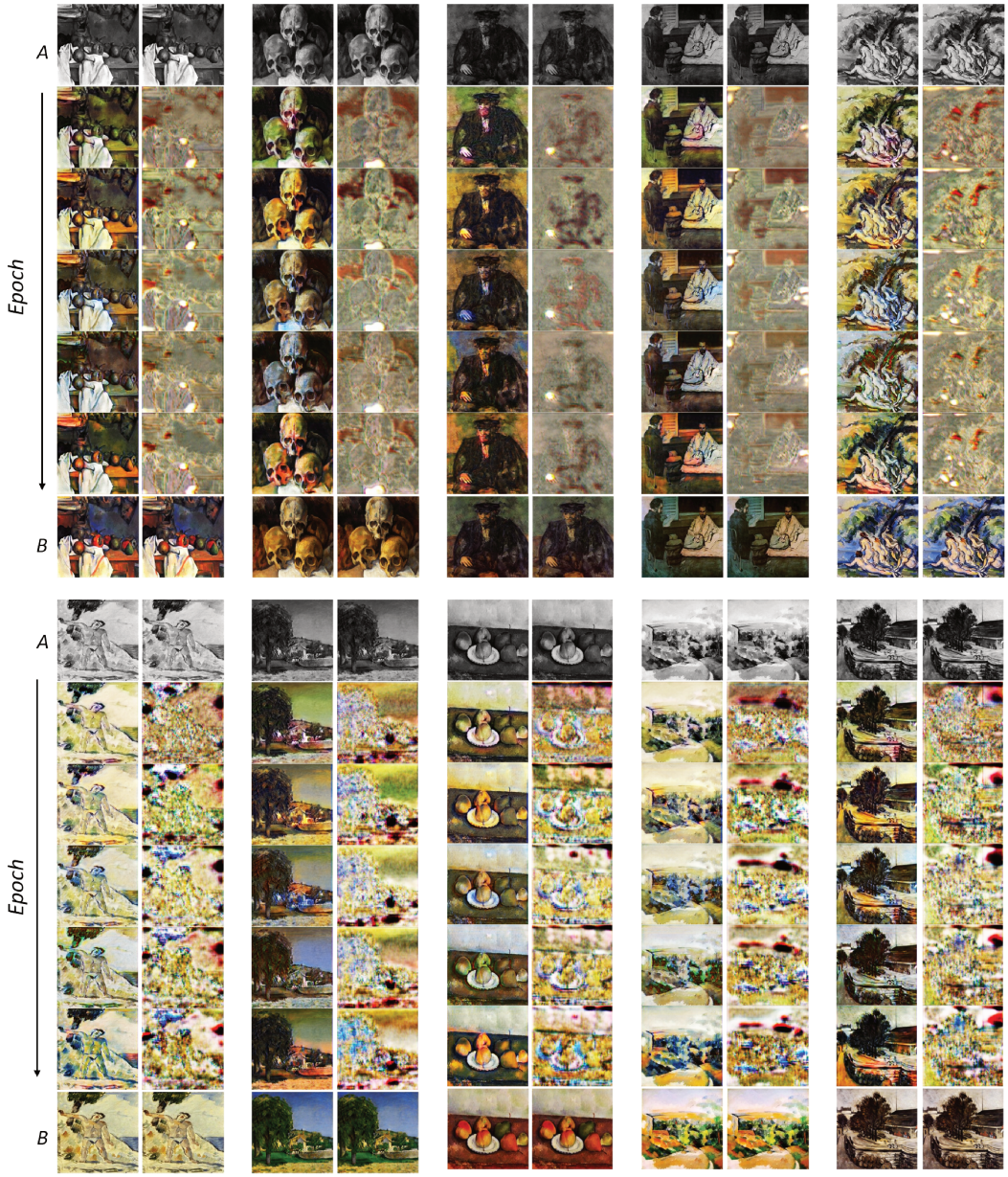


FIGURE 4.15 – Sorties intermédiaires de SpyncoGan sur des peintures de Cézanne à différentes *epochs*. Ligne du haut : $G_2(I)$. Ligne du bas : $G_3(I)$. Colonne de gauche : sorties après un entraînement à l'aide de $\mathscr{L}^{1,2,3}$. Colonne de droite : sorties après un entraînement à l'aide de $\mathscr{L}^1$, tel que décrit dans la section 4.4.4.

le nombre accru de caractéristiques utilisées dans le décodeur de SpyncoGan par rapport à ColCycle, et l'utilisation de l'OSP. Comme espéré, on observe que la qualité des sorties $G_i(I_A)$ de l'OSP évolue de façon proportionnelle à $\frac{1}{i}$. Cependant, on observe sur les sous-figures 4.16 (a), (c) et (g) que $G_2(I_A)$ permet parfois d'obtenir des résultats en MSE meilleurs que $G_1(I_A)$. Nous pensons que ces résultats sont dus à la nature même de l'OSP, qui impose des contraintes plus fortes aux caractéristiques internes du réseau. Aux regards des résultats obtenus en SSIM, ces contraintes ne semblent cependant pas suffisantes pour permettre la préservation des hautes fréquences sans utiliser des filtres de convolutions supplémentaires après ré-échantillonnage.

En nous basant sur ces observations, nous pouvons ici conclure que l'utilisation d'une *Handcrafted Translation* H_t est une solution viable pour remplacer l'un des deux GAN utilisé par les réseaux de neurones cycliques dans un contexte de colorisation. De plus, l'utilisation de pyramides spatiales de sortie pour contraindre les couches intermédiaires du générateur vers un même optimal semble permettre l'amélioration graduelle des résultats entre les différentes sorties malgré un

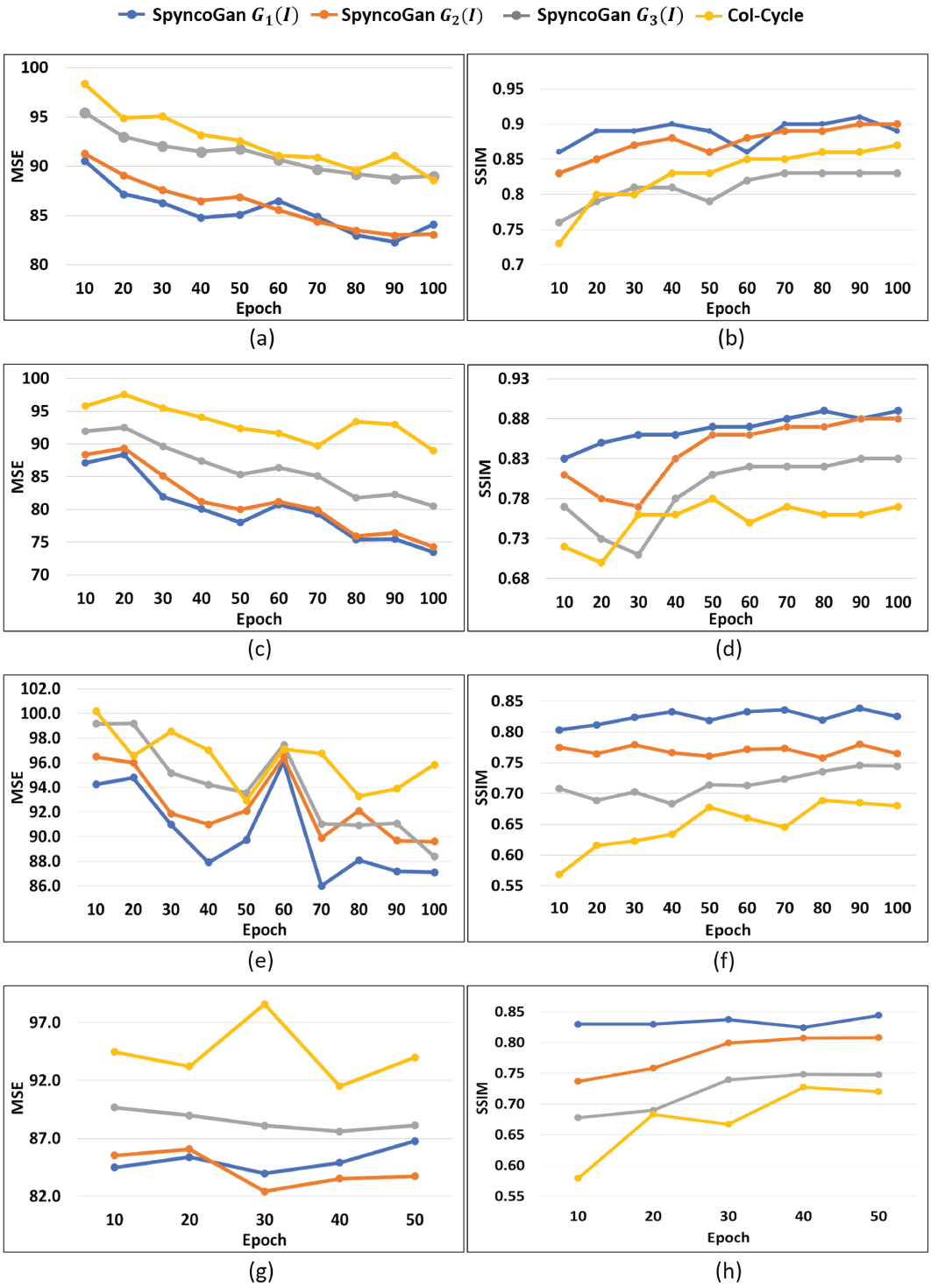


FIGURE 4.16 – *Mean Square Error* (MSE) et *Structural Similarity Measure* (SSIM) entre les images colorisées et les images réelles des jeux de données Cifar-10 (a,b), UCMerced Land Use (c,d), peintures de Cézanne (e,f) et *Landscape* photos (g,h).

gain observé relativement faible.

Comparaisons complémentaires

Nous comparons ici les résultats obtenus par Col-Cycle et SpyncoGan avec ceux de CycleGan sur les jeux de données Cézanne et UCMerced Land Use. On observe sur le tableau 4.3 que, sur

ces jeux de données, SpyncoGan tend à obtenir de meilleurs résultats (MSE plus faible, SSIM plus élevée).

TABLEAU 4.3 – Comparaison des colorisations produites par Col-Cycle, CycleGan (9 couches résiduelles), et SpyncoGan sur les jeux de données UCMerced Land Use et les peintures de Cézanne (meilleurs résultats parmi les 50 premières *epochs*).

Métrique	Jeu de données	Col-Cycle	CycleGan	SpyncoGan
MSE	UCMerced Land Use	92.4	90.3	78.0
SSIM	UCMerced Land Use	77.6	77.8	87.3
MSE	Cézanne	92.9	91.7	87.9
SSIM	Cézanne	68.6	70.4	83.5

4.4.5 Application à la classification

Nous nous sommes demandés si les colorisations générées par SpyncoGan pouvaient alors permettre d'obtenir des gain en classification.

Classification inter-domaines

Nous souhaitions dans un premier temps étudier la généralisation en classification d'un réseau de neurones à convolutions à des images représentées avec des couleurs légèrement différentes. Il s'agit ici de répondre à la question : dans quelle mesure est-il possible d'appliquer un réseau de neurones entraîné sur des images en couleurs pour classifier des images colorisées, et inversement ? Cette information nous permettrait d'envisager l'utilisation de jeux de données récents pour la classification d'images historiques panchromatiques via la colorisation.

Afin d'investiguer cette question, nous avons considéré les images générées par SpyncoGan à toutes les 10 *epochs* d'entraînement comme étant représentées avec des couleurs différentes. On

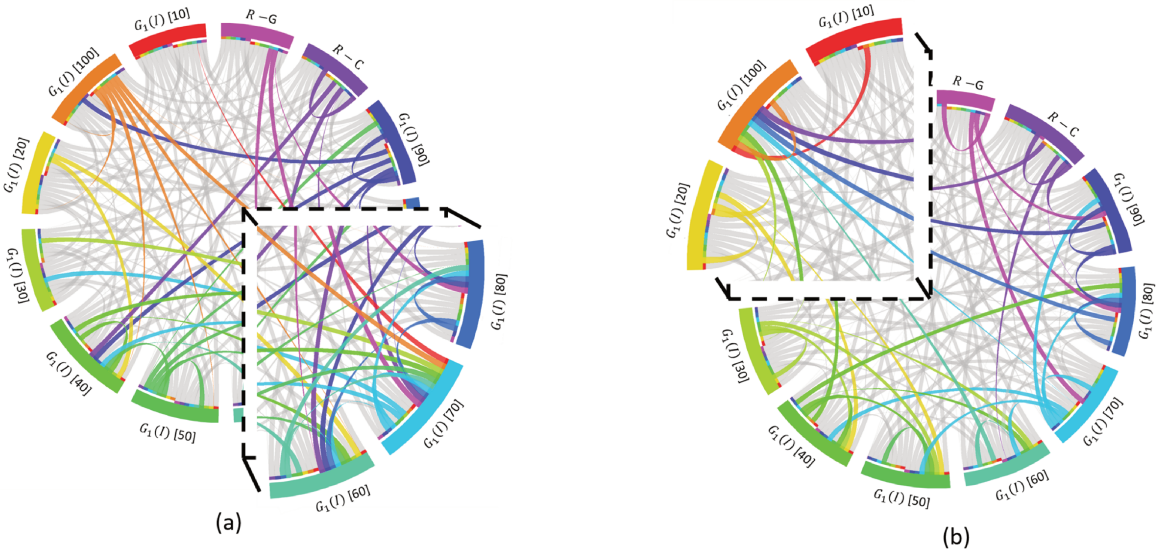


FIGURE 4.17 – Diagrammes de cordes pour la classification inter-domaine sur (a) UCMerced Land Use et (b) Cifar-10 avec $G_1(I)$. R-G est niveaux de gris (*Real-Gray*). R-C est pour couleurs réelles (*Real-Color*). [Nombre] indique l'*epoch* de colorisation.

considère alors que les couleurs générées sont représentatives d'un domaine couleur légèrement différent des images réelles en couleurs, mais aussi de tous les autres domaines couleur générés. Cette observation se vérifie en pratique, que ce soit sur la figure 4.14, ou sur les figures complémentaires présentées en section 4.6. A l'aide de ces données, nous avons entraîné des réseaux de neurones à convolutions sur les images colorisées, les vraies images en couleurs, et les images en niveaux de gris des jeux de données d'entraînement de Cifar-10 et de UCMerced Land Use. Ainsi, pour un même jeu de données, un même réseau a été entraîné séparément sur chacun des 12 domaines couleur (1 domaine couleur généré toutes les 10 epochs, 1 pour les vraies couleurs, et 1 pour les niveaux de gris). Le but est alors d'évaluer le pouvoir discriminant de ces réseaux entraînés sur l'ensemble des domaines couleur disponibles (*i.e.*, évaluation inter-domaines) en utilisant les données de test. En pratique, nous avons utilisé AlexNet [KSH12] avec la normalisation par *batch* sur le jeu de données Cifar-10, et VGG-16 [SZ14] sur UCMerced Land Use. Pour l'entraînement de ces réseaux, toutes les images ont été redimensionnées à 256 × 256 pixels. Le taux d'apprentissage a été fixé à 0.0001, avec une décroissance d'un facteur 10 à 33% et 66% de l'entraînement. AlexNet a été entraîné durant 20 *epochs*, et VGG-16 durant 40 *epochs*.

Les résultats obtenus en classification inter-domaine pour $G_1(I_A)$ sont représentés à l'aide d'un diagramme de cordes [KSB+09] sur la figure 4.17. Sur ce diagramme, chaque arc de cercle correspond au même jeu de données mais représenté dans un domaine couleur différent. Les cordes indiquent quant à elles les relations entre les jeux d'entraînement et de test des différents domaines (*i.e.*, réseau entraîné sur un domaine couleur puis évalué sur un autre). Une corde attachée à un arc de cercle indique que le domaine correspondant à l'arc de cercle a été utilisé pour l'entraînement. Une corde séparée par un blanc de l'arc de cercle indique que le domaine correspondant à l'arc de cercle a été utilisé pour le test. Seules les cordes correspondant au premier quartile des taux de bonne classification obtenus (les 25% taux les plus élevés parmi tous) sont en couleur, les autres étant grisés. Afin d'identifier les domaines couleur qui sont les plus aptes à permettent à un réseau de neurones à convolutions de généraliser à d'autres domaines couleur, il suffit alors de compter le nombre de cordes colorées attachées à chaque arc. L'arc ayant le compte le plus grand correspond au domaine couleur qui a permis la meilleure généralisation. On observe ainsi qu'entraîner VGG-16 sur les images d'UCMerced Land Use colorisées par SpyncoGan à l'*epoch* 70 permet une meilleure généralisation qu'avec les autres domaines couleur. Pour Cifar-10, les images colorisées par SpyncoGan à l'*epoch* 100 sont celles qui permettent la meilleure généralisation d'AlexNet. Il semblerait ici qu'entraîner un réseau de neurones classifieur sur des données colorisées permette d'obtenir des représentations plus robustes aux variations de domaines couleur qu'un entraînement réalisé à l'aide de données réellement en couleurs. Afin d'accompagner ces observations, nous présentons les résultats moyennés par domaine d'entraînement sur le tableau 4.4. La valeur moyenne très faible obtenue sur Cifar-10 par un AlexNet entraîné sur les images en niveaux de gris et évalué sur l'ensemble des jeux de données pourrait s'expliquer par la présence d'un fort biais dans les représentations couleurs. Ce point mériterait cependant la réalisation d'expériences supplémentaires afin de mieux comprendre les tenants et les aboutissants de ce résultat qui nous a particulièrement surpris.

Classification sur HistAerial

Nous reproduisons ici une partie des expériences réalisées avec Col-Cycle à l'aide de SpyncoGan afin de voir si les colorisations générées par ce réseau permettent d'améliorer la classification des images aériennes historiques. Pour cela, nous avons entraîné SpyncoGan sur le même jeu de données que Col-Cycle et avec le même taux d'apprentissage. Comme pour Col-Cycle, nous avons appliqué le remplacement de textures lors de l'inférence. Les résultats obtenus après 120 *epochs* sont présentés sur le tableau 4.5. Nous présentons également les résultats obtenus avec les statistiques couleur générées seules (*i.e.*, sans la texture). Les deux réseaux semblent permettre de générer des couleurs permettant d'améliorer légèrement les taux de bonne classification, sans différence notable lorsque ces couleurs sont combinées aux caractéristiques de texture. Lorsqu'uti-

TABLEAU 4.4 – Taux de classification (%) inter-domaine moyennés sur tous les domaines couleur. (1) VGG-16 entraîné pour 40 *epochs* sur UCMerced Land Use et (2) AlexNet entraîné pour 20 *epochs* sur Cifar-10.

Ensemble d'entraînement	Col. *epoch*	Avg. % (1)	Avg. % (2)
SpyncoGan ($G_1(I_A)$)	10	94.7	74.8
SpyncoGan ($G_1(I_A)$)	20	95.2	77.0
SpyncoGan ($G_1(I_A)$)	30	95.4	78.7
SpyncoGan ($G_1(I_A)$)	40	96.2	79.2
SpyncoGan ($G_1(I_A)$)	50	95.3	79.5
SpyncoGan ($G_1(I_A)$)	60	96.3	79.9
SpyncoGan ($G_1(I_A)$)	70	97.0	78.7
SpyncoGan ($G_1(I_A)$)	80	96.1	79.9
SpyncoGan ($G_1(I_A)$)	90	95.2	79.5
SpyncoGan ($G_1(I_A)$)	100	95.1	81.0
Couleurs réelles	/	92.4	75.7
Niveaux de gris	/	92.5	22.1

lisées seules, les couleurs générées par Col-Cycle semblent cependant avoir un pouvoir discriminant nettement supérieur, ce qui tend à montrer qu'une colorisation que l'on pourrait qualifier de plus grossière (SSIM plus faible pour Col-Cycle en moyenne) n'est pas un problème pour améliorer la classification des images aériennes historiques. Dans les deux cas, les couleurs générées seules permettent d'obtenir des taux de bonne classification beaucoup plus élevés qu'un choix aléatoire ($\approx$ 14.3% pour 7 classes d'occupation du sol). Ce dernier point indique que les couleurs générées ne sont pas incohérentes sémantiquement les unes par rapport aux autres. Des exemples de colorisations d'images aériennes historiques avec SpyncoGan sont présentées sur la figure 4.18, montrant que les résultats visuels obtenus semblent aussi intéressants visuellement que ceux obtenus avec Col-Cycle (voir figure 4.9).

TABLEAU 4.5 – Comparaison de l'apport des couleurs générées par Col-Cycle et SpyncoGan à la classification des images aériennes historiques de HistAerial.

Texture	Réseau	Col. *epoch*	Taux de bonne classification (%)
CLBP	/	/	88.1
CLBP	Col-Cycle	120	89.2
CLBP	SpyncoGan	120	89.2
LCOLBP	/	/	89.3
LCOLBP	Col-Cycle	120	89.5
LCOLBP	SpyncoGan	120	89.4
/	Col-Cycle	120	58.6
/	SpyncoGan	120	49.0
Aléatoire	/	/	14.3

4.4.6 Conclusion partielle

Nous avons développé SpyncoGan, une nouvelle approche non-supervisée combinant réseaux de neurones profonds à convolutions et méthodes classiques pour la colorisation d'images panchromatiques. Nous avons appliqué cette méthode sur plusieurs jeux de données différents, ce qui nous a permis de mettre en avant sa capacité à générer des images relativement réalistes par rapport à l'existant. Une étude par ablation nous a permis de montrer les forces et les faiblesses des blocs constituant SpyncoGan. En particulier, le gain procuré par l'utilisation d'une pyramide

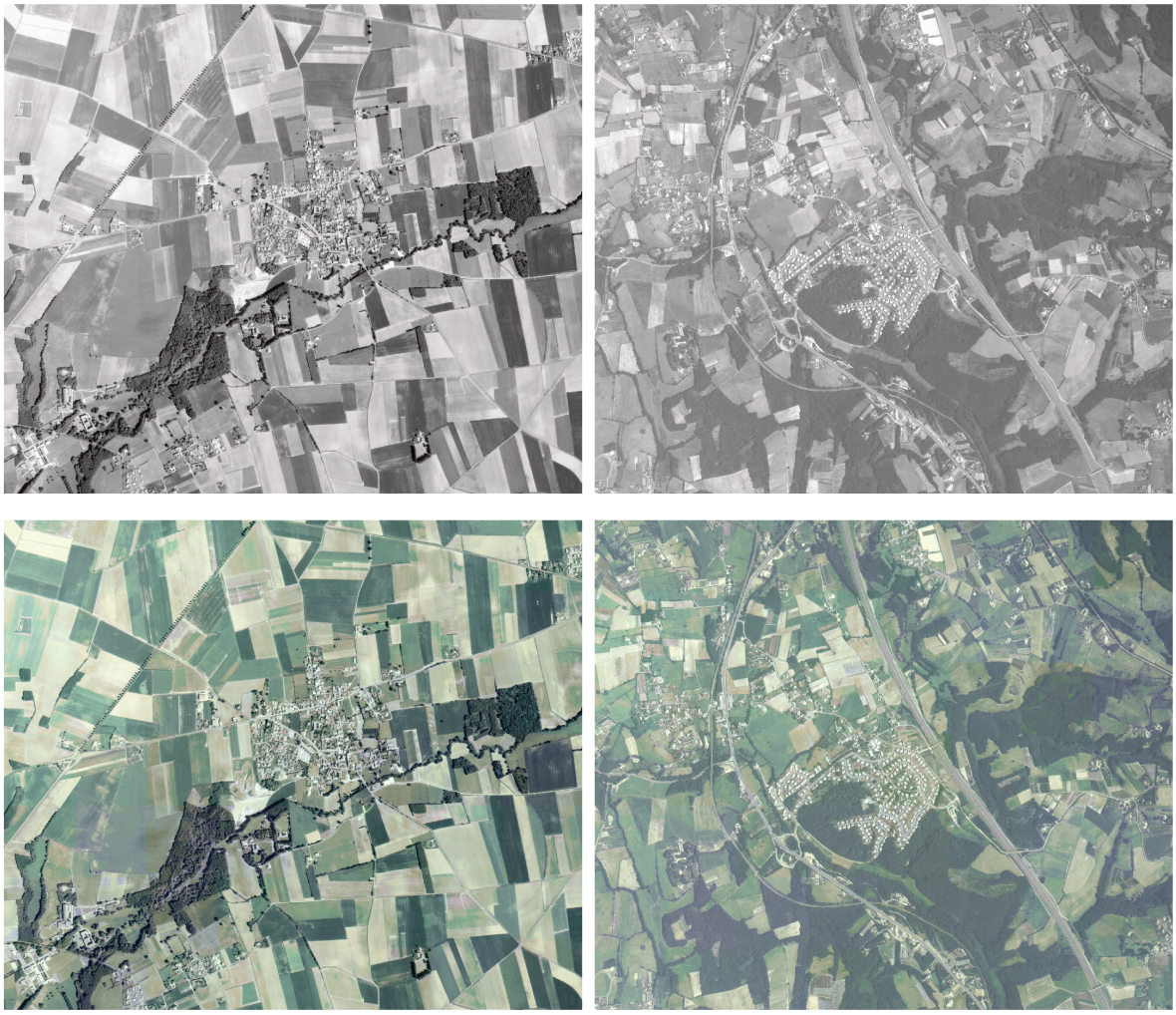


FIGURE 4.18 – Exemples d'images aériennes historiques colorisées avec SpyncoGan après 120 *epochs*. Ligne du haut : images panchromatiques. Ligne du bas : colorisation par imagette avec remplacement de textures.

spatiale de sortie semble limité dans un contexte de colorisation, et ce malgré les contraintes intuitives qu'elle impose sur les représentations internes du réseau. Le fait de remplacer l'un des deux GAN utilisés dans les réseaux de neurones cycliques par une fonction définie manuellement (H_t) semble être une alternative viable pour la colorisation non-supervisée. L'utilisation de cette approche dans le cas général semble néanmoins limitée par la capacité que nous avons à définir des fonctions "artisanales" pour un ensemble de tâches données (*e.g.*, passer d'une segmentation sémantique à une image réaliste). Enfin, nous avons évalué l'intérêt de la colorisation pour la classification. Nous avons montré qu'entraîner des réseaux de neurones à convolutions sur des images colorisées permettait d'obtenir des résultats proches de ceux obtenus avec des images réellement en couleurs dans un contexte de classification inter-domaines couleur. Ce point permet d'envisager l'utilisation conjointe de données récentes et historiques pour améliorer les résultats que l'on peut espérer obtenir en classification. Enfin, nous avons montré que SpyncoGan permettait, comme Col-Cycle, de générer des couleurs adaptées à la classification des images aériennes historiques.

4.5 Conclusion

Résumé des travaux réalisés. Nous nous sommes intéressés à la colorisation non-supervisée d'images panchromatiques. Dans un premier temps, nous avons cherché à coloriser les images aériennes historiques de très hautes résolutions afin d'améliorer la visualisation de ces données pour les géomaticiens. Nous avons pu mettre en avant un effet mosaïque apparaissant lors de la colorisation par imagettes, que nous avons partiellement résolu à l'aide d'un remplacement de textures. Devant les résultats encourageants obtenus, nous nous sommes placés dans un cadre plus général afin de proposer une nouvelle approche de colorisation non-supervisée basée sur un *a priori* empirique et une représentation pyramidale. Nous avons ainsi pu montrer que remplacer l'un des deux GAN utilisé dans les réseaux de neurones cycliques était possible pour la colorisation. Nous avons également montré que la représentation pyramidale permettait de contraindre les représentations internes du réseau de neurones (visualisation réalistes), mais que son gain quantitatif semble limité. Enfin, nous avons pu évaluer l'intérêt de la colorisation pour la classification. Nous avons ainsi montré qu'un réseau de neurones classifieur entraîné sur des images colorisées pouvait mieux se généraliser à d'autres domaines couleur que lorsqu'il était entraîné sur des données en vraies couleurs. Nous avons également montré l'intérêt de la colorisation pour améliorer légèrement la classification des images aériennes historiques, les couleurs générées occasionnant de légers gains sur HistAerial lorsque combinées avec des caractéristiques de texture.

Vision critique sur les travaux réalisés. L'utilisation de méthodes non-supervisées basées sur des représentations cycliques est un choix que nous avons fait afin de pouvoir entraîner les réseaux de neurones générateurs à l'aide des images à coloriser elles-mêmes. Il aurait cependant pu être intéressant de comparer les méthodes développées avec des approches supervisées de la littérature afin de pouvoir mieux positionner nos méthodes. Par ailleurs, l'utilisation de H_t est ici limitée au cas de la colorisation. Il aurait pu être intéressant de chercher à développer des fonctions H_t pour différentes tâches de translation d'image à image, ce qui représente une problématique que nous jugeons particulièrement complexe. De plus, nous avons ici étudié l'utilisation de pyramides spatiales de sorties en nous basant sur des observations intuitives, mais nous n'avons pas comparé cette approche à d'autres formulations multi-échelles. Nous pensons que ce point mériterait d'être approfondi. Enfin, les comparaisons réalisées à différentes *epochs* fixées ne peuvent qu'être indicatives : les colorisations générées lors de l'entraînement de réseaux différents pris à une même *epoch* n'ont pas de raison, *a priori*, d'être similaires (*i.e.*, dans tous les cas, nous comparons des minima locaux). Il est par ailleurs possible que des couleurs générées soient très éloignées d'une image réelle en couleurs tout en étant discriminantes sémantiquement et perceptuellement appréciables pour l'être humain : une image en couleurs parfaite n'existe pas. A ce titre, nous avons ici uniquement utilisé des métriques par rapport à des images de références. Il aurait pu être intéressant d'évaluer la diversité des couleurs générées afin de déterminer si l'un des réseaux a tendance a apprendre des représentations plus variées qu'un autre.

4.6 Visualisations supplémentaires

Cette section est constituée de visualisations supplémentaires (figures 4.19, 4.20, 4.21 et 4.22) mettant en avant les différentes colorisations générées au cours de l'entraînement de SpyncoGan. On y constate une forte variabilité, ainsi que des représentations beaucoup plus réalistes que d'autres. Nous recommandons une visualisation électronique pour apprécier les qualités et les défauts de ces images. Ces visualisations sont également disponibles ici :
`http://liris.univ-lyon2.fr/SpyncoGan/files/ratajczak-SpyncoGan19supp.pdf`.



FIGURE 4.19 – Exemples de peintures de Cézanne colorisées avec SpyncoGan durant l'entraînement ($G_1(I_A) = O_{d_1}^{W \times H}$).

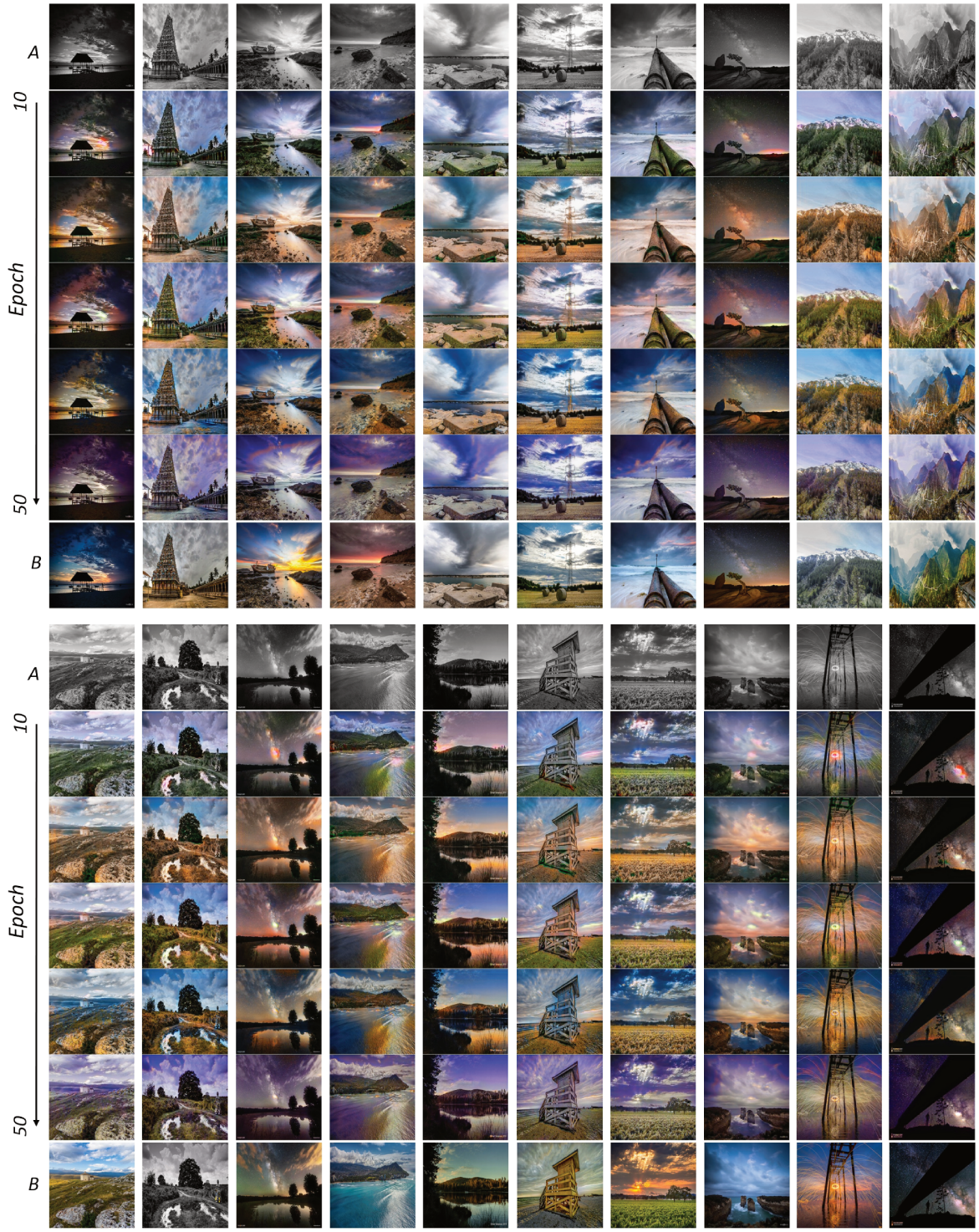


FIGURE 4.20 – Exemples de photographies de paysages colorisées avec SpyncoGan durant l'entraînement ($G_1(I_A) = O_{d_1}^{W \times H}$).

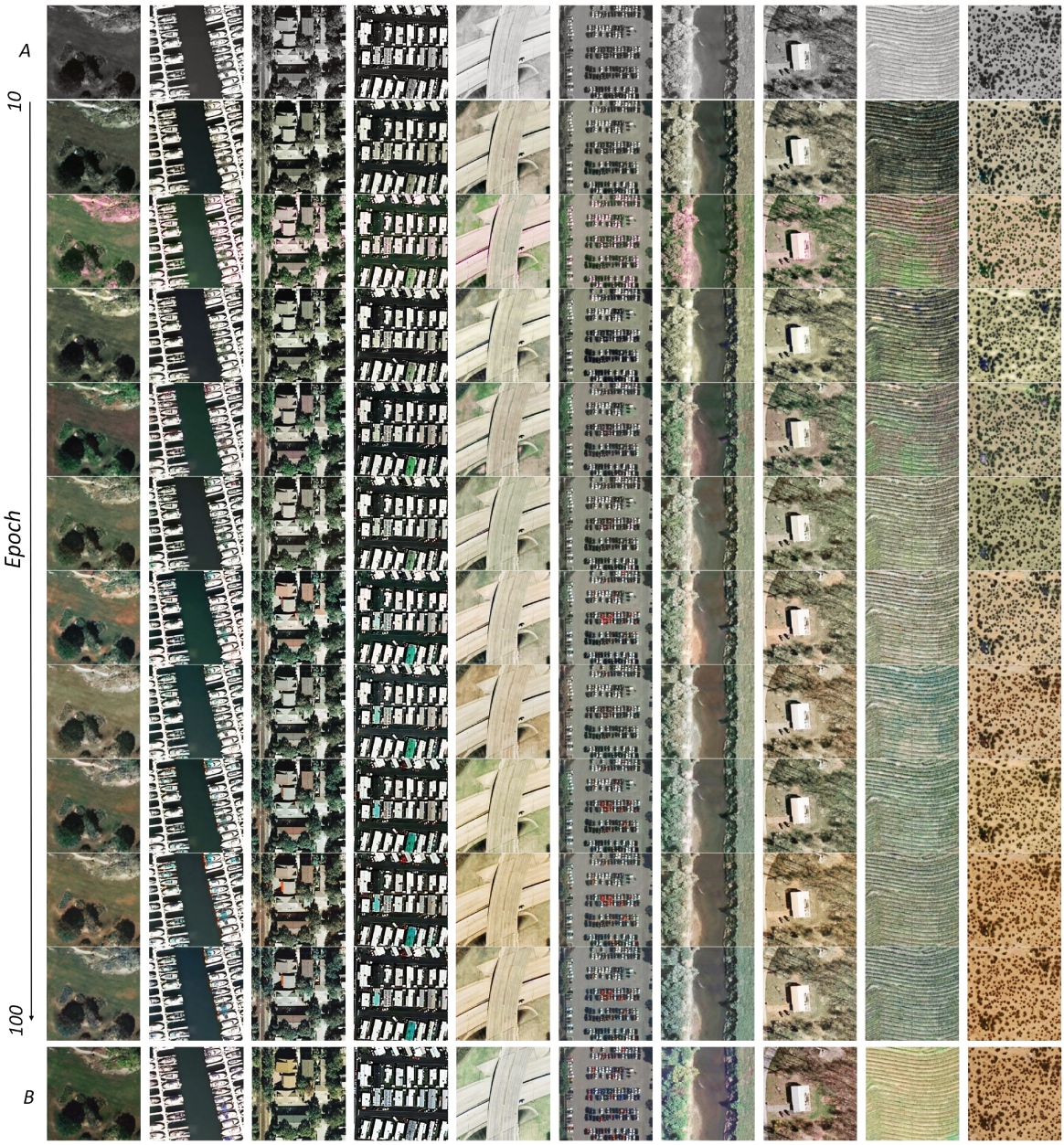


FIGURE 4.21 – Exemples d'images aériennes du jeu de données UCMerced Land Use colorisées avec SpyncoGan durant l'entraînement ($G_1(I_A) = O_{d_1}^{W \times H}$).

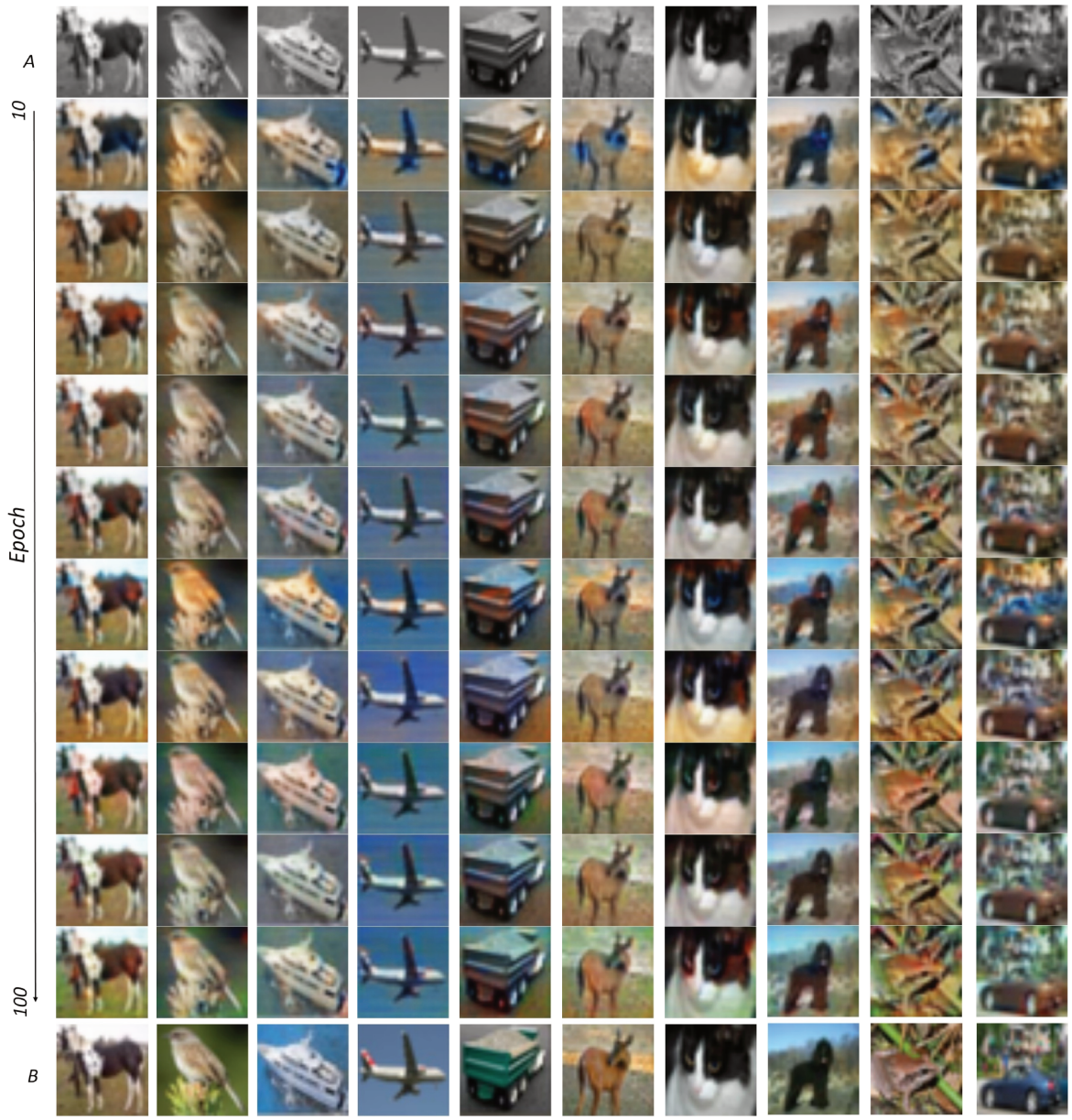


FIGURE 4.22 – Exemples d'images du jeu de données de Cifar-10 colorisées avec SpyncoGan durant l'entraînement ($G_1(I_A) = O_{d_1}^{W \times H}$).

Chapitre 5

Segmentation sémantique et post-traitement

Ce chapitre présente nos travaux sur le post-traitement de segmentations d'images aériennes historiques. Notre but est d'améliorer les cartes d'occupation du sol générées à l'aide d'un logiciel tel que Gouramic (voir Annexe A). D'une part, nous avons cherché à générer des superpixels qui prennent en compte les séparations sémantiques entre les parcelles des images aériennes historiques afin d'obtenir des groupes de pixels réalistes. D'autre part, nous avons étudié l'intégration de l'information portée par ces superpixels au sein d'un champ aléatoire conditionnel afin de contraindre l'inférence de ces modèles graphiques. Face aux résultats encourageants que nous avons obtenus, nous nous sommes alors demandé dans quelle mesure la colorisation automatique, étudiée dans le chapitre précédent, pouvait avoir un intérêt pour le post-traitement des données historiques.

Sommaire

5.1 Introduction 120
5.2 Travaux connexes 121
5.2.1 Bords et bords profonds 121
5.2.2 Champs aléatoires conditionnels 122
5.3 Méthode 123
5.3.1 Détection de bords profonds et représentations basées superpixels 124
5.3.2 Intégration au sein d'un champ aléatoire conditionnel 126
5.4 Expériences et résultats 127
5.4.1 Mise en place 127
5.4.2 Expériences 129
5.4.3 Apport de la colorisation 133
5.5 Conclusion 136

5.1 Introduction

La segmentation sémantique est une tâche qui consiste à attribuer une étiquette à chaque pixel d'une image. Elle permet d'obtenir des informations de hauts niveaux (*i.e.*, qui ont un sens pour l'humain) dans un large éventail d'applications. Par exemple, en télédétection, la segmentation sémantique peut correspondre à la génération automatique de cartes d'occupation du sol. Elle permet d'accélérer significativement les analyses à moindre coût, en réduisant le nombre d'interventions humaines. Dans d'autres thématiques, telles que la conduite de véhicules autonomes, les informations de hauts niveaux que l'on aimerait obtenir peuvent correspondre à la localisation de zones libres, où le véhicule peut circuler, et d'obstacles. Ces éléments permettent d'envisager des stratégies avancées dans les prises de décisions pour la conduite du véhicule. Pour le traitement d'images médicales, la segmentation sémantique permet, par exemple, de cibler les zones d'intérêts à observer afin de qualifier ou de quantifier certaines pathologies.

De par leurs nombreuses applications, les méthodes de segmentation sémantique ont reçu une attention particulière au sein de la communauté de vision par ordinateur. L'apparition de nombreuses méthodes basées sur des réseaux de neurones profonds à convolutions témoigne de cet engouement. Cependant, les résultats obtenus à l'aide de méthodes de segmentation sémantique ne s'attachent pas toujours correctement aux bords des objets présents dans l'image, et ils peuvent parfois manquer de cohérence spatiale (*e.g.*, pixels isolés mal étiquetés). Ces deux cas de figures se retrouvent notamment dans les algorithmes de classification au pixel près (*i.e.*, chaque pixel est classifié indépendamment de ses voisins). Des observations similaires ont pu être réalisées sur les résultats obtenus à l'aide de DCNN. Afin de tenir compte de ces problématiques, des méthodes basées sur des superpixels ont été proposées. Elles consistent à calculer puis à classifier des superpixels plutôt que des pixels (*i.e.*, on attribue une même étiquette à tous les pixels d'un groupe homogène de pixels). Pour rappel, ces méthodes sont généralement référencées sous le terme *Object Based Image Analysis* (OBIA) en télédétection (voir chapitre 2). L'utilisation d'algorithmes de post-traitement a également été explorée afin d'améliorer les pipelines de segmentation. Contrairement à l'utilisation de superpixels, ces algorithmes sont utilisés après avoir obtenu un premier résultat. Il s'agit alors de raffiner le résultat obtenu en regardant à la fois la segmentation sémantique initiale et l'image qui a été utilisée pour la générer. En particulier, l'utilisation de champs aléatoires conditionnels (*Conditional Random Fields*, CRF) a gagné en popularité en proposant de représenter une image à l'aide d'un graphe permettant de moduler une segmentation préalablement obtenue en exploitant les corrélations entre les pixels. Ces approches de post-traitement ont l'avantage de pouvoir être, généralement, appliquées sans avoir connaissance de l'algorithme de segmentation initial.

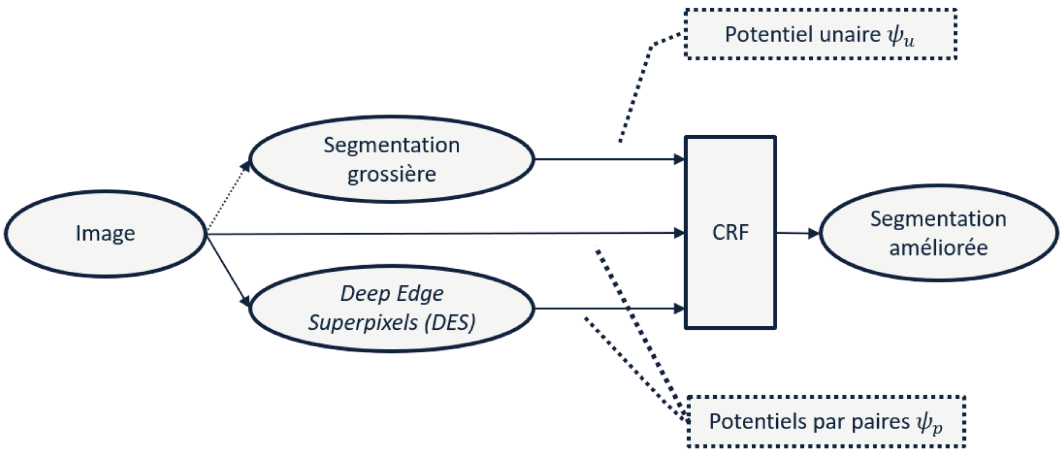


FIGURE 5.1 – Schéma générique de l'approche proposée, sans colorisation.

Dans le cadre nos travaux, nous souhaitions étudier l'intérêt de combiner les champs aléatoires conditionnels et les superpixels pour améliorer les cartes d'occupation du sol obtenues à l'aide du logiciel Gouramic (voir Annexe A). Nous avons cependant remarqué que les algorithmes de génération de superpixels sont en général construits à l'aide d'un nombre non négligeable de paramètres, et ont tendance à générer des superpixels relativement petits (*i.e.*, sur-segmentation) qui ne tiennent pas forcément compte des bords sémantiques entre les objets présents dans l'image (*e.g.*, beaucoup de superpixels se retrouvent au sein d'une même parcelle d'occupation du sol). Afin de tenir compte de ces potentielles faiblesses, nous avons dans un premier temps cherché à générer des superpixels à partir de la sortie d'un réseau de neurones à convolutions entrainé pour détecter des bords sémantiquement intéressants, dits bords profonds (*deep edges*). Fortement inspirés par les travaux de [SAA18], nous avons ensuite étudié l'intégration de l'information portée par ces superpixels au sein d'un champ aléatoire conditionnel dense (voir figure 5.1), que nous décrivons dans ce chapitre. Enfin, nous nous sommes intéressés à l'apport de la colorisation pour le post-traitement à l'aide de champs aléatoires conditionnels.

5.2 Travaux connexes

Nous présentons ici les travaux connexes à l'utilisation de bords profonds et de champs aléatoires conditionnels. Nous reviendrons plus en détails sur les méthodes utilisées dans la section suivante.

5.2.1 Bords et bords profonds

Un bord représente une séparation spatiale entre deux pixels au sein d'une image. Il est représenté par un gradient d'intensité. Les bords sont à distinguer des contours. D'après la définition des contours aux sens de Canny [Can86] : les contours doivent être binaires (seuillage) et constitués d'un unique pixel afin de satisfaire un critère de localisation (*i.e.*, un contour ne peut pas être plus épais que l'unité de base qu'est le pixel). Les bords n'ont tout simplement pas ces contraintes : ils ont des valeurs continues et non discrètes, et ils peuvent être épais. En cela, un bord est généralement considéré comme une représentation grossière d'un contour.

Pour détecter les bords, les approches usuelles se basent sur des filtres de convolutions représentant des dérivées spatiales, tels que le filtre de Sobel ou le Laplacien de la gaussienne. Ces filtres ont des réponses linéaires en intensité : ils génèrent des valeurs identiques pour des changements d'intensités de même amplitude, et ce même si ces changements d'intensités correspondent à des sémantiques différentes. On peut néanmoins mettre en opposition les bords contenus au sein d'un objet d'intérêt, correspondant à des attributs internes à l'objet (*e.g.*, la texture d'un champ ou de l'écorce d'un arbre), et les bords correspondants aux séparations des objets que l'on souhaite

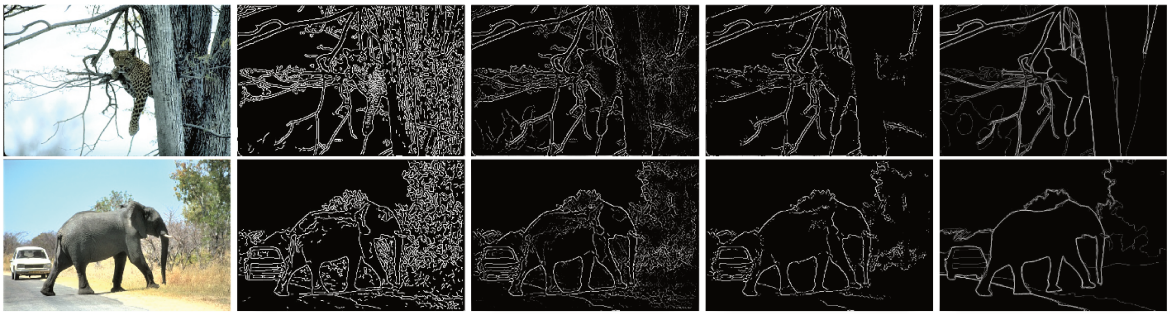


FIGURE 5.2 – Illustration de la différence entre contours, bords profonds, et bord profonds seuillés. De gauche à droite : image d'entrée, filtre de Canny, bords profonds, bords profonds seuillés, vérité terrain. Image extraite de [BST15].

détecter (*e.g.*, séparation entre une parcelle de forêt et une parcelle de prairie).

Afin de tenir compte de la sémantique des objets et de détecter prioritairement des bords correspondants aux entités que l'on aimerait détourer, des approches combinant traitement d'image et apprentissage machine ont été proposées [AMFM11]. Dans ce manuscrit, nous nous intéressons particulièrement aux approches exploitant des réseaux de neurones profonds à convolutions. Ces approches ont permis d'obtenir des résultats compétitifs sur plusieurs jeux de données standards [XT15]. L'idée derrière l'utilisation de réseaux de neurones à convolutions pour détecter des bords est qu'ils vont permettre d'apprendre des filtres aptes à donner plus d'importance aux gradients d'intensités ayant une connotation sémantique, et de "gommer" les autres. Pour cela, les détecteurs de bords sont entrainés à segmenter une image en deux catégories, à savoir les pixels de fond et les pixels de bords. Une fois entraîné, un détecteur génère une carte de probabilités, indiquant pour chaque pixel sa probabilité d'appartenir à un bord sémantiquement intéressant (voir figure 5.2).

Les bords profonds ont à ce jour trouvé moult applications en télédétection. Marmanis *et al.* [MSW+18] ont proposé d'intégrer des bords profonds sous forme de canaux supplémentaires aux images IRGB et RGBD des jeux de données IPSRS Potsdam et Vaihingen. Les images avec ces canaux supplémentaires ont été utilisées afin d'entraîner des réseaux de neurones entièrement convolutifs tels que SegNet [BKC17] et FCN-8 [LSD15], populairement utilisés pour la segmentation sémantique. Les résultats obtenus ont montré l'intérêt d'intégrer explicitement des bords profonds afin de contraindre la génération de segmentations sémantiques plus proches de la vérité terrain. Chen *et al.* [CBP+16] ont étudié l'intérêt de prédire des bords profonds en plus des cartes de segmentation sémantique à l'aide d'un même réseau de neurones. Le but était ici de contraindre l'apprentissage de représentations cohérentes par rapport aux bords afin d'améliorer implicitement les résultats du réseau pour la segmentation. Les bords profonds ont également trouvé des applications pour le détourage de parcelles de champs de cultures [MPT20; GPLSREGM19], la génération automatique de cartes cadastrales [XPK19; CKYV19], ou encore l'estimation de réseaux routiers [XXFC18].

Nos travaux se positionnent ici à mi-chemin entre le détourage de parcelles et l'intégration de bords profonds pour l'amélioration de segmentations sémantiques. Dans notre cas, le détourage de parcelles constitue une étape intermédiaire à l'amélioration de segmentations obtenues *a priori*.

5.2.2 Champs aléatoires conditionnels

Les champs aléatoires conditionnels ont été largement étudiés dans la littérature pour le post-traitement de segmentations sémantiques. Pour cela, une image I est représentée à l'aide d'un graphe G dont les vertex sont les pixels de l'image. A chaque pixel sont associées des caractéristiques et à une étiquette estimée. L'étiquette estimée est généralement assortie d'une probabilité, qui peut-être obtenue soit en sortie d'un algorithme de segmentation, soit fixée manuellement (cas où l'algorithme de segmentation n'est pas accessible). Cette probabilité est régulièrement appelée potentiel unaire (*i.e.*, potentiel indépendant pour chaque pixel). Le but des algorithmes de type CRF est alors de moduler le potentiel unaire en tenant compte des relations d'adjacences entre les pixels (étape d'inférence). Celles-ci sont représentées à l'aide de potentiels par paires, qui indiquent la proximité des pixels dans l'espace dans caractéristiques (*e.g.*, différence entre l'intensité de deux pixels connexes sur le graphe). En se basant sur ce principe, plusieurs approches ont vu le jour.

Triggs et Verbeek [TV08] ont cherché à agréger les informations portées par des représentations à plusieurs échelles afin de tenir compte des observations locales et globales en se basant sur des sous-ensembles de pixels rectangulaires (imagettes, *patch*). Krahenbühl *et al.* [KK11] ont

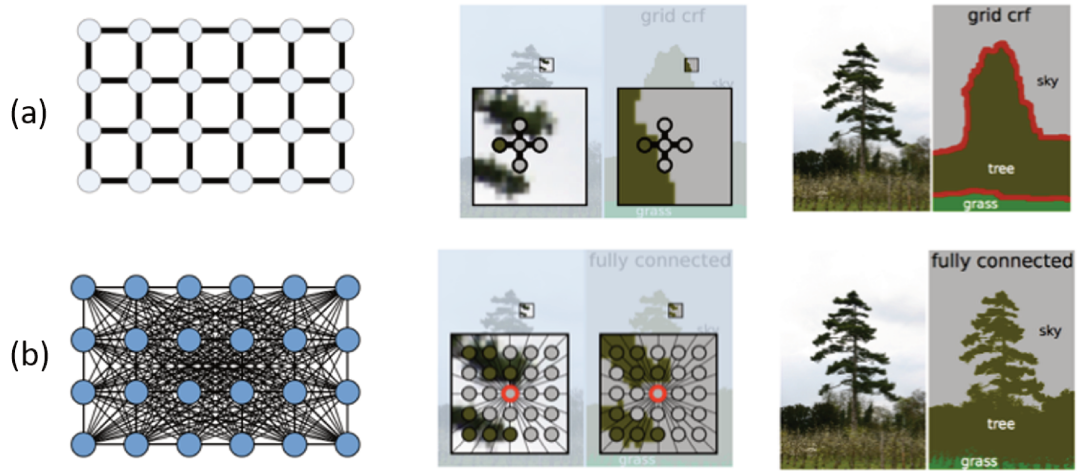


FIGURE 5.3 – Illustration d'un CRF basé sur des relations d'adjacence selon une grille régulière (a) et d'un CRF dense (b). Le fait de pouvoir tenir compte des relations entre pixels éloignés permet d'obtenir des résultats plus réalistes. Images extraites de la présentation de [KK11].

décrit un algorithme efficace pour traiter le cas où tous les pixels de l'image seraient considérés comme étant connectés entre eux (voir figure 5.3). Le fait d'utiliser un CRF entièrement connecté (*fully-connected* CRF), aussi appelé CRF dense (DenseCRF), permet de tenir compte de l'information portée par des pixels éloignés les uns des autres, sans se restreindre à un voisinage local. Cependant, cette représentation est particulièrement coûteuse en ressources puisqu'il faut tenir compte de l'intégralité des pixels, et ce pour chaque pixel analysé. Afin de réduire ce coût algorithmique, l'idée de Krahenbühl *et al.* était de se baser sur des filtres gaussiens pour calculer les potentiels par paires, et de réaliser une approximation du processus d'inférence à l'aide de l'algorithme de champs moyen (*mean field approximation*). Cela a permis aux auteurs de réduire les temps de traitement de plusieurs heures à quelques secondes comparé aux méthodes précédentes basées sur des modèles denses. Kohli *et al.* [KT$^+$09] ont quant à eux poposé le modèle P^N Potts, qui modélise un CRF à l'aide de superpixels afin de limiter les calculs à réaliser (il y a moins de superpixels que de pixels dans une image). Sulimowicz *et al.* [SAA18] ont formulé l'intégration de superpixels au sein d'un CRF dense en attribuant la valeur moyenne de chaque superpixel aux pixels qui le compose, et en utilisant le résultat obtenu sous la forme d'un potentiel par paires supplémentaire. L'idée est ici de contraindre les pixels d'un même superpixel à avoir la même étiquette (potentiel nul), et de moduler les différences inter-superpixels. Il a été montré [SAA18] que cette formulation est équivalente à celle utilisée par le modèle P^N Potts, mais exprimé avec le formalisme du DenseCRF. Ce dernier point permet l'intégration *ad-hoc* de l'information portée par les superpixels au sein de l'algorithme de [KK11]. Zheng *et al.* [ZJRP$^+$15] ont quant à eux proposé de formuler l'algorithme d'inférence du DenseCRF à l'aide d'un réseau de neurones récurrents (CRFasRNN) afin de pouvoir optimiser simultanément le réseau de neurones et les paramètres des filtres gaussiens.

Dans nos travaux, nous nous sommes inspirés de la méthode proposée par Sulimowicz *et al.* [SAA18] afin d'intégrer l'information portée par les superpixels au sein d'un CRF dense. A la différence de Sulimowicz *et al.*, les superpixels que nous utilisons sont générés à partir de bords profonds. Nous montrons l'intérêt de ces superpixels sur nos données.

5.3 Méthode

Notre méthode peut se décomposer en deux blocs. Le premier consiste à générer des représentations issues de superpixels, eux-mêmes générés à partir de bords profonds. Le second est le CRF exploitant ces représentations afin de raffiner les segmentations grossières.

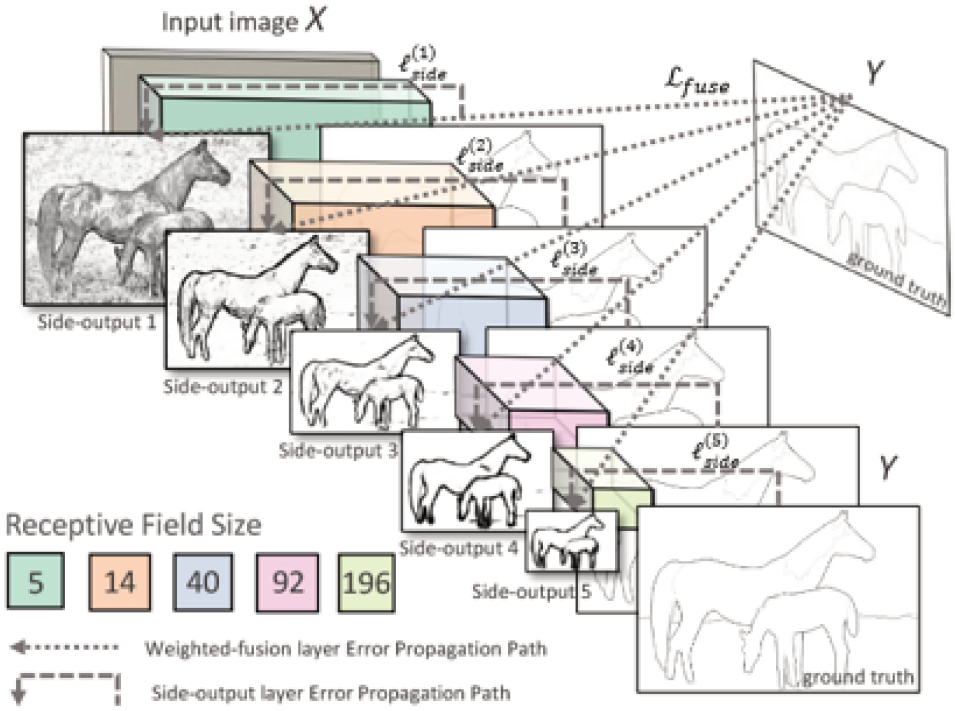


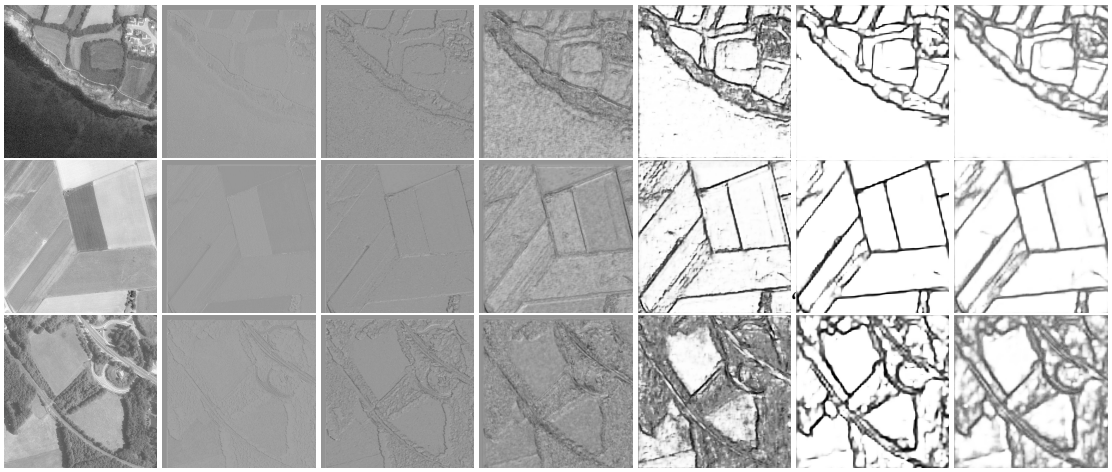
FIGURE 5.4 – Illustration du réseau de neurones HED. Image extraite de [XT15]. A partir d'une image source, des bords profonds sont générés à plusieurs échelles, puis fusionnés à l'aide d'un filtre de convolutions.

5.3.1 Détection de bords profonds et représentations basées superpixels

Détection des bords profonds

Nous avons cherché à générer des bords profonds pour nos images aériennes historiques de très haute résolution (VHR) comme étape intermédiaire à la création de superpixels. Pour cela, nous nous sommes basé sur le modèle de détection de bords heuristiques (*Holistic Edge Detector*, HED) [XT15]. HED est un réseau de neurones entièrement convolutif [LSD15] basé sur le réseau VGG16 [SZ14] auquel on aurait retiré les couches entièrement connectées. Il permet de générer des bords profonds à plusieurs échelles (5 échelles) en transformant les cartes d'activations intermédiaires obtenues avant chaque couche de *pooling* vers l'espace de sortie en utilisant des convolutions transposées. Ces bords profonds multi-échelles sont ensuite concaténés dans la dimension des canaux et fusionnés ensemble par un filtre de convolutions. Cette sortie fusionnée représente la sortie finale du réseau. Elle est basée sur l'idée que les cartes d'activations plus profondes transportent plus d'informations sémantiques que les cartes d'activations moins profondes, mais au prix de représentations plus grossières (moins bien localisées). Fusionner ces informations à l'aide d'un filtre convolutif devrait permettre de mixer finesse des résultats et aspects sémantiques. Le réseau HED avec un exemple d'image traitée est présenté sur la figure 5.4.

Dans nos travaux, nous avons suivit [MSW+18] et utilisé la sortie fusionnée pour représenter nos bords profonds. Étant donné que les bords que nous aimerions détecter sont naturellement sous-représentés par rapport à l'arrière-plan, la fonction de coût utilisée pour entraîner HED est l'entropie croisée équilibrée par classe (voir équation (5.1)) [XT15]. Elle permet de pondérer l'importance de l'erreur calculée pour les différentes classes afin d'éviter de négliger la génération de pixels de bords (*i.e.*, empiriquement, le réseau pourrait vouloir tricher en ne prédisant jamais les pixels de bords car il n'y en a pas beaucoup). Pour l'optimisation, l'algorithme de descente stochastique du gradient est utilisé.



(a) (b) (c) (d) (e) (f) (g)

FIGURE 5.5 – Résultats obtenus avec HED pour la détection de bords sur une imagette de 1024x1024 pixels après 10000 itérations d'entraînement. (a) Imagette, (b) à (f) les sorties intermédiaires du réseau, (g) résultat de la fusion linéaire apprise par le réseau et appliquée sur les 5 sorties.

$$\begin{aligned} \mathscr{L} = &-\beta \sum_{i \in \mathrm{Y}+} log(p(y_i = 1|\mathrm{X};\mathrm{W};w^m) \\ &-(1-\beta) \sum_{i \in \mathrm{Y}-} log(p(y_i = 0|\mathrm{X};\mathrm{W};w^m) \end{aligned} \tag{5.1}$$

Dans l'équation (5.1), W représente les poids du réseau de base (VGG-16 ici), w^m représente les poids associés à une sortie intermédiaire m, Y+ sont les étiquettes de vérité terrain pour le fond, Y− sont des étiquettes de vérité terrain pour les bords, et $\beta = \frac{\mathrm{Y}-}{\mathrm{Y}}$ et $1-\beta = \frac{\mathrm{Y}+}{\mathrm{Y}}$.

Nous présentons des exemples de bords profonds obtenus avec les différentes sorties de HED après 10 000 *epochs* d'entraînement sur la figure 5.5. Ces bords profonds ont été générés pour des images aériennes historiques. Nous avons ici inversé les valeurs des résultats obtenus pour que les bords détectés soient plus facilement visibles sur papier blanc. On peut observer que les sorties les moins profondes (b)-(d) préservent effectivement de nombreuses hautes fréquences qui ont tendance à s'effacer par la suite. On remarque également que les dernières sorties permettent d'obtenir une meilleure séparation des classes (bords, non bords). La sortie fusionnée (g) fournie une représentation lissée de la sortie (f). On distingue en effet que les bords sont moins crénelés sur (g) que sur (f), mais ils sont plus flous. Comme attendu, cette sortie fusionnée semble également intégrer ces informations provenant des sorties précédentes : une image (g) possède plus de pixels sombres qu'une image (f). Ce dernier point permet de préserver des bords ou des bouts de bords qui auraient autrement été tronqués.

Des bords profonds aux superpixels

Une fois les bords profonds générés, nous avons choisi d'utiliser l'algorithme de partage des eaux [BM93] afin d'obtenir des superpixels. Pour rappel, cet algorithme va créer des groupes de pixels en se basant sur une carte de gradients. Cette dernière est ici simplement remplacée par les bords profonds. Nous supposons que ceux-ci permettent de réduire la prise en compte des hautes fréquences qui ne correspondent pas à des séparations sémantiquement intéressantes comparés aux opérateurs de gradients classiques. Ce processus est fortement inspiré des travaux réalisés par [AMFM11]. Les auteurs proposaient d'utiliser les résultats obtenus en sortie d'un détecteur de bords afin de générer une hiérarchie de contours à l'aide de la *Oriented Watershed Transform*. Pour cela, les auteurs se basaient sur des caractéristiques extraites à l'aide de filtres "artisanaux" et de techniques d'apprentissage automatique. Cette idée fût ensuite reprise par [XKP19] afin d'esti-

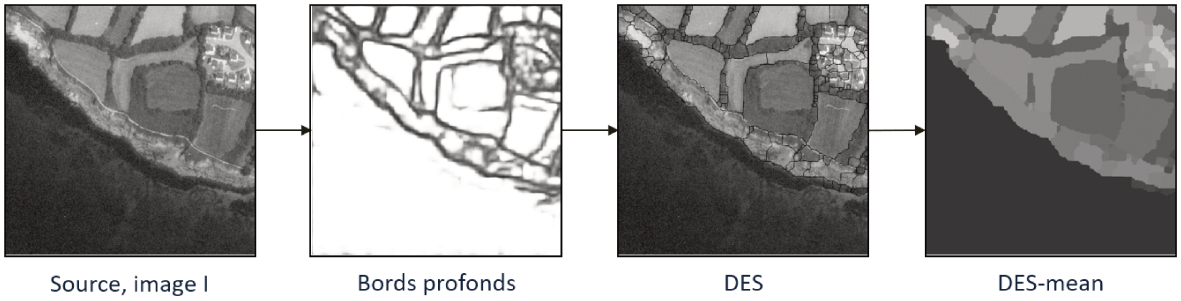


FIGURE 5.6 – Schéma illustrant la génération de représentations lissées à l'aide de superpixels extrait de bords profonds.

mer des cartes cadastrales à partir de bords profonds. Ici, l'algorithme de partage des eaux permet d'obtenir des groupes de pixels que nous avons nommés *Deep Edge Superpixels* (DES).

Dans notre cas, nous ne nous intéressons pas aux contours des DES générés, mais à l'information contenue dans les zones qu'ils représentent. En particulier, nous utilisons les DES pour générer une représentation lissée de l'image, d'une façon tout à fait similaire à l'approche proposée par [SAA18] avec des superpixels plus classiques. Pour cela, nous calculons la valeur moyenne de l'intensité des pixels de chaque superpixel, et nous assignons cette valeur à tous les pixels du superpixel. Chaque superpixel est alors représenté par sa valeur moyenne, ce qui signifie que la différence d'intensité entre deux pixels au sein d'un même superpixel sur la représentation lissée est nulle. A noter que dans le cadre de nos expériences, nous avons également étudié l'intérêt d'utiliser la valeur médiane à la place de la moyenne. Nous nommons respectivement ces deux représentations lissées DES-*mean* et DES-*median*. Une illustration de ce processus est présentée sur la figure 5.6 pour la valeur moyenne.

5.3.2 Intégration au sein d'un champ aléatoire conditionnel

Nous définissons ici la façon dont nous avons intégré l'information portée par les DES au sein d'un champ aléatoire conditionnel.

L'estimation des étiquettes à l'aide d'un champ aléatoire conditionnel est réalisée en minimisant une énergie de Gibbs, représentée comme la somme d'un potentiel unaire ψ_u et de potentiels par paires ψ_p. D'après la formulation du CRF dense de [KK11], nous pouvons noter i et j les indices de deux pixels, avec des étiquettes x_i et x_j. Alors, l'énergie de Gibbs peut-être définie par l'équation (5.2).

$$E(x) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j) \tag{5.2}$$

Pour rappel, le potentiel unaire ψ_u représente la probabilité qu'une étiquette particulière soit associée au pixel i. Afin de moduler cette probabilité, les potentiels par paires prennent une forme générique donnée par l'équation (5.3), qui représente une combinaison linéaire de N noyau gaussiens (un par potentiel) : $k^{(n)}$, $n \in \{1,...,N\}$. Dans cette équation, $\mu(x_i, x_j) = 1_{[x_i \neq x_j]}$ (0 sinon) défini un modèle de Potts, et f_i et f_j sont les vecteurs de caractéristiques associés aux pixels i et j.

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \overbrace{\sum_{n=1}^{N} \omega^{(n)} k^{(n)}(f_i, f_j)}^{k(f_i, f_j)} \tag{5.3}$$

Ici, nous modélisons les potentiels par paires à l'aide de N = 3 noyaux gaussiens, tels que représentés par l'équation (5.4). Les deux premiers noyaux de l'équation (5.4) modélisent les poten-

tiels sensibles au contraste (*contrast-sensitive*), comme définis par [KK11]. Ils permettent d'intégrer l'information portée par la couleur (ou l'intensité du niveau de gris dans notre cas) I_i et I_j, ainsi que l'information portée par la position des pixels P_i et P_j. L'idée est ici que des pixels spatialement proches avec des couleurs similaires devraient avoir la même étiquette. Ces deux filtres ont pour paramètres θ_γ, θ_α et θ_β qui correspondent aux déviations standards du filtre gaussien. En complément, nous intégrons un troisième noyau qui représente l'information portée par les pixels de l'image DES-*mean*, dont les intensités sont représentées par DES_i et DES_j. Son but est de pénaliser deux pixels appartenant à des superpixels différents. Nous avons dans un premier temps fixé ce noyau sous la forme d'un potentiel générique (pas de paramètres, voir équation (5.4)). Nous avons ensuite étendu cette formulation à l'utilisation d'un potentiel bilatéral, similaire au noyau $k^{(2)}$.

$$\begin{aligned} k(f_i, f_j) =& \omega^{(1)} exp(-\frac{P_i - P_j}{2\theta_\gamma^2}) \\ &+ \omega^{(2)} exp(-\frac{|P_i - P_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}) \\ &+ \omega^{(3)} exp(\frac{-|DES_i - DES_j|^2}{2}) \end{aligned} \tag{5.4}$$

En pratique, les poids ω peuvent être optimisés. Dans notre cas, nous les avons fixés manuellement afin d'étudier l'importance relative des caractéristiques de l'image initiale et de l'image lissée en tenant compte des bords profonds (voir section 5.4.3).

5.4 Expériences et résultats

Nous présentons dans cette section les expériences que nous avons réalisées, ainsi que les résultats que nous avons obtenus en post-traitement.

5.4.1 Mise en place

Données initiales

Afin de réaliser nos expériences, nous nous sommes basés sur 17 images aériennes historiques de très hautes résolutions annotées manuellement à l'aide de 7 classes d'occupation du sol, plus une classe supplémentaire représentant la catégorie "autre" (*e.g.*, stade) qui a été ignorée dans nos traitements. Parmi ces images, nous en avons réservées 9 pour entraîner le détecteur de bords HED, et 8 que nous avons utilisées pour évaluer les résultats obtenus en segmentation sémantique. Nous avons nommé ces deux ensembles de données d_e (*dataset edges*) et d_s (*dataset segmentation*). La distribution des étiquettes pour ces images est donnée sur la figure 5.7. Cette répartition nous montre que les images possèdent, globalement, des répartitions de classes similaires, ce qui devrait permettre d'améliorer les performances du détecteur de bords lors de l'inférence.

Génération de segmentations sémantiques grossières

Afin d'obtenir des résultats initiaux de segmentation sémantique, nous avons choisi de mimer le processus d'utilisation du logiciel Gouramic (voir Annexe A). Pour chaque image aérienne historique de d_s, nous échantillonnons aléatoirement 300 pixels par classe. Nous extrayons ensuite une imagette de 100 × 100 pixels centrée sur chaque pixel échantillonné, à partir de laquelle nous calculons un histogramme de textures à l'aide du filtre LCoLBP présenté dans le chapitre 3. Chaque imagette de 100 × 100 pixels est ici considérée comme étant annotée par son pixel central. Les histogrammes calculés sont ensuite utilisés pour entraîner une forêt aléatoire d'arbres de décisionnels composée de 100 arbres. Nous entraînons une forêt aléatoire d'arbres décisionnels

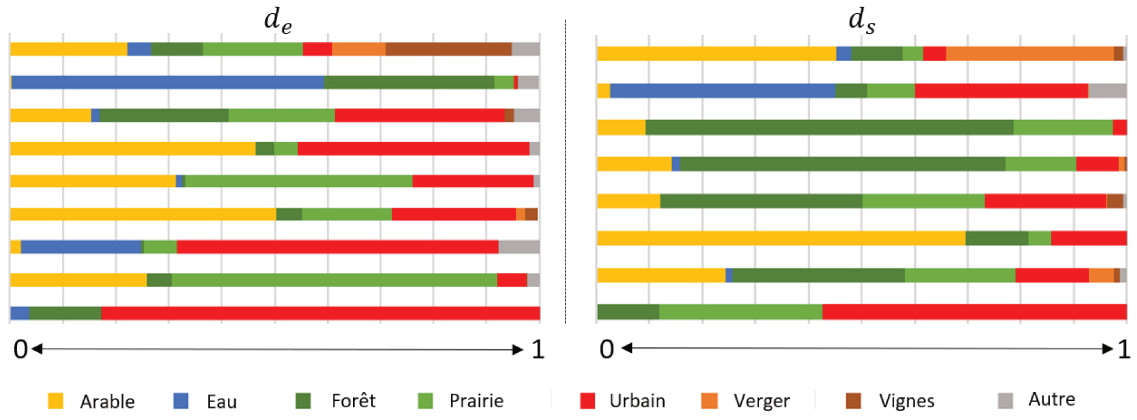


FIGURE 5.7 – Distribution des étiquettes/classes dans les jeux de données utilisés.

pour chaque image aérienne (voir chapitre 3). Nous générons ensuite un résultat de classification sur l'ensemble de l'image à l'aide d'une fenêtre glissante parcourant une grille régulière. La taille de la fenêtre glissante est égale à la taille des imagettes utilisées pour entraîner la forêt d'arbres décisionnels. Le pas entre deux positions de la fenêtre est quant à lui fixé à 25 pixels, horizontalement comme verticalement. Le résultat obtenu par classification est ici attribué à tous les pixels se trouvant dans une aire de 25×25 pixels, centrée sur la fenêtre glissante (taille d'une cellule). Il s'agit ici d'une extension au niveau de l'imagette des algorithmes de classification au pixel près. L'utilisation d'une approche à l'imagette près présente l'avantage d'accélérer significativement les temps de traitements (*e.g.*, il y a ici $25 \times 25 - 1$ fois moins de calculs à réaliser) par rapport à une approche au pixel près. En contrepartie, les résultats ont un niveau de finesse moindre, défini par la taille du pas de la fenêtre glissante dans notre cas. Des exemples de segmentations grossières ainsi obtenues sont visibles sur la figure 5.8. On y constate le crénelage occasionné par la classification à l'imagette près, ainsi que la présence d'imagettes incorrectement classifiées. Ces résultats grossiers représentent nos segmentations sémantiques de base.

Détection des bords profonds et des DES-*mean* associées

Afin d'entraîner le détecteur de bords profonds en tenant compte des contraintes mémoires, nous avons choisi de travailler avec des imagettes de 1024×1024 pixels. Pour cela, nous avons extrait 1545 imagettes de 1024×1024 pixels de d_e avec 75% de recouvrement entre deux imagettes. Pour chaque imagette, nous avons obtenu des bords vérité terrain en appliquant un filtre de Sobel sur les annotations manuelles de segmentation sémantique, puis nous avons binarisé le résultat (seul fixé à 1). Nous avons ensuite entraîné le réseau de neurones HED pour 10000 itération avec un taux d'apprentissage initiale de $1e^{-6}$ à l'aide du code fournit par les auteurs.

Une fois le détecteur de bords entraîné, nous découpons des imagettes de 1024×1024 pixels sans recouvrement à partir des images aériennes panchromatiques de d_s (130 imagettes), ainsi que les segmentations sémantiques grossières correspondantes. Nous appliquons le détecteur de bords entraîné sur ces imagettes, avant de générer des DES et leurs représentations moyennes / lissées correspondantes.

Résumé des données

A ce stade, nous avons donc 130 imagettes (d_s) de 1024×1024 pixels, pour lesquelles nous avons une segmentation grossière (et une vérité terrain), des DES et une image de DES-*mean*. Notre but va maintenant être de déterminer quel est l'apport de chacun de ces éléments pour le post-traitement des segmentations grossières.

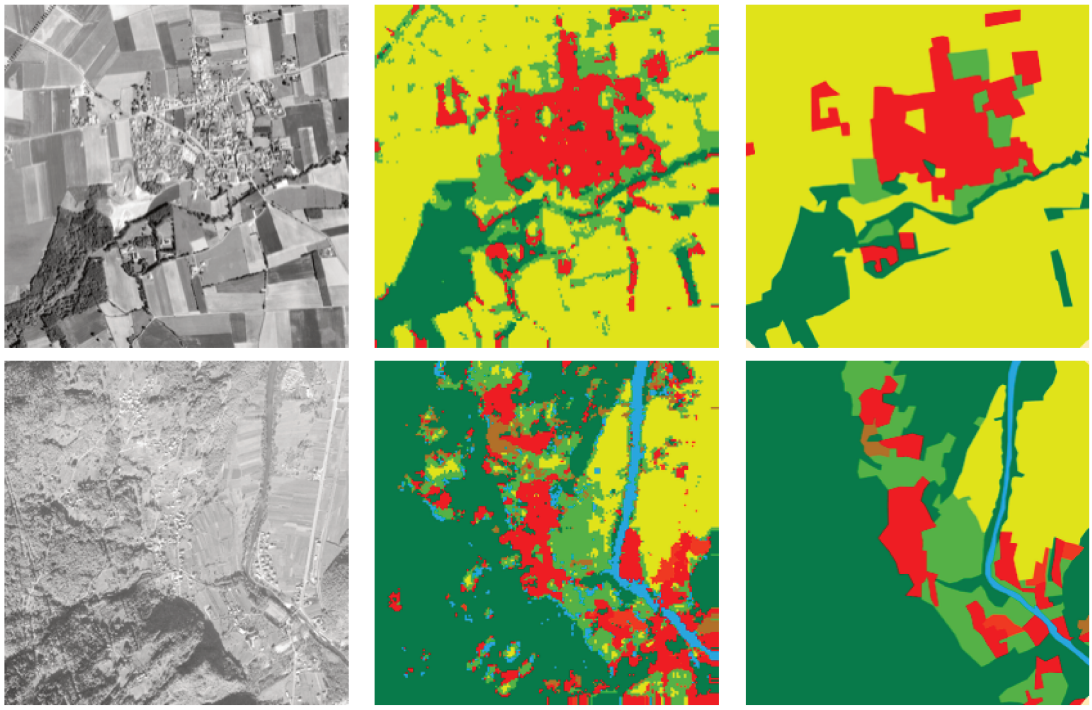


FIGURE 5.8 – Exemples d'images aériennes segmentées grossièrement (taille originale : 4500 × 4500 pixels). Gauche : images aériennes panchromatiques. Milieu : segmentations grossières. Droite : Vérités terrains.

5.4.2 Expériences

Utilisation des Deep Edge Superpixels : vote majoritaire

Nous avons dans premier temps étudié l'intérêt des DES par rapport à d'autres algorithmes de superpixels. Pour cela, nous avons commencé par réaliser une étude par vote majoritaire. Pour chaque superpixel, nous comptons le nombre de fois qu'une étiquette apparait, et nous associons l'étiquette la plus fréquente à tous les pixels du superpixel. L'idée est ici de mettre en avant quel algorithme est le plus prompt à générer des groupes de pixels qui permettent de réduire les écarts entre étiquettes estimées et étiquettes réelles.

Nous avons ainsi comparé les DES aux algorithmes de SLIC, ETPS et FH présentés dans le chapitre 2 [ASS+12; YBFU15; FH04]. Pour cela, nous avons utilisé les implémentations proposées par Stutz *et al.* [SHL18] avec les paramètres par défaut des méthodes (ceux de OpenCV pour FH). Pour les algorithmes SLIC et ETPS, nous avons fait varier le nombre de superpixels entre 300 et 3000 (sur imagettes de 1024 × 1024 pixels) avec un nombre d'itérations fixé à 10 et sans appliquer d'algorithme de fusion des superpixels. Il va de soi que modifier les paramètres des algorithmes, modifier le nombre d'itérations et tenter de fusionner les superpixels entre eux devrait permettre d'obtenir des résultats différents. Seuls les meilleurs résultats sont présentés ici pour ne pas polluer la lecture. Nous avons cependant remarqué que moins nous avions de superpixels, meilleurs étaient les résultats pour un post-traitement de type vote majoritaire. Cela tend à indiquer que nos données sont particulièrement sensibles au phénomène de sur-segmentation.

Les résultats obtenus par vote majoritaire sont présentés sur le tableau 5.1, à l'aide de plusieurs métriques usuellement utilisées en segmentation sémantique, à savoir l'intersection sur l'union moyenne (m-IoU), l'intersection sur l'union pondérée (f-IoU) et le taux de bonne classification au pixel près. Pour l'ensemble de ces métriques, plus la valeur est élevée, meilleur est le résultat. On constate que les résultats obtenus avec les DES sur d_s sont supérieurs à ceux obtenus avec les autres algorithmes (1.6% supérieurs en taux de bonne classification par rapport à FH). On observe

également que les algorithmes SLIC et ETPS ne permettent pas d'obtenir des résultats au niveau de la méthode FH. Ces résultats tendent à indiquer que, sur nos données, l'utilisation de superpixels ayant des tailles variables semble être à privilégier (FH, DES) afin d'améliorer les résultats grossiers de segmentations sémantiques.

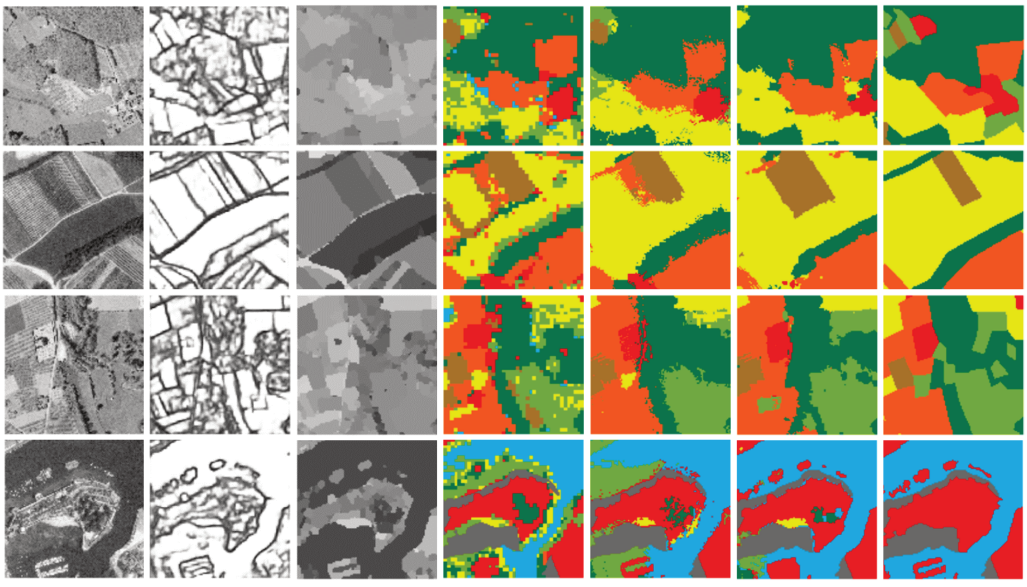
Intégration des Deep Edge Superpixels dans le CRF

Nous avons étendu notre évaluation à la chaîne de traitements complète présentée sur la figure 5.1.

Pour cela, nous avons comparé l'utilisation du CRF dense avec et sans intégration de l'information portée par les DES-*mean*. Nous avons aussi évalué l'intérêt d'intégrer l'information portée par des superpixels plus classiques (algorithme SLIC). Pour cela, nous avons fait varier les paramètres θ_α et θ_β (minimum égal à 3) afin de comparer les méthodes à leur meilleur point de fonctionnement. Les poids associés aux noyaux gaussiens ont été fixés de manière à donner autant d'importance à l'image initiale qu'à l'information portée par les superpixels : $\omega^{(1)} = 3$, $\omega^{(2)} = 10$ and $\omega^{(3)} = 10$. Le taux de confiance (*i.e.*, la probabilité) associée à chaque pixel des résultats grossiers a été fixée à 0.55. L'inférence réalisée pour chaque variante de CRF (avec et sans superpixels) a été réalisée pour 10 itérations à l'aide de l'algorithme de [KK11]. Les résultats que nous avons obtenus sont présentés sur la figure 5.10. Nous pouvons y observer que le CRF seul permet d'obte-

TABLEAU 5.1 – Résultats obtenus avec un post-traitement par vote majoritaire par superpixel sur d_s.

Méthode	m-IoU ↑	f-IoU ↑	Taux de bonne classification ↑
Base	59.5	60.6	74.3
Slic [ASS+12]	61.2	62.1	75.5
Etps [YBFU15]	61.6	62.6	76.0
FH [FH04]	65.9	67.2	79.6
DES	68.7	69.4	81.2



(a) (b) (c) (d) (e) (f) (g)

FIGURE 5.9 – Exemples de résultats. (a) image aérienne panchromatique, (b) bords profonds, (c) DES-*mean*, (d) résultats de segmentation grossière, (e) post-traitement avec CRF dense, (f) post-traitement avec CRF dense et DES-*mean*, (g) vérité terrain.

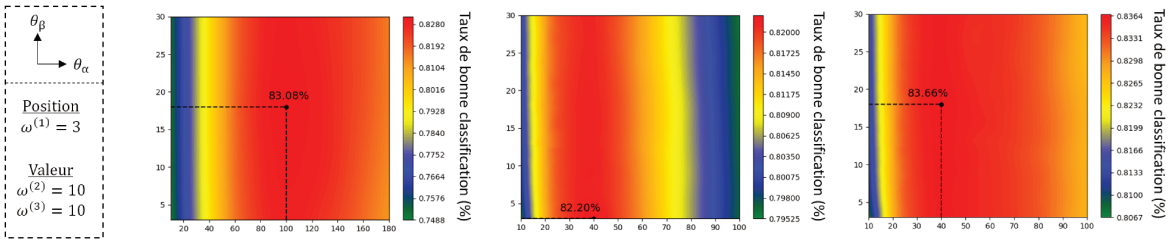


FIGURE 5.10 – Résultats obtenus en intégrant l'information portée par les superpixels au sein d'un champ aléatoire conditionnel dense. (a) CRF dense classique (83.08%), (b) CRF avec superpixels SLIC (82.20%), (c) CRF avec DES (83.66%).

nir des résultats plus élevés qu'avec un vote majoritaire (voir sous-section précédente). Incorporer l'information portée par les DES tend à améliorer ces résultats de 0.58%. Des exemples de résultats obtenus sont présentés sur la figure 5.9. Étonnamment, intégrer l'information portée par les superpixels SLIC semble diminuer l'efficacité du post-traitement sur nos données.

Modification du noyau

Nous nous sommes interrogés sur la forme du noyau gaussien utilisé pour représenter les DES. En particulier, nous avons comparé les résultats obtenus à l'aide d'un noyau générique, tel qu'utilisé dans la section précédente, et ceux obtenus à l'aide d'un noyau bilatéral. L'utilisation du noyau bilatéral était préconisée par Sulimowicz *et al.* [SAA18]. En pratique, passer du noyau générique au noyau bilatéral se traduit par remplacer l'équation (5.4) à l'aide de l'équation (5.5). Dans cette configuration, les paramètres du filtre gaussien sont fixés pour être identiques à ceux utilisés pour le noyau $k^{(2)}$.

$$\begin{aligned} k(f_i, f_j) =& \omega^{(1)} exp(-\frac{P_i - P_j}{2\theta_\gamma^2}) \\ &+\omega^{(2)} exp(-\frac{|P_i - P_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}) \\ &+\omega^{(3)} exp(-\frac{|P_i - P_j|^2}{2\theta_\alpha^2} - \frac{|DES_i - DES_j|^2}{2\theta_\beta^2}) \end{aligned} \tag{5.5}$$

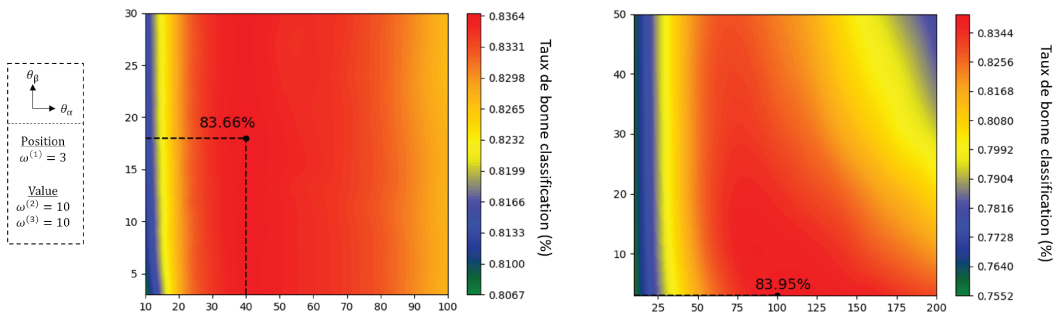


FIGURE 5.11 – Comparaison entre le filtre générique (à gauche, 83.66%) et le filtre bilatéral (à droite, 83.95%) pour l'intégration de l'information portée par les DES.

Les résultats que nous avons obtenus sont présentés sur la figure 5.11. Nous observons un gain de 0.29% avec l'utilisation du noyau bilatéral par rapport à l'utilisation du noyau générique.

Le noyau bilatéral semble donc être à privilégier pour le post-traitement de segmentations sémantiques à l'aide des DES-*mean*. L'ensemble des résultats présentés par la suite seront donc basés sur ce noyau.

Utilisation de la médiane

Nous avons jusqu'à présent utilisé la valeur moyenne des superpixels pour représenter les DES. Nous nous sommes cependant demandé si l'utilisation d'une autre statistique pouvait avoir un intérêt. En particulier, la valeur médiane est régulièrement décrite comme étant plus représentative d'un ensemble que la valeur moyenne. Nous avons de fait comparé les résultats obtenus à l'aide des représentations DES-*mean* et DES-*median* intégrées dans un CRF dense à l'aide d'un filtre bilatéral. D'un point de vue visualisation, la différence entre DES-mean et DES-median est visible sur la figure 5.12. A titre indicatif, nous avons également affiché l'image correspondant à la déviation standard sur cette figure (DES-*std*). On constate que les images DES-*mean* et DES-*median* se ressemblent beaucoup, avec des différences difficilement distinguables à l'œil nu. L'image de déviation standard est quant à elle particulièrement disparate : de nombreux éléments se retrouvent perceptuellement fusionnés alors qu'ils ne correspondent pas aux mêmes

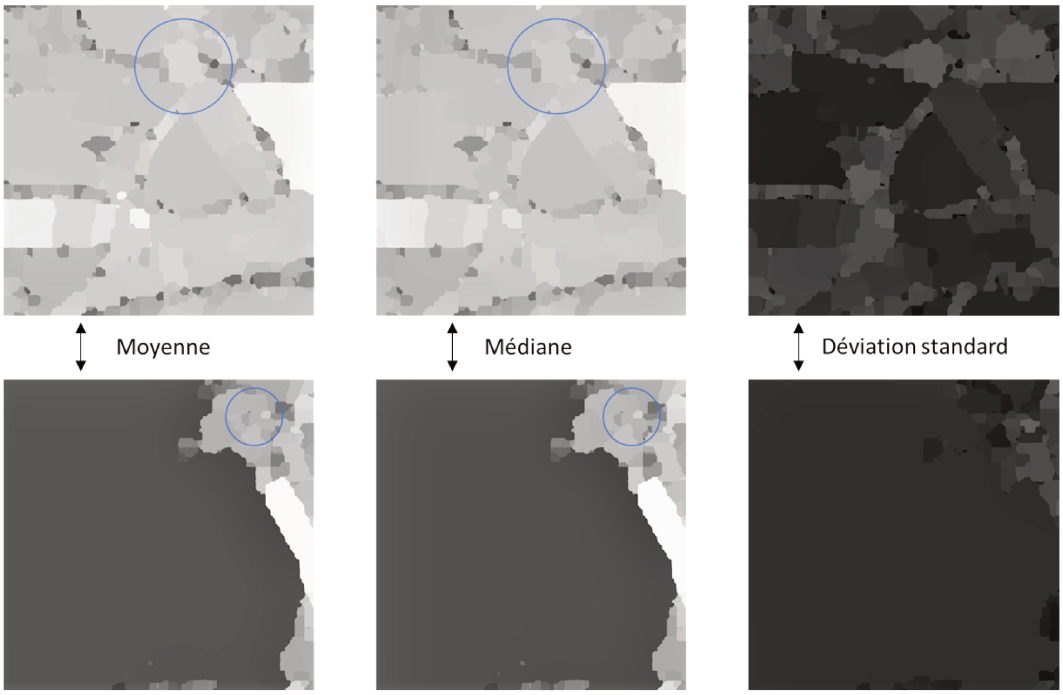


FIGURE 5.12 – Comparaison visuelle entre DES-*mean*, DES-*median* et DES-*std*.

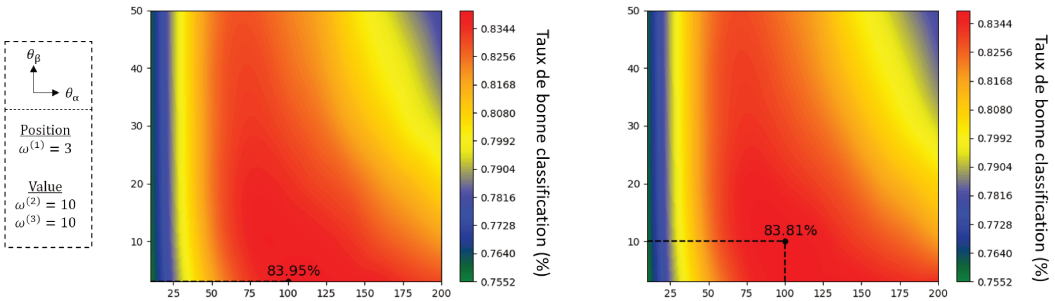


FIGURE 5.13 – Comparaison entre l'utilisation de DES-*mean* (à gauche, 83.95%) et de DES-*median* (à droite, 83.81%) pour le post-traitement.

types d'occupation du sol (image du bas).

Les résultats obtenus en utilisant ces deux représentations pour le post-traitement sont présentés sur la figure 5.13. On y constate que, sur nos données, la représentation DES-*mean* permet d'obtenir des taux de bonne classification légèrement plus élevés que DES-*median*. L'utilisation de la valeur moyenne de chaque superpixel ne semble donc pas contre-indiquée ici.

5.4.3 Apport de la colorisation

Face aux résultats encourageants que nous avons obtenus sur les images en niveaux de gris, nous avons souhaité étudier l'apport de la colorisation sur le post-traitement (voir figure 5.14). Il s'agit ici d'étudier l'intérêt de coloriser l'image source avant d'en extraire des DES-*mean* utilisables au sein d'un CRF dense. Intuitivement, nous nous disions que l'information portée par la couleur devrait permettre d'avoir des potentiels par paires plus discriminants. Nous souhaitions également comparer l'utilisation d'images colorisées par rapport à l'utilisation d'images en couleurs réelles.

Mise en place

Pour cela, nous avons travaillé sur un jeu de données constitué de 18 images aériennes en couleurs acquises entre 1991 et 2003. Comme précédemment, ces images ont été annotées manuellement par un géomaticien du Centre Léon Bérard. La moitié des images a été réservée pour entraîner le détecteur de bords, et l'autre moitié a été utilisée pour évaluer les pipelines de post-traitements. Les images aériennes ont ensuite été découpées en imagettes de 1024 × 1024 pixels, pour un total de 1389 imagettes avec 75% de recouvrement pour entraîner le détecteur de bords, et de 100 imagettes sans recouvrement pour l'évaluation.

Chaque imagette a ensuite été convertie en niveaux de gris avant d'être colorisée à l'aide de Col-Cycle dans sa version entraînée après 60 *epochs* (voir chapitre 4). Le choix d'utiliser ce réseau là pour la colorisation a été fait afin de tenir compte de l'étude par note moyenne d'opinions réalisée dans le chapitre précédent. Empiriquement, nous pensions que si les images colorisées sont réalistes pour un être humain, elles devraient également contenir des informations couleur spatialement pertinentes comparées à des images réellement en couleurs. De plus, les taux de bonne classification obtenus sur les images colorisées à l'aide de Col-Cycle et de SpyncoGan se sont avérés relativement proches lorsque combinés à la texture (voir chapitre 4), ne mettant pas en avant une colorisation particulièrement plus efficace qu'une autre pour discriminer les classes d'occupation du sol. Il est néanmoins à noter que les couleurs générées par Col-Cycle semblaient être plus discriminantes que celles de SpyncoGan (tous deux comparés après 120 epochs d'entraînement) lorsqu'utilisées seules pour la classification.

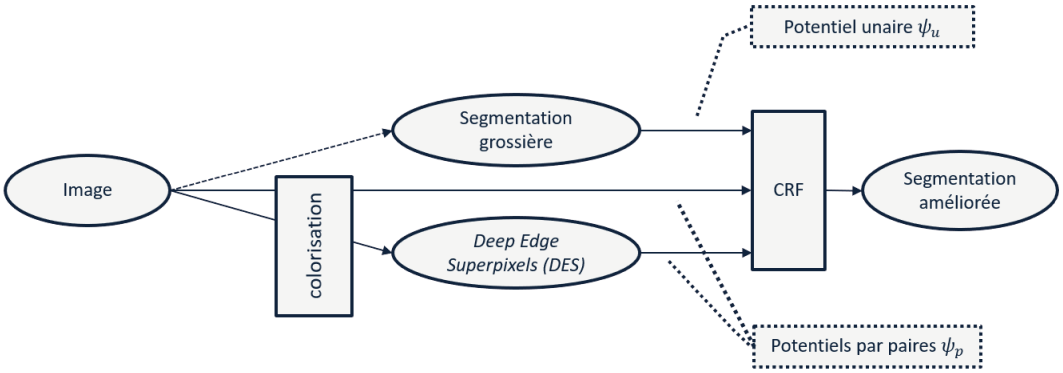


FIGURE 5.14 – Schéma générique de l'approche proposée, avec colorisation.

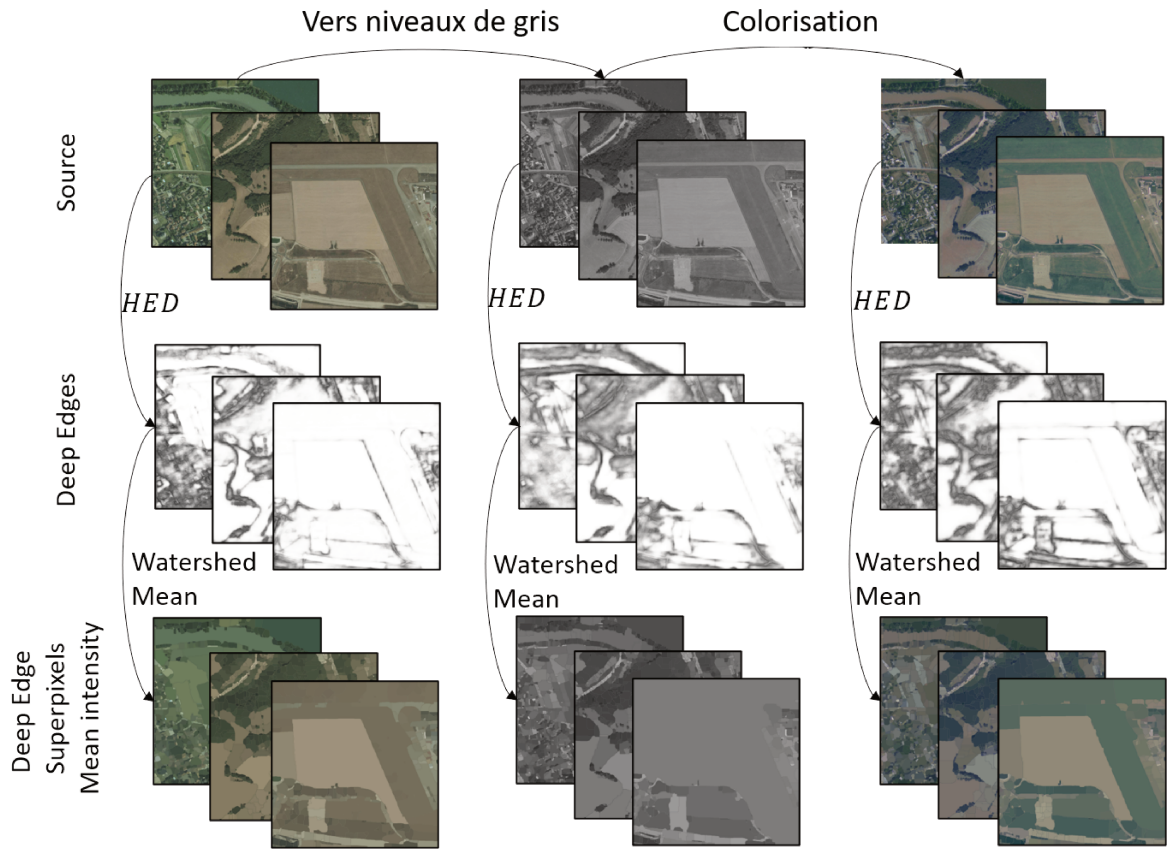


FIGURE 5.15 – Schéma illustrant la génération de représentations lissées (DES-*mean*) à l'aide de superpixels extrait de bords profonds à partir d'images en couleurs, niveaux de gris et colorisées.

Une fois la colorisation réalisée, on se retrouve alors avec les mêmes instances d'images, représentées dans trois domaines couleur différents : niveaux de gris, couleurs réelles, et fausses couleurs.

Pour chacun de ces domaines, nous raffinons l'entraînement du détecteur de bords HED utilisé précédemment, et ce pour 4000 itérations. Le but est ici de profiter d'une initialisation proche de l'optimum souhaité pour chacun des domaines couleur, tout en ayant la même procédure d'entraînement pour chacun d'entre eux. Il est néanmoins possible qu'un biais existe pour les images en niveaux de gris, domaine couleur sur lequel le réseau a initialement été entraîné (risque de sur-apprentissage, ou de sous-apprentissage pour les autres domaines). Les détecteurs de bords raffinés sont ensuite appliqués sur les imagettes de leurs domaines couleur respectifs. Cela nous permet de générer des représentations de type DES-*mean* pour chacun des domaines couleur. On se retrouve alors avec des images, des DES et des DES-*mean* différents pour les trois domaines couleur considérés. Ce processus est illustré sur la figure 5.15. On y constate la diversité des représentations obtenues.

Intérêt de la colorisation sans DES-mean

Nous avons dans un premier temps évalué l'intérêt de la colorisation pour le post-traitement à l'aide d'un CRF dense sans intégrer l'information portée par les DES-*mean* (*i.e.*, $w^{(1)} = 3$, $w^{(2)} = 10$, $w^{(3)} = 0$). Les résultats obtenus en faisant varier θ_α et θ_β sont présentés sur la figure 5.16. On y observe que le post-traitement à l'aide de potentiels par paires calculés à partir d'images en couleurs permet d'obtenir les résultats les plus élevés (82.77% de taux de bonne classification). Les potentiels colorisés permettent quant à eux d'obtenir des résultats proches des potentiels couleur

(82.65%), et supérieurs à ceux obtenus à l'aide des potentiels en niveaux de gris (82.04%). Ces résultats nous indiquent que coloriser les images en niveaux de gris peut permettre d'améliorer l'efficacité d'un post-traitement à l'aide d'un CRF dense.

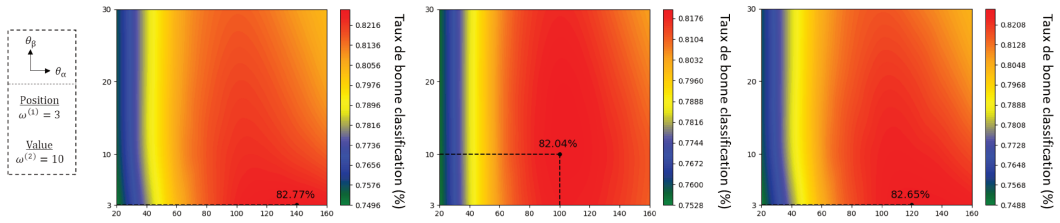


FIGURE 5.16 – Résultats obtenus en post-traitement avec un CRF dense sans DES-*mean*. De gauche à droite : image source en couleurs (82.77%), niveaux de gris (82.04%), et colorisée (82.65%).

Intégration des DES-mean et variation des poids relatifs

Nous avons évalué l'intérêt d'intégrer les DES-*mean* de chaque domaine couleur pour le post-traitement à l'aide d'un CRF dense. Les potentiels par paires ont été définis à l'aide d'un filtre bilatéral, en accord avec les observations réalisées dans les sous-sections précédentes. Les poids $w^{(2)}$ et $w^{(3)}$ ont d'abord été fixés pour donner la même importance à chaque potentiel par paires, tels que $w^{(2)} = w^{(3)} = 10$. Nous avons ensuite fait varier ces poids de façon relative afin de déterminer s'il est préférable de donner plus d'importance à l'image ou à sa représentation lissée. Les résultats obtenus sont présentés sur le tableau 5.2. On y remarque que les potentiels en couleurs et colorisés permettent d'obtenir des résultats plus élevés que les potentiels en niveaux de gris sur nos données. On remarque également que la configuration permettant d'obtenir les taux de classification les plus élevés correspond à un équilibre relatif entre les poids attribués à chaque potentiel par paires (poids identiques). Par ailleurs, donner plus d'importance à l'image source par rapport à la DES-*mean* ne diminue que légèrement les résultats obtenus. En revanche, donner plus d'importance à la DES-*mean* par rapport à l'image source semble diminuer les résultats de façon plus importante. Ces observations semblent indiquer que l'information lissée est moins discriminante que l'information brute (image source). L'utilisation des DES-*mean* en complément de l'image source semble être à préconiser.

TABLEAU 5.2 – Taux de bonne classification (%) obtenus en intégrant les DES-*mean* au sein d'un CRF dense et en faisant varier $w^{(2)}$ et $w^{(3)}$. Les valeurs reportées correspondent aux meilleurs résultats obtenus en faisant varier θ_α et θ_β.

	Poids ($w^{(2)}$, $w^{(3)}$)		
Domaine couleur	(10,10)	(30,10)	(10,30)
Niveaux de gris	**82.68**	82.16	82.12
Couleurs réelles	**83.34**	83.28	82.36
Fausses couleurs	**83.26**	83.11	82.42

Superpixels d'un autre domaine couleur

Nous avons observé des gains pour les taux de bonne classification liés à la colorisation. Cependant, nous avons jusqu'à présent généré les superpixels à partir de bords profonds détectés sur des images de domaines différents. Or, les superpixels utilisés sont *a priori* différents d'un domaine à l'autre. Afin d'évaluer si les résultats observés sont le fruit de la colorisation ou de la forme des superpixels, nous avons générés des DES-*mean* pour chaque domaine couleur à l'aide

des superpixels des autres domaines (*i.e.*, DES-*mean* de l'image colorisée en utilisant la forme des superpixels extraits de l'image en niveaux de gris). Les résultats que nous avons obtenus sont présentés sur le tableau 5.3. On observe que les meilleurs taux de bonne classification sont obtenus à l'aide des DES calculés à partir des images en niveaux gris et en couleurs. Ces résultats semblent cohérents avec notre intuition. Les images colorisées ont en effet été hallucinées par un réseau de neurones à convolutions à partir des images en niveaux de gris, ce qui implique que les structures spatiales ont pu être légèrement modifiées (*e.g.*, aberrations chromatiques locales). Néanmoins, pour chaque type de superpixel, on observe des taux de bonne classification plus élevés en utilisant les représentations moyennes en vraies et fausses couleurs. Ce dernier point tend à montrer que les gains observés sont effectivement dus à la colorisation, et non à la forme des superpixels.

TABLEAU 5.3 – Taux de bonne classification (%) en utilisant les DES-*mean* obtenus à l'aide de DES de différents domaines couleur avec $w^{(1)} = 3$ and $w^{(2)} = w^{(3)} = 10$. Les valeurs reportées correspondent aux meilleurs résultats obtenus en faisant varier θ_α et θ_β.

	Superpixels		
DES-*mean*	*Niveaux de gris*	*Couleurs réelles*	*Fausses couleurs*
Niveaux de gris	**82.68**	82.63	82.64
Couleurs réelles	83.32	**83.34**	83.16
Fausses couleurs	**83.52**	83.34	83.26

5.5 Conclusion

Résumé des travaux réalisés. Nous nous sommes intéressés à l'utilisation d'algorithmes de post-traitements pour améliorer les résultats de segmentation sémantique d'images aériennes. Nous avons montré l'intérêt d'extraire des bords profonds afin de générer des superpixels et intégrer l'information qu'ils portent au sein d'un CRF dense. Nous avons également mis en avant l'intérêt de la colorisation pour le post-traitement.

Vision critique sur les travaux réalisés. Nous avons travaillé dans un cadre exploratoire afin de déterminer l'intérêt de certains algorithmes de post-traitement pour améliorer les cartes d'occupation du sol. Nous nous sommes limités au cas de l'inférence à l'aide de champs aléatoires conditionnels denses. Nous n'avons pas exploré l'utilisation d'autres algorithmes de post-traitement. Nous aurions pu, par exemple, remplacer le CRF dense par un réseau de neurones à convolutions. De la même manière, nous n'avons pas cherché à optimiser les poids associés à chaque potentiel par paires, nous contentant d'une évaluation de l'importance relative de chacun d'eux. Il aurait pu être intéressant d'optimiser ces poids de manière plus rigoureuse. Enfin, nous n'avons pas appliqué les algorithmes de post-traitement sur les images aériennes entières, mais seulement sur des imagettes de 1024×1024 pixels sans recouvrement. Les relations entre les pixels de deux imagettes connexes n'ont donc pas été prises en compte. La prise en compte de ces relations aurait pu avoir un impact sur les résultats obtenus.

Conclusion générale et perspectives

Dans ce manuscrit, nous avons abordé l'utilisation d'algorithmes de vision par ordinateur afin d'analyser automatiquement les images aériennes historiques dans le cadre d'une étude épidémiologique portant sur l'impact sur la santé de l'exposition aux pesticides associée aux cultures agricoles. Nous avons vu comment classifier, coloriser, et segmenter ces données en plusieurs classes d'occupation du sol. Nous avons notamment pu mettre en avant la faisabilité de ce type d'approches pour faciliter le travail de photo-interprétation des géomaticiens. Nos travaux ne sont cependant encore qu'une esquisse du champ des possibles, que ce soit à cause de la nature complexe des images étudiées (peu de modalités, différences de résolutions, différentes dates, *etc.*), ou de l'état actuel des ressources disponibles (peu de données annotées, mise à disposition récente des images).

Classification de textures

Nous avons étudié l'utilisation de descripteurs de textures et de réseaux de neurones profonds à convolutions afin de classifier automatiquement les images aériennes historiques en plusieurs classes d'occupation du sol. Nous avons mis en avant l'efficacité des deux types d'approches pour cette tâche, les descripteurs de textures ayant des temps de traitements plus rapides que les DCNN, pour des taux de bonne classification légèrement inférieurs sur le jeu de données HistAerial. Face à des résultats encourageants, nous avons étendu nos travaux à des images d'écorces d'arbres dans le cadre d'une collaboration avec une autre doctorante, mettant en avant l'intérêt de combiner texture et couleur. La rapidité de ces algorithmes basés sur la texture nous a poussé à les utiliser dans le cadre du projet TESTIS afin qu'ils puissent bénéficier aux géomaticiens du Centre Léon Bérard, ne disposant pas de cartes graphiques aptes à accélérer l'exécution des DCNN.

Les méthodes développées dans le cadre de cette thèse, ainsi que le jeu de données HistAerial, peuvent avoir un intérêt pour la classification de textures en général. L'utilisation d'autres combinaisons de filtres et de caractéristiques pourrait également avoir un intérêt afin d'améliorer les résultats obtenus, non seulement sur HistAerial, mais aussi sur d'autres jeux de données. Parmi les pistes possibles, la combinaison de caractéristiques extraites par des descripteurs de textures et des réseaux de neurones profonds à convolutions nous parait intéressante. En particulier, il pourrait être pertinent de contraindre l'entraînement d'un DCNN pour la génération de caractéristiques complémentaires à celles extraites par les filtres classiques. Pour cela, la concaténation des caractéristiques de textures et des caractéristiques profondes durant l'entraînement est une piste envisagée. Il s'agirait alors d'avoir un ensemble de caractéristiques pré-définies, et un semble de caractéristiques qui seraient apprises en complément. Nous avons par ailleurs vu que l'utilisation de modalités générées par un réseau de neurones à convolutions (*i.e.*, la couleur dans notre cas) permettait d'améliorer légèrement les taux de bonne classification sur HistAerial. Ces résultats nous poussent à croire que la génération d'autres modalités pourrait avoir un intérêt pour l'analyse des images aériennes historiques. En particulier, les informations de profondeur, qui peuvent être générée par stéréoscopie, devraient nous donner des indices visuels complémentaires pour analyser ces images (*e.g.*, distinction vignes / vergers en fonction des pentes ou de la hauteur des cultures). La fusion d'informations non issues d'images (*e.g.*, saison de l'acquisition,

coordonnées géographiques, registres cadastraux, statistiques de recensements) est une piste de recherche qui nous semble également pertinente pour inclure des *a priori* supplémentaires quant aux données observées (*e.g.*, certaines régions ne possèdent pas de vignes). En particulier, le fait de tenir compte des coordonnées géographiques pourrait nous permettre de guider la génération automatique de cartes d'occupation du sol en adaptant les méthodes en fonction des zones observées. Enfin, l'utilisation de séries temporelles est une piste que nous avions temporairement exclue dû à la faible période temporelle entre deux acquisitions, et à la volonté de générer des résultats à un instant donné (étude TESTIS). Il pourrait néanmoins être intéressant de combiner les informations d'images multi-temporelles, et éventuellement multi-spectrales, afin d'améliorer les taux de bonne classification.

Colorisation automatique

Afin d'annoter les images aériennes historiques, que ce soit manuellement ou à l'aide du logiciel Gouramic, les géomaticiens ont besoin de déterminer quel est le contenu des images. Ce contenu est particulièrement difficile à analyser lorsqu'il n'est disponible qu'en niveaux de gris. Afin de les aider dans cette tâche, nous avons étudié l'utilisation de réseaux de neurones générateurs adversaires cycliques afin de coloriser automatiquement les images anciennes. Nous avons montré que les colorisations générées étaient réalistes pour les êtres humains, mais permettaient aussi d'améliorer légèrement les taux de bonne classification par rapport à la texture seule (*i.e.*, les couleurs générées semblent positivement corrélées aux classes d'occupation du sol). Afin de tenter d'améliorer les colorisations générées, nous avons proposé une méthode dite pseudo-cyclique, qui consiste à remplacer l'un des deux GAN par une fonction définie empiriquement. Nous avons montré qu'une telle approche, pseudo-cyclique, permettait d'obtenir des résultats au moins au niveau des autres méthodes comparées, sans pour autant que les couleurs générées permettent d'obtenir un gain supplémentaire en classification.

Les travaux que nous avons menés sur la colorisation automatique se cantonnent à l'utilisation d'approches entièrement automatiques, sans tenir compte de la géolocalisation des images ni de la représentation actuelle des sols des lieux observés. Il serait intéressant d'étudier, d'une part, l'intérêt d'algorithmes supervisés pour coloriser les images aériennes historiques, et, d'autre part, d'évaluer l'intérêt des méthodes utilisées sur des jeux de données plus disparates que ceux avec lesquels nous avons travaillés. Une autre perspective intéressante, selon nous, serait de guider le processus de colorisation en appariant les images historiques aux images récentes dans le cadre d'une approche hybride, à mi-chemin entre le transfert de couleur et la colorisation. De même, guider le processus de colorisation en intégrant des contraintes liées à la classification est une piste relativement populaire dans la littérature, qu'il pourrait être intéressant de suivre. A l'inverse, il pourrait être pertinent de s'inspirer des Auto-Encodeurs pour étudier l'efficacité des caractéristiques profondes générées pour la colorisation afin de réaliser d'autres tâches (*e.g.*, segmentation). A ce propos, nous pourrions aussi nous demander si ces caractéristiques sont très différentes de celles obtenues avec un auto-encodeur, et pourquoi ? Par ailleurs, l'utilisation de méthodes plus avancées pour gérer l'effet mosaïque est également une piste envisagée. On pourrait, par exemple, tenir compte de la cohérence spatiale des colorisations lors de l'entraînement en utilisant des imagettes avec recouvrement, et en imposant des contraintes pour que les pixels de deux imagettes qui se recouvrent aient la même couleur. De plus, l'utilisation d'approches cycliques pour générer des représentations spectralement plus complètes (*e.g.*, infrarouge, profondeur) est une piste qui pourrait avoir un intérêt pour la visualisation et la classification des images aériennes historiques (*i.e.*, apprentissage d'une relation entre texture et infrarouge). A noter que l'utilisation de la colorisation et de la profondeur (élévation du terrain) pour la visualisation est une idée qui a été mis en place dans le cadre d'une collaboration avec le laboratoire Environnement Ville Société de Saint-Etienne, France. Enfin, l'utilisation de la colorisation pour l'adaptation de domaines est une piste qui nous semble prometteuse. Il s'agirait ici de convertir des images acquises avec des

capteurs différents vers une représentation en niveaux de gris, et de toutes les coloriser à l'aide du même algorithme. Les domaines couleur rattachés aux niveaux de gris et aux images colorisées serviraient alors d'intermédiaires entre les différents capteurs, sans avoir nécessairement besoin d'entraîner un algorithme pour chaque type d'images.

Post-traitement

Dans le but d'améliorer les résultats obtenus par segmentation sémantique au pixel ou à l'imagette près, nous avons étudié l'utilisation de méthodes de post-traitement. Nous nous sommes basés sur des algorithmes de sur-segmentation et des champs aléatoires conditionnels. En particulier, nous avons proposé d'extraire des bords profonds afin de générer des superpixels basés sur des informations supposées sémantiquement intéressantes. Nous avons montré l'intérêt de ces approches pour améliorer les segmentations obtenues sur des images aériennes historiques et récentes. Nous avons également évalué l'intérêt de la colorisation pour le post-traitement, mettant en avant l'intérêt potentiel de générer des représentations colorisées pour cette tâche.

Les perspectives liées au post-traitement portent, d'une part, sur l'intégration des algorithmes de post-traitement au sein des chaînes de traitements utilisées par le Centre Léon Bérard via le logiciel Gouramic. D'autre part, nous nous sommes ici intéressés au post-traitement de segmentations grossières obtenues par une classification par imagette. Nous n'avons pas étudié l'intérêt de segmenter les images aériennes historiques à l'aide de DCNN. Il pourrait être intéressant d'étudier l'intérêt de ces approches pour la segmentation automatique des images aériennes historiques. De même, nous n'avons pas appliqués nos algorithmes sur d'autres types d'images que des images aériennes. Une perspective pourrait être d'étudier l'intérêt des superpixels issus de bords profonds pour le post-traitement d'images d'autres catégories (*e.g.*, images de vie courante, image médicales). En particulier, il serait possible d'étendre les approches développées au cas 3D afin d'analyser des volumes complexes. Il serait alors intéressant d'observer dans quelle mesure les bords profonds peuvent être exploités afin de générer des groupes de voxels (pixels volumiques). Par ailleurs, nous avons ici travaillé uniquement à l'amélioration de segmentations existantes. Nous n'avons pas cherché à développer d'approches bout en bout, incluant à la fois l'algorithme de segmentation et celui de post-traitement. L'étude de ce type d'approches, que ce soit à l'aide d'algorithmes classiques ou de réseaux de neurones profonds à convolutions, pourrait avoir un intérêt pour l'analyse automatique des images aériennes historiques. De la même manière, il pourrait être intéressant d'exploiter des réseaux de neurones profonds à convolutions pour post-traiter les segmentations grossières. Une piste qui nous semble intéressante consisterait à entraîner un réseau de neurones pour à la fois raffiner des segmentations et segmenter des images. Pour cela, il serait possible de donner la segmentation grossière et l'image source en entrée du réseau, et de remplacer aléatoirement la segmentation grossière par du bruit blanc lors de l'entraînement. Ainsi, lorsque la segmentation grossière sera présente, le réseau pourra s'en servir pour extraire des informations sémantiques qui vont lui permettre d'améliorer l'existant. Lorsque du bruit blanc sera présent, il aura alors pour tâche de segmenter l'image source.

Apport pour le projet TESTIS

Nous avons contribué au projet TESTIS en développant Gouramic, un logiciel d'aide à l'annotation des images aériennes historiques. Notre logiciel a permis de réduire le temps consacré par un géomaticien sur chaque image à environ, 20 minutes, contre 6 à 10 heures auparavant. Ce temps inclut l'ensemble de la chaîne de traitements, de l'ouverture du fichier à la sauvegarde du résultat sur le disque, en passant par l'annotation partielle des images. Afin de fonctionner, Gouramic nécessite des annotations partielles (des traces) fournies par l'utilisateur. Ce choix a été fait afin de permettre une vérification et une amélioration des résultats par l'utilisateur : il lui suffit de fournir un plus grand nombre d'annotations en cas de résultat insatisfaisant. La récolte de

l'ensemble des annotations partielles réalisées sur les images de l'année de naissance des sujets de l'étude TESTIS a permis le développement d'une approche automatique qui est actuellement en cours d'évaluation. Les résultats générés par l'étude TESTIS à l'aide de Gouramic devraient permettre de mettre en place des pistes de réflexion sur l'impact sur la santé de l'exposition aux pesticides liée à la proximité de résidences aux cultures agricoles, volet intéressant particulièrement l'Agence De l'Environnement et de la Maîtrise de l'Energie et le Centre Léon Bérard.

D'un point de vue utilisation, le logiciel Gouramic a été éprouvé par les géomaticiens du Centre Léon Bérard, ainsi que par un groupe d'étudiants en géomatique de l'Université Jean Monnet de Saint-Étienne. Ces utilisateurs ont trouvé que le logiciel était ergonomique et facile d'utilisation. D'un point de vue améliorations, l'intégration d'outils de visualisation supplémentaires et d'un outil plus performant pour gommer les traces réalisées sont envisagés. D'un point de vue algorithmes, la mise en place d'approches basées sur des superpixels est une piste qui nous semble prometteuse. Les méthodes de colorisation nous semblent quant à elle pré-destinées à être utilisée en amont de Gouramic. Par ailleurs, l'utilisation préalable d'un algorithme automatique avant que l'utilisateur ait le besoin de réaliser des traces devrait accélérer les traitements réalisés. L'évaluation de l'impact des traces utilisateurs sur les résultats, ainsi que la dépendance des résultats en fonction de l'utilisateur sont des questions qui sont actuellement à l'étude. De même, la comparaison des résultats générés à l'aide de Gouramic avec d'autres bases d'occupation du sol est en cours (Corine Land Cover, Hilda, *etc.*).

Références

[AFA+16] M. A. Aguilar, A. Fernandez, F. Aguilar, F. Bianconi, and A. Lorca. Classification of urban areas from geoeye-1 imagery through texture features based on histograms of equivalent patterns. *European Journal of Remote Sensing*, 49 :93–120, 03 2016.

[AKG17] A. Albert, J. Kaur, and M.C. Gonzalez. Using convolutional networks and satellite imagery to identify patterns in urban environments at a large scale. In *International Conference on Knowledge Discovery and Data Mining*, page 1357–1366. ACM, 2017.

[AKvdW+18] R.M. Anwer, F.S. Khan, J. van de Weijer, M. Molinier, and J. Laaksonen. Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 138 :74–85, 2018.

[ALSL16] N. Audebert, B. Le Saux, and S. Lefèvre. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In *Asian Conference on Computer Vision (ACCV)*, pages 180–196. Springer, 2016.

[AMFM11] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(5) :898–916, Mai 2011.

[AS93] M.F. Augusteijn and T.L. Skufca. Identification of human faces through texture-based feature recognition and neural network technology. In *IEEE International Conference on Neural Networks*, pages 392–398. IEEE, 1993.

[ASS+12] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(11) :2274–2282, 2012.

[ATY+19] Md. Z. Alom, T. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. Nasrin, M. Hasan, B. Essen, A. Awwal, and V. Asari. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8 :292, 03 2019.

[BAC+18] S. Bertrand, R.B. Ameur, G. Cerutti, D. Coquin, L. Valet, and L. Tougne. Bark and leaf fusion systems to improve automatic tree species recognition. *Ecological Informatics*, 42 :1574–9541, 2018.

[BBB+13] R. Béranger, J. Blain, E. Billoir, M.-L. Bayle, J. Nuckols, J. Schüz, B. Combourieu, and B. Fervers. Sigexpo project : pesticides exposure level in the french context : burden of environmental exposures and domestic habits. *ISEE Conference Abstracts*, 2013 :5043, 09 2013.

[BBB+14] R. Béranger, J. Blain, C. Baudinet, E. Faure, A. Fléchon, H. Boyle, V. Chasles, B. Charbotel, J. Schüz, and B. Fervers. Testicular germ cell tumours and early exposures to pesticides : The TESTEPERA pilot study. *Bulletin du Cancer*, 101(3) :225–235, Mars 2014.

[BCT17] S. Bertrand, G. Cerutti, and L. Tougne. Bark recognition to improve leaf-based classification in didactic tree species identification. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2017.

[BFOS84] L. Breiman, J. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth and Brooks/Cole Advanced Books, 1984.

[BHvdM+17] M. Brouwer, A. Huss, M. van der Mark, P. Nijssen, W. Mulleners, A. Sas, T. van Laar, G.R. Snoo, H Kromhout, and R Vermeulen. Environmental exposure to pesticides and the risk of parkinson's disease in the netherlands. *Environment International*, 107 :100–110, 10 2017.

[BKC17] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet : A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(12) :2481–2495, 2017.

[BKD+16] S. Basu, M. Karki, R. DiBiano, S. Mukhopadhyay, S. Ganguly, R. Nemani, and S. Gayaka. A theoretical analysis of deep neural networks for texture classification. In *2016 International Joint Conference on Neural Networks (IJCNN)*, volume 97, pages 992–999, 05 2016.

[BKH16] J.L. Ba, J.R. Kiros, and G.E. Hinton. Layer normalization. *arXiv preprint arXiv :1607.06450*, 2016.

[Bla10] T. Blaschke. Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(1) :2–16, 2010.

[BM93] S. Beucher and F. Meyer. The morphological approach to segmentation : the watershed transformation. *Mathematical morphology in image processing*, 34 :433–481, 1993.

[BMOL+13] V. Bakić, S. Mouine, S. Ouertani-Litayem, A. Verroust-Blondet, I. Yahiaoui, H. Goëau, and A. Joly. Inria's participation at imageclef 2013 plant identification task. In *CLEF Working Notes*, 2013.

[BPB+14] R. Béranger, O. Pérol, L. Bujan, E. Faure, J. Blain, C. Le Cornet, A. Flechon, B. Charbotel, T. Philip, J. Schüz, and B. Fervers. Studying the impact of early life exposures to pesticides on the risk of testicular germ cell tumors during adulthood (TESTIS project) : study protocol. *BMC Cancer*, 14(1), Août 2014.

[Bra00] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[Bre01] L. Breiman. Random forests. *Machine learning*, 45(1) :5–32, 2001.

[BST15] G. Bertasius, J. Shi, and L. Torresani. Deepedge : A multi-scale bifurcated deep network for top-down contour detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4380–4389, 2015.

[BYB18] S. Boudra, I. Yahiaoui, and A. Behloul. Plant identification from bark : A texture description based on statistical macro binary pattern. In *International Conference on Pattern Recognition (ICPR)*, 2018.

[Bé14] R. Béranger. *Tumeurs germinales du testicule : étudier l'impact des expositions professionnelles et environnementales aux pesticides*. PhD thesis, 2014. Thèse de doctorat dirigée par Fervers, Béatrice et Schüz, Joachim Santé publique. Épidemiologie Lyon 1 2014.

[Can86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, (6) :679–698, 1986.

[CBP+16] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2016.

[CCK+18] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan : Unified generative adversarial networks for multi-domain image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8789–8797, 2018.

[CGDC+14] I. Champion, C. Germain, J.P. Da Costa, A. Alborini, and P. Dubois-Fernandez. Retrieval of forest stand age from sar image texture for varying distance and orientation values of the gray level co-occurrence matrix. *IEEE Geoscience and Remote Sensing Letters*, 11(1) :5–9, 2014.

[CJL+16] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10) :6232–6251, 2016.

[CKYV19] S. Crommelinck, M. Koeva, M.Y. Yang, and G. Vosselman. Application of deep learning for delineation of visible cadastral boundaries from remote sensing imagery. *Remote sensing*, 11(21) :2505, 2019.

[CM02] D. Comaniciu and P. Meer. Mean shift : A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(5) :603–619, 2002.

[CM19] Y. Chen and D. Ming. Superpixel Classification of High Spatial Resolution Remote Sensing Image Based on Multi-scale CNN and Scale Parameter Estimation. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W13 :681–685, Juin 2019.

[CMM08] C.M.R. Caridade, A.R.S. Marçal, and T. Mendonça. The use of texture for image classification of black & white air photographs. *International Journal of Remote Sensing*, 29(2) :593–607, Janvier 2008.

[CMV15] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[CMZ+11] M. Cockburn, P. Mills, X. Zhang, J. Zadnick, D. Goldberg, and B. Ritz. Prostate cancer and ambient pesticide exposure in agriculturally intensive areas in california. *American Journal of Epidemiology*, 173(11) :1280–1288, Mars 2011.

[CPSA17] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv :1706.05587*, 2017.

[CRN+19] B. Canovas, M. Rombaut, A. Nègre, S. Olympieff, and D. Pellerin. A coarse and relevant 3d representation for fast and lightweight rgb-d mapping. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2019.

[CTK15] Z. Camlica, H.R. Tizhoosh, and F. Khalvati. Medical image classification via svm using lbp features from saliency-based folded data. In *International Conference on Machine Learning and Applications*, pages 128–132, 2015.

[CV95] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3) :273–297, 1995.

[CZZY17] Y. Cao, Z. Zhou, W. Zhang, and Y. Yu. Unsupervised diverse colorization via generative adversarial networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 151–166. Springer, 2017.

[DK05] K.-B. Duan and S.S. Keerthi. Which is the best multiclass svm method ? an empirical study. In N C Oza, R Polikar, J Kittler, and F Roli, editors, *Multiple Classifier Systems*, pages 278–285. Springer Berlin Heidelberg, 2005.

[DP13] H. Dubey and V. Pudi. Class based weighted k-nearest neighbor over imbalance dataset. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 305–316. Springer, 2013.

[DP19] P. Dusanek and J. Potuckova, M. adn Hodac. Historical aerial images of czechia-archiving and applications in landscape studies. *EuroSDR Workshop, Geoprocessing and Archiving of Historical Aerial Images*, June 2019.

[dRslCC20] Centre International de Recherche sur le Cancer (CIRC). Outils de visualisation en ligne des statistiques liées aux cancers, 2020. `http://gco.iarc.fr/today/online-analysis-map` (accès : 2020-04-03).

[eDDU20] Université Virtuelle Environnement et Développement Durable (UVED). Suivi de l'environnement par télédétection, 2020. `https://e-cours.univ-paris1.fr/modules/uved/envcal/html/msg/index.html` (accès : 2020-04-06).

[FÁB13] A. Fernández, M.X. Álvarez, and F. Bianconi. Texture description through histograms of equivalent patterns. *Journal of Mathematical Imaging and Vision*, 45(1) :76–102, 2013.

[FBB+13] D. Forman, F. Bray, D.H. Brewster, C. Gombe Mbalawa, B. Kohler, M. Pineros, and et al. *Cancer Incidence in Five Continents*. Vol. X. IARC Scientific Publication ed. Lyon : IARC., 2013.

[FBF+18] E. Faure, R. Béranger, B. Fervers, J. Schüz, and J. Blain. A gis-based method to define geographical determinants of environmental exposure to agricultural pesticides in france. *Revue d'Épidémiologie et de Santé Publique*, 66 :S333, July 2018.

[FDCC+17] E. Faure, A.M.N. Danjou, F. Clavel-Chapelon, M.-C. Boutron-Ruault, L. Dossus, and B. Fervers. Accuracy of two geocoding methods for geographic information system-based exposure assessment in epidemiological studies. *Environmental Health*, 16(1), Février 2017.

[Fel08] D.R. Feldman. Medical treatment of advanced testicular cancer. *Journal of the American Medical Association (JAMA)*, 299(6) :672, Février 2008.

[FH04] P. F. Felzenszwalb and D.P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision (IJCV)*, 59(2) :167–181, 2004.

[FHV+14] R. Fuchs, M. Herold, P.H. Verburg, J.G.P.W. Clevers, and J. Eberle. Gross changes in reconstructions of historic land cover/use for europe between 1900 and 2010. *Global Change Biology*, 21(1) :299–313, Septembre 2014.

[FLG15] Q. Feng, J. Liu, and J. Gong. Uav remote sensing for urban vegetation mapping using random forest and texture analysis. *Remote sensing*, 7(1) :1074–1094, 2015.

[FRCJ+18a] E. Faure, R. Ratajczak, C.F. Crispim-Junior, O. Pérol, L. Tougne, and B. Fervers. Development of a software based on automatic multi-temporal aerial images classification to assess retrospective environmental exposures to pesticides in epidemiological studies. *Epidemiology and Public Health / Revue d'Epidémiologie et de Santé Publique*, 66, Juilliet 2018.

[FRCJ+18b] E. Faure, R. Ratajczak, C.F. Crispim-Junior, O. Pérol, L. Tougne, and B. Fervers. Development of a software based on automatic multi-temporal aerial images classification to assess retrospective environmental exposures to pesticides in epidemiological studies. In *CLARA 2018 Cancer Research Forum*, Lyon, France, Avril 2018.

[FRCJ+19] E. Faure, R. Ratajczak, C.F. Crispim-Junior, A. Danjou, O. Perol, L. Tougne, and B. Fervers. GOURAMIC : A Software to Estimate Historical Land Use in Epidemiological Studies. *Environmental Epidemiology*, 3 :118, Octobre 2019.

[FVCH15] R. Fuchs, P.H. Verburg, J.G.P.W. Clevers, and M. Herold. The potential of old maps and encyclopaedias for reconstructing historic european land cover/use change. *Applied Geography*, 59 :43–55, Mai 2015.

[GEB16] L.A. Gatys, A.S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 2414–2423, 2016.

[GGLM16] C. Gonzalo-Martin, A. Garcia-Pedrero, M. Lillo-Saavedra, and E. Menasalvas. Deep learning for superpixel-based classification of remote sensing images. In *GEOBIA 2016 : Solutions and Synergies*. University of Twente Faculty of Geo-Information and Earth Observation (ITC), September 2016.

[GGvdWB18] A. Gonzalez-Garcia, J. van de Weijer, and Y. Bengio. Image-to-image translation for cross-domain disentanglement. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1287–1298, 2018.

[GGY+18] Y. Gan, J. Gong, M. Ye, Y. Qian, K. Liu, and S. Zhang. GANs with multiple constraints for image translation. *Complexity*, 2018 :1–12, dec 2018.

[GLBM18] S. Giordano, A. Le Bris, and C. Mallet. Toward Automatic Georeferencing of Archival Aerial Photogrammetric Surveys. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2 :105–112, Mai 2018.

[GM19] S. Giordano and C. Mallet. Archiving and geoprocessing of historical aerial images : current status in europe. *Official Publication N°70 of the European Spatial Data Research (EuroSDR)*, 02 2019.

[GPAM+14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680. Curran Associates, Inc., 2014.

[GPGBC19] D. Gominski, M. Poreba, V. Gouet-Brunet, and L. Chen. Challenging deep image descriptors for retrieval in heterogeneous iconographic collections. In *SUMAC Workshop @ ACM Multimedia 2019*. ACM, Octobre 2019.

[GPLSREGM19] A. García-Pedrero, M. Lillo-Saavedra, D. Rodríguez-Esparragón, and C. Gonzalo-Martín. Deep learning for automatic outlining agricultural parcels : Exploiting the land parcel identification system. *IEEE Access*, 7 :158223–158236, 2019.

[GTP17] R. Giraud, V.-T. Ta, and N. Papadakis. Superpixel-based color transfer. In *IEEE International Conference on Image Processing (ICIP)*, pages 700–704. IEEE, 2017.

[GTR18] Z. Gharibbafghi, J. Tian, and P. Reinartz. Modified superpixel segmentation for digital surface model refinement and building extraction from satellite stereo imagery. *Remote Sensing*, 10(11) :1824, Novembre 2018.

[GWA+11] R.B. Gunier, M.H. Ward, M. Airola, E.M. Bell, J. Colt, M. Nishioka, P.A. Buffler, P. Reynolds, R.P. Rull, A. Hertz, C. Metayer, and J.R. Nuckols. Determinants of agricultural pesticide concentrations in carpet dust. *Environmental Health Perspectives*, 119(7) :970–976, Juilliet 2011.

[GZZ10] Z. Guo, L. Zhang, and D. Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing (ICIP)*, 19(6) :1657–1663, 2010.

[HAGM15] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 447–456, 2015.

[Har79] R.M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5) :786–804, 1979.

[HCLD16] L. Huang, C. Chen, W. Li, and Q. Du. Remote sensing image scene classification using multi-scale completed local binary patterns and fisher vectors. *Remote Sensing*, 8(6) :483, 2016.

[HDL+18] M. He, D.Chen, J. Liao, P.V. Sander, and L. Yuan. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)*, 37(4) :1–16, jul 2018.

[HHD+06] Z.-K. Huang, D.-S. Huang, J.-X. Du, Z.-H. Quan, and S.-B. Guo. Bark classification based on gabor filter features using rbpnn neural network. In *International Conference on Neural Information Processing Systems (NIPS)*, pages 80–87. Springer, 2006.

[HLZ14] X. Huang, X. Liu, and L. Zhang. A multichannel gray level co-occurrence matrix for multi/hyperspectral image texture representation. *Remote Sensing*, 6(9) :8424–8445, 2014.

[HPS06] M. Heikkila, M. Pietikainen, and C. Schmid. Description of interest regions with center-symmetric local binary patterns. In *5th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'06)*, volume 4338, pages 58–69. Springer-Verlag, 2006.

[HSD73] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, Novembre 1973.

[HSS12] G. Hinton, N. Srivastava, and K. Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. 2012.

[HW90] D.-C. He and L. Wang. Texture unit, texture spectrum, and texture analysis. *IEEE transactions on Geoscience and Remote Sensing*, 28(4) :509–512, 1990.

[HZ18] J. Hodač and A. Zemánková. Historical Orthophotos Created on Base of Single Photos - Specifics of Processing. *Stavební obzor - Civil Engineering Journal*, 27(3) :425–438, 2018.

[HZC+17] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets : Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv :1704.04861*, 2017.

[HZRS16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[IGN20a] IGN. Géoportail, 2020. `https://www.geoportail.gouv.fr/` (accès : 2020-04-02).

[IGN20b] IGN. Remonterletemps, 2020. `https://remonterletemps.ign.fr/` (accès : 2020-04-02).

[IHM+16] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, and K. Keutzer. Squeezenet : Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. 2016. cite arxiv :1602.07360Comment : In ICLR Format.

[INC19] INCA. *Les cancers en France en 2018 - L'essentiel des faits et chiffre.* 2019.

[IP81] J.R. Irons and G.W. Petersen. Texture transforms of remote sensing data. *Remote Sensing of Environment*, 11 :359–370, 1981.

[IS15] S. Ioffe and C. Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv :1502.03167*, 2015.

[ISSI16] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be Color ! : Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Transactions on Graphics (TOG)*, 2016.

[IZZE17] P. Isola, J.-Y. Zhu, T. Zhou, and A.A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, July 2017.

[JSD+14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe : Convolutional architecture for fast feature embedding. *arXiv preprint arXiv :1408.5093*, 2014.

[JSL+18] V. Jampani, D. Sun, M.-Y. Liu, M.-H. Yang, and J. Kautz. Superpixel sampling networks. In *European Conference on Computer Vision (ECCV)*, pages 352–368, 2018.

[Jul62] B. Julesz. Visual pattern discrimination. *IRE Transactions on Information Theory*, 8(2) :84–92, 1962.

[JW17] N. Li J. Wang, Y. Fan. Combining fine texture and coarse color features for color texture classification. *Journal of Electronic Imaging*, 2017.

[KB14] D.P. Kingma and J. Ba. Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*, 2014.

[KHH17] S. Kwak, S. Hong, and B. Han. Weakly supervised semantic segmentation using superpixel pooling network. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[KK11] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems (NIPS)*, pages 109–117, 2011.

[KLSS17] N Kussul, N Lavreniuk, S Skakun, and A Shelestov. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, PP :1–5, 03 2017.

[KLSY16] N. Kussul, M. Lavreniuk, A. Shelestov, and B. Yailymov. Along the season crop classification in ukraine based on time series of optical and sar images using ensemble of neural network classifiers. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 7145–7148, July 2016.

[KMT18] B. Kellenberger, D. Marcos, and D. Tuia. Detecting mammals in uav images : Best practices to address a substantially imbalanced dataset with deep learning. *Remote sensing of environment*, 216 :139–153, 2018.

[KP19] G.-H. Kwak and N.-W. Park. Impact of texture information on crop classification with machine learning and uav images. *Applied Sciences*, 9(4) :643, 2019.

[KRH19] C. Kruse, F. Rottensteiner, and C. Heipke. Marked Point Processes for the Automatic Detection of Bomb Craters in Aerial Wartime Images. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W13 :51–60, Juin 2019.

[Kri09] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[KSB+09] M.I. Krzywinski, J.E. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S.J. Jones, and M.A. Marra. Circos : An information aesthetic for comparative genomics. *Genome Research*, 2009.

[KSH12] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems (NIPS)*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[KT+09] P. Kohli, P.H. Torr, et al. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision (IJCV)*, 82(3) :302–324, 2009.

[KZ02] W.K. Kong and D. Zhang. Palmprint texture analysis based on low-resolution images for personal authentication. In *Object recognition supported by user interaction for service robots*, volume 3, pages 807–810 vol.3, Aug 2002.

[Lak98] R Lakmann. Barktex benchmark database of color textured images. *Koblenz-Landau University*, 1998.

[LBBH98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998.

[LBK17] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 700–708. Curran Associates, Inc., 2017.

[LCF+19] L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, and M. Pietikainen. From bow to cnn : Two decades of texture representation for texture classification. *International Journal of Computer Vision (IJCV)*, 127(1) :74–109, 2019.

[LCLTF+14] C. Le Cornet, J. Lortet-Tieulent, D. Forman, R. Béranger, A. Flechon, B. Fervers, J. Schüz, and F. Bray. Testicular cancer incidence to rise by 25% by 2025 in europe ? model-based predictions in 40 countries using population-based registry data. *European Journal of Cancer*, 50(4) :831–839, Mars 2014.

[LDG+17] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, July 2017.

[LFG+17] L. Liu, P. Fieguth, Y. Guo, X. Wang, and M. Pietikäinen. Local binary features for texture classification : Taxonomy and experimental study. *Pattern Recognition (PR)*, 62 :135–160, 2017.

[LFW+17] D. Lin, K. Fu, Y. Wang, G. Xu, and X. Sun. Marta gans : Unsupervised representation learning for remote sensing image classification. *IEEE Geoscience and Remote Sensing Letters*, 14(11) :2092–2096, 2017.

[LGCL18] Y. Liu, Y. Guo, W. Chen, and M.S. Lew. An extensive study of cycle-consistent generative networks for image-to-image translation. In *International Conference on Pattern Recognition (ICPR)*, pages 219–224. IEEE, Août 2018.

[LLF+16] L. Liu, S. Lao, P. W. Fieguth, Y. Guo, X. Wang, and M. Pietikäinen. Median robust extended local binary pattern for texture classification. *IEEE Transactions on Image Processing (TIP)*, 25(3) :1368–1381, March 2016.

[LMS16] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision (ECCV)*, pages 577–593, 2016.

[Low04] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2) :91–110, Novembre 2004.

[LSD15a] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[LSD15b] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[LSS+17] M.G. Lubner, A.D. Smith, K. Sandrasegaran, D.V. Sahani, and P.J. Pickhardt. CT texture analysis : Definitions, applications, biologic correlates, and challenges. *RadioGraphics*, 37(5) :1483–1503, Septembre 2017.

[LTYD03] L. Ma, T. Tan, Y. Wang, and D. Zhang. Personal identification based on iris texture analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(12) :1519–1533, Dec 2003.

[LYF+13] L. Liu, B. Yang, P. Fieguth, Z. Yang, and Y. Wei. Brint : A binary rotation invariant and noise tolerant texture descriptor. In *International Conference on Image Processing (ICIP)*, pages 255–259, 09 2013.

[LZL+12] L Liu, L Zhao, Y Long, G Kuang, and P Fieguth. Extended local binary patterns for texture classification. *Image and Vision Computing*, 30(2) :86–99, 2012.

[MAN10] S.K. Maxwell, M. Airola, and J.R. Nuckols. Using landsat satellite data to support pesticide exposure assessment in california. *International Journal of Health Geographics*, 9(1) :46, 2010.

[Mar11] U. Marmol. Use of gabor filters for texture classification of airborne images and lidar data. *Archiwum Fotogrametrii, Kartografii i Teledetekcji*, 22, 2011.

[Mat17] B. Mathieu. *Interactive multi-class image segmentation using superpixel classification and factor graph-based optimisation.* PhD thesis, Université Paul Sabatier - Toulouse III, Novembre 2017.

[Mau74] H. Maurer. Quantification of textures-textural parameters and their significance for classifying agricultural crop types from colour aerial photographs. *Photogrammetria*, 30(1) :21–40, Février 1974.

[MCVS16] G. Masi, D. Cozzolino, L. Verdoliva, and G. Scarpa. Pansharpening by convolutional neural networks. *Remote Sensing*, 8(7) :594, 2016.

[MFWCM18] S. Ma, J. Fu, C. Wen Chen, and T. Mei. Da-gan : Instance-level image translation by deep attention generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5657–5666, 2018.

[MG12] F. Mamalet and C. Garcia. Simplifying convnets for fast learning. In *International Conference on Artificial Neural Networks*, pages 58– –65. Springer, 2012.

[MGRT19] C. Mallet, S. Giordano, F. Remondino, and J. A. Trollvik. Workshop on geoprocessing and archiving of historical aerial images. *Workshop summary*, July 2019.

[MGS+19] F. Ma, F. Gao, J. Sun, H. Zhou, and A. Hussain. Weakly supervised segmentation of SAR imagery using superpixel and hierarchically adversarial CRF. *Remote Sensing*, 11(5) :512, Mars 2019.

[MLX+17] X. Mao, Q. Li, H. Xie, R. YK Lau, Z. Wang, and S. Paul Smolley. Least squares generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2794–2802, 2017.

[MM96] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 18(8) :837–842, 1996.

[MNDR19] P.-O. Mazagol, P. Niogret, M. Depeyre, and J. Riquier. Géovisualisation 3D du patrimoine englouti des Gorges de la Loire. In *Colloque international interdisciplinaire "Patrimoines et Territoires"*, Roanne, France, Novembre 2019.

[MPD+18] E. Maltezos, E. Protopapadakis, N. Doulamis, A. Doulamis, and C. Ioannidis. Understanding historical cityscapes from aerial imagery through machine learning. In *Euro-Mediterranean Conference*, pages 200–211. Springer, 2018.

[MPT20] K.M. Masoud, C. Persello, and V.A. Tolpekin. Delineation of agricultural field boundaries from sentinel-2 images using a novel super-resolution contour detector based on fully convolutional networks. *Remote Sensing*, 12(1) :59, 2020.

[MSW+18] D. Marmanis, K. Schindler, J.D. Wegner, S. Galliani, M. Datcu, and U. Stilla. Classification with an edge : Improving semantic image segmentation with boundary detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 135 :158–172, Janvier 2018.

[MTCA16] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. Fully convolutional neural networks for remote sensing image classification. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, Juilliet 2016.

[NAS20a] NASA. Base publique d'images de la NASA, 2020. `https://www.flickr.com/photos/nasacommons` (accès : 2020-03-31).

[NAS20b] NASA. Landsatlooks, 2020. https://earthobservatory.nasa.gov/features/LandsatLooks (accès : 2020-03-31).

[NNE18] K. Nazeri, E. Ng, and M. Ebrahimi. Image colorization using generative adversarial networks. In *International conference on articulated motion and deformable objects*, pages 85–94. Springer, 2018.

[Nvi19] Nvidia. Digits, 2019. https://developer.nvidia.com/digits (accès : 2019-01).

[O+13] D. Otair et al. Approximate k-nearest neighbour based spatial clustering using kd tree. *arXiv preprint arXiv :1303.1951*, 2013.

[ODO16] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10) :e3, 2016.

[oGiSS20] Laboratory of Geo-information Science and Remote Sensing. Hilda dataset, 2020.
https://www.wur.nl/en/Research-Results/Chair-groups/Environmental-Sciences/ (accès : 2020-04-02).

[OMP+02] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen. Outex - new framework for empirical evaluation of texture analysis algorithms. pages 701 – 706, 2002.

[OPM00] T. Ojala, M. Pietikäinen, and T. Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. In *European Conference on Computer Vision (ECCV)*, pages 404–420, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[OPM01] T. Ojala, M. Pietikäinen, and T. Mäenpää. A Generalized Local Binary Pattern Operator for Multiresolution Gray Scale and Rotation Invariant Texture Classification. In *Advances in Pattern Recognition, ICAPR 2001*, pages 399–408. Springer Berlin Heidelberg, 2001.

[OPM02] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(7) :971–987, 2002.

[OR19] E. Ozdemir and F. Remondino. Wwii air strike data analysis for risk mapping. *EuroSDR Workshop, Geoprocessing and Archiving of Historical Aerial Images*, June 2019.

[Pan18] X. Pan. *Shape and texture analysis : application to coin identification and coin grading*. Theses, Université Lyon 2 Lumière, Juin 2018.

[PBMS18] T. Postadjian, A. Le Bris, C. Mallet, and H. Sahbi. Superpixel partitioning of very high resolution satellite images for large-scale classification perspectives with deep convolutional neural networks. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, Juilliet 2018.

[PCS19] P. Picuno, G. Cillis, and D. Statuto. Investigating the time evolution of a rural landscape : How historical maps may provide environmental information when processed using a GIS. *Ecological Engineering*, 139 :105580, Novembre 2019.

[PGBPH19] A.T. Pinto, J.A. Gonçalves, P. Beja, and J. Pradinho Honrado. From archived historical aerial imagery to informative orthophotos : A framework for retrieving the past in long-term socioecological research. *Remote Sensing*, 11(11) :1388, Juin 2019.

[PHVH18] A. Porebski, V.T. Hoang, N. Vandenbroucke, and D. Hamad. Multi-color space local binary pattern-based feature selection for texture classification. *Journal of Electronic Imaging*, 27 :27 – 27 – 15, 2018.

[PNDS15] O.AB. Penatti, K. Nogueira, and J.A. Dos Santos. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains ? In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 44–51, 2015.

[PPC+17] C. Pouchieu, C. Piel, C. Carles, A. Gruber, C. Helmer, S. Tual, E. Marcotullio, P. Lebailly, and I. Baldi. Pesticide use in agriculture and parkinson's disease in the AGRICAN cohort study. *International Journal of Epidemiology*, 47(1) :299–310, Novembre 2017.

[PS06] M. Petrou and P.G. Sevilla. *Image processing : dealing with texture*. Wiley, 2006.

[PSM+19] D. Poli, M. Strudl, K. Moe, F. Baumann, E. Bollmann, and C. Casarotto. Historical 3d glacier modelling. *EuroSDR Workshop, Geoprocessing and Archiving of Historical Aerial Images*, June 2019.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12 :2825–2830, 2011.

[PVMH14] A. Porebski, N. Vandenbroucke, L. Macaire, and D. Hamad. A new benchmark image test suite for evaluating colour texture classification schemes. *Multimedia Tools and Applications*, 70(1) :543–556, 2014.

[PZ15] M. Pietikäinen and G. Zhao. Two decades of local binary patterns. In *Advances in Independent Component Analysis and Learning Machines*, pages 175–210. Elsevier, 2015.

[Qui86] J.R. Quinlan. Induction of decision trees. *Machine learning*, 1(1) :81–106, 1986.

[QZS+16] X Qi, G Zhao, L Shen, Q Li, and M Pietikäinen. Load : local orientation adaptive descriptor for texture and material classification. *Neurocomputing*, 184 :28–35, 2016.

[RAHI13] Md A. Rahim, Md S. Azam, N. Hossain, and Md R. Islam. Face recognition using local binary patterns (lbp). *Global Journal of Computer Science and Technology*, 2013.

[RBCJT19] R. Ratajczak, S. Bertrand, C.F. Crispim-Junior, and L. Tougne. Efficient Bark Recognition in the Wild. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, Feb 2019.

[RBLG16] O. Regniers, L. Bombrun, V. Lafon, and C. Germain. Supervised Classification of Very High Resolution Optical Images Using Wavelet-Based Textural Features. *IEEE Transactions on Geoscience and Remote Sensing*, 54(6) :3722–3735, 2016.

[RCJF+18] R. Ratajczak, C.F. Crispim-Junior, E. Faure, B. Fervers, and L. Tougne. Reconstruction automatique de l'occupation du sol à partir d'images aériennes historiques monochromes : une étude comparative. In *Conférence Française de Photogrammétrie et de Télédétection*, Marne-la-Vallée, France, Juin 2018.

[RCJF+19a] R. Ratajczak, C.F. Crispim-Junior, E. Faure, B. Fervers, and L. Tougne. Automatic Land Cover Reconstruction From Historical Aerial Images : An Evaluation of Features Extraction and Classification Algorithms. *IEEE Transactions on Image Processing (TIP)*, 2019.

[RCJF+19b] R. Ratajczak, C.F. Crispim-Junior, E. Faure, B. Fervers, and L. Tougne. Toward an Unsupervised Colorization Framework for Historical Land Use Classification. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, Juilliet 2019.

[RCJF+19c] R. Ratajczak, C.F. Crispim-Junior, B. Fervers, E. Faure, and L. Tougne. Pseudo-Cyclic Network for Unsupervised Colorization with Handcrafted Translation and Output Spatial Pyramids. In *SUMAC @ ACM Multimedia*. ACM, Octobre 2019.

[RCJF+20] R. Ratajczak, C.F. Crispim-Junior, B. Fervers, E. Faure, and L. Tougne. Semantic Segmentation Refinement with Deep Edge Superpixels to Enhance Historical Land Cover. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, (to appear), September 2020. IEEE.

[RDD08] G. Rabatel, C. Delenne, and M. Deshayes. A non-supervised approach using gabor filters for vine-plot detection in aerial images. *Computers and Electronics in Agriculture*, 62(2) :159–168, Juilliet 2008.

[RDS+15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3) :211–252, 2015.

[RFB15] O. Ronneberger, P. Fischer, and T. Brox. U-net : Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015.

[RHW86] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088) :533–536, 1986.

[RMC15] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[Ros58] F. Rosenblatt. The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6) :386, 1958.

[RRVB17] R. Raghavendra, K.B. Raja, S. Venkatesh, and C. Busch. Transferable deep-cnn features for detecting digital and print-scanned morphed face images. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1822–1830. IEEE, 2017.

[SAA18] L. Sulimowicz, I. Ahmad, and A. Aved. Superpixel-enhanced pairwise conditional random field for semantic segmentation. In *International Conference on Image Processing (ICIP)*, Oct 2018.

[SBD19] J. Shepherd, P. Bunting, and J. Dymond. Operational large-scale segmentation of imagery based on iterative elimination. *Remote Sensing*, 11(6) :658, Mars 2019.

[SBF15] C. Silva, T. Bouwmans, and C. Frélicot. An extended center-symmetric local binary pattern for background modeling and subtraction in videos. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2015.

[SCB+17] I. Slimene, N. Chehata, J-S. Bailly, I. Farah, and P. Lagacherie. Parcel-based active learning for large extent cultivated area mapping. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pages 1–10, 09 2017.

[SD16] F. Sandid and A. Douik. Robust color texture descriptor for material recognition. *Pattern Recognition Letters (PRL)*, 2016.

[SFYW14] J. Sun, G. Fan, L. Yu, and X. Wu. Concave-convex local binary features for automatic target recognition in infrared imagery. *EURASIP Journal on Image and Video Processing*, 2014(1) :23, 2014.

[SHL18] D. Stutz, A. Hermans, and B. Leibe. Superpixels : An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166 :1–27, 2018.

[SLF+17] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler : Controlling deep image synthesis with sketch and color. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5400–5409, 2017.

[SPMV13] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector : Theory and practice. *International Journal of Computer Vision (IJCV)*, 105(3) :222–245, Juin 2013.

[SRDMM01] N.E. Skakkebæk, E. Rajpert-De Meyts, and K.M. Main. Testicular dysgenesis syndrome : an increasingly common developmental disorder with environmental aspects : Opinion. *Human Reproduction*, 16(5) :972–978, Mai 2001.

[SSSJ10] S. Schmiedel, J. Schüz, N.E. Skakkebæk, and C. Johansen. Testicular germ cell cancer incidence in an immigration perspective, denmark, 1978 to 2003. *Journal of Urology*, 183(4) :1378–1382, Avril 2010.

[SZ14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[TT10] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing (TIP)*, 19(6) :1635–1650, 2010.

[TV08] B. Triggs and J.J. Verbeek. Scene segmentation with crfs learned from partially labeled images. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1553–1560, 2008.

[UVL16] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization : The missing ingredient for fast stylization. *arXiv preprint arXiv :1607.08022*, 2016.

[UVL18] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9446–9454, 2018.

[VBt18] T. Verelst, M. Berman, and et al. Generating superpixels with deep representations. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2018.

[VRB20] P. Vitoria, L. Raad, and C. Ballester. Chromagan : Adversarial picture colorization with semantic class distribution. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2434–2443, 2020.

[VS08] A. Vedaldi and . Soatto. Quick shift and kernel methods for mode seeking. In *European Conference on Computer Vision (ECCV)*, pages 705–718, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[Š14] M. Švab. *Computer-vision-based tree trunk recognition.* PhD thesis, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2014.

[WB18] S.L. Woldu and A. Bagrodia. Update on epidemiologic considerations and treatment trends in testicular cancer. *Current Opinion in Urology*, 28(5) :440–447, Septembre 2018.

[WDH+04] Y.-Y. Wan, J.-X. Du, D.-S. Huang, Z. Chi, Y.-M. Cheung, X.-F. Wang, and G.-J. Zhang. Bark texture feature extraction based on statistical texture analysis. In *International Symposium on Intelligent Multimedia, Video and Speech Processing.* IEEE, 2004.

[Wer90] P.J. Werbos. Backpropagation through time : what it does and how to do it. *Proceedings of the IEEE*, 78(10) :1550–1560, 1990.

[WFZ+18] M. Wang, X. Fei, Y. Zhang, Z. Chen, X. Wang, J.Y. Tsou, D. Liu, and X. Lu. Assessing texture features to classify coastal wetland vegetation from high spatial resolution imagery using completed local binary patterns (CLBP). *Remote Sensing*, 10(5) :778, Mai 2018.

[WHT08] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Workshop on faces in real-life images : Detection, alignment, and recognition*, 2008.

[Wil19] A. Williams. The digital transformation of the national collection of aerial photography (NCAP) : achievements to-date & next stage. *EuroSDR Workshop, Geoprocessing and Archiving of Historical Aerial Images*, June 2019.

[WLH+17] J. Wang, C. Luo, H. Huang, H. Zhao, and S. Wang. Transferring pre-trained deep CNNs for remote scene classification with general features learned from linear PCA network. *Remote Sensing*, 9(3) :225, Mars 2017.

[WLZ+18] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8798–8807, 2018.

[WSFW15] X. Wu, J. Sun, G. Fan, and Z. Wang. Improved local ternary patterns for automatic target recognition in infrared imagery. *Sensors*, 15(3) :6399, 2015.

[WSG11] A. Wendel, S. Sternig, and M. Godec. Automated identification of tree species from images of the bark, leaves and needles. In *Computer Vision Winter Workshop*, 2011.

[WTYM18] J.D. Wegner, D. Tuia, M. Yang, and C. Mallet. Foreword to the theme issue on geospatial computer vision. *ISPRS Journal of Photogrammetry and Remote Sensing*, Janvier 2018.

[XKP19] X. Xia, M. Koeva, and C. Persello. Extracting cadastral boundaries from uav images using fully convolutional networks. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 2455–2458. IEEE, 2019.

[XPK19] X. Xia, C. Persello, and M. Koeva. Deep fully convolutional networks for cadastral boundary detection from uav images. *Remote sensing*, 11(14) :1725, 2019.

[XT15] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1395–1403, 2015.

[XXC12] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *Advances in neural information processing systems (NIPS)*, pages 341–349, 2012.

[XXFC18] Y. Xu, Z. Xie, Y. Feng, and Z. Chen. Road extraction from high-resolution remote sensing imagery using deep learning. *Remote Sensing*, 10(9) :1461, 2018.

[YBFU15] J. Yao, M. Boben, S. Fidler, and R. Urtasun. Real-time coarse-to-fine topologically preserving segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2947–2955, 2015.

[YGZS17] Y. Yu, Z. Gong, P. Zhong, and J. Shan. Unsupervised representation learning with deep convolutional neural network for remote sensing images. In *International Conference on Image and Graphics*, pages 97–108. Springer, 2017.

[YN10] Y. Yang and S. Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *International Conference on Advances in Geographic Information Systems - GIS (SIGSPATIAL)*. ACM Press, 2010.

[YR14] L. Yan and D.P. Roy. Automated crop field extraction from multi-temporal web enabled landsat data. *Remote Sensing of Environment*, 144 :42–64, 2014.

[YXW18] X. Yang, D. Xie, and X. Wang. Crossing-domain generative adversarial networks for unsupervised multi-domain image-to-image translation. In *ACM International Conference on Multimedia (ACMMM)*, MM '18, pages 374–382, New York, NY, USA, 2018. ACM.

[ZAMP11] G. Zhao, T. Ahonen, J. Matas, and M. Pietikainen. Rotation-invariant image and video description with local binary pattern features. *IEEE Transactions on Image Processing (TIP)*, 21(4) :1465–1477, 2011.

[ZBC13] C. Zhu, C.-E. Bichot, and L. Chen. Image region description using orthogonal combination of local binary patterns enhanced with color information. *Pattern Recognition (PR)*, 46(7) :1949–1963, 2013.

[ZIE16] R. Zhang, P. Isola, and A.A. Efros. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*, pages 649–666. Springer, 2016.

[ZJRP+15] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P.HS. Torr. Conditional random fields as recurrent neural networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1529–1537, 2015.

[ZMLS07] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories : A comprehensive study. *International Journal of Computer Vision (IJCV)*, 73(2) :213–238, 2007.

[ZPIE17] J.-Y. Zhu, T. Park, P. Isola, and A.A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2223–2232, 2017.

[ZY98] C. Zhu and X. Yang. Study of remote sensing image texture analysis and classification using wavelet. *International Journal of Remote Sensing*, 19(16) :3197–3203, 1998.

[ZYS09] H. Zhou, Y. Yuan, and C. Shi. Object tracking using sift features and mean shift. *Computer Vision and Image Understanding (CVIU)*, 113(3) :345–352, 2009.

[ZZI+17] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG)*, 9(4), 2017.

[ZZS+19] A. Zhang, S. Zhang, G. Sun, F. Li, H. Fu, Y. Zhao, H. Huang, J. Cheng, and Z. Wang. Mapping of coastal cities using optimized spectral–spatial features based multi-scale superpixel classification. *Remote Sensing*, 11(9) :998, Avril 2019.

Annexe A

Gouramic

Cette section Annexe contient une description illustrée du logiciel Gouramic.

Interface

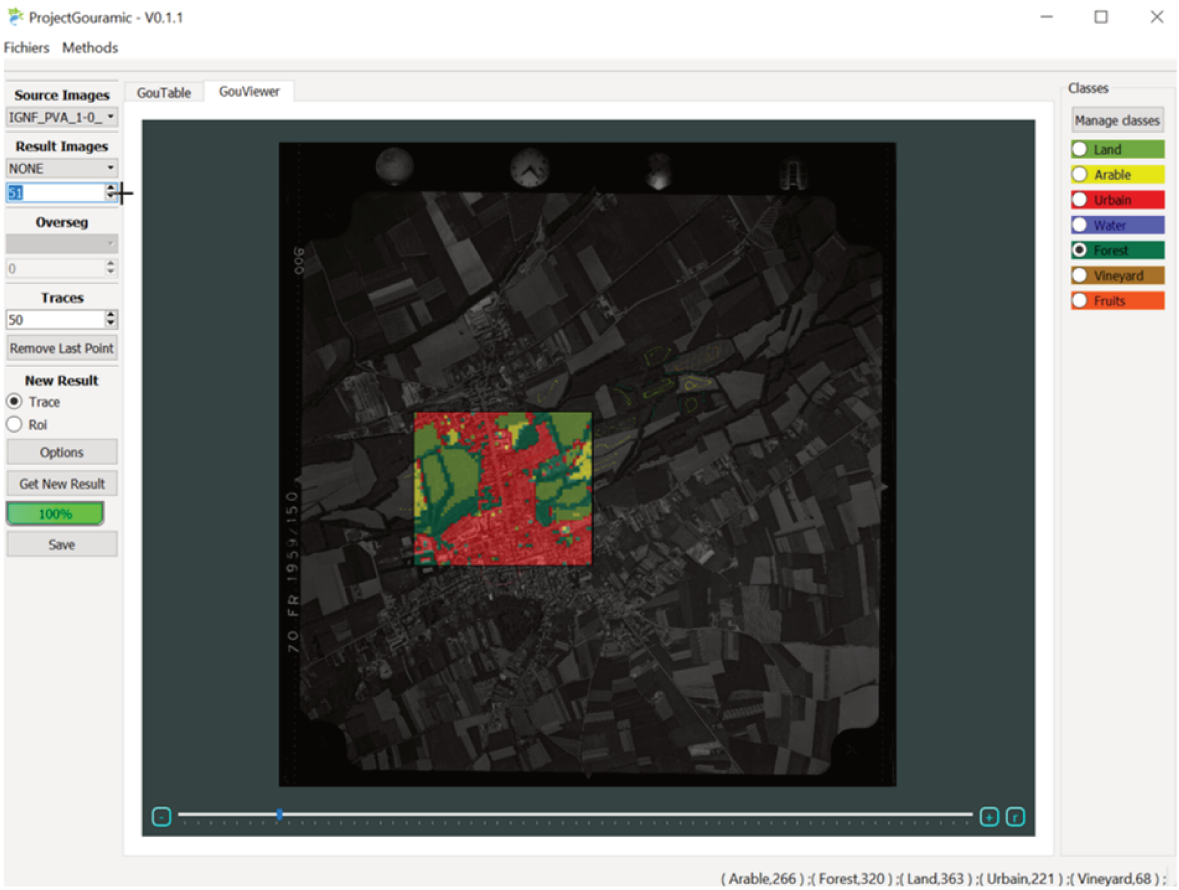


FIGURE A.1 – Illustration de l'interface du logiciel Gouramic.

L'interface de Gouramic a été pensée avec les géomaticiens du Centre Léon Bérard afin d'avoir un logiciel relativement facile à utiliser. Elle est présentée sur la figure A.1. Celle-ci permet à l'utilisateur d'accéder à un dossier contenant des images aériennes historiques encodées au format JPEG2000 afin de les visualiser et de les traiter. Ce logiciel offre la possibilité de zoomer et de dézoomer sur les images. Il permet également de modifier les classes d'occupation du sol que l'utilisateur désire détecter. Le lancement des traitements pour l'image courante se fait d'un simple click sur un bouton [1].

1. Exemple d'utilisation : `https://youtu.be/VJ7zR9o8oxM` (2020-07-01)

Fonctionalités

D'un point de vue fonctionnalités, le logiciel Gouramic a été développé afin d'accélérer l'annotation des images aériennes historiques. Il se base sur nos travaux présentés dans le chapitre 3 de ce manuscrit. Gouramic intègre des chaines de traitements basées sur l'extraction et la classification de caractéristiques de texture. Le processus d'utilisation générique est résumé sur la figure A.2 et décrit ci-après.

L'utilisateur peut commencer par sélectionner une zone d'intérêt sur laquelle il aimerait obtenir un résultat (par défaut, l'image entière est la zone d'intérêt). On propose ensuite à l'utilisateur d'annoter partiellement les images aériennes historiques afin d'entraîner un classifieur indépendant pour chaque image. Ces annotations partielles sont représentées par des traces, qui sont tout simplement des pixels colorés. La couleur de chaque pixel correspond à une classe d'occupation du sol définie par l'utilisateur.

Une fois que l'utilisateur a fini de réaliser ses traces, il peut sélectionner les paramètres des méthodes à utiliser. L'utilisateur a le choix d'utiliser un ou plusieurs filtres de texture (le LCoLBP par défaut), ainsi qu'un classifieur au choix parmi 4 (SVM, MLP, *Random Forest*, KNN). Le SVM est sélectionné par défaut afin de mitiger qualité des résultats et temps d'entraînement. Les paramètres des classifieurs sont supposés fixés dans le logiciel, mais un utilisateur expérimenté a la possibilité de les modifier via un fichier csv (*e.g.*, nombre d'arbres dans la forêt d'arbres de décisions). Pour chaque pixel annoté par l'utilisateur, une imagette de taille S × S va être extraite. Le descripteur de texture va être appliqué sur chaque imagette annotée, et les vecteurs de caractéristiques résultants vont permettre d'entraîner le classifieur. Une fois cette étape d'entraînement réalisée, l'inférence est faite à l'aide d'une fenêtre glissante sur la zone d'intérêt préalablement sélectionnée par l'utilisateur. Le pas P_a de la fenêtre glissante correspond à la finesse du résultat final. En pratique, on classifie une imagette de taille S × S, et la classe obtenue est affichée sur une surface de taille $P_a \times P_a$. Les deux surfaces S × S et $P_a \times P_a$ sont centrées sur le même pixel. Ce principe est illustré sur la figure A.3. Si $P_a = 1$, alors on obtient une étiquette pour chaque pixel. Le fait de considérer $P_a > 1$ permet de mitiger la précision spatiale des résultats et le temps de traitement (il

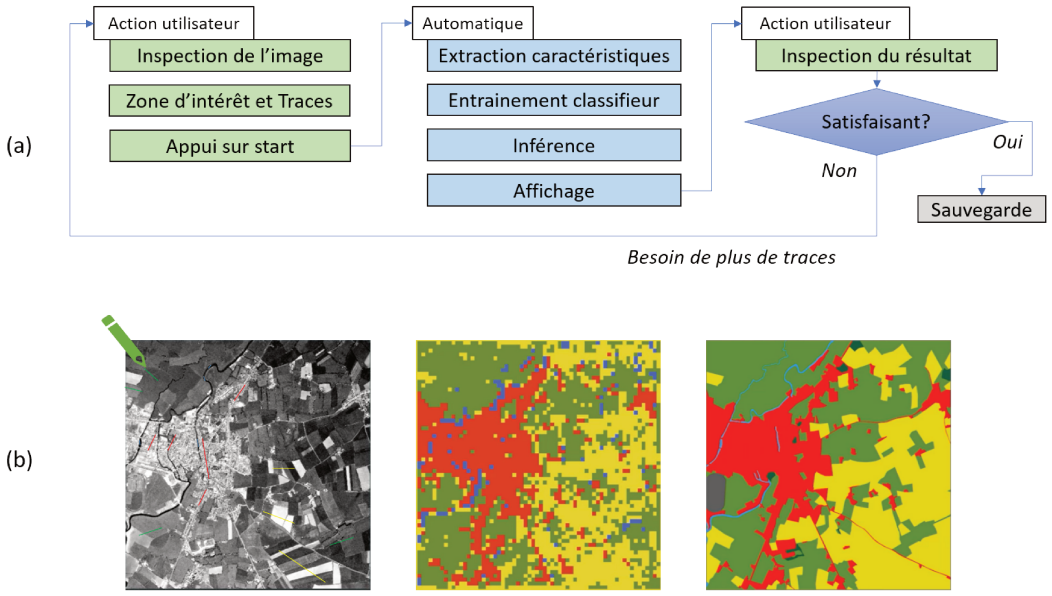


FIGURE A.2 – Schéma illustrant l'utilisation du logiciel Gouramic. (a) Pipeline utilisateur. (b) Traces schématiques à gauche, exemple de résultat au centre, vérité terrain à droite.

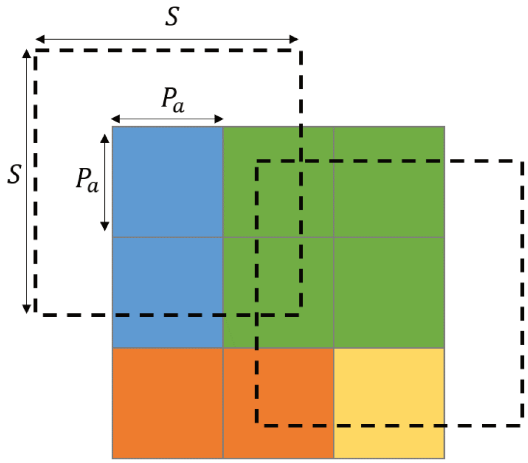


FIGURE A.3 – Schéma illustrant la différence entre S et P_a pour un résultat à l'imagette près.

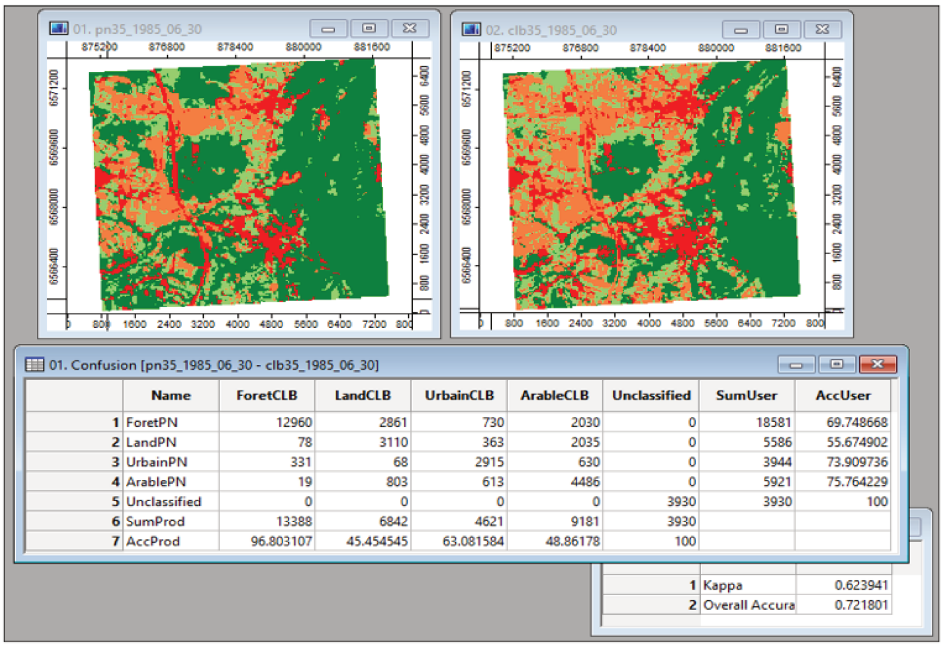
y a moins de calculs à réaliser). Les résultats obtenus ne sont alors plus au pixel près, mais à l'imagette près : une même étiquette est attribuée à tous les pixels de l'imagette $P_a \times P_a$. Le choix des valeurs de S et P_a est laissé à l'utilisateur en fonction de ses besoin (par défaut, S = 100 et $P_a = 50$). Après avoir obtenu un premier résultat, l'utilisateur peut le visualiser, en zoomant au besoin. Si le résultat est satisfaisant, il peut le sauvegarder. Sinon, il peut réaliser plus de traces et relancer le processus afin d'obtenir un résultat de meilleur qualité. Ce processus permet de créer une boucle de retour pour l'utilisateur, lui permettant de vérifier et d'améliorer les résultats obtenus autant que de besoin.

Performances qualitatives

Gouramic a été pris en mains par des étudiants en géomatique de l'Université Jean Monet, Saint-Étienne, France. Ces étudiants devaient dans un premier temps réaliser un retour d'expérience quant à l'utilisation du logiciel pour des novices. Ces derniers ont trouvé l'interface intuitive et facile à utiliser. Ils n'ont pas tenté de modifier les méthodes sélectionnées par défaut car ils ne comprenaient pas les différences entre celles-ci. D'un point de vue fonctionnalités, ils ont trouvé que le logiciel répondait correctement au besoin. Ils s'accordent sur ce point avec les géomaticiens plus expérimentés du Centre Léon Bérard. Ils se sont néanmoins interrogés sur l'interprétation des valeurs RVB représentées sur les images de résultats : ils s'attendaient à ne trouver qu'une seule valeur. Cela a mis en avant la nécessité de pouvoir encoder les résultats générés pour qu'ils soient plus facilement utilisables au sein d'autres outils SIG. De plus, les étudiants se sont intéressés à la variabilité des résultats obtenus lorsque plusieurs utilisateurs réalisent des traces différentes (voir figure A.4). Ils n'ont cependant pas eu le temps de mener ces expériences préliminaires sur de vastes ensembles de données ou avec de nombreux utilisateurs. Ce dernier point est actuellement en cours d'approfondissement dans le cadre du projet GOURAMIC soutenu par le LabEx Institut des Mondes Urbains (IMU).

Performances quantitatives

Les résultats de Gouramic ont été comparés par Matthieu Dubuis, géomaticien, avec les cartes d'occupation du sol de Corine Land Cover pour certains sujets de l'étude TESTIS nés autour des années 1990. Le but était ici d'avoir une idée de comment le logiciel fonctionne par rapport à ce jeu de données considéré comme étant un standard pour le territoire européen, et ce malgré des différences notables dans la taille minimale des parcelles générées ($P_a \times P_a$ pixels pour Gouramic, contre 25 hectares pour Corine Land Cover). Dans ce cas de figure, il s'agit uniquement de vérifier



	Name	ForetCLB	LandCLB	UrbainCLB	ArableCLB	Unclassified	SumUser	AccUser
1	ForetPN	12960	2861	730	2030	0	18581	69.748668
2	LandPN	78	3110	363	2035	0	5586	55.674902
3	UrbainPN	331	68	2915	630	0	3944	73.909736
4	ArablePN	19	803	613	4486	0	5921	75.764229
5	Unclassified	0	0	0	0	3930	3930	100
6	SumProd	13388	6842	4621	9181	3930		
7	AccProd	96.803107	45.454545	63.081584	48.86178	100		

1	Kappa	0.623941
2	Overall Accura	0.721801

FIGURE A.4 – Comparaison de deux résultats obtenus avec Gouramic par deux utilisateurs différents.

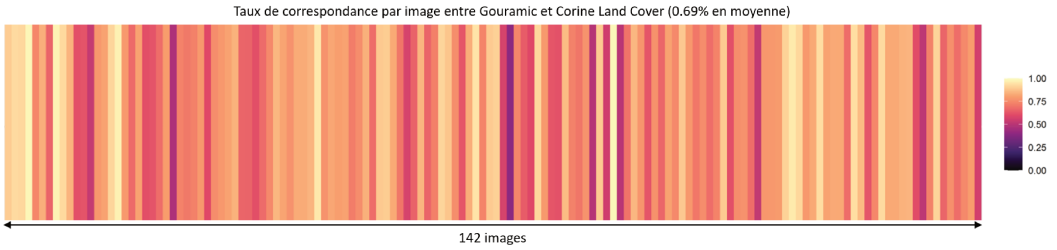


FIGURE A.5 – Comparaison des résultats issus de Gouramic avec les cartes d'occupation du sol de Corine Land Cover.

que les deux types d'occupation du sol sont, globalement, en accords. La figure A.5 résume les résultats obtenus pour 142 images. On constate un taux de concordance de 69%, ce qui indique un accord positif entre les deux sources de données. Les raisons des divergences entre les deux jeux de données n'ont pas encore été déterminées (*i.e.*, qui entre CLC et Gouramic a raison ? Quel est l'impact de la taille des parcelles ?). Une comparaison plus adaptée nécessiterait la génération entièrement manuelle de cartes d'occupation du sol afin de posséder une vérité terrain adaptée à l'étude TESTIS. L'utilisation de données plus récentes, telles que le Registre Parcellaire Graphique qui possède des parcelles annotées de différentes tailles, est également à l'étude pour évaluer le logiciel Gouramic.

Liste des figures

1.1 Distribution des sujets recrutés dans l'étude TESTIS à leur année de naissance, France métropolitaine (Matthieu Dubuis, 2020). 4
1.2 Taux d'incidence standardisé pour l'âge (*Age-Standardized Rates*, ASR) du cancer du testicule en 2018. Données fournies par le CIRC [dRslCC20]. 5
1.3 Taux de mortalité standardisé pour l'âge (*Age-Standardized Rates*, ASR) lié au cancer du testicule en 2018. Données fournies par le CIRC [dRslCC20]. 6
1.4 Évolution de la consommation moyenne de pesticides dans le monde par hectare de surface cultivée de 1990 à 2017. Données fournies par la FAO. 7
1.5 Illustration de l'étude SIGEXPO. 8
1.6 Diagramme de flux simplifié de l'étude TESTIS. 9
1.7 Exemples de cartes d'occupation du sol générées manuellement à partir d'images aériennes historiques au niveau de la commune de Die, France, en 1978. 11
1.8 Exemple d'annotations issues du jeu de données CLC 2018. 13
1.9 Exemple d'annotations issues du jeu de données HILDA [FHV+14; FVCH15] pour les années 1910 et 1990. 14
1.10 Première photographie acquise par le satellite Explorer 6 en 1959. 15
1.11 Image Landsat-1 multispectrale en fausses couleurs au niveau de Garden City, Kansas, USA. 16
1.12 Etat des lieux des images aériennes disponibles en Europe. Image issue de [GM19]. . 18
1.13 Cartes de chaleur des acquisitions d'images aériennes par l'IGN (a) entre 1919 et 1970, et (b) entre 1970 et 2000. 20
1.14 Nombre relatif d'images aériennes disponibles sur remonterletemps [IGN20b] par type d'acquisition pour la période 1970-2000. 20
1.15 Exemple d'une image aérienne historique de 1956 mis en correspondance avec une image aérienne récente de 2015 au niveau de Strasbourg, France. 21

2.1 Schéma générique de l'obtention d'un vecteur de caractéristiques à partir d'une image et de son utilisation pour différentes tâches. 26
2.2 Exemples de d'images texturées dans divers domaines d'applications. Image extraite de [LCF+19]. 27
2.3 Exemple de la construction d'une matrice de cooccurrences de niveaux de gris. . . . 28
2.4 Schéma représentant le pipeline générique pour l'extraction de caractéristiques à l'aide de LBP. 30
2.5 Schéma représentant différents voisinages (P,R) pour les motifs binaires de type LBP. 31
2.6 Schéma représentant les 58 motifs binaires uniformes pour un degré d'uniformité égal à 2. 32
2.7 Exemples de superpixels. 35
2.8 Schéma illustrant le principe des K plus proches voisins avec 3 classes. 40
2.9 Schéma illustrant le principe des forêts aléatoires d'arbres décisionnels. 42
2.10 Schéma illustrant le principe d'un réseau de neurones. 43
2.11 Schéma illustrant le principe d'un réseau de neurones à convolutions pour la classification. 45

2.12 Schéma illustrant le principe d'un réseau de neurones entièrement convolutif (FCN) de type encodeur-décodeur. 46
2.13 Schéma illustrant le principe de la convolution classique. 47
2.14 Schéma illustrant plusieurs filtres de convolutions 2D. 48
2.15 Schéma illustrant le principe du *pooling* avec une taille de 2 pixels et un pas de 2 pixels. 49

3.1 Deux exemples d'images aériennes historiques acquises en France (a)(c) et leurs occupations du sol annotées manuellement par des géomaticiens (b)(d). 52
3.2 Processus d'extraction d'imagettes pour la création de HistAerial. 55
3.3 Exemples d'imagettes de différentes tailles dans HistAerial. 57
3.4 Exemples de filtres et de voisinages utilisés avec les filtres de type LBP. 59
3.5 Différents types de filtres LBP et leurs codes binaires obtenus sur un même voisinage (P,R). 64
3.6 Représentation schématique des filtres utilisés dans le LCoLBP. 65
3.7 Schéma de la génération d'un histogramme de textures avec le LCoLBP appliqué sur un voisinage (P,R) = $(8,\{1,2,3\})$. 65
3.8 Schéma simplifié d'un block résiduel. 67
3.9 Processus générique d'évaluation sur le jeu de données HistAerial. 68
3.10 Composition des jeux d'entraînement, de validation et de test de HistAerial. 69
3.11 Matrices de confusion normalisées pour le filtre LCoLBP sur le jeu de données équilibré en taille. 73
3.12 Matrices de confusion normalisées pour AlexNet sur le jeu de données équilibré en taille. 74
3.13 Exemples d'images d'écorces d'arbres du jeu de données Bark-101 [RBCJT19]. 76
3.14 Schéma de l'algorithme de réduction de l'histogramme de teintes appliqué pour 3 itérations. 77
3.15 Schéma représentant l'extraction de $N_s = 7$ statistiques tardives à partir de l'histogramme du filtre LCoLBP. 78

4.1 Exemples de peintures, de paysages et d'images aériennes colorisées avec SpyncoGan. 84
4.2 Illustration du principe d'un réseau de neurones cyclique non-supervisé basé sur deux GAN (a) exploitant la consistance cyclique (b)(c). Schéma adapté de [ZPIE17]. 86
4.3 Illustration des méthodes de colorisation proposées par (a) Zhang *et al.* [ZIE16] et (b) Iizuka *et al.* [ISSI16]. 87
4.4 Méthode de colorisation non-supervisée d'images aériennes historiques proposée. 88
4.5 Architecture des générateurs et des discriminateurs utilisés dans Col-Cycle. 90
4.6 Résultats visuels mettant en avant l'effet mosaïque. 92
4.7 Exemples d'imagettes utilisées pour l'entraînement de Col-Cycle. 93
4.8 Résultats de l'évaluation par note moyenne d'opinions visant à déterminer la qualité de la couleur des images générées. Plus la valeur est élevée, meilleure est la qualité. 94
4.9 Exemples d'images aériennes VHR colorisée avec Col-Cycle. 95
4.10 Taux de bonne classification sur le jeu de données HistAerial (7 classes d'occupation du sol) à l'aide de filtres de textures et de statistiques couleur. 96
4.11 Schéma d'un réseau de neurones cyclique [ZPIE17] (a) et d'un réseau de neurones pseudo-cyclique. 97
4.12 Schéma d'une pyramide spatiale de sortie (OSP). 99
4.13 Schéma de SpyncoGan. 101
4.14 Résultats qualitatifs obtenus durant l'entraînement de SpyncoGan sur le jeu de données Cifar-10. 105
4.15 Sorties intermédiaires de SpyncoGan sur des peintures de Cézanne à différentes *epochs*.108
4.16 *Mean Square Error* (MSE) et *Structural Similarity Measure* (SSIM) entre les images colorisées et les images réelles des jeux de données Cifar-10 (a,b), UCMerced Land Use (c,d), peintures de Cézanne (e,f) et *Landscape* photos (g,h). 109

4.17 Diagrammes de cordes pour la classification inter-domaine sur (a) UCMerced Land Use et (b) Cifar-10 avec $G_1(I)$. R-G est niveaux de gris (*Real-Gray*). R-C est pour couleurs réelles (*Real-Color*). [Nombre] indique l'*epoch* de colorisation. 110
4.18 Exemples d'images aériennes historiques colorisées avec SpyncoGan après 120 *epochs*. Ligne du haut : images panchromatiques. Ligne du bas : colorisation par imagette avec remplacement de textures. 113
4.19 Exemples de peintures de Cézanne colorisées avec SpyncoGan durant l'entraînement ($G_1(I_A) = O_{d_1}^{W\times H}$). 115
4.20 Exemples de photographies de paysages colorisées avec SpyncoGan durant l'entraînement ($G_1(I_A) = O_{d_1}^{W\times H}$). 116
4.21 Exemples d'images aériennes du jeu de données UCMerced Land Use colorisées avec SpyncoGan durant l'entraînement ($G_1(I_A) = O_{d_1}^{W\times H}$). 117
4.22 Exemples d'images du jeu de données de Cifar-10 colorisées avec SpyncoGan durant l'entraînement ($G_1(I_A) = O_{d_1}^{W\times H}$). 118

5.1 Schéma générique de l'approche proposée, sans colorisation. 120
5.2 Illustration de la différence entre contours, bords profonds, et bord profonds seuillés. 121
5.3 Illustration d'un CRF basé sur des relations d'adjacence selon une grille régulière (a) et d'un CRF dense (b). Le fait de pouvoir tenir compte des relations entre pixels éloignés permet d'obtenir des résultats plus réalistes. Images extraites de la présentation de [KK11]. 123
5.4 Illustration du réseau de neurones HED. Image extraite de [XT15]. A partir d'une image source, des bords profonds sont générés à plusieurs échelles, puis fusionnés à l'aide d'un filtre de convolutions. 124
5.5 Résultats obtenus avec HED pour la détection de bords sur une imagette de 1024x1024 pixels. 125
5.6 Schéma illustrant la génération de représentations lissées à l'aide de superpixels extrait de bords profonds. 126
5.7 Distribution des étiquettes/classes dans les jeux de données utilisés. 128
5.8 Exemples d'images aériennes segmentées grossièrement. 129
5.9 Exemples de résultats. 130
5.10 Résultats obtenus en intégrant l'information portée par les superpixels au sein d'un champ aléatoire conditionnel dense. 131
5.11 Comparaison entre le filtre générique (à gauche, 83.66%) et le filtre bilatéral (à droite, 83.95%) pour l'intégration de l'information portée par les DES. 131
5.12 Comparaison visuelle entre DES-*mean*, DES-*median* et DES-*std*. 132
5.13 Comparaison entre l'utilisation de DES-*mean* (à gauche, 83.95%) et de DES-*median* (à droite, 83.81%) pour le post-traitement. 132
5.14 Schéma générique de l'approche proposée, avec colorisation. 133
5.15 Schéma illustrant la génération de représentations lissées (DES-*mean*) à l'aide de superpixels extrait de bords profonds à partir d'images en couleurs, niveaux de gris et colorisées. 134
5.16 Résultats obtenus en post-traitement avec un CRF dense sans DES-*mean*. 135

A.1 Illustration de l'interface du logiciel Gouramic. I
A.2 Schéma illustrant l'utilisation du logiciel Gouramic. (a) Pipeline utilisateur. (b) Traces schématiques à gauche, exemple de résultat au centre, vérité terrain à droite. II
A.3 Schéma illustrant la différence entre S et P_a pour un résultat à l'imagette près. III
A.4 Comparaison de deux résultats obtenus avec Gouramic par deux utilisateurs différents. IV
A.5 Comparaison des résultats issus de Gouramic avec les cartes d'occupation du sol de Corine Land Cover. IV

Liste des tableaux

1.1 Métadonnées correspondant au progamme Corine Land Cover (CLC). 12

3.1 Le jeu de données HistAerial complet. 56
3.2 Le sous ensemble équilibré en taille du jeu de données HistAerial. 56
3.3 Le sous ensemble équilibré en classe du jeu de données HistAerial. 56
3.4 Meilleurs résultats obtenus sur le sous ensemble équilibré en taille du jeu de données HistAerial pour des imagettes de 25 pixels × 25 pixels. 70
3.5 Meilleurs résultats obtenus sur le sous ensemble équilibré en taille du jeu de données HistAerial pour des imagettes de 50 pixels × 50 pixels. 71
3.6 Meilleurs résultats obtenus sur le sous ensemble équilibré en taille du jeu de données HistAerial pour des imagettes de 100 pixels × 100 pixels. 72
3.7 Meilleurs résultats obtenus sur le sous ensemble de HistAerial équilibré en classe. . 75
3.8 Caractéristiques de différents jeux de données d'écorces d'arbres considérés. 77
3.9 Étude par ablation des statistiques tardives appliquées aux filtres LCoLBP et CLBP sur le jeu de données BarkTex. 80
3.10 Résultats obtenus avec un 1-NN sur les jeux de données BarkTex, AFF et Trunk12. . 81
3.11 Résultats obtenus sur les jeux de données NewBarkTex et Bark-101. 81

4.1 Résultat de l'ablation des sorties. Métriques calculées toutes les 10 *epochs* (entraînement de 50 *epochs*) puis moyennées (*Avg.*). 106
4.2 Résultats de l'ablation de la fonction de coût liée aux hautes fréquences sur les peintures de Cézanne. Métriques calculées toutes les 10 *epochs* (entraînement de 50 *epochs*) puis moyennées (Avg.). 107
4.3 Comparaison des colorisations produites par Col-Cycle, CycleGan (9 couches résiduelles), et SpyncoGan sur les jeux de données UCMerced Land Use et les peintures de Cézanne (meilleurs résultats parmi les 50 premières *epochs*). 110
4.4 Taux de classification (%) inter-domaine moyennés sur tous les domaines couleur. (1) VGG-16 entraîné pour 40 *epochs* sur UCMerced Land Use et (2) AlexNet entraîné pour 20 *epochs* sur Cifar-10. 112
4.5 Comparaison de l'apport des couleurs générées par Col-Cycle et SpyncoGan à la classification des images aériennes historiques de HistAerial. 112

5.1 Résultats obtenus avec un post-traitement par vote majoritaire par superpixel sur d_s. 130
5.2 Taux de bonne classification (%) obtenus en intégrant les DES-*mean* au sein d'un CRF dense et en faisant varier $w^{(2)}$ et $w^{(3)}$. 135
5.3 Taux de bonne classification (%) en utilisant les DES-*mean* obtenus à l'aide de DES de différents domaines couleur avec $w^{(1)} = 3$ and $w^{(2)} = w^{(3)} = 10$. 136