ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : **3298**

THESE

En vue de l'obtention du : ***DOCTORAT***

Structure de Recherche : **Laboratoire de Matière Condensée et Sciences Interdisciplinaires (LaMCScI)**
Discipline : **Informatique**
Spécialité : **Informatique**

Présentée et soutenue le **29/02/2020** ***par :***

Abdelali ZBAKH

Exploration et analyse des données massives : algorithmes et applications

JURY

Abdelillah BENYOUSSEF,	PES, Membre Résident de l'académie Hassan II des Sciences et Techniques – Rabat	Président / Rapporteur
Abdelouahid LYHYAOUI ,	PES, ENSA – Université Abdelmalek Essaadi- Tanger	Rapporteur / Examinateur
Mohamed LAZAAR ,	PH, ENSIAS –Université Mohammed V- Rabat	Rapporteur / Examinateur
Mohammed ALACHHAB ,	PH, ENSA – Université Abdelmalek Essaadi- Tétouan	Examinateur
Mourad ELYADARI ,	PH, EST-Université Moulay Ismaïl- Meknès	Co-Directeur de thèse
Abdellah EL KENZ ,	PES, Faculté des sciences –Université Mohammed V- Rabat	Directeur de thèse

Année Universitaire : 2019-2020

Remerciements

Gloire et louange à Dieu, le tout Puissant, de m'avoir donné courage et persévérance pour achever cette thèse.

Avant de commencer l'écriture de ces petits mots, j'avoue que c'est la partie de la thèse la plus difficile à écrire, vu que les personnes qui ont une influence sur ce travail sont nombreux.

Ce travail est effectué dans le laboratoire « Laboratoire de Matière Condensée et Sciences Interdisciplinaires - (LaMCScI) » sous la direction du Pr Abdellah EL KENZ et la co-direction du Pr Mourad EL YADARI.

Je souhaite remercier en premier lieu mon directeur de thèse, M. Abdellah EL KENZ, Professeur à la Faculté des sciences, Université Mohammed V-Rabat. Je lui suis également reconnaissant pour le temps conséquent qu'il m'a accordé, ses qualités pédagogiques et scientifiques, sa franchise et sa sympathie. J'ai beaucoup appris à ses côtés et je lui adresse ma gratitude pour tout cela.

Je remercie aussi mon co-directeur de thèse, M. Mourad EL YADARI, Professeur à EST, Université Moulay Ismaïl –Meknes pour la qualité d'encadrement, la disponibilité et le soutien.

Je remercie le Pr Abdelillah BENYOUSSEF, Membre Résident de l'académie Hassan II des Sciences et Techniques - Rabat , d'avoir accepté d'être président du jury et de rapporter cette thèse. La version finale de ce mémoire a bénéficié de ses remarques précieuses

Je remercie M. Abdelouahid LYHYAOUI, Professeur à l'ENSA – Université Abdelmalek Essaadi- Tanger, d'avoir accepté d'examiner et de rapporter cette thèse. Ses conseils de rédaction ont été très précieux.

Je remercie M. Mohamed LAZAAR, Professeur à ENSIAS –Université Mohammed V- Rabat, d'avoir accepté d'examiner et de rapporter cette thèse en passant beaucoup de temps à m'orienter avec de judicieuses remarques pour améliorer la qualité de ce rapport.

Je remercie M. Mohammed ALACHHAB, Professeur à ENSA – Université Abdelmalek Essaadi-Tétouan, d'avoir accepté d'examiner cette thèse.

Je tiens à remercier aussi l'équipe du labo composé de Mlle Zoubida Alaoui Mdaghri et Ayman Quodad pour l'extraordinaire environnement de travail et pour le partage du savoir et du savoir-faire qui on a fait durant la préparation de cette thèse.

Je ne peux pas terminer ces remerciements sans dire un grand merci à mes amis : Mohamed Naoum, Adnan Souri, Otman El Hichami et Mohammed Taj Bennani pour les bons moments qu'on a vécus ensemble durant la préparation de cette thèse et pour les milliers de discussions qu'on a eues pour réussir ce travail.

Résumé

Les données massives (Big Data), désignent les jeux de données qui ne peuvent pas être traités efficacement à l'aide des outils traditionnels existants. Les Big Data sont caractérisés par les 3Vs (**V**olume, **V**ariété et **V**élocité) : Le volume croissant de données, La vitesse de traitement doit être la plus rapide possible et les données sont de formats très variés et ne sont pas toujours structurées. Les Big Data apparaissent dans de nombreuses applications importantes, telles que la recherche sur Internet, les réseaux sociaux et la télédétection. Les Big Data attirent, de plus en plus, l'attention des chercheurs en matière d'exploration et d'analyse.

Dans cette thèse, nous avons proposé et développé un ensemble d'algorithmes et de modèles liés au domaine d'exploration et d'analyse de données massives. Initialement, nous avons commencé par le choix des jeux de données sur lesquels, on va appliquer nos contributions. Pour cela, nous avons choisi les Images Hyperspectrales (HSI) comme premier jeu de données et un corpus arabe comme deuxième jeu de données.

Sur les HSI, on a proposé une version distribuée parallèle de l'algorithme de réduction de dimension ACP. L'algorithme est implémenté dans un environnement distribué parallèle nommé Apache Spark. En utilisant la méthode de transformation et en se basant sur l'ACP distribué parallèle, nous avons proposé, dans la deuxième contribution, un algorithme de visualisation des HSI dans l'environnement Apache Spark. La troisième contribution concerne la proposition d'un modèle de classification spectrale des HSI en utilisant l'apprentissage en profondeur (Deep Learning) : Les réseaux de neurones convolutifs(CNN)

Sur le corpus arabe, nous avons proposé un modèle de prédiction des textes manquants dans des documents arabes, en utilisant les réseaux de neurones convolutifs.

Mots clés : Données massives ; Image Hyperspectrale ; TALN ; CNN ; Classification ; Visualisation ; ACP ; Spark ; Apprentissage en profondeur

Abstract

Big Data refers to datasets that cannot be processed efficiently using existing traditional tools. Big Data is characterized by the 3Vs (**V**olume, **V**ariety and **V**elocity) : Increasing volume of data, the processing speed must be as fast as possible and the data are very different formats and are not always structured. Big Data appears in many important applications, such as Internet search, social networking and remote sensing. Big Data is increasingly attracting the interest of researchers in data mining and analysis.

In this thesis, we proposed and developed a set of algorithms and models related to the field of big data mining and analysis.

Initially, we started by choosing the data sets on which we will apply our contributions. To do this, we chose Hyperspectral Images (HSI) as the first dataset and an Arabic corpus as the second dataset.

On the HSI, a parallel distributed version of the PCA dimension reduction algorithm has been proposed. The algorithm is implemented in a parallel distributed environment called Apache Spark. Using the transformation method and based on the parallel distributed PCA, we proposed, in the second contribution, an algorithm for visualizing HSI in the Apache Spark environment. The third contribution concerns the proposition of an HSI spectral classification model using Deep Learning : Convolutional Neural Networks (CNN)

On the Arabic corpus, we proposed a model for predicting missing texts in Arabic documents, using the convolutional neural networks.

Keywords : Big Data ; Hyperspectral image ; NLP ; CNN ; Classification ; Visualization ; PCA ; Spark ; Deepth learning

« *“Créer une intelligence artificielle serait le plus grand évènement de l’histoire humaine.*

Malheureusement, ce pourrait être le dernier,

à moins que nous découvrions comment éviter les risques.” »

***

« "*Success in creating AI would be the biggest event in human history. Unfortunately, it might also be the last, unless we learn how to avoid the risks.*" »

***

Stephen Hawking, Physicien théoricien et cosmologiste

إهداء

إلى والدي الكريمين، أطال الله عمرهما، اللذين لم يبخلا علي يوما بصالح الدعاء.

إلى زوجتي الغالية، شريكة حياتي، التي ساندتني في مسيرتي العلمية والبحثية، متحملة مسؤولياتي المنزلية بدلا عني وموفرة لي مناخا صحيا للعمل.

شكر خاص لأولادي الصغار هبة، مراد والرضيعة رقية، من اجل الوقت الذي عاشوه بدوني، وأنا منشغل عنهم بدراستي.

Liste des publications

— **Spectral Classification of a Set of Hyperspectral Images using the Convolutional Neural Network, in a Single Training**

Abdelali Zbakh, Zoubida Alaoui Mdaghri, Abdelillah Benyoussef, Abdellah El Kenz, Mourad El Yadari

International Journal of Advanced Computer Science and Applications

DOI : http ://doi.org/10.14569/IJACSA.2019.0100634

2019 - **indexé scopus**

— **Convolutional Neural Networks in Predicting Missing Text in Arabic**

Adnan Souri, Mohamed Alachhab, Badr Eddine Elmohajir and Abdelali Zbakh

International Journal of Advanced Computer Science and Applications

DOI : http ://doi.org/10.14569/IJACSA.2019.0100668

2019 - **indexé scopus**

— **Proposition of a Parallel and Distributed Algorithm for the Dimensionality Reduction with Apache Spark**

Abdelali Zbakh, Zoubida Alaoui Mdaghri, Abdelillah Benyoussef, Abdellah El Kenz, Mourad El Yadari

Lecture Notes in Networks and Systems,Innovations in Smart Cities and Applications. SCAMS 2017. vol 37. Springer, Cham

DOI : https ://doi.org/10.1007/978-3-319-74500-8_46

2018 - **indexé scopus**

— **Visualization of Hyperspectral Images on Parallel and Distributed Platform : Apache Spark**

Abdelali Zbakh, Zoubida Alaoui Mdaghri, Abdelillah Benyoussef, Abdellah El Kenz, Mourad El Yadari

Date d'envoi :30/12/2017 (**En cours de révision**)

International Journal of Intelligent Enterprise : Inderscience Publishers

indexé scopus

— **A new Leach protocol based on ICH-Leach for adaptive image transferring using DWT**

Abdelali Zbakh, Mohamed Taj Bennani

TELKOMNIKA, Vol.17, No.5, October 2019, pp.2418 2426
DOI : http ://dx.doi.org/10.12928/telkomnika.v17i5.12446
2019 - **indexé scopus**

— **Leach routing protocol for image transfer using Castalia simulator**
M Taj Bennani, A Zbakh
https ://hal.archives-ouvertes.fr/hal-02174671
2019- *TELECOM 2019- 11èmes JFMMA, SAIDIA, MAROC*

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

CNN : Convolutional Neural Network
SVM : Support Vector Machine
HSI : hyperspectral image
USVC : Unsupervised classification
SVC : Supervised classification
RNN : Recurrent Neural Network
LSTM : Long Short Term Memory
OA : Overall Accuracy
PCA : Principal Component Analysis
HDFS : Hadoop Distributed File System
RDD : Resilient Distributed Datasets
AVIRIS : Airborne Visible Infra-Red Imaging Spectrometer
PCA : Principal Component Analysis
ACP : Analyse en composantes principales
NLP : Natural Language Processing
TALN : Traitement automatique du langage naturel
LLE : Locally Linear Embedding
sPCA : Scalable principal component analysis
HSL : Hue,Saturation,Lightness
RGB : Red, Green, Blue
CMF : Color Matching Functions
ICA : Independent Component Analysis
USVC : Unsupervised classification
SVC : Supervised classification
MLP : Multilayer Perceptron
HSI : Hyperspectral imagery
AA : Average Accuracy
OA : Overall Accuracy
GPU : Graphics Processing Unit
ICA : Independent Component Analysis
ROSIS : Reflective Optics System Imaging Spectrometer
CA : Class-specific Accuracy
EOSDIS : Earth Observing System Data and Information System
GEO : Group on Earth Observations
GBDX : Geospatial Big Data
VHR : Very High Resolution

EWT : leaf Equivalent Water Thickness
SST : Standford Sentiment Treebank
RNTN : Recursive Neural Tensor Network
MTD : Multilingual Twitter Dataset
NMT : Neural Machine Translation
OD : Opinosis Dataset
MT : Machine Translation
RBMT : Rule Based Machine Translation
SMT : Statistical Machine Translation
XBRL : Extensible Business Reporting Language
IT : information technology
CoS : Classify or send for classifcation
GA : Genetic Algorithm
MSF : Multiple Species Flocking
IF : Infrared Radiation
NoSQL : Not only SQL

Chapitre 1

Introduction générale

Sommaire

1.1 Contexte et Problématique

Au cours de ces dernières décennies, la génération des grandes quantités de données massives à partir des sources de données différentes a connu une grande augmentation. La taille des données générées par jour sur Internet a déjà dépassé deux exaoctets. En une minute, 72 heures de vidéos sont téléchargées sur Youtube, environ 30 000 nouveaux messages sont créés sur la plateforme de blogs Tumblr, soit plus de 100 000 Tweets sont partagés sur Twitter et plus de 200.000 photos sont affichées sur Facebook [1]. Les problèmes liés au Big Data soulèvent plusieurs questions de recherche, telles que : comment concevoir des environnements évolutifs ? comment assurer la tolérance aux pannes ? et comment concevoir des solutions efficaces ? La plupart des outils existants pour le stockage, le traitement et l'analyse des données sont inadéquats pour des volumes massifs de données hétérogènes. Par conséquent, il existe un besoin urgent de solutions Big Data plus avancées et plus appropriées.

De nombreuses définitions des Big Data ont été proposées dans la documentation. La plupart d'entre elles ont convenu que les problèmes de données massives partagent quatre caractéristiques principales, appelées les quatre V (volume, variété, vélocité et véracité). Le volume fait référence à la taille des jeux de données disponibles qui nécessitent généralement un stockage et un traitement distribués. La variété fait référence au fait que le Big Data est composé de plusieurs types de données tels que le texte, le son, l'image et la vidéo. La vélocité fait référence à la vitesse d'augmentation du volume de données volumineuses et aide les entreprises à comprendre la croissance relative de leurs données massives et la rapidité avec laquelle ces données parviennent aux utilisateurs, aux applications et aux systèmes. La véracité se rapporte aux biais, au bruit et à l'anomalie dans les données.

Les capteurs hyperspectraux (souvent appelés spectromètres d'imagerie) sont des instruments qui acquièrent des images dans de nombreuses bandes spectrales contiguës très étroites dans toutes les parties du spectre : visibles, proches de l'IR, de l'IR moyen et de l'IR thermique. Ces systèmes collectent généralement 200 bandes de données ou plus, ce qui permet la construction d'un spectre de réflectance effectivement contigu de chaque pixel de la scène. Ces caractéristiques permettent de faire la distinction entre les caractéristiques de la surface terrestre qui ont des propriétés d'absorption et de réflexion diagnostiques sur des intervalles de longueur d'onde étroits. Les données hyperspectraux collectées par les capteurs peuvent être considérées comme un cube, ayant deux dimensions l'une représentant la position spatiale et une autre qui représente la longueur d'onde.

L'exploration, l'analyse et même l'affichage de données hyperspectrales sont des tâches compliquées en raison des difficultés à sélectionner les bandes appropriées et les plus informatives à traiter. En ce qui concerne la qualité et les capacités de production de données des capteurs hyperspectraux, la communauté des utilisateurs peut maintenant obtenir des informations extrêmement utiles à partir des gros volumes de données.

Bien que les données hyperspectrales puissent être interprétées avec une bonne qualité, les systèmes traditionnels actuels ne sont pas en mesure de répondre à la demande des utilisateurs opérationnels dans le traitement de la grande masse de données de télédétection. Ce problème est davantage lié au manque de techniques et d'algorithmes efficaces nécessaires pour interpréter les jeux de données hyperspectraux avec un niveau suffisant de détails et de précision. Ceux-ci mettent en évidence la nécessité de concevoir de nouveaux algorithmes et de nouvelles techniques pratiques, permettant ainsi l'analyse de haute qualité des images acquises par les spectromètres d'imagerie.

Dans cette thèse, nous nous sommes intéressés à proposer des solutions pour des problèmes liés au domaine de big data (volume, variété, vélocité).

La première problématique traitée dans cette thèse est l'analyse des données hyperspectrales de télédétection : réduction de dimensions, visualisation et classification.

La deuxième problématique traitée dans cette thèse est le traitement automatique des langues naturelles avec des réseaux de neurones convolutifs : prédiction d'un texte manquant dans des documents en arabe.

1.2 Contributions et originalités de notre thèse

Les principales contributions de cette thèse sont résumées à la figure 1.1, qui décrit les contributions sur le jeu de données :les Images Hyperspectrales (HSI) et sur le jeu de données : le corpus arabe.

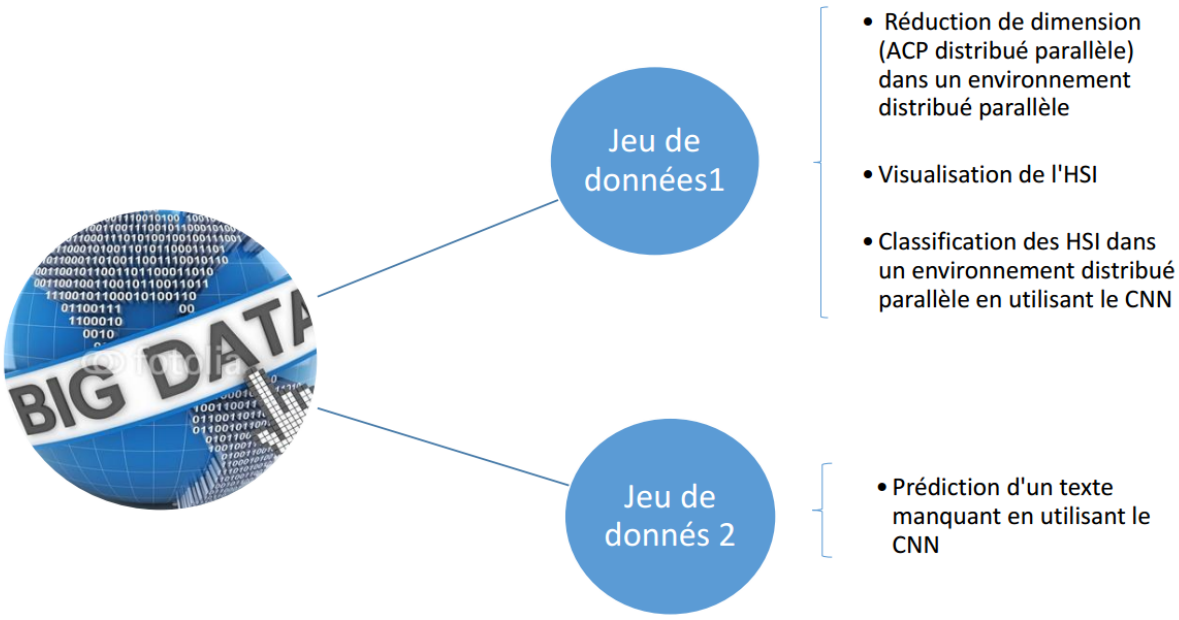


FIGURE 1.1 – Contributions de la thèse

Nous donnerons par la suite, un aperçu sur chaque contribution :

— **Contribution 1 : Réduction de dimension dans un environnement parallèle, distribué**

Le domaine de stockage et de traitement de données a connu, ces dernières années, une évolution radicale, en raison de la grande masse de données générées chaque minute. En conséquence, les outils et les algorithmes traditionnels sont devenus incapables de suivre cette évolution exponentielle et de produire des résultats dans un délai raisonnable. Parmi les solutions pouvant être adoptées pour résoudre ce problème, figure l'utilisation de stockage distribué de données et le traitement parallèle de données. Dans cette contribution, nous avons utilisé comme source de donnée massive,

les **images Hyperspectrales** et pour le traitement, nous avons utilisé la plateforme distribuée **Apache Spark**. En effet, le traitement d'une image hyperspectrale, tel que la visualisation et l'extraction de caractéristiques, doit prendre en compte la grande dimensionnalité de l'image. Dans la littérature, on trouve plusieurs techniques de réduction de dimensions (comme ACP). Dans cette contribution, nous avons proposé une version distribuée parallèle de l'algorithme de l'analyse en composantes principales (ACP).

— **Contribution 2 : Visualisation de données : Application aux images Hyperspectrales**

La visualisation des images Hyperspectrales représente un défi lorsque le nombre de bandes dépasse trois bandes, puisque la visualisation directe à l'aide du système trivial RGB ou HSL n'est pas possible. Parmi les solutions qui peuvent être adoptées pour résoudre ce problème est la réduction de la dimensionnalité de l'image à trois dimensions et par la suite l'attribution de chaque dimension à une couleur. Les outils et les algorithmes traditionnels sont devenus incapables de produire des résultats dans un délai raisonnable. Dans cette contribution, nous avons présenté une nouvelle méthode distribuée de visualisation de l'image hyperspectrale basée sur l'algorithme **ACP distribué parallèle** qu'on a développé dans la première contribution. La visualisation des grandes images hyperspectrales avec la méthode proposée, est faite dans un temps plus court et avec les mêmes performances que la méthode classique de visualisation .

— **Contribution 3 : Classification par les réseaux de neurones convolutifs :Application aux images Hyperspectrales**

L'imagerie hyperspectrale a connu une grande évolution ces dernières années. Par conséquent, plusieurs domaines (médical, agricole, géosciences) ont besoin de faire la classification automatique de ces images hyperspectrales avec un taux élevé et dans un temps acceptable. L'état de l'art présente plusieurs algorithmes de classification basés sur les réseaux de neurones convolutifs (en anglais Convolutional Neural Network (CNN)) et chaque algorithme s'entraîne sur une partie de l'image puis effectue la prédiction sur le reste de l'image. Cette contribution propose un nouvel algorithme de classification spectrale, rapide et basé sur CNN. Il permet de construire une image composite à partir de multiples images hyperspectrales, puis entraine le modèle de classification une seule fois sur l'image composite. Après l'entraînement, le modèle peut prédire chaque image séparément. Pour tester la validité de l'algorithme proposé, deux images hyperspectrales libres sont prises. Le temps d'entraînement obtenu par le modèle proposé, sur l'image composite est meilleur que le temps obtenu sur les modèles existants dans l'état de l'art.

— **Contribution 4 : Prédiction par les CNNs : Application aux TALN**

La prédiction du texte manquant est l'une des principales préoccupations de la communauté d'apprentissage en profondeur de « Traitement automatique de la langue ». Cependant, la plupart des recherches liées à la prédiction de textes sont effectuées dans d'autres langues, mais pas en arabe. Notre contribution est la prédiction du texte manquant dans des documents textes, en appliquant les réseaux de neurones convolutionnels (CNN) sur les modèles de la langue arabe. Nous avons construit des modèles linguistiques basés sur CNN répondant à des paramètres spécifiques en relation avec la langue arabe. Nous avons préparé notre jeu de données à partir d'une grande quantité de documents textes téléchargés gratuitement à partir de "Arab World Books", "Hindawi foundation", et "Shamela datasets". Pour calculer la précision des prédictions, nous avons comparé des documents textes complets avec les mêmes documents mais avec le texte manquant. Nous avons réalisé les étapes d'entrainement, de validation

et de test à trois expériences différentes visant à augmenter la performance de la prédiction. Le modèle a été entrainé lors de la première expérience sur des documents du même auteur, puis dans la deuxième expérience, il a été entrainé sur des documents du même jeu de données. A la troisième expérience, le modèle a été entrainé sur tous les documents confondus. Les étapes d'entrainement, de validation et de test ont été répétées plusieurs fois en changeant à chaque fois l'auteur, le jeu de données et la combinaison auteur-jeu de données, respectivement. Nous avons également utilisé la technique d'élargissement des données d'entrainement en alimentant le modèle CNN à chaque fois par une grande quantité de texte. Le modèle a donné une haute performance de la prédiction de texte arabe en utilisant des réseaux neuronaux convolutionnels avec une précision de 97,8% dans le meilleur des cas.

1.3 Organisation de la thèse

Le reste de la thèse est divisé en deux parties :

La première partie est consacrée à l'état de l'art, on y trouve :

— **Chapitre 2 : Données massives**
On va commencer, dans ce chapitre, par définir la notion de données massives (big data) comme étant un ensemble de V (Le Volume, La Vélocité, La Variété, La Véracité, La Valeur, ...). , ensuite on va présenter l'analyse de données massives : cycle de vie de l'analyse, les plates-formes utilisées et les algorithmes d'analyse. On terminera ce chapitre par les défis de big data.

— **Chapitre 3 : Les images hyperspectrales**
Dans ce chapitre, on va présenter les données massives de télédétection et spécialement les images hyperspectrales. On va donner pour ces images, la représentation en cube, les domaines d'application, le traitement (Visualisation, Classification) et les défis.

— **Chapitre 4 : Traitement Automatique des Langues Naturelles (TALN)**
Dans ce chapitre on va présenter une introduction au domaine de TALN et ses applications, puis on va traiter le problème du TALN pour la langue arabe.

La deuxième partie est consacrée aux contributions scientifiques, parmi lequelles on trouve :

— **Chapitre 5 : Réduction de dimension dans un environnement parallèle, distribué**
La contribution qui sera présentée dans ce chapitre propose une méthode distibuée de réduction de dimension de l'image hyperspectrale.

— **Chapitre 6 : Visualisation de données : Application aux images Hyperspectrales**
La contribution qui sera présentée dans ce chapitre traite la visualisation des images hyperspectrales

— **Chapitre 7 : Classification par les réseaux de neurones convolutifs :Application aux images Hyperspectrales**
La contribution qui sera présentée dans ce chapitre propose un modèle de classification spectrale en se basant sur les réseaux de neurones convolutifs pour les images hyperspectrales.

— **Chapitre 8 : Prédiction par les CNNs :Application aux TALN**
La contribution qui sera présentée dans ce chapitre propose un modèle de prédiction des textes manquants dans les documents en arabe.

On terminera la thèse par une conclusion et quelques pistes à suivre pour les chercheurs.

Première partie

État de l'art

Chapitre 2

Les données massives :BIG DATA

Sommaire

2.1 Introduction

Nous assistons à un tsunami de données de différents types et de différents formats. La gestion, le traitement, le stockage et le transport de ces données devient un véritable défi. Les données massives (en anglais Big Data) font référence aux stratégies non conventionnelles et aux technologies novatrices utilisées par les entreprises et les organisations pour capturer, gérer, traiter et donner du sens à un grand volume de données [2]. Twitter, Facebook, Amazon, Verizon, Macy's, et Whole Foods sont toutes des entreprises qui gèrent leurs activités en utilisant l'analyse des données massives.

Chaque jour, nous créons plus de deux quintillions d'octets de données (2EB) (voir la fig 2.1) et les chercheurs estiment que plus de 90% des données ont été générées au cours des dernières années [3] :

Décimal			Binaire		
Valeur	Préfixe SI		Valeur	Préfixe IEC	
1000	k	kilo	1024	Ki	kibi
1000^2	M	mega	1024^2	Mi	mebi
1000^3	G	giga	1024^3	Gi	gibi
1000^4	T	téra	1024^4	Ti	tebi
1000^5	P	péta	1024^5	Pi	pebi
1000^6	E	exa	1024^6	Ei	exbi
1000^7	Z	zetta	1024^7	Zi	zebi
1000^8	Y	yotta	1024^8	Yi	yobi

FIGURE 2.1 – Les unités de mesure de données

La nécessité de comprendre et de donner un sens à ces grandes quantités de données a donné naissance à la notion de Big Data.

2.2 Dimensions du Big Data

En 2001, Doug Laney [4], un analyste au sein de la société de conseil Meta Group Inc (rachetée par Gartner), a introduit l'idée des trois Vs (Volume, Variété et Vélocité). Les 3V ont été utilisés en tant que cadre commun pour décrire les caractéristiques de Big Data [5, 6].

Dans la suite de ce chapitre, nous allons décrire les 3V et d'autres dimensions supplémentaires de Big Data proposées dans le secteur Informatique.

Volume :

Le volume (voir la Fig 2.2) fait référence à la quantité de données qu'une organisation ou un individu collecte et/ou génère. En 2016, chaque jour, on estimait que 5,5 millions de nouveaux appareils étaient connectés pour collecter, analyser et partager des données. Dans [7],

l'auteur a prévu que 20,8 milliards d'appareils connectés seraient utilisés d'ici 2020. Donc La taille minimale pour être qualifiée de données volumineuses dépend du développement technologique.

Variété :

La variété (voir la Fig 2.2) fait référence au nombre de types de données. Les progrès technologiques permettent aux organisations de générer divers types de données : structurées, semi-structurées et non structurées. Le texte, la photo, l'audio, la vidéo, les données de flux de clics et les données des capteurs sont des exemples de données non structurées, qui n'ont pas la structure normalisée requise pour un calcul efficace. Les données semi-structurées ne sont pas conformes aux spécifications de la base de données relationnelle, mais peuvent être spécifiées pour répondre à certains besoins structurels des applications.

Vélocité :

La vélocité (voir la Fig 2.2) fait référence à la vitesse à laquelle les données sont générées et traitées. La vélocité des données augmente avec le temps. Initialement, les entreprises ont analysé les données à l'aide des systèmes de traitement par lots en raison de la lenteur et du coût élevé du traitement des données.

véracité :

IBM [1] a ajouté la **véracité** en tant que quatrième dimension (voir la Fig 2.2). La véracité représente le manque de fiabilité et l'incertitude dans les sources de données. L'incertitude et le manque de fiabilité résultent de l'incomplétude, de l'inexactitude, de la latence, de l'incohérence, de la subjectivité et de la tromperie dans les données. Les gestionnaires ne font pas confiance aux données lorsque des problèmes de véracité sont répandus. Des outils et des techniques statistiques ont été développés pour traiter l'incertitude et le manque de fiabilité des données massives avec des niveaux ou des intervalles de confiance précis [8, 9].

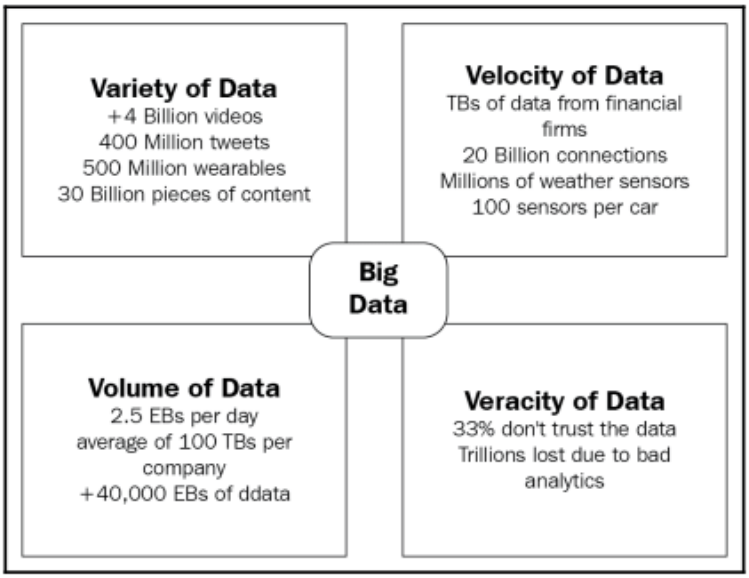


FIGURE 2.2 – Les quatre V de Big Data

1. **https://www.ibm.com/fr-fr/it-infrastructure/solutions/big-data**.

variabilité & complexité :

SAS[2] a ajouté deux dimensions supplémentaires au Big Data : la **variabilité** et la **complexité**. La variabilité fait référence à la variation des débits de données. En plus de la vitesse croissante et de la variété des données, les flux de données peuvent fluctuer avec des pics et des creux imprévisibles. Les données de pointe déclenchées par des événements imprévisibles sont difficiles à gérer avec des ressources informatiques limitées. D'autre part, l'investissement dans les ressources pour répondre à la demande informatique de pointe sera coûteux en raison de la sous-utilisation globale des ressources. La complexité fait référence au nombre de sources de données. Les données massives sont collectées à partir de nombreuses sources de données. La complexité rend difficile la collecte, le nettoyage, le stockage et le traitement de données hétérogènes. Il est nécessaire de réduire la complexité avec les sources ouvertes, les plates formes standard et le traitement en temps réel des données en continu.

valeur :

Oracle[3] a introduit la **valeur** en tant que dimension supplémentaire du Big Data. Les entreprises doivent comprendre l'importance de l'utilisation du Big Data pour augmenter les revenus, réduire les coûts opérationnels et mieux servir les clients, tout en tenant compte du coût d'investissement d'un projet Big Data. Les données seraient de faible valeur dans leur forme originale, mais l'analyse des données transformera les données en un actif stratégique de grande valeur. Les professionnels de l'IT doivent évaluer les avantages et les coûts de la collecte et/ou de la génération de données volumineuses, choisir des sources de données à haute valeur ajoutée et créer des outils analytiques capables de fournir des informations à valeur ajoutée aux gestionnaires.

2.3 Analyse des big data

L'analyse des données massives diffère de l'analyse traditionnelle des données, principalement en raison des caractéristiques des données massives : le volume, la vélocité et la variété . Pour répondre aux exigences distinctes en matière d'analyse des données massives, une méthodologie, étape par étape, est nécessaire pour organiser les activités et les tâches liées à l'acquisition, au traitement, à l'analyse et à la réutilisation des données.

Dans la section suivante, on va présenter le cycle de vie suivi pour l'analyse des données volumineuses.

2.3.1 Cycle de vie d'analyse de données volumineuses

Le cycle de vie de l'analyse des données massives peut être divisé en neuf étapes, comme le montre la figure 2.3 :

— Évaluation de l'étude de cas
— Identification des données
— Acquisition et filtrage des données
— Extraction des données

2. **https://www.sas.com/fr_ma/insights/big-data/what-is-big-data.html**.
3. **https://www.oracle.com/fr/big-data/**.

— Validation et nettoyage des données
— Agrégation et représentation des données
— Analyse des données
— Visualisation des données
— Utilisation des résultats de l'analyse

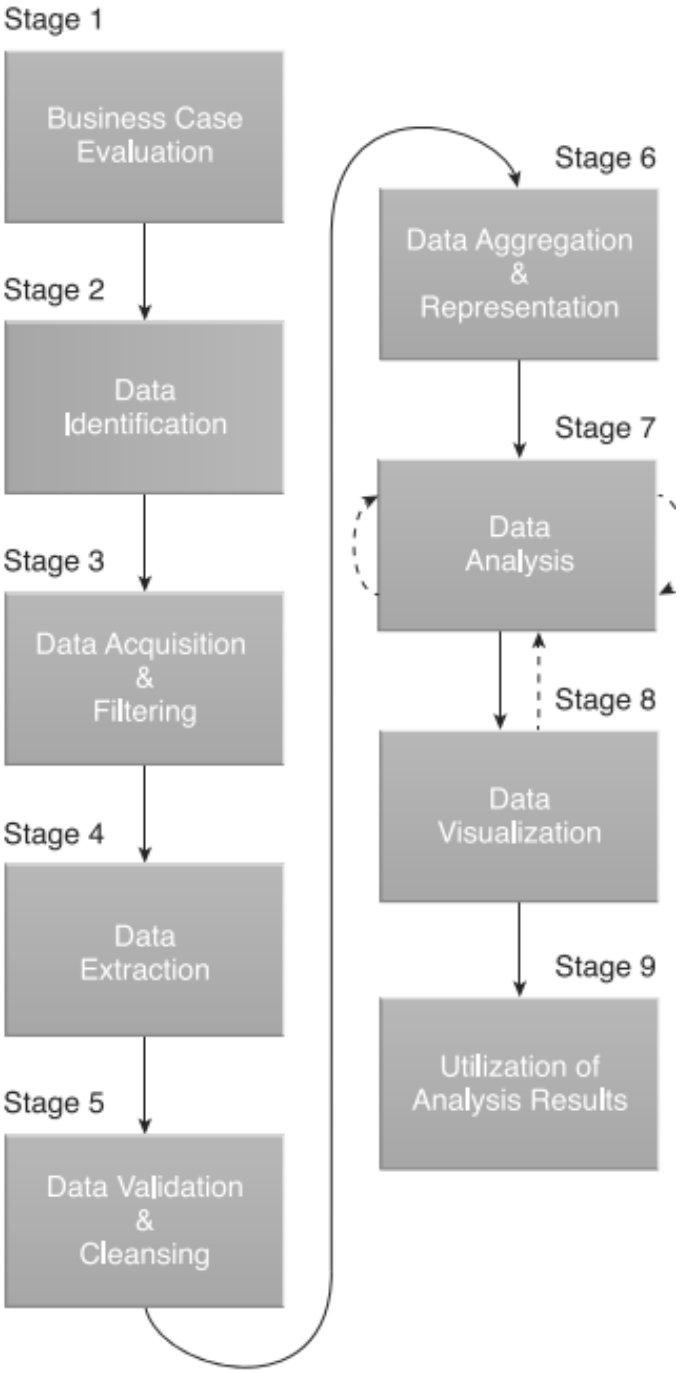


FIGURE 2.3 – Les neuf étapes du cycle de vie de l'analyse Big Data

1. Évaluation de l'étude de cas

Chaque cycle de vie de l'analyse de Big Data doit commencer par une étude de cas bien définie qui présente une compréhension claire de la justification, de la motivation et des objectifs de la réalisation de l'analyse. En fonction des exigences métiers documentées dans l'étude de cas, il est possible de déterminer si les problèmes métier à résoudre sont réellement des problèmes de Big Data. Pour être qualifié de problème Big Data, un problème métier doit être directement lié à une ou à plusieurs des caractéristiques Big Data telles que le volume, la vélocité ou la variété. À noter également qu'un autre résultat de cette étape est la détermination du budget nécessaire pour mener à bien le projet d'analyse. Tout achat requis, tel que des outils, du matériel et l'entrainement, doit être compris à l'avance, de sorte que l'investissement prévu puisse être mis en balance avec les avantages escomptés de la réalisation des objectifs.

2. Identification des données

La phase d'identification des données permet d'identifier les jeux de données nécessaires au projet d'analyse et leurs sources. Identifier une grande variété de sources de données peut augmenter la probabilité de trouver des modèles et des corrélations cachés, en particulier si ce qu'il faut rechercher exactement n'est pas clair. Selon le champ d'activité du projet d'analyse et la nature des problèmes métiers traités, les jeux de données requis et leurs sources peuvent être internes et/ou externes à l'entreprise.

3. Acquisition et filtrage des données

Lors de l'étape d'acquisition et de filtrage des données, ces dernières sont collectées à partir de toutes les sources de données identifiées lors de l'étape précédente. Les données acquises sont ensuite soumises à un filtrage automatisé en vue de la suppression des données corrompues ou considérées comme n'ayant aucune valeur pour les objectifs de l'analyse. Selon le type de source de données, les données peuvent provenir d'une collection de fichiers, tels que les données achetées d'un fournisseur de données tiers, ou peuvent nécessiter l'intégration d'API, comme avec Twitter. Dans de nombreux cas, en particulier lorsqu'il s'agit de données externes non structurées, certaines ou la plupart des données acquises peuvent ne pas être pertinentes (bruit) et peuvent être éliminées dans le cadre du processus de filtrage. Comme le montre la figure 2.4, des métadonnées peuvent être ajoutées via une automatisation aux données provenant de sources de données internes et externes afin d'améliorer la classification et l'interrogation. Comme exemples de métadonnées ajoutées, on trouve la taille et la structure du jeu de données, les informations source, la date et l'heure de la création ou de la collecte et des informations spécifiques à la langue. Il est essentiel que les métadonnées soient lisibles par machine et transmises au cours des étapes d'analyse ultérieures. Cela permet de préserver la provenance des données tout au long du cycle de vie de l'analyse Big Data, ce qui permet d'établir et de préserver la précision et la qualité des données.

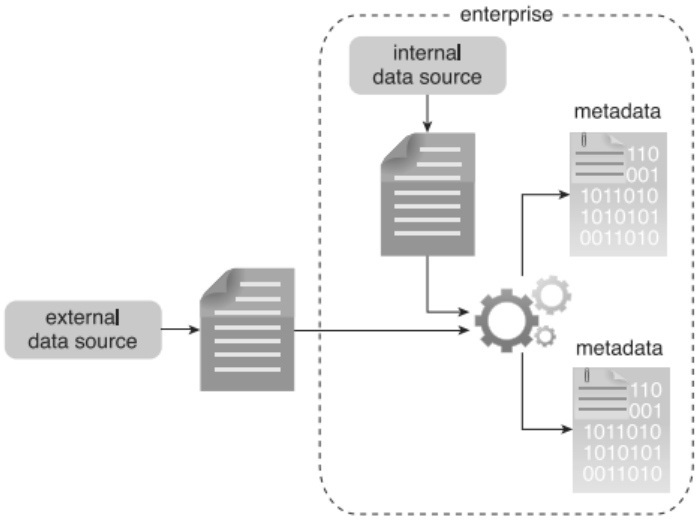


FIGURE 2.4 – Des métadonnées sont ajoutées aux données de sources internes et externes

4. Extraction des données

Certaines des données identifiées comme entrées pour l'analyse peuvent arriver dans un format incompatible avec la solution Big Data. La nécessité de traiter des types de données disparates est plus probable avec des données provenant de sources externes. L'étape de l'extraction des données, consiste à extraire des données disparates et à les transformer en un seul format que la solution Big Data sous-jacente peut utiliser pour l'analyse des données. L'étendue de l'extraction et de la transformation requises dépend des types d'analyse et des capacités de la solution Big Data. Par exemple, l'extraction des champs requis à partir de données textuelles délimitées, comme avec les fichiers journaux du serveur Web, peut ne pas être nécessaire si la solution Big Data sous-jacente peut déjà traiter directement ces fichiers. La Figure 2.5 illustre l'extraction de commentaires et d'un ID utilisateur intégré à un document XML sans qu'il soit nécessaire de procéder à une transformation plus poussée.

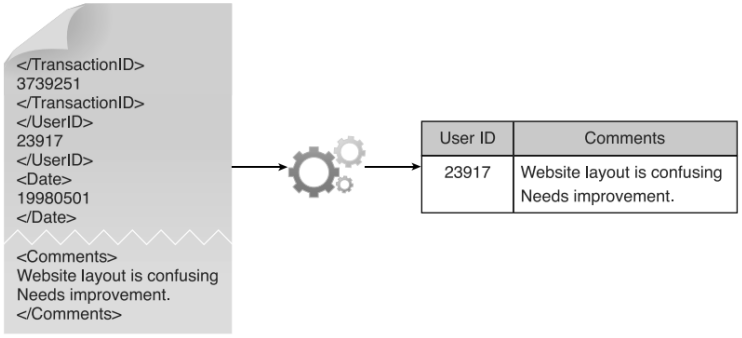


FIGURE 2.5 – Les commentaires et les ID utilisateurs sont extraits d'un document XML

La figure 2.6 illustre l'extraction des coordonnées de latitude et de longitude d'un utilisateur à partir d'un seul champ JSON. Une transformation supplémentaire est nécessaire pour séparer les données en deux champs distincts, comme requis par la solution Big Data.

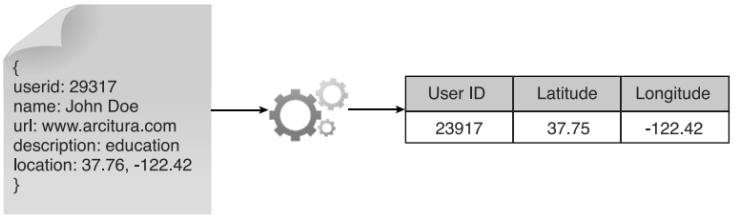


FIGURE 2.6 – L'ID utilisateur et les coordonnées d'un utilisateur sont extraits d'un seul champ JSON.

5. Validation et nettoyage des données

Des données invalides peuvent fausser et falsifier les résultats de l'analyse. Contrairement aux données traditionnelles d'une entreprise , où la structure de données est prédéfinie et les données prévalidées, les données entrées dans les analyses Big Data peuvent être non structurées sans aucune indication de validité. En raison de sa complexité, il peut être difficile d'en arriver à un ensemble de contraintes de validation appropriées. La phase de validation et de nettoyage des données, permet d'établir des règles de validation souvent complexes et de supprimer les données invalides connues. Les solutions Big Data reçoivent souvent des données redondantes sur différents jeux de données. Cette redondance peut être exploitée pour explorer des jeux de données interconnectés afin d'assembler des paramètres de validation et de remplir des données valides manquantes. Par exemple, comme illustré à la figure 2.7 :

— La première valeur du jeu de données B est validée par rapport à la valeur correspondante du jeu de données A.

— La deuxième valeur du jeu de données B n'est pas validée par rapport à la valeur correspondante du jeu de données A.

— Si une valeur est manquante, elle est insérée à partir du jeu de données A.

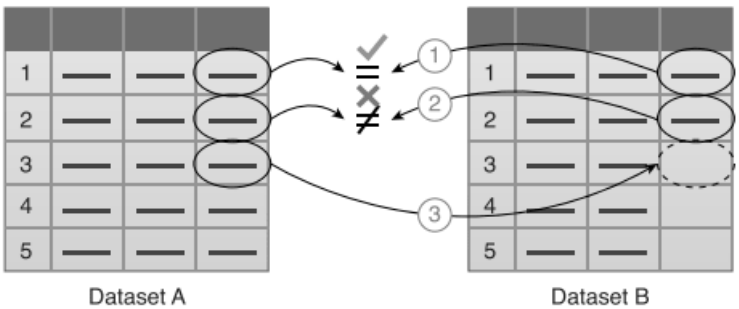


FIGURE 2.7 – La validation des données

6. Agrégation et représentation des données

Les données peuvent être réparties sur plusieurs jeux de données, ce qui exige que les ensembles de données soient regroupés au moyen de champs communs, par exemple la date ou l'ID. Dans d'autres cas, les mêmes champs de données peuvent apparaître dans plusieurs jeux de données, tels que la date de naissance. Dans les deux cas, une méthode de rapprochement des données est requise. L'étape d'agrégation et de représentation des données est dédiée à l'intégration de plusieurs jeux de données afin d'obtenir une vue unifiée. Effectuer cette étape peut devenir compliqué à cause des différences dans :

— Structure de données : bien que le format des données puisse être identique, le modèle de données peut être différent.
— Sémantique : une valeur étiquetée différemment dans deux jeux de données différents peut signifier la même chose, par exemple «code élève» et «num élève».

l'agrégation des grands volumes de données traités par les solutions Big Data est une opération qui demande beaucoup de temps et d'efforts. La réconciliation de ces différences peut nécessiter une logique complexe, une exécution automatique et sans intervention humaine. Les besoins futurs en matière d'analyse des données doivent être pris en compte à cette étape pour favoriser la réutilisation des données. Que l'agrégation des données soit nécessaire ou non, il est important de comprendre que les mêmes données peuvent être stockées sous de nombreuses et différentes formes. Une forme peut être mieux adaptée à un type particulier d'analyse qu'une autre. Une structure de données normalisée par la solution Big Data peut servir de dénominateur commun pouvant être utilisé pour toute une gamme de techniques d'analyse et de projets. Cela peut nécessiter la création d'un référentiel d'analyse central centralisé, comme une base de données NoSQL (voir la figure 2.8).

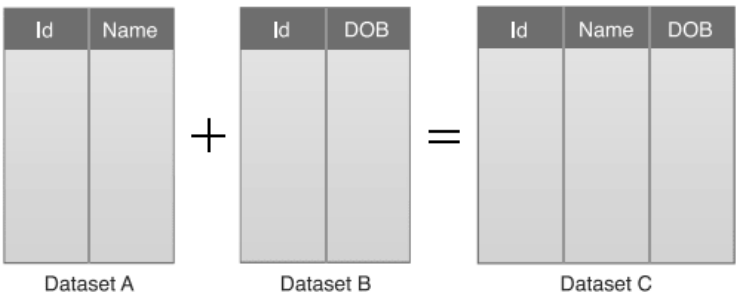


FIGURE 2.8 – Exemple simple d'agrégation de données

7. Analyse des données

L'étape de l'analyse des données, est consacrée à l'exécution de la tâche d'analyse proprement dite, qui comprend habituellement un ou plusieurs types d'analyse. Cette étape peut être itérative par nature, surtout si l'analyse des données est exploratoire, auquel cas l'analyse est répétée jusqu'à ce que le modèle approprié ou la corrélation soient découverts.

En fonction du type de résultat d'analyse requis, cette étape peut être aussi simple que d'interroger un jeu de données afin de calculer une agrégation à des fins de comparaison. D'autre part, il peut être aussi difficile comme l'exemple de combiner l'exploration de données et des techniques d'analyse statistique complexes pour découvrir des modèles et des

anomalies. L'analyse de données peut être classifiée en analyse de confirmation ou en analyse exploratoire (voir la figure 2.9).

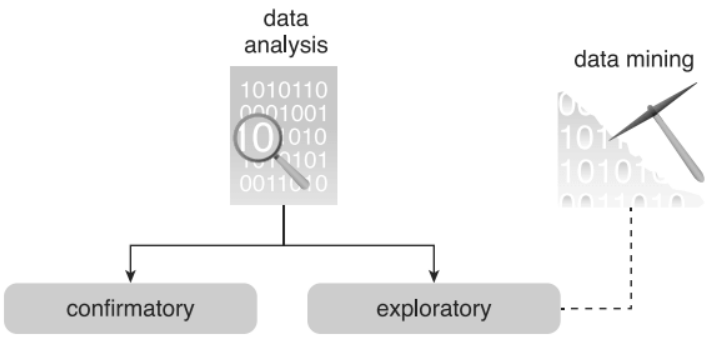


FIGURE 2.9 – L'analyse des données.

L'analyse des données de confirmation est une approche déductive où la cause du phénomène étudié est proposée au préalable. La cause ou l'hypothèse proposée est appelée hypothèse. Les données sont ensuite analysées afin de prouver ou de réfuter l'hypothèse et de fournir des réponses définitives à des questions spécifiques. Les techniques d'échantillonnage des données sont généralement utilisées. Les découvertes ou les anomalies inattendues sont généralement ignorées car une cause prédéterminée a été supposée.

L'analyse exploratoire des données est une approche inductive étroitement associée à l'exploration de données. Aucune hypothèse n'est générée. Les données sont plutôt analysées afin de mieux comprendre la cause du phénomène. Bien qu'elle ne fournisse peut-être pas de réponses définitives, cette méthode fournit une orientation générale pouvant faciliter la découverte de modèles ou d'anomalies.

8. Visualisation des données

La capacité d'analyser des quantités massives de données et de trouver des informations utiles a peu de valeur, si les seuls qui peuvent interpréter les résultats sont les analystes. L'étape de la visualisation des données est consacrée à l'utilisation de techniques et d'outils de visualisation des données afin de communiquer graphiquement les résultats de l'analyse pour une interprétation efficace par les utilisateurs professionnels. Les utilisateurs professionnels doivent être en mesure de comprendre les résultats afin d'obtenir une valeur ajoutée de l'analyse, puis de pouvoir fournir une réaction, comme indiqué par la ligne pointillée pointant de l'étape 8 à l'étape 7.

La visualisation des données donne aux utilisateurs la possibilité d'effectuer une analyse visuelle, permettant de découvrir des réponses à des questions qu'ils n'ont même pas encore formulées.

9. Utilisation des résultats de l'analyse

Selon la nature des problèmes d'analyse abordés, il est possible que les résultats de l'analyse produisent des « modèles » qui résument de nouvelles idées et de nouvelles interprétations sur la nature des modèles et des relations existantes dans les données analysées. Un modèle peut ressembler à une équation mathématique ou à un ensemble de règles.

2.3.2 Les plates-formes d'analyse des big data

Diverses solutions ont été présentées pour l'analyse des données volumineuses, qui peuvent être divisées en [10] :

— Traitement/Calcul : Hadoop [4], Nvidia CUDA [5] ou Twitter Storm [6]

— Stockage : Titan ou HDFS

— Analyse : MLPACK [11] ou Mahout [7].

Bien qu'il existe des produits commerciaux pour l'analyse des données [11, 12], la plupart des études sur l'analyse traditionnelle des données sont axées sur la conception et l'élaboration de « moyens» efficaces pour trouver les éléments utiles à partir des données. Mais lorsque nous entrons dans l'ère des big data, la plupart des systèmes informatiques actuels ne seront pas en mesure de gérer l'ensemble des données en une seule fois ; par conséquent, la façon de concevoir une bonne structure logicielle d'analyse des données et la façon de concevoir des méthodes d'analyse sont deux choses importantes pour le processus d'analyse des données.

Dans l'état de l'art, des enquêtes approfondies ont été menées pour discuter les infrastructures logicielles de données volumineuses [13, 14, 15]. Dans cette section, on va présenter certaines infrastructures logicielles populaires dans le domaine de données massives.

2.3.2.1 Apache Hadoop

Hadoop est un projet Apache fondé en 2008 par Doug Cutting chez Yahoo et Mike Cafarella à l'Université du Michigan. Hadoop est constitué de deux composants principaux :

— Hadoop Distributed File System (**HDFS**) : pour le stockage des données

— Hadoop **Mapreduce** : pour le traitement de données [16].

Par la suite, on va présenter le modèle de programmation **Mapreduce** et le système de fichiers **HDFS**

Modèle de programmation MapReduce :

MapReduce est un modèle de programmation conçu pour traiter en parallèle de grands jeux de données. MapReduce a été proposé par Google en 2004 [16] comme une abstraction permettant d'effectuer des calculs simples tout en masquant les détails de la parallélisation, du stockage distribué, de l'équilibrage de la charge et de la tolérance aux pannes.

Les caractéristiques centrales du modèle de programmation Mapreduce sont deux fonctions, écrites par l'utilisateur : **Map** et **Reduce**.

La fonction **Map** prend une seule paire clé-valeur comme entrée et produit une liste de paires clé-valeur intermédiaires. Les valeurs intermédiaires associées à la même clé intermédiaire sont regroupées et transmises à la fonction Reduce.

4. **http://hadoop.apache.org/**.
5. **https://developer.nvidia.com/**.
6. **http://storm.apache.org/**.
7. **http://mahout.apache.org/**.

La fonction **Reduce** prend comme entrée une clé intermédiaire et un ensemble de valeurs pour cette clé. Elle fusionne ces valeurs pour former un ensemble plus petit de valeurs. L'aperçu du système de Mapreduce est illustré par la Figure 2.10

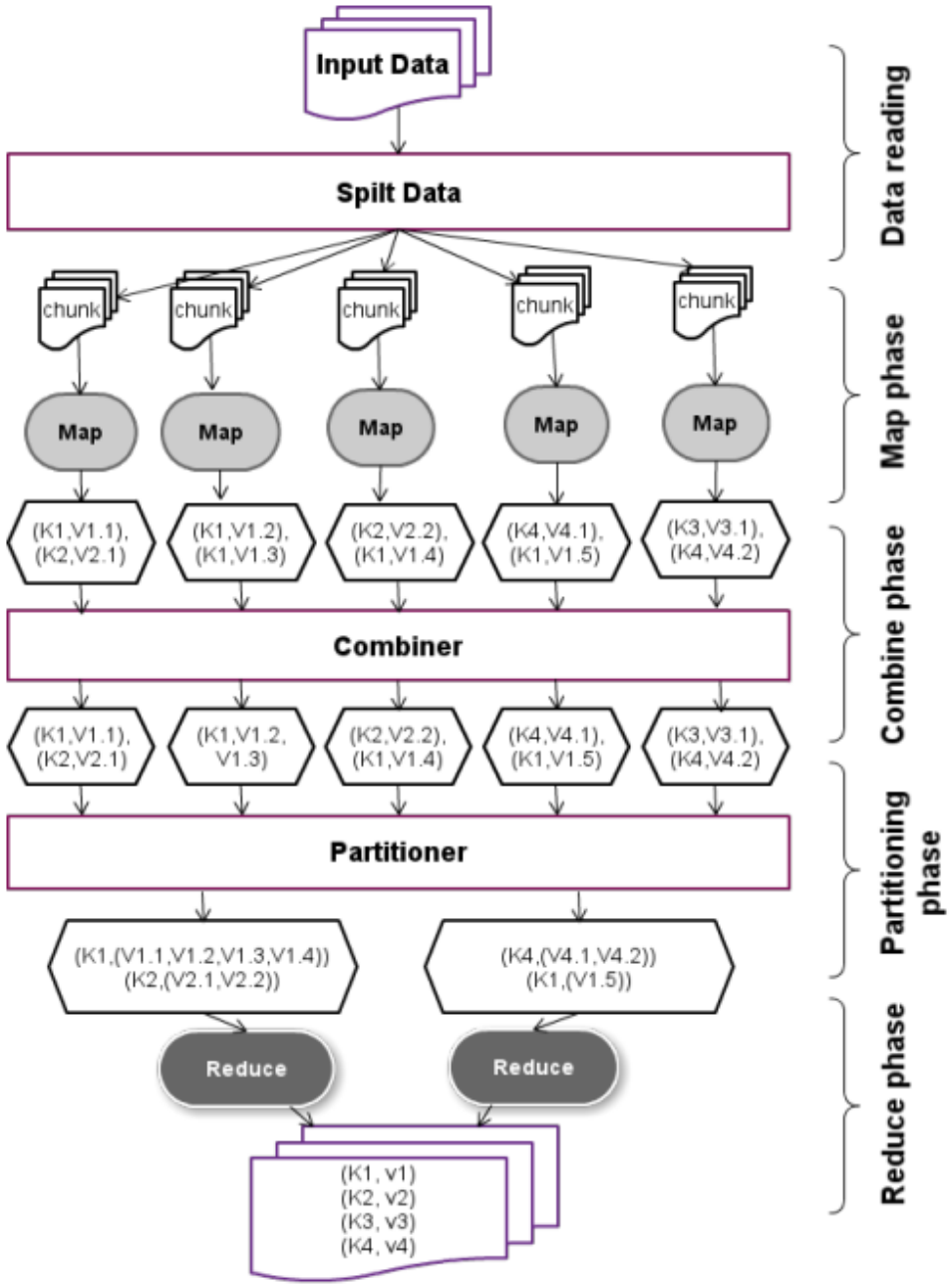


FIGURE 2.10 – L'architecture de MapReduce

Les étapes de base d'un programme MapReduce sont les suivantes :

— Lecture des données : dans cette phase, les données d'entrée sont transformées en un ensemble de paires clé-valeur. Les données d'entrée peuvent provenir de diverses

sources telles que des systèmes de fichiers, des systèmes de gestion de bases de données (SGBD) ou la mémoire principale (RAM). Les données d'entrée sont divisées en plusieurs morceaux de taille fixe. Chaque bloc est traité par une instance de la fonction Map.

— Phase de Map : pour chaque bloc ayant la structure clé-valeur, la fonction de Map correspondante est déclenchée et produit un ensemble de paires clé-valeur intermédiaires.

— Phase de combinaison : cette étape a pour but de regrouper toutes les paires clé-valeur intermédiaires associées à la même clé intermédiaire.

— Phase de partitionnement : après leur combinaison, les résultats sont répartis entre les différentes fonctions de Reduce.

— Phase de Reduce : la fonction Reduce fusionne les paires de valeurs clés ayant la même clé et calcule un résultat final.

HDFS :

HDFS est une implémentation open source du système de fichiers distribué de Google (GFS) [17]. Il fournit un système de fichiers distribués évolutifs pour stocker de grands fichiers sur des machines distribuées de manière fiable et efficace [18]. Dans la Figure 2.11, nous montrons l'architecture abstraite de l'HDFS et ses composants.

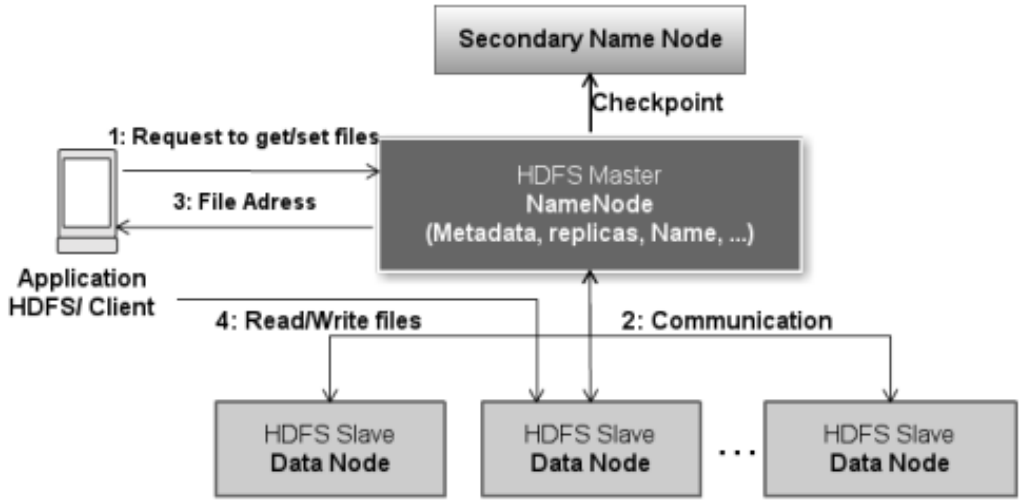


FIGURE 2.11 – Architecture de HDFS

HDFS se compose d'une architecture maître/esclave avec un maître nommé "**Name Node**" et plusieurs esclaves nommés "**Data Nodes**" . Le Name Node est responsable de l'attribution de l'espace physique pour stocker les fichiers volumineux envoyés par le client HDFS. Si le client souhaite récupérer des données à partir du HDFS, il envoie une demande au Name Node. Le Name Node cherchera leur emplacement dans son système d'indexation et renverra ensuite leur adresse au client. Le Name Node renvoie au client HDFS les métadonnées (nom de fichier, emplacement du fichier, etc.) associées aux fichiers stockés. Un Name Node secondaire enregistre périodiquement l'état du Name Node. Si le Name Node échoue, le Name Node secondaire prend automatiquement le relais.

2.3.2.2 Apache Spark

Apache Spark est une puissante infrastructure logicielle de traitement offrant un outil simple d'utilisation pour l'analyse efficace de données hétérogènes. Il a été développé à l'origine à l'UC Berkeley en 2009 [19]. Spark présente plusieurs avantages par rapport aux autres infrastructures Big Data telles que Hadoop et Storm. Spark est utilisé par de nombreuses sociétés telles que Yahoo, Baidu et Tencent. Un concept clé de Spark est constitué par les jeux de données distribués résilients (Resilient Distributed Datasets : RDD). Un RDD est essentiellement une collection immuable d'objets répartis sur un cluster Spark. Dans Spark, il existe deux types d'opérations sur les RDD : transformations et actions.

— Les transformations consistent à créer de nouveaux RDD à partir des RDD existants en utilisant des fonctions comme map, filter, union et join.

— Les actions se composent du résultat final des calculs RDD.

Dans la Fig. 2.12, nous présentons un aperçu de l'architecture Spark.

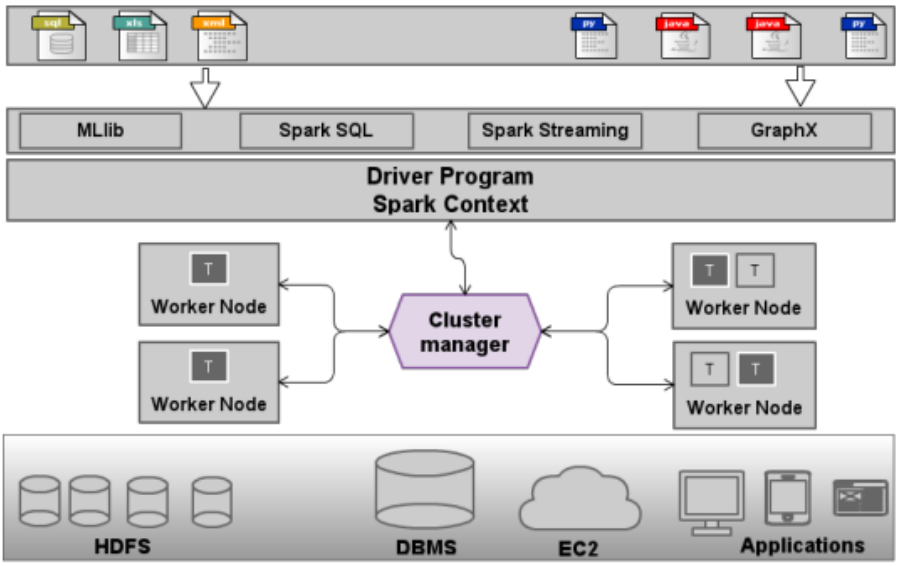


FIGURE 2.12 – Aperçu du système Spark

Un cluster Spark est basé sur une architecture maître/esclave avec trois composantes principales :

— **Driver Program :**
ce composant représente le nœud esclave dans un cluster Spark. Il gère un objet appelé SparkContext qui gère et supervise les applications en cours d'exécution.

— **Cluster Manager :**
Ce composant est responsable de l'orchestration du flux de travail de l'application attribuée par Driver Program aux **workers**. Il contrôle et supervise également toutes les ressources du cluster et renvoie leur état au Driver Program.

— **Worker Nodes :**
chaque Worker Node représente un conteneur d'une seule opération lors de l'exécution d'un programme Spark.

En plus des opérations existantes dans Hadoop (MapReduce), Apache Spark offre la possibilité de travailler avec des requêtes SQL, le streaming, le traitement de graphes, la machine d'apprentissage (voir figure 2.13). Spark propose les interfaces de programmation d'applications suivantes (API) [19].

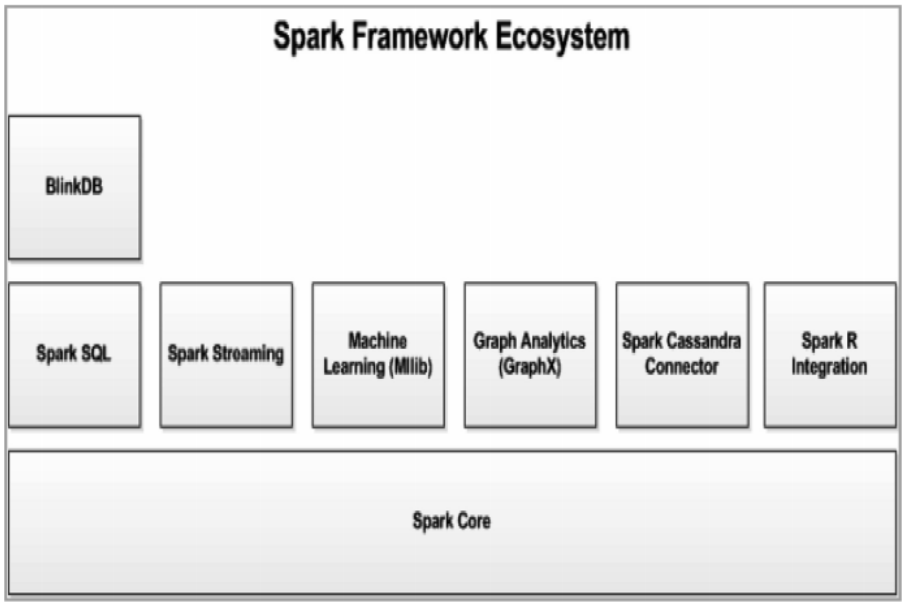


FIGURE 2.13 – Bibliothèques d'infrastructure Spark

— **SparkCore :**
Spark Core est le moteur d'exécution général de la plate-forme Spark. Toutes les autres fonctionnalités et extensions sont construites au dessus de SparkCore. Spark Core fournit des capacités de calcul en mémoire et un modèle d'exécution généralisé pour prendre en charge une grande variété d'applications, ainsi que des API Java, Scala et Python pour faciliter le développement.

— **SparkStreaming :**
Spark Streaming permet de réaliser de puissantes applications interactives et analytiques sur les données historiques et en streaming, tout en héritant de la facilité d'utilisation de Spark et de ses caractéristiques de tolérance aux pannes. Il peut être utilisé avec une grande variété de sources de données populaires, y compris HDFS, Flume [20], Kafka [21], et Twitter [19].

— **SparkSQL :**
Spark offre une gamme de fonctionnalités pour structurer les données extraites de plusieurs sources. Il permet ensuite de les manipuler en utilisant le langage SQL [22].

— **SparkMLLib :**
Spark fournit une bibliothèque d'apprentissage machine évolutive fournissant à la fois des algorithmes de haute qualité et une vitesse élevée (jusqu'à 100 fois plus rapide que MapReduce) [19].

— **GraphX :**
Graphx [23] est une API Spark pour le calcul parallèle de graphes (par exemple, algorithme PageRank et filtrage collaboratif). A un haut niveau, Graphx étend l'abstraction Spark RDD en introduisant le **Resilient Distributed Property Graph** : un multigraphe

orienté avec des propriétés attachées à chaque sommet et arréte. Pour le calcul des graphes, Graphx fournit un ensemble d'opérateurs fondamentaux (par exemple, subgraph, joinVertices, et MapReduceTriplets)) ainsi qu'une variante optimisée de l'API Pregel [24]. De plus, Graphx inclut une collection croissante d'algorithmes de graphes (par exemple, PageRank, composants connectés, propagation d'étiquettes et comptage de triangle) pour simplifier les tâches d'analyse des graphes.

Il existe d'autres infrastructures logicielles pour le traitement de Big data. La figure 2.14 présente pour chaque infrastructure logicielle, les caractéristiques suivantes :

— le modèle de programmation,
— les langages de programmation pris en charge,
— le type de sources de données
— la possibilité de permettre le traitement itératif des données
— la compatibilité avec les bibliothèques d'apprentissage automatique existantes
— la stratégie de tolérance aux pannes.

	Hadoop	**Spark**	**Storm**	**Flink**	**Samza**
Data format	Key-value	Key-value, RDD	Key-value	Key-value	Events
Processing mode	Batch	Batch and Stream	Stream	Batch and Stream	Stream
Data sources	HDFS	HDFS, DBMS and Kafka	HDFS, HBase and Kafka	Kafka, Kinesis, message queus, socket streams and files	Kafka
Programming model	Map and Reduce	Transformation and Action	Topology	Transformation	Map and Reduce
Supported programming language	Java	Java, Scala and Python	Java	Java	Java
Cluster manager	YARN	Standalone, YARN and Mesos	YARN or Zookeeper	Zookeeper	YARN
Comments	Stores large data in HDFS	Gives several APIs to develop interactive applications	Suitable for real-time applications	Flink is an extension of MapReduce with graph methods	Based on Hadoop and Kafka
Iterative computation	Yes (by running multiple MapReduce jobs)	Yes	Yes	Yes	YES
Interactive Mode	No	Yes	No	No	No
Machine learning compatibility	Mahout	SparkMLlib	Compatible with SAMOA API	FlinkML	Compatible with SAMOA API
Fault tolerance	Duplication feature	Recovery technique on the RDD objects	Checkpoints	Checkpoints	Data partitioning

FIGURE 2.14 – Une étude comparative des frameworks Big Data populaires

2.3.3 Algorithmes d'analyse

2.3.3.1 Algorithmes d'exploration de données

Fan et Bifet [25] ont souligné que les termes «données volumineuses» [26] et «exploration de données volumineuses» (en anglais : Big Data Mining) [27] avaient été présentés pour la

première fois en 1998. La recherche de quelque chose dans le Big Data sera l'une des tâches principales de ce domaine de recherche. Les algorithmes de Data Mining pour l'analyse des données jouent un rôle vital dans l'analyse des big data, en ce qui concerne le coût de calcul, les besoins en mémoire et la précision des résultats finaux.

2.3.3.2 Algorithmes de regroupement (clustering)

À l'ère des big data, les algorithmes de clustering traditionnels deviendront encore plus limités qu'auparavant. Ils exigent généralement que toutes les données soient dans le même format et soient chargées dans la même machine de façon à trouver des choses utiles à partir des données entières.

Bien que le problème d'analyse du jeu de données à grande échelle et de grande dimension [28] ait attiré de nombreux chercheurs de diverses disciplines au cours du siècle dernier et que plusieurs solutions [29, 30] aient été présentées ces dernières années, les caractéristiques du big data soulèvent encore plusieurs nouveaux défis pour les problèmes de clustering de données, la réduction de la complexité des données est l'un des problèmes majeurs du clustering Big Data.

Dans [31], les auteurs ont divisé le regroupement des big data en deux catégories : le regroupement sur une seule machine (solutions d'échantillonnage et de réduction des dimensions) et le regroupement sur plusieurs machines (solutions parallèles et Mapreduce). Cela signifie que les solutions traditionnelles de réduction peuvent également être utilisées à l'ère des big data, parce que la complexité et l'espace mémoire nécessaires au processus d'analyse des données seront réduits en utilisant des méthodes d'échantillonnage et de réduction des dimensions.

Plus précisément, l'échantillonnage peut être considéré comme une réduction de la « quantité de données » entrées dans un processus d'analyse des données, tandis que la réduction des dimensions peut être considérée comme une « réduction de la taille de jeu des données ». Les dimensions non pertinentes seront éliminées avant que le processus d'analyse des données ne soit effectué.

Cloudvista [32, 33] est une solution représentative pour le clustering des big data, qui utilise le cloud computing pour exécuter le processus de clustering en parallèle. L'utilisation de GPU pour améliorer les performances d'un algorithme de clustering est une autre solution prometteuse pour l'exploration de données volumineuses. Le Multiple Species Flocking (MSF) [34] a été appliqué à la plate-forme CUDA de NVIDIA pour réduire le temps de calcul de l'algorithme de clustering dans [35]. Les résultats de la simulation montrent que le facteur d'accélération peut être augmenté de 30 à 60 en utilisant le GPU pour le regroupement de données.

Étant donné que la plupart des algorithmes de regroupement traditionnels (p. ex., k-means) exigent un calcul centralisé, la principale préoccupation de Feldman et al. [36], est de savoir comment les rendre capables de traiter les problèmes de regroupement de big data.

2.3.3.3 Algorithmes de classification

Semblable à l'algorithme de regroupement pour l'exploration de big data, plusieurs études ont également tenté de modifier les algorithmes de classification traditionnels pour les faire fonctionner sur un environnement de calcul parallèle ou pour développer de nouveaux algorithmes de classification qui fonctionnent naturellement sur un environnement de calcul

parallèle. Dans [37], la conception de l'algorithme de classification a pris en compte les données d'entrée collectées par des sources de données distribuées et elles seront traitées par un ensemble hétérogène d'apprenants. Dans cette étude, Tekin et al. ont présenté un nouvel algorithme de classification appelé « classifier ou envoyer pour classification ». Ils ont supposé que chaque apprenant pouvait traiter les données d'entrée de deux manières différentes. Les informations seront échangées entre les différents apprenants. Ce type de solutions peut être considéré comme un apprentissage coopératif visant à améliorer la précision de la résolution du problème de la classification des données volumineuses. Une solution intéressante utilise l'informatique quantique pour réduire l'espace mémoire et le coût de calcul d'un algorithme de classification. Par exemple, dans [38], Rebentrost et al. ont présenté une version de SVM basée sur le quantique pour la classification des big data et ont soutenu que l'algorithme de classification qu'ils ont proposé peut être mis en œuvre avec une complexité temporelle O(log NM) où N est le nombre de dimensions et M est le nombre de données d'entrainement. Il y a de grandes perspectives pour l'exploration de big data en utilisant un algorithme de recherche quantique lorsque le matériel informatique quantique deviendra mature.

2.3.3.4 Algorithmes d'extraction de modèles fréquents

La plupart des recherches sur l'extraction de modèles fréquents (c'est-à-dire les règles d'association et l'extraction de modèles séquentielle) était axée dès le début sur la gestion d'un jeu de données à grande échelle, au tout début parce que certaines approches précoces ont été tentées pour analyser les données de transaction des grands centres commerciaux. Étant donné que le nombre de transactions dépasse généralement les « dizaines de milliers », les questions relatives à la façon de traiter les données à grande échelle ont été étudiées pendant plusieurs années, comme l'arbre FP [39] qui utilise la structure de l'arbre pour inclure les motifs fréquents afin de réduire davantage le temps de calcul de l'exploration des règles d'association. En plus des algorithmes traditionnels d'exploitation de modèles fréquents, bien entendu, les technologies d'informatique parallèle et le cloud computing ont également attiré des chercheurs dans ce domaine de recherche. Parmi eux, la solution de MapReduce a été utilisée pour les études [40, 41] afin d'améliorer les performances de l'algorithme d'extraction de modèle fréquent. En utilisant le modèle map-reduce pour l'algorithme d'exploration de motif fréquent, on peut facilement s'attendre à ce que son application à la « plateforme cloud » [42, 43] devienne une tendance populaire dans l'avenir. L'étude de [40] a non seulement utilisé le modèle de Map-Reduce, elle a également permis aux utilisateurs d'exprimer leurs contraintes d'intérêt spécifique dans le processus d'exploration de modèle fréquent. La performance de ces méthodes en utilisant le modèle map-reduce pour l'analyse des méga-données est, sans aucun doute, meilleure que les algorithmes traditionnels d'exploration de motifs fréquents fonctionnant sur une seule machine.

2.3.3.5 Apprentissage automatique pour l'exploration de big data

Le potentiel de l'apprentissage automatique pour l'analyse des données se trouve facilement dans les premières publications [44, 45]. Les algorithmes d'apprentissage machine peuvent être utilisés pour les différents problèmes d'exploration et d'analyse, car ils sont généralement utilisés comme algorithme de «recherche» de la solution requise. Comme la plupart des algorithmes d'apprentissage machine peuvent être utilisés pour trouver une solution approximative au problème d'optimisation, ils peuvent être utilisés pour la plupart des problèmes d'analyse de données, si les problèmes d'analyse de données peuvent être formulés en tant que problèmes d'optimisation. Par exemple, l'algorithme génétique (GA), l'un des algorithmes d'apprentissage machine, peut non seulement être utilisé pour résoudre

le problème de regroupement [46], mais aussi pour résoudre le problème de l'extraction de modèles fréquents [47].

Une étude récente [48] montre que certains algorithmes d'exploitation traditionnels, certaines méthodes statistiques, certaines solutions de prétraitement et même l'interface graphique ont été appliqués à plusieurs outils et plateformes représentatifs pour l'analyse des données volumineuses. Les résultats montrent clairement que les algorithmes d'apprentissage automatique seront l'une des parties essentielles de l'analyse des données volumineuses. L'un des problèmes liés à l'utilisation des méthodes actuelles d'apprentissage automatique pour l'analyse des données volumineuses est similaire à celui de la plupart des algorithmes d'exploration de données traditionnels conçus pour l'informatique séquentielle ou centralisée. Cependant, l'une des solutions les plus possibles est de les faire fonctionner pour le calcul parallèle.

Heureusement, certains des algorithmes d'apprentissage automatique (par exemple, des algorithmes basés sur une population) peuvent essentiellement être utilisés pour le calcul parallèle, ce qui est démontré depuis plusieurs années, comme la version de calcul parallèle de l'algorithme génétique (PGA) [49]. Comme le montre la figure 2.15(a), la population de l'algorithme génétique du modèle d'île, l'un des GA parallèles, peut être divisée en plusieurs sous-populations, comme le montre la figure 2.15(b). Cela signifie que les sous-populations peuvent être affectées à différents threads ou nœuds d'ordinateur pour un calcul parallèle, par une simple modification de l'AG.

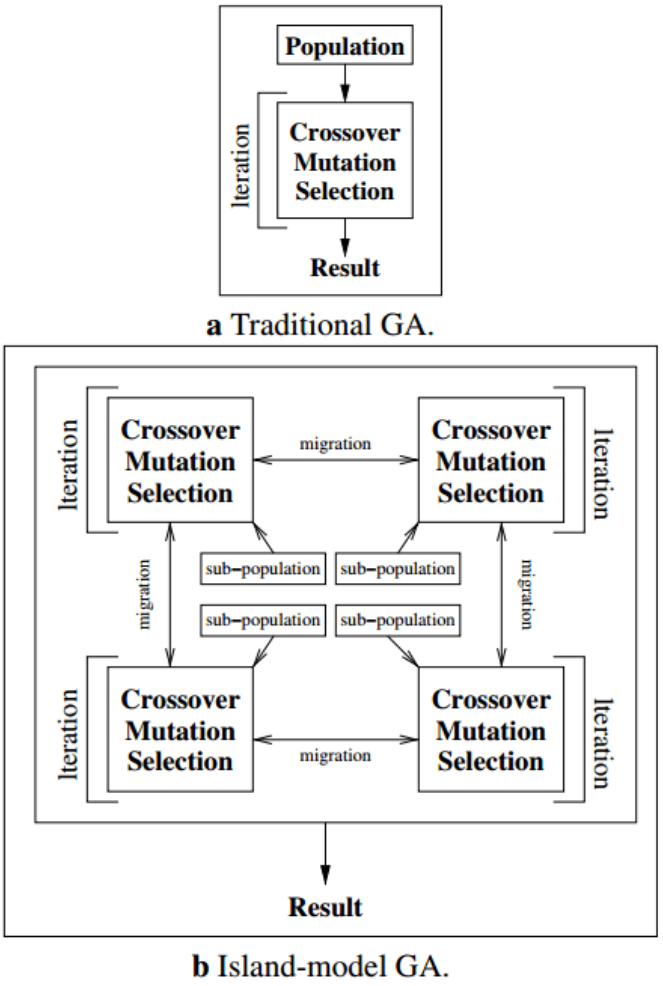


FIGURE 2.15 – L'AG traditionnel (TGA) et l'algorithme génétique parallèle (PGA)

Pour cette raison, dans [50], Kiran et Babu ont expliqué que la structure logicielle pour l'algorithme d'exploration de données distribuées doit encore agréger les informations provenant de différents nœuds d'ordinateur. Comme le montre la figure 2.16, la conception commune de l'algorithme d'exploration de données distribuées est la suivante : chaque algorithme d'extraction sera exécuté sur un nœud d'ordinateur (Worker) qui a ses données localement cohérentes, mais pas sur l'ensemble des données.

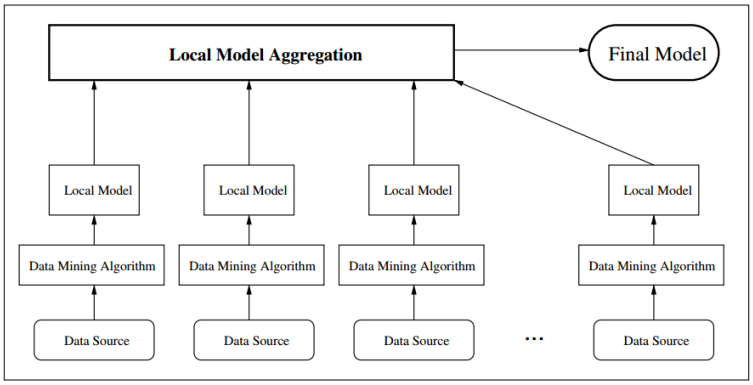


FIGURE 2.16 – Un exemple simple d'une Structure logicielle d'exploration de données distribuées
parencitecurtin2013mlpack

Pour construire une connaissance globalement significative après que chaque algorithme d'exploration ait trouvé son modèle local, le modèle local de chaque nœud informatique doit être agrégé et intégré dans un modèle final afin de représenter la connaissance complète. Bu et al. [51] ont trouvé des problèmes de recherche en essayant d'appliquer des algorithmes d'apprentissage automatique à des plates-formes informatiques parallèles. Par exemple, la première version de la structure logicielle map-reduce ne prend pas en charge l'itération (c.-à-d., la récursion). Mais la bonne nouvelle est que certains travaux récents [24, 52] ont porté une attention particulière à ce problème et ont essayé de le modifier. À l'instar des solutions pour améliorer les performances des algorithmes traditionnels d'exploration de données, l'une des solutions possibles pour améliorer les performances d'un algorithme d'apprentissage machine est d'utiliser CUDA, c.-à-d., un GPU, pour réduire le temps de calcul de l'analyse des données. Une autre étude [53] a tenté d'appliquer l'algorithme basé sur les fourmis à la plate-forme de calcul en grille. L'algorithme d'exploitation proposé étant étendu par l'algorithme de classification des fourmis de Deneubourg et al. [54], Ku-Mahamud a modifié le comportement de cet algorithme de classification de fourmis pour la classification de grandes données. Autrement dit, chaque fourmi sera placée au hasard sur la grille. Cela signifie que l'algorithme de groupement de fourmis peut alors être utilisé sur un environnement de calcul parallèle. Les tendances des études sur l'apprentissage automatique pour l'analyse des big data peuvent être divisées en deux volets : on tente de faire fonctionner des algorithmes d'apprentissage automatique sur des plateformes parallèles, comme Radoop[8], Mahout[9] et PIMRU [51] ; l'autre consiste à remanier les algorithmes d'apprentissage automatique pour les rendre adaptés au calcul parallèle ou à l'environnement de calcul parallèle, comme les algorithmes de réseau neural pour GPU [55] et l'algorithme à base de fourmis pour la grille [53]. En résumé, les deux permettent d'appliquer les algorithmes d'apprentissage automatique à l'analyse des big data, même si de nombreux problèmes de recherche doivent encore être résolus, comme le coût de la communication pour les nœuds informatiques différents [11] et le coût de calcul élevé que la plupart des algorithmes d'apprentissage automatique exigent [55]. La figure suivante 2.17, décrit quelques structures logicielles utilisées pour l'analuyse de données massives :

8. https ://rapidminer.com/products/radoop/
9. https ://mahout.apache.org/

$\mathcal{P}$	Name	References	Year	Description	$\mathcal{T}$
Analysis framework	DOT	[88]	2011	Add more computation resources via scale out solution	Framework
	GLADE	[89]	2011	Multi-level tree-based system architecture	
	Starfish	[92]	2012	Self-tuning analytics system	
	ODT-MDC	[96]	2012	Privacy issues	
	MRAM	[91]	2013	Mobile agent technologies	
	CBDMASP	[94]	2013	Statistical computation and data mining approaches	
	SODSS	[97]	2013	Decision support system issues	
	BDAF	[93]	2014	Data centric architecture	
	HACE	[95]	2014	Data mining approaches	
	Hadoop	[83]	2011	Parallel computing platform	Platform
	CUDA	[84]	2007	Parallel computing platform	
	Storm	[85]	2014	Parallel computing platform	
	Pregel	[125]	2010	Large-scale graph data analysis	
	MLPACK	[86]	2013	Scalable machine learning library	ML
	Mahout	[87]	2011	Machine-learning algorithms	
	MLAS	[124]	2012	Machine-learning algorithms	
	PIMRU	[124]	2012	Machine Learning algorithms	
	Radoop	[129]	2011	Data analytics, machine learning algorithms, and R statistical tool	
Mining algorithm	DBDC	[144]	2004	Parallel clustering	CLU
	PKM	[145]	2009	Map-reduce-based *k*-means clustering	
	CloudVista	[111]	2012	Cloud computing for clustering	
	MSFCUDA	[113]	2013	GPU for clustering	
	BDCAC	[127]	2013	Ant on grid computing environment for clustering	
	Corest	[114]	2013	Use a tree construction for generating the coresets in parallel for clustering	
	SOM-MBP	[126]	2013	Neural network with CGP for classification	CLA
	CoS	[115]	2013	Parallel computing for classification	
	SVMGA	[72]	2014	Using GA for reduce the number of dimensions	
	Quantum SVM	[116]	2014	Quantum computing for classification	
	DPSP	[121]	2010	Applied frequent pattern algorithm to cloud platform	FP
	DHTRIE	[120]	2011	Applied frequent pattern algorithm to cloud platform	
	SPC, FPC, and DPC	[117]	2012	Map-reduce model for frequent pattern mining	
	MFPSAM	[119]	2014	Concerned the specific interest constraints and applied map-reduce model	

$\mathcal{P}$ perspective, $\mathcal{T}$ taxonomy, *ML* machine learning, *CLU* clustering, *CLA* classification, *FP* frequent pattern

FIGURE 2.17 – structures logicielles utilisées pour l'analuyse de données massives

2.4 Défis de l'analyse de données volumineuses

De nombreuses études se sont concentrées sur l'utilisation de techniques d'analyse telles que l'exploration de données, la visualisation, l'analyse statistique et l'apprentissage automatique. Cependant, il est nécessaire de développer de nouvelles approches analytiques afin de relever les défis liés au Big Data, tels que le temps requis pour le traitement lorsque le volume de données est très important [56]. Dans [56] les auteurs présentent les difficultés d'application des solutions analytiques actuelles, y compris l'apprentissage automatique, l'apprentissage en profondeur, les approches progressives et l'informatique granulaire.

Les auteurs de [57] ont abordé de la même manière les applications, les opportunités et les défis liés au big data, et ont examiné plusieurs techniques permettant de relever les défis du big data, telles que le cloud computing et l'informatique quantique, afin d'examiner leur efficacité.

Les auteurs de [58] ont expliqué le potentiel et les applications des big data. Ils présentaient les techniques des données massives et offraient un certain contexte aux approches analytiques des données massives.

Par la suite, on va parenciter quelques défis qu'on doit résoudre dans cette thèse pour exploiter efficacement les données massives

2.4.1 Stockage des données, capture des données et qualité des données :

Capturer et stocker des données n'est pas facile, d'autant plus que la taille et la complexité des ensembles de données augmentent de plus en plus. Il n'y a souvent pas assez d'espace pour stocker de telles données massives et de nombreux secteurs et domaines, tels que les domaines financier et médical, sont forcés de supprimer des données. La capture et la création de données précieuses ne se font qu'à un coût élevé [57].

Les auteurs de [56] ont discuté des caractéristiques du big data en termes de traitement par de nombreux outils d'analyse et de visualisation. La couche des plateformes de données, ses composants et ses technologies ont été expliqués. En termes de capacités, différentes technologies ont été comparées, et les systèmes de big data ont été catégorisés en fonction de leurs caractéristiques et des services fournis aux utilisateurs. Ils ont montré que l'utilisation des données massives a encore de nombreux problèmes techniques à résoudre. Ils ont également présenté les défis des systèmes informatiques Big Data, en examinant les difficultés à différents niveaux «comprenant la capture, le stockage, la recherche, le partage, l'analyse, la gestion et la visualisation de données». Cela comprenait l'examen des questions de sécurité et de confidentialité. La taille des données massives augmente de façon exponentielle, ce qui empêche la technologie actuelle de gérer de tels ensembles de données.

Les défis actuels du Big Data incluent donc la gestion du Big Data, qui consiste à collecter, intégrer et stocker des données avec des exigences minimales (matériel et logiciel). La gestion des données volumineuses nécessite également le nettoyage des données pour des raisons de fiabilité, puis l'agrégation des données provenant de différentes sources avant de les encoder à des fins de sécurité et de confidentialité [57, 59]. Le défi du nettoyage des données volumineuses réside dans la complexité des données : vitesse, volume et variété [60]. Les défis de l'agrégation de Big Data sont impliqués dans la synchronisation de sources de données externes et de plateformes Big Data distribuées (comprenant des applications,

des référentiels, des capteurs, des réseaux, etc.) dans un système cohérent [56]. De plus, en raison du déséquilibre des capacités des systèmes, le défi réside dans l'architecture et la capacité des ordinateurs, car les capacités déséquilibrées des systèmes pourraient avoir une incidence sur les performances des applications de big data [57]. En outre, le défi des big data déséquilibrées est de savoir comment classifier les jeux de données déséquilibrés comme des techniques d'apprentissage classiques ne sont pas adaptées à des jeux de données déséquilibrés » [56]. Les défis de l'analyse de données volumineuses résident dans l'analyse complexe des données nécessaire pour comprendre les relations entre les caractéristiques des données. Certaines analyses de données nécessitent une analyse en temps réel, telles que la navigation, les réseaux sociaux, la finance, la biomédecine, l'astronomie et les systèmes de transport intelligents, tandis que d'autres analyses nécessitent des résultats précis mais pas nécessairement les mêmes vitesses. Le défi de l'analyse de données volumineuses provient principalement des 5V et de leurs effets sur les performances des ensembles de données [61]. Une solution pour le défi de stockage est l'utilisation de Hadoop (plate-forme Apache), qui est une plate-forme de traitement de données distribuée open-source avec la puissance de traiter des quantités extrêmement grandes de données. Pour ce faire, Hadoop divise les données en parties plus petites, puis spécifie certaines parties des jeux de données pour des serveurs séparés (nœuds) [62]

2.4.2 Analyse de données et visualisation

Les problèmes d'analyse des données découlent de la complexité des données (des types et des structures complexes). Les techniques d'analyse de données standard rencontrent des difficultés pour traiter de telles données massives, car il est plus difficile de comprendre les lois de distribution de ces données [63].

Les défis de la visualisation des données volumineuses proviennent des dimensions et de la taille élevées des données. L'objectif principal de la visualisation de données est d'expliquer efficacement les connaissances à l'aide de diagrammes, afin de transférer facilement des informations à l'utilisateur. Les connaissances cachées dans les jeux de données complexes et à grande échelle sont rendues visibles. Toutefois, pour une analyse plus précise des données, il est utile de résumer les informations dans des formats schématiques, y compris des caractéristiques ou des variables représentant des unités d'information. Néanmoins, en raison de la grande taille et des grandes dimensions du Big Data, il peut également être difficile de gérer la visualisation de données dans des applications Big Data [57, 63].

[56] a discuté l'utilité des méthodes d'exploration de données dans plusieurs domaines. Les méthodes d'exploration de données sont importantes lorsqu'elles sont utilisées pour découvrir des modèles et extraire de la valeur cachée dans des jeux de données volumineux.

L'application des techniques traditionnelles d'exploration de données, comme l'extraction par association, le regroupement et la classification, aux big data est toutefois inefficace et inexacte. Le volume, la vitesse et la variabilité de ces données la rendent impropre au stockage et à l'analyse à long terme. Plusieurs méthodes d'exploration de données ont ainsi été adaptées pour contenir des techniques de détection afin de prendre en compte l'environnement de données.

[64] a noté que certaines études empiriques et quelques idées anciennes ont beaucoup caractérisé la réalisation de la valeur des big data ; l'étude a examiné six débats identifiés en termes de « comment les organisations réalisent la valeur sociale et économique des big data qui nécessitent l'attention des recherches futures ». Deux autres caractéristiques des

big data ont également été identifiées, la portabilité et l'interconnectivité, et ces caractéristiques ont été utilisées pour montrer l'effet de la réalisation de la valeur des big data dans les organisations. Plusieurs suggestions pour une étude plus approfondie ont également été présentées :

— Des systèmes pertinents, tels que Hadoop, capables de travailler à la fois avec des données volumineuses et des données plus traditionnelles identifiées pour différents cas parencite ekbia2015big ;
— Examiner la dépendance à l'égard de la taille des organisations qui peuvent adopter les mégadonnées [65].
— Examiner les modèles organisationnels appropriés pour créer et s'approprier la valeur du Big Data [65, 66]
— Enquêtes complémentaires sur deux questions clés :
  - un accès Big Data contrôlé et ouvert lorsque l'analyse des données peut être considérés comme un avantage concurrentiel, les entreprises pouvant s'opposer à l'échange de données avec des concurrents perçus [67]
  - minimiser et contrer les risques sociaux liés à la réalisation de la valeur des données volumineuses [68].

2.5 Conclusion

Dans le premier chapitre de l'état de l'art, nous avons introduit le monde de big data qui est caractérisé par les 3Vs (Volume, Variété et Vélocité). Les solutions traditionnelles d'analyse et de traitement de données sont devenues incapables de donner des résultats dans un temps acceptable lorsqu'on travaille avec des données massives. Dans ce chapitre, nous avons donné les démarches à suivre et les outils à utiliser pour traiter les données massives. Dans cette thèse, nous avons choisi comme jeux de données massives : les images hyperspectrales et un corpus de la langue arabe. Dans le chapitre 3, nous allons détailler les images hyperspectrales et dans le chapitre 4, nous allons présenter le traitement automatique des langues naturelles et spécialement la langue arabe

Chapitre 3

Les Images Hyperspectrales

Sommaire

3.1 Télédétection

La télédétection est le balayage de la Terre par satellite ou par un avion volant à haute altitude afin d'obtenir des informations à ce sujet. Aujourd'hui, notre capacité d'acquérir des données de télédétection a connu une évolution radicale. Nous sommes entrés dans l'ère des Big Data. Ces données montrent clairement les caractéristiques des Big Data : Volume, Variété, Vélocité. De nombreux pays se sont empressés de lancer leurs propres satellites. La figure 3.1 résume le nombre de satellites de télédétection lancés par les principaux pays entre 1962 et 2014. On voit que les Etats-Unis, l'Inde et la Russie sont les trois pays qui ont lancé la plupart des satellites de télédétection. Pour la plupart des pays et des régions, presque tous les satellites de télédétection ont été lancés entre 2001 et 2014.

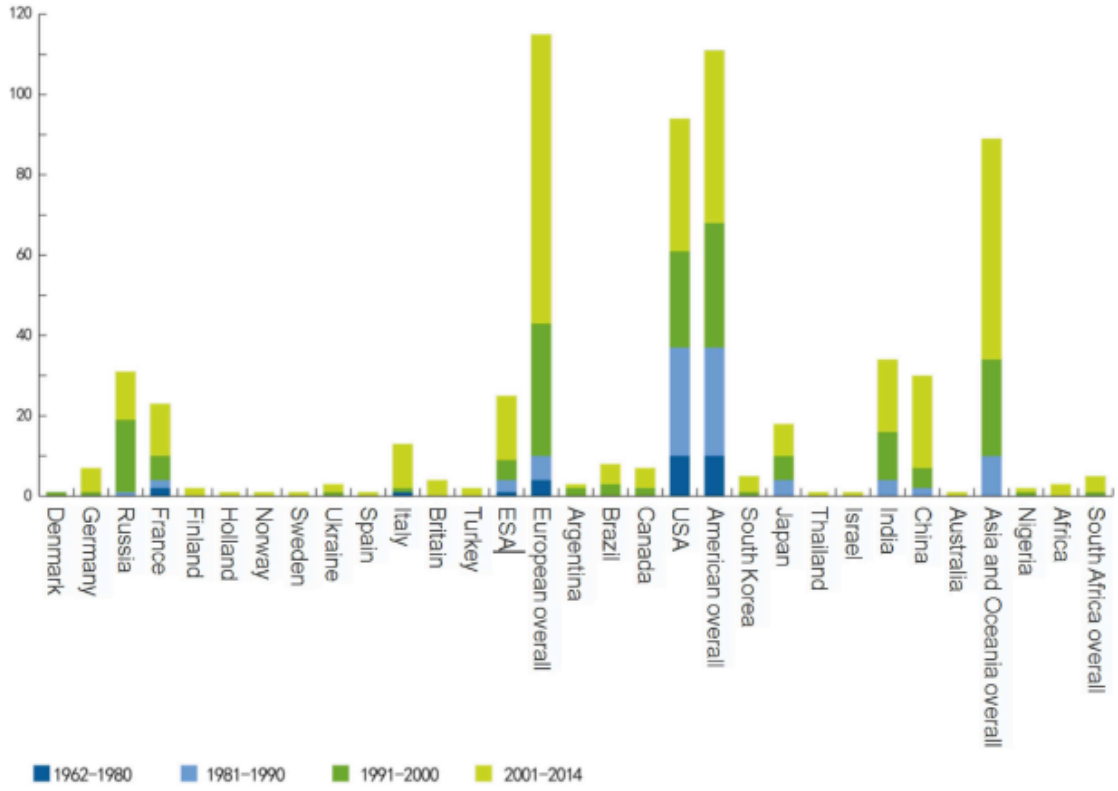


FIGURE 3.1 – Résumé des satellites de télédétection par pays ou par région

Les données massives de télédétection attirent de plus en plus l'attention des projets gouvernementaux, des applications commerciales et les domaines académiques. Les exemples suivants montrent l'utilisation des données massives dans le domaine de télédétection :

le "**Earth Observing System Data and Information System**" (EOSDIS) [69] : est l'un des plus importants projets du gouvernement américain. Le projet fournit la capacité de gérer, de bout en bout, les données scientifiques de la Terre de la « NASA » provenant de diverses sources.

L' "**Agence spatiale européenne**" a organisée en 2019 une conférence nommée « Big Data from Space » [70] . La conférence vise à stimuler les interactions et à réunir des chercheurs,

des ingénieurs, des utilisateurs, des fournisseurs d'infrastructure et de services, intéressés par l'exploitation du « Big Data from Space »

Le "**Group on Earth Observations** " (GEO) [71], la plus grande organisation intergouvernementale de coopération multilatérale, fait également la promotion du développement des Big Data.

Dans le domaine des applications commerciales, "Google Earth" pourrait être l'un des exemples de succès des Big Data de télédétection. De nombreuses applications de télédétection telles que la détection de cibles, la couverture terrestre, la ville intelligente, etc, peuvent être développées facilement en se basant sur Google Earth.

Avec l'écosystème GBDX [72] de la plate-forme géospatiale de DigitalGlobe, la société DigitalGlobale crée rapidement des empreintes de construction en tirant parti de l'apprentissage automatique combiné à la bibliothèque d'images de 100 pétaoctets basée sur le cloud de DigitalGlobe.

D'autres grandes entreprises telles que Microsoft (Redmond, Washington, États-Unis) et Baidu (Beijing, Chine) développent toutes leurs cartes électroniques qui prennent en charge les données volumineuses de télédétection et les vues de la rue. Les applications commerciales sur le Big Data changent la vie des gens.

Dans les domaines académiques, le «big data de télédétection» est également l'un des sujets les plus populaires. Beaucoup de grands journaux ont publié leurs numéros spéciaux sur le « Big Data de télédétection ».

— **IEEE JSTARS** [73] a publié le numéro spécial sur "Big Data in Remote Sensing" en 2015.

— **Journal of Applied Remote Sensing** [74] a publié le numéro spécial sur la "Management and Analytics of Remotely Sensed Big Data" en 2015.

— Le magazine **IEEE Geoscience and Remote Sensing** [75] a publié le numéro spécial sur le « Big Data from Space » en 2016.

— **GeoInformatica** [76] de Springer a lancé le numéro spécial sur la "Big Spatial and Spatiotemporal Data Management and Analytics" en 2016.

— **Environmental Remote Sensing** [77] a lancé le numéro spécial sur les "Big Remotely Sensed Data : Tools, Applications and Experiences" en 2017.

— **Remote Sensing MDPI** [78] lance un appel à contributions sur des numéros spéciaux sur "Advanced Machine Learning and Big Data Analytics in Remote Sensing for Natural Hazards Management", "SAR in the Big Data Era and Analysis of Big Data in Remote Sensing" en 2018.

— **International Journal of Digital Earth** [79] lance un appel à contributions pour le numéro spécial sur la "Social Sensing and Big Data Computing for Disaster Management" en 2018.

La télédétection hyperspectrale est l'outil optique par excellence pour augmenter la connaissance et la compréhension de la surface de la Terre. La télédétection hyperspectrale, est une technologie relativement nouvelle sur laquelle les chercheurs et les scientifiques étudient actuellement la détection et l'identification des minéraux, de la végétation terrestre, des matériaux, etc.

La télédétection hyperspectrale, également appelée spectroscopie d'imagerie, combine l'imagerie et la spectroscopie dans un système unique qui comprend souvent de grands jeux

de données et nécessite de nouvelles méthodes de traitement. Les jeux de données hyperspectrales sont généralement composés d'environ 100 à 200 bandes spectrales de bandes passantes relativement étroites (5 à 10 nm) et de haute résolution spatiale (1-5 m), tandis que les jeux de données multispectrales sont généralement composés d'environ 5 à 10 bandes de bandes passantes relativement grandes (70 à 400 nm).

La télédétection hyperspectrale est utilisée en laboratoire par les physiciens et les chimistes depuis plus de 100 ans pour identifier les matériaux et leur composition. La spectroscopie peut être utilisée pour détecter des caractéristiques d'absorption individuelles en raison de liaisons chimiques spécifiques dans un solide, un liquide ou un gaz.

3.2 Caractéristiques des données massives de télédétection

Les données massives (en anglais Big Data) font référence à une collection de jeux de données très vastes et complexes qu'il est difficile de traiter avec des algorithmes et des modèles traditionnels. Les défis incluent l'acquisition, le stockage, la recherche, le partage, le transfert, l'analyse et la visualisation des données. Les scientifiques rencontrent régulièrement des limitations dues aux grands jeux de données dans de nombreux domaines, tels que la géoscience et la télédétection, les simulations physiques complexes et la recherche biologique et environnementale. Lorsque nous parlons des caractéristiques des données massives, il est populaire de se référer aux trois V [4] : une croissance significative du volume, de la vitesse et de la variété des données. Cependant, le terme trois V est trop général. Le big data de la télédétection présente plusieurs caractéristiques concrètes et particulières. Les données ont des caractéristiques : multi-sources, multi-échelle, haute dimension, état dynamique, isomère et non-linéarité.

La caractéristique multi-source des données massives de télédétection est évidente. La raison fondamentale de la caractéristique multi-source est que nous utilisons souvent différents instruments pour acquérir les données. De plus, les significations physiques des données multi-sources peuvent être totalement différentes. Du point de vue du mécanisme d'imagerie, les principaux types de données sont les données optiques, les données micro-ondes et les données en nuage de points. D'autres types de données de télédétection comprennent les paires stéréographiques créées à partir de plusieurs photographies (souvent utilisées pour créer des cartes tridimensionnelles ou topographiques) et les données de gravité qui montrent la situation de gravité et la quantité d'eau disponible dans une région.

Les données multi-sources nous permettent d'utiliser et de comprendre des informations sous différents angles. Cependant, elles sont parfois source de confusion dans la mesure où nous devons décider quel type de données est le plus approprié et le plus efficace pour une application donnée.

On fait souvent référence aux multiples échelles des données massives de la télédétection. L'échelle d'observation, aussi appelée échelle de mesure, fait référence à la résolution, à l'intervalle de temps, à la plage spectrale, à l'angle solide ou à la direction de la polarisation [80]. L'échelle spatiale se réfère à la résolution spatiale et peut être considérée comme la taille des plus petits objets qui peuvent être distingués par des capteurs. Une bonne observation dépend souvent de l'échelle spatiale appropriée. En conséquence, nous avons un grand nombre de satellites et de capteurs avec différentes résolutions spatiales. Du point de vue de la résolution spatiale, il y a les satellites à haute résolution comme Quickbird (résolution de

0,61 m) [81], les satellites à moyenne résolution comme les satellites Landsat (30 m) [82], [83] et les satellites à basse résolution comme MODIS (250 m) [84]. Les caractéristiques multi-échelles des données massives de télédétection signifient qu'il est important de choisir une échelle appropriée et de tenir compte des effets d'échelle dans l'analyse et le traitement des données.

La caractéristique de grande dimension des données massives de télédétection se reflète principalement dans les dimensions spectrales et temporelles des données. Par exemple, l'AVIRIS (Air-borne Visible/Infrared Imaging Spectrometer) de la NASA mesure les réponses spectrales dans 224 bandes spectrales contiguës acquises dans les régions visibles et dans le proche infrarouge [85]. La figure 3.2 donne quelques exemples sur les dimensions des données massives de télédétection [86].

Sensor	Organization /Country	Optical Subsystem	Spectral Bands	Spectral Range (µm)	Spectral Resolution	Spatial Coverage
Landsat-8	NASA, US.	VNIR-TIR	8	0.45-12.50	8	Global
MODIS	NASA, US.	VNIR-TIR	36	0.40-14.40	250-1000	Global
MERIS	ESA, EU.	VNIR	15	0.39-1.040	300	Global
ASTER	NASA, US & METI, Japan.	VNIR-TIR	15	0.52-11.65	15-90	Global
Hyperion	NASA, US.	VNIR-SWIR	242	0.40-2.500	30	Regional
ALOS	JAXA, Japan.	VIS	1	0.52-0.77	2.5	Local
AVIRIS	NASA, US.	VNIR	224	0.38-2.500	4-20	Local
HyMap	Integrated Spectronics Pty Ltd, Australia.	VNIR-SWIR	128	0.45-2.480	2-10	Local
ROSIS	DLR, Germany.	VNIR	115	0.42-0.873	2	Local
DAIS-7915	GER Corp, US.	VNIR-TIR	79	0.45-12	3-10	Local
AISA	SPECIM, Finland.	VNIR	286	0.45-0.9	2.9	Local
CASI	Itres Research, Canada.	VNIR	288	0.43-0.87	2	Local

FIGURE 3.2 – Capteurs spectraux actuels fournissant des données pour la cartographie des terres

L'analyse des données d'une image de grande dimension présente à la fois de nouvelles possibilités et de nouveaux défis. Les données de grande dimension nous fournissent plus d'informations sur la surface de la Terre mais soulèvent aussi de nombreuses difficultés. La première difficulté est la malédiction de la dimensionnalité [87]. La complexité de nombreux algorithmes de Data Mining existants est exponentielle par rapport au nombre de dimensions. Avec une dimensionnalité croissante, ces algorithmes deviennent rapidement intraitables par calcul et donc inapplicables dans de nombreuses applications réelles. La deuxième

difficulté est l'hétérogénéité. Avoir un peu de points dans les données de grande dimension rend « l'apprentissage efficace », difficile dans ce qu'on appelle le phénomène de l'espace vide. En fait, le phénomène de l'espace vide est un cas particulier de l'hétérogénéité des données massives. Apparemment, les données de grande dimension sont beaucoup plus difficiles à analyser que les données de faible dimension dans la plupart des cas.

3.3 Imagerie hyperspectrale

Les capteurs hyperspectraux permettent l'acquisition simultanée de plusieurs informations, notamment avec l'acquisition quasi continue de données spectrales pour tous les pixels de la scène, de l'ultraviolet au proche infrarouge, qui correspond à plusieurs centaines d'images associées à différentes bandes spectrales pour la même scène. Ce très grand nombre de données augmente inévitablement la complexité du traitement, rendant les méthodes traditionnelles de traitement d'image moins efficaces. Pour exploiter ces informations (visualisation ou classification), il a fallu développer de nouvelles méthodes de traitement d'images hyperspectrales (HSI).

3.3.1 Représentation de l'image hyperspectrale

Les images classiques en couleur sont représentées avec trois couches (ou bandes : Rouge, Vert et Blue) qui ont chacune des informations différentes. Il est possible de faire encore plus de couches en utilisant des bandes de longueur d'onde plus petites.

Les images hyperspectrales peuvent être considérées comme une pile d'images avec différents intervalles de longueur d'onde (canaux spectraux) provenant de la même scène de la surface de la Terre. Sur la base de cette interprétation, les images hyperspectrales peuvent être référées à des cubes de données hyperspectrales.

Chaque canal spectral représente une image en niveaux de gris et toutes les images forment un cube hyperspectral en trois dimensions. La figure 3.3 illustre un exemple d'un cube de données hyperspectrales. En raison de cette représentation cubique des données, il est naturel de considérer l'utilisation d'un tenseur d'ordre 3 comme modèle mathématique pour l'analyse des images hyperspectrales. En général, les dimensions spatiales sont respectivement associées au L et C du tenseur et la dimension spectrale est associée à d du tenseur.

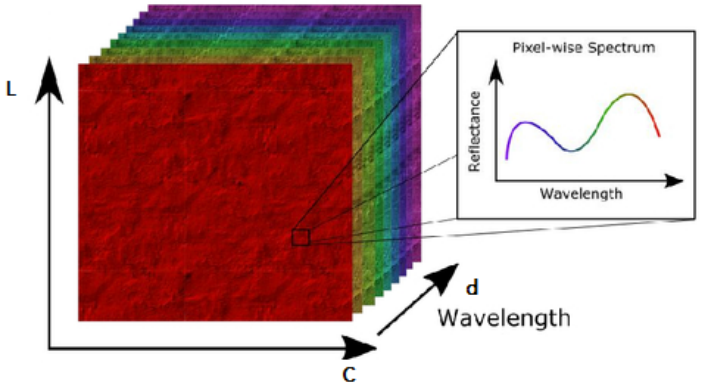


FIGURE 3.3 – Tenseur d'une image hyperspectrale

De façon plus détaillée, une image hyperspectrale peut être introduite à partir de l'une des perspectives suivantes :

— **Perspective spectrale (ou dimension spectrale) :**
Dans ce cas, un cube de données hyperspectrales se compose de plusieurs pixels et chaque pixel est un vecteur de d valeurs (nommé ultérieurement vecteur pixel). Chaque pixel (ou vecteur pixel) correspond au rayonnement réfléchi de la région spécifique de la Terre et possède de multiples valeurs dans les bandes spectrales. Ces informations spectrales détaillées peuvent être utilisées pour analyser avec précision, les différents matériaux. Dans la figure 3.3, on trouve à droite une courbe d'un pixel vecteur, avec des d valeurs pour chaque bande dans la dimension spectrale. Dans ce domaine, les points suivants sont importants :

- En général, les vecteurs de différents pixels appartenant à un matériel similaire ont presque les mêmes valeurs (même signature). Différentes techniques de classification supervisées et non supervisées sont utilisées afin de regrouper les vecteurs avec presque la même caractéristique.
- En général, dans chaque vecteur, les pixels de voisinage dans différents canaux spectraux ont une forte corrélation. Différentes techniques de réduction des caractéristiques supervisées et non supervisées sont utilisées afin de réduire la dimensionnalité du cube de données hyperspectrales.

— **Perspective spatiale (ou dimension spatiale)** :
Dans ce cas, un cube de données hyperspectrales se compose de d images en niveaux de gris d'une taille de L x C. Les valeurs de tous les pixels dans une seule bande spectrale font une image en niveau de gris avec deux dimensions L x C (voir Fig. 3.3) Dans la dimension spatiale, les pixels adjacents appartiennent très souvent au même objet (en particulier pour les données à très haute résolution (VHR)). Cette dimension fournit des informations précieuses sur la taille et la forme des différentes structures et objets sur la terre. Il existe plusieurs façons d'extraire l'information spatiale (p. ex., la segmentation)

— **Résolution temporelle** :
Dans la télédétection hyperspectrale, la résolution temporelle dépend des caractéristiques orbitales du capteur d'imagerie. Elle est généralement définie comme le temps nécessaire à la plateforme du capteur pour revisiter et obtenir des données à partir du même emplacement [88]. La résolution temporelle est dite élevée si la fréquence de visite de la plate-forme de capteur pour le même emplacement est élevée et si la fréquence de visite est faible. Elle est normalement définie en jours.

3.3.2 Applications modernes de l'imagerie hyperspectrale

L'imagerie hyperspectrale (HSI) est de plus en plus utilisée pour une grande variété d'applications commerciales, industrielles et militaires. On trouve l'utilisation de HSI dans :

— Évaluation de la qualité et de la sécurité des aliments
— Diagnostic médical
— Gestion des ressources en eau et des inondations
— **Agriculture de précision**

Bien que les domaines d'utilisation des HSI soient nombreux pour chaque domaine d'application on trouve, dans l'état de l'art des documentations riches, On va seulement, détailler dans la suite que l'utilisation de HSI dans le domaine de l'Agriculture de précision :

De nombreuses études ont montré que la production agricole mondiale doit être doublée d'ici à la fin de 2050 en raison de la croissance rapide de la population mondiale [89]. Cependant, diverses études ont montré que les rendements des cultures n'augmentent plus à un rythme permettant de répondre aux besoins croissants de la population [90]-[91]. Des études récentes ont également indiqué que l'augmentation du rendement des cultures sans utiliser davantage de terres à la culture est le moyen le plus efficace pour assurer la sécurité alimentaire [92]-[93]. La pauvreté et la sous-alimentation dans le monde peuvent être directement réduites en augmentant la production agricole ; de plus, la majorité de la population pauvre et sous-alimentée est constituée d'agriculteurs eux-mêmes [94].

L. Zhang et al. [95] ont donné un état de l'art sur les techniques d'apprentissage en profondeur les plus avancées pour l'extraction de caractéristiques représentatives et la compréhension de la scène dans les images hyperspectrales de télédétection.

Traditionnellement, la surveillance de la maladie des cultures et les attaques des insectes était effectuée par une inspection manuelle et visuelle à partir du sol. Ces méthodes ont été limitées par le fait que les symptômes visuels apparaissent souvent à des stades ultérieurs de la maladie, ce qui rend difficile le rétablissement de la santé des plantes. L'avancée des méthodes HSI aéroportées et terrestres a rendu possible l'évaluation du stress des cultures, en analysant les caractéristiques du sol et de la végétation de manière rentable, remplaçant ainsi les méthodes de dépistage traditionnelles.

Le stress dû à la sécheresse est un facteur important qui influe sur le rendement des cultures. La détection rapide des stress liés à l'eau peut considérablement augmenter les chances de la réussite d'une culture. Les niveaux élevés du stress de l'eau sont perceptibles dans les variations des pigments photosynthétiques. Ces changements entraînent une teinte jaunâtre dans les cultures, en raison de l'augmentation de la réflectance de la longueur d'onde rouge. Contrairement à l'œil humain, les capteurs HSI peuvent détecter ces changements à un stade précoce.

Colombo et al. [96] ont indiqué que des modifications de EWT (leaf equivalent water thickness) étaient responsables de la modification de la réflectance des feuilles dans les spectres visible et infrarouge. Ils ont déclaré que les indices de régression hyperspectrale calculés à partir de HSI étaient des outils puissants pour l'estimation de la teneur en eau au niveau de la feuille et du paysage. Rascher et al. [97] ont utilisé un système HSI portable et un indice de réflectance photochimique pour estimer le stress hydrique dans les feuilles d'arbres tropicaux et ont observé les effets temporels de la déshydratation sur les feuilles des arbres. Rossini et al. [98] ont trouvé que le HSI était utile pour détecter le stress dû à la sécheresse au niveau de la ferme avec le maïs.

Les carences en éléments nutritifs et la contamination du sol provoquent divers symptômes qui peuvent être évalués par HSI. Schuerger et al. [99] ont utilisé HSI pour observer une carence en zinc et la toxicité afin d'identifier les niveaux de chlorophylle liés aux symptômes de stress. Ils ont indiqué que les méthodes traditionnelles d'échantillonnage direct sont beaucoup plus coûteuses que l'HSI.

La surveillance de la croissance des cultures a permis de prévoir la production. Liu et al. [100] ont amélioré la prévision du rendement du blé d'hiver en utilisant de nouveaux paramètres spectraux. La technologie de la classification fine en agriculture a beaucoup évolué.

La figure **??** montre la classification fine fondée sur l'HSI des régions productrices de légumes.

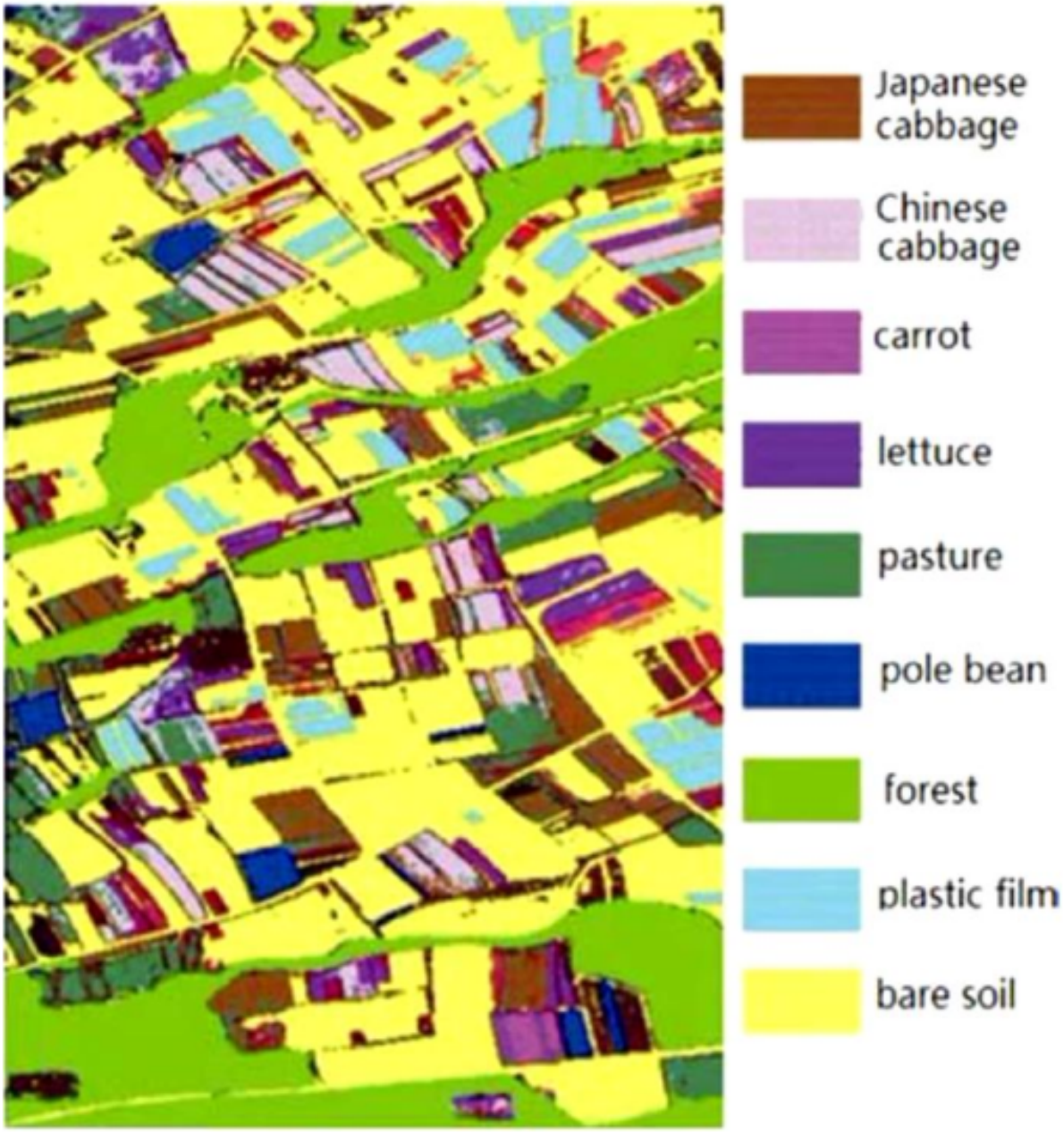


FIGURE 3.4 – Classification fine, basée sur HSI, des régions productrices de légumes

3.3.3 Les contraintes de l'hyperspectrale

Les principales contraintes sont le coût et la complexité :

— Des ordinateurs rapides, des détecteurs sensibles et de grandes capacités de stockage de données sont nécessaires pour analyser les données hyperspectrales.

— Dans la dimension élevée, il est difficile d'effectuer une estimation précise des paramètres, par exemple pour la visualisation des données hyperspectrales.

3.4 Conclusion

L'imagerie hyperspectrale a été initialement développée pour la télédétection ; cependant, les chercheurs ont commencé, dernièrement, à voir son potentiel dans d'autres domaines aussi comme le diagnostic médical et la détection des maladies. Les images hyperspectrales fournissent des quantités massives de données sur les objets qu'elles étudient, ce qui pose des défis lors du traitement de l'information.

La réduction des dimensions, la visualisation, la classification par les outils d'apprentissage automatique, tels que les réseaux neuronaux convolutionnels (CNN), l'utilisation des architectures de Big Data comme les plateformes distribuées parallèles sont les principaux axes de recherche que nous avons traités durant cette thèse et qui seront détaillés dans les chapitres suivants.

Chapitre 4

Traitement automatique du langage naturel

Sommaire

TABLE 4.1 – Méthodes populaires d'apprentissage en profondeur en TALN

Paper	NLP Tasks	Architecture	Datasets
Socher et al. 2013 [101]	Sentiment Analysis	RNTN	SST
Kim 2014 [102]	Sentiment Analysis, General Classifcation	CNN	SST
Wehrmann et al. 2017 [103]	Sentiment Analysis	Conv-Char-S	MTD
Bahdanau et al. 2014 [104]	Translation	Bidir RNN Encoder Decoder	WMT-14-EF
Cho et al. 2014 [105]	Translation	RNN Encoder Decoder	WMT-14-EF
Wu et al. 2016 [106]	Translation	GNMT	WMT-14-EF WMT-14-EG
Socher et al. 2011 [107]	Paraphrase Identifcation	Unfolding RAE	MSRP
Yin et al. 2016 [108]	Paraphrase Identifcation, Question & Answer	ABCNN	WikiQA MSRP
Kågebäck et al. 2014 [109]	Summarization	Unfolding RAE	OD
Dong et al. 2015 [110]	Question & Answer	MCCNN	WQ
Feng et al. 2015 [111]	Question & Answer	CNN	IQA

4.1 Introduction

Le traitement automatique du langage naturel (TALN) est une série d'algorithmes et de techniques qui se concentrent principalement sur l'enseignement des ordinateurs pour comprendre le langage humain. Certaines tâches du TALN incluent la classification des documents, la traduction, la paraphrase, la similarité des textes, la synthèse et la réponse aux questions. Le développement de TALN est difficile en raison de la complexité et de la structure ambiguë du langage humain et spécialement pour la langue Arabe qui présente des spécificités grammaticales et syntaxiques. De plus, le langage naturel est très spécifique au contexte, où les significations littérales changent en fonction de la forme des mots et de la spécificité du domaine. Les méthodes d'apprentissage en profondeur ont récemment permis de démontrer plusieurs tentatives réussies pour atteindre une grande précision dans les tâches de TALN. La table 4.1 présente un résumé de certaines principales solutions pour l'apprentissage en profondeur de TALN, leurs architectures et leurs jeux de données.

La plupart des modèles NLP suivent un prétraitement similaire :

— Étape 1 : le texte d'entrée est décomposé en mots par tokénisation

— Étape 2 : ces mots sont reproduits sous forme de vecteurs, ou n-grammes.

Représenter des mots dans une faible dimension est une opération importante pour créer une perception précise des similarités et des différences entre les différents mots. Le défi arrive quand il faut décider de préciser la longueur des mots contenus dans chaque n-gramme. Cette procédure est spécifique au contexte et nécessite une connaissance préalable du domaine.

Dans la partie suivante, on va présenter quelques-unes des approches très percutantes pour résoudre les tâches les plus connues du TALN.

4.2 Applications du TALN

4.2.1 Analyse de sentiment

Cette branche du TALN consiste à examiner un texte et à classifier le sentiment ou l'opinion de l'auteur. La plupart des jeux de données pour l'analyse des sentiments sont étiquetés comme positifs ou négatifs, et les phrases neutres sont supprimées par les méthodes de classification de subjectivité. Un exemple populaire est le Standford Sentiment Treebank (SST) [101], un jeu de données des critiques de films étiquetés dans cinq catégories (allant de très négatif à très positif). Parallèlement à l'introduction à la SST, Socher et al. [101] proposent un réseau de tenseur neural récursif (RNTN) qui utilise des vecteurs de mots et analyse un arbre pour représenter une phrase, capturant les interactions entre les éléments ayant une fonction de composition à base de tenseur. Cette approche récursive est avantageuse lorsqu'il s'agit de la classification au niveau de la phrase puisque la grammaire présente souvent une structure arborescente.

Kim [102] améliore la précision pour SST en suivant une approche différente. Même si les modèles CNN ont été créés initialement dans un souci de la reconnaissance et de la classification des images, leurs implémentation dans le cadre du TALN s'est avérée un succès et a donné d'excellents résultats. Kim présente un modèle CNN simple en utilisant une couche de convolution au-dessus de vecteurs entrainés dans une architecture BoW. Les modèles ont été maintenus relativement simples avec un petit nombre d'hyperparamètres pour l'entrainement. Les réseaux sociaux sont une source populaire de données pour étudier les sentiments. Le Multilingual Twitter Dataset (MTD) [112] est l'un des plus grands jeux de données publiques, contenant plus de 1,6 million de tweets annotés manuellement en 13 langues. Il est difficile d'appliquer l'analyse des sentiments aux tweets, en raison de la nature courte du texte. Pour aborder la question d'un jeu de données multilingue avec une petite quantité de texte, [103] propose Conv-Char-S, une architecture basée sur les caractères et qui est exempte de la dépendance aux langues. Bien que l'approche n'ait pas été en mesure de surpasser les architectures d'incorporation de mots, les auteurs affirment que sa simplicité et sa consommation d'énergie prédictive sont un bon compromis.

4.2.2 Traduction automatique

La traduction automatique (en anglais : Machine Translation- MT) est essentiellement une application permettant de mapper d'une langue humaine (langue source) à une autre langue (langue cible). Les différentes approches de la MT peuvent être regroupées en deux catégories : la symbolique basée sur les règles (RBMT) et la statistique basée sur le corpus (SMT).

L'apprentissage profond a joué un rôle important dans l'amélioration des méthodes traditionnelles de traduction automatique. Cho et al. [105] ont introduit une nouvelle architecture de codage et de décodage basée sur RNN afin d'entrainer les mots dans une « Neural Machine Translation » (NMT). L'architecture de « RNN Encoder-Decoder » utilise deux RNN : l'un mappe une séquence d'entrée en vecteurs de longueur fixe, tandis que l'autre RNN décode le vecteur dans les symboles cible. L'inconvénient du « RNN Encoder-Decoder » est la dégradation des performances lorsque la séquence de symboles d'entrée devient plus grande. Bahdanau et al. [104] traitent ce problème en introduisant un vecteur de longueur dynamique et en apprenant conjointement les procédures d'alignement et de traduction. Leur approche consiste à effectuer une recherche binaire pour rechercher les parties du discours les plus prédictives pour la traduction. Néanmoins, les systèmes de traduction récemment proposés sont coûteux en calcul et inefficaces dans le traitement des phrases contenant des mots rares. Ainsi, dans [106], « Google's Neural Machine Translation » (GNMT) est proposé, introduisant un équilibre entre la flexibilité fournie par les modèles de niveau caractères et l'efficacité des modèles de niveau mots. GNMT est un réseau LSTM profond utilisant huit couches de codeur et de huit couches de décodeur connectées à l'aide du mécanisme basé sur l'attention. La méthode basée sur l'attention a été introduite pour améliorer les NMT en général. La MT arabe-anglais a reçu beaucoup d'attention ces dernières années, ce qui a permis des progrès importants en termes de ressources créées et de systèmes construits [Annexe C]. Il existe plusieurs grands corpus parallèles et de nombreux dictionnaires pour l'arabe-anglais, et d'autres langues, par exemple, le corpus des Nations Unies a des documents parallèles en arabe, anglais, chinois, espagnol, français et russe . La majorité des recherches sur la MT arabe porte sur l'arabe-anglais ; cependant, des publications ont été publiées en anglais-arabe [113], [114], [115], [116] arabe-français[hasan2006creating] et même arabe-chinois [117] et danois-arabe [118]. Diverses entreprises ont des différents systèmes de MT pour les différentes paires de langues ; le plus notable est Google Translate qui permet la traduction bidirectionnelle à travers plus de 50 langues, y compris l'arabe. D'autres systèmes publics importants comprennent Microsoft Bing Trasnslator et Tarjim de Sakhr.

4.2.3 Résumé automatique de texte

La synthèse automatique permet d'extraire les informations les plus significatives et les plus pertinentes des documents volumineux. Un résumé bien représenté réduit efficacement la taille du texte sans perdre les informations les plus importantes. Cela peut considérablement réduire le temps et les calculs requis pour analyser de grands jeux de données en texte. Kågebäck et al. [109] proposent un modèle basé sur la représentation vectorielle continue pour les phrases. Leur modèle évalue de multiples combinaisons et compositions pour obtenir des représentations significatives. Ganesan et al. [119] utilisent un modèle basé sur un graphe et qui produit de brefs résumés à partir du jeu de données d'opinion connu sous le nom de « Opinosis Dataset » (OD). Leur modèle cible les opinions des utilisateurs en termes de commentaires, de critiques de produits et de rapports de satisfaction de la clientèle, sans perte de matériel pédagogique.

4.2.4 Système Question-Réponses

Un système automatique de Question-Réponses devrait être en mesure d'interpréter une question en langage naturel et d'utiliser le raisonnement pour retourner une réponse appropriée. Les bases de connaissances modernes, telles que le célèbre jeu de données FREEBASE,

permettent à ce domaine de disparaître et de sortir du temps où les fonctionnalités et les ensembles de règles étaient conçus à la main pour des domaines spécifiques.

4.3 Application de TALN pour la langue Arabe

4.3.1 Les caractéristiques de la langue Arabe.

L'arabe est l'une des langues les plus courantes avec plus de 420 millions de locuteurs dans le monde. C'est l'une des six langues officielles utilisées par l'Organisation des Nations Unies [1]. Contrairement à l'anglais, l'arabe n'a pas de majuscule,c'est une langue écrite de droite à gauche, dans un style cursif, et qui comprend 28 lettres. Elle se distingue également des autres langues naturelles par la présence de diacritiques qui représentent une petite voyelle comme «fatha, kasra, damma, sukun, shadda et tanween». Le système orthographique de la langue arabe est basé sur l'effet des diacritiques, chaque type spécifique de diacritique produisant des mots différents avec des significations différentes. Cette langue a des lettres spécifiques appelées voyelles arabes (waw, yaa, alf) qui nécessitent un système spécial de morphologie et de grammaires. Ce qui distingue également l'arabe, c'est l'énorme quantité de vocabulaires et de concepts [120]. Bien que les textes arabes soient considérés comme les plus difficiles, il existe peu d'études sur le traitement des textes arabes pour des raisons liées aux caractéristiques linguistiques de la langue arabe [121]. Dans la suite, on va présenter une variété d'applications TALN qui ont récemment émergé pour manipuler des langues telles que l'arabe, Anglais, et Ourdou. Une de ces applications est la classification des textes (TC), qu'on va l'utiliser dans la contribution [chapitre 8] pour compléter le texte manquant dans les documents arabes

4.3.2 Travaux connexes

Il existe de nombreux algorithmes de classification qui ont été appliqués aux textes arabes. Dans leur application de l'algorithme Naive Bayes (NB) pour classifier 1500 documents en arabe, El-Kourdi et al trouvent cinq grandes catégories dont les résultats indiquent que la précision est d'environ 68,78

El-Halees et al. ont également classifié 300 documents arabes en appliquant différents algorithmes tels que le modèle d'espace vectoriel (VSM), les algorithmes K-Nearest Neighbor (KNN) et Naïve Bayes (NB). La précision de la classification était de 74,41

Une quantité considérable de travaux a porté sur l'application de CNN à la langue Arabe ainsi qu'à d'autres langues.

Dans [122], les auteurs proposent le codage automatiquement du texte au niveau octet en utilisant CNN avec une architecture récursive. Des expériences ont été effectuées sur des jeux de données en Arabe, en Chinois et en Anglais. La motivation était de rechercher s'il était possible d'avoir une génération de texte évolutive et homogène au niveau des octets de manière non séquentielle via la tâche simple d'encodage automatique. Les travaux ont montré qu'une génération de texte non séquentielle à partir d'une représentation de longueur fixe est non seulement possible, mais permet également d'obtenir de meilleurs résultats de codage automatique que l'utilisation de RNN.

1. https ://www.un.org/ar/

Dans [123], les auteurs ont utilisé CNN pour résoudre trois problèmes de classification démographique (genre, justesse et combinaison du genre et de la justesse). La recherche a été effectuée sur deux bases de données d'écriture publiques, IAM et KHATT, contenant respectivement du texte anglais et du texte arabe. CNN s'est avéré plus apte à extraire des fonctionnalités d'écriture manuscrite pertinentes que celles qui avaient été créées à la main pour le problème de la transcription automatique de texte. Les auteurs de [123] ont utilisé une configuration unique de CNN avec des paramètres spécifiques pour les trois problèmes démographiques considérés. Enfin, la méthode de prédiction proposée selon le sexe reste relativement robuste pour plus d'un alphabet.

La référence [124] a utilisé CNN comme classificateur d'apprentissage approfondi pour la reconnaissance de texte de scène arabe. Les auteurs supposent qu'une telle approche est plus appropriée dans les scripts cursifs. Ainsi, leur modèle avait été appliqué à l'apprentissage de modèles d'images visuelles dans lesquels un texte arabe était écrit. Les résultats expérimentaux indiquent que CNN peut améliorer la précision des jeux de données volumineux et variables.

Deuxième partie

Contributions

Chapitre 5

Réduction de dimension dans un environnement parallèle, distribué

Sommaire

5.1 Introduction

Les données recueillies aujourd'hui par les capteurs, augmentent rapidement et en particulier les données hyperspectrales, qui permettent de donner plus d'informations physiques sur la zone observée. L'image hyperspectrale est une image qui représente la même scène avec des centaines de bandes spectrales contiguës dans différentes gammes de longueurs d'onde. Les données d'une image hyperspectrale sont organisées sous la forme d'un cube à trois dimensions : deux dimensions, notées x et y, pour les dimensions spatiales et une dimension, notée z, pour la dimension spectrale (voir Fig 5.1) [125].

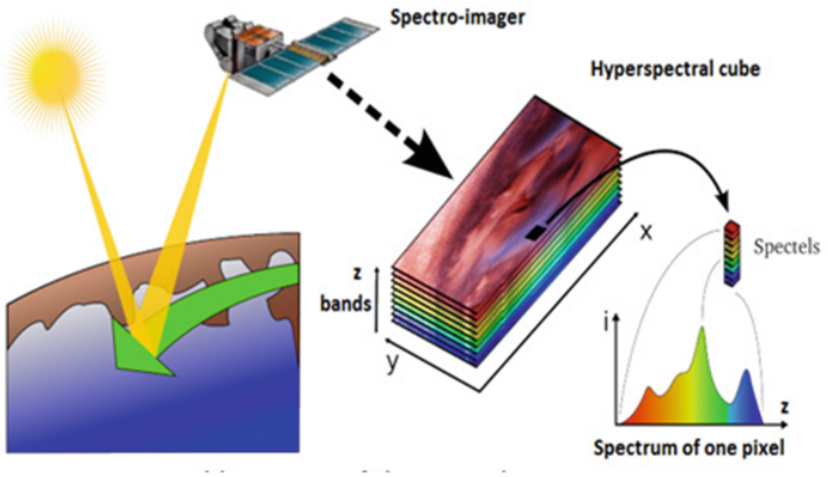


FIGURE 5.1 – Acquisition et décomposition d'une image hyperspectrale

On notera qu'il existe des images multispectrales composées d'une dizaine de bandes, tandis que l'image hyperspectrale dépasse une centaine de bandes, ce qui implique une exigence importante en termes de traitement et de stockage des données.

A la différence de l'image couleur classique, l'image Hyperspectrale donne plus d'informations physiques pour chaque objet observé. La technique de l'imagerie Hyperspectrale est utilisée dans plusieurs domaines : La géologie, l'agriculture, l'urbanisme, la foresterie et le domaine militaire.

Pour préparer l'image hyperspectrale à la visualisation ou pour une analyse plus poussée, comme la classification, il est nécessaire de réduire les dimensions de l'image à des dimensions analysables par les humains. Plusieurs techniques de réduction des dimensions existent. Nous trouvons des versions itératives et également parallèles [126].

Dans ce travail, nous avons proposé une version distribuée et parallèle de l'algorithme de réduction de dimension (**ACP**), qui a été testé sur la plateforme **Apache Spark** en utilisant le paradigme **MapReduce**.

5.2 Plateformes parallèles et distribuées

Afin de traiter les images hyperspectrales , nous avons utilisés des calculs parallélisés et distribués afin d'obtenir des résultats dans un délai raisonnable. Les plateformes, les plus reconnues qui effectuent ce type de traitement sont : **Apache Hadoop** (voir chapitre 2) et **Apache Spark** (voir chapitre 2).

5.3 Réduction de dimension

Maintenant, pour comprendre les informations cachées dans le cube de l'image hyperspectrale par les humains, ou extraire une partie utile de l'image, nous avons souvent recours à la visualisation. Cependant, la perception humaine étroite ne peut pas visualiser plus de 3 bandes hyperspectrales. Ainsi, avant de commencer la visualisation de notre image hyperspectrale, nous devons commencer par la réduction des bandes spectrales de l'image à 3, sans perdre la qualité de l'information. Au cours des dernières années, plusieurs techniques de réduction de la dimensionnalité ont été mises au point pour réduire les données hyperspectrales à un espace de dimension inférieure. Parmi les exemples importants, on trouve : ISOMAP, LLE [127], Laplacian eigenmap embedding, Hessian eigenmap embedding, conformal maps, diffusion maps et Principal Components Analysis (PCA)[128].

Dans cette contribution, nous avons utilisé ACP, la technique la plus populaire dans plusieurs domaines : la réduction de la dimensionnalité, le traitement d'image, la visualisation des données et la découverte des modèles cachés dans les données [129].

5.3.1 Version classique de l'algorithme : ACP

L'analyse des composantes principales est une technique de réduction des dimensions d'une matrice de données quantitatives. Cette méthode permet d'extraire les profils dominants de la matrice [130]. La description de la version classique de l'algorithme ACP est comme suite :

Nous supposons que notre image hyperspectrale est une matrice X de taille $(m = LxC, N)$ où L est le nombre de lignes dans l'image, C est le nombre de colonnes et N est le nombre de bandes avec $m \gg N$ (voir la figure 5.2)

$$X \begin{bmatrix} X11 & \cdots & X1N \\ \vdots & \ddots & \vdots \\ Xm1 & \cdots & XmN \end{bmatrix}$$

FIGURE 5.2 – Représentation matricielle d'une image Hyperspectrale

Chaque ligne de la matrice X représente le vecteur de pixel. Par exemple, le premier pixel est représenté par le vecteur : $[X_{11}, X_{12}, \ldots, X_{1N}]$, avec X_{1j} est la valeur du pixel 1 prise par le spectre numéro j.

Chaque colonne de la matrice X représente les valeurs de tous les pixels de l'image prise par un spectre. Par exemple $X_{i1} = [X11, X21, \ldots, Xm1]$ représente les données de l'image prise par le spectre numéro 1.

Pour appliquer l'algorithme ACP à l'image hyperspectrale X, on suit les étapes suivantes :

— Étape 1 : Calculer la matrice centrée réduite de X, notée : XRC

$$\text{XRC}_{ij} = \frac{\text{Xij} - \bar{X}j}{\sigma j} \text{ Pour chaque i} = 1..\text{m et Pour chaque j} = 1..\text{N (1)}$$

$$\text{Avec } \bar{X}j = \frac{1}{\text{m}} \sum_{\text{i}=1}^{\text{m}} \text{Xij et } \sigma j^2 = \frac{1}{\text{m}} \sum_{i=1}^{m} (Xij - \bar{X}j)^2$$

Dans la formule 1, $\bar{X}j$ désigne la moyenne de la colonne j et σj désigne l'écart type de la colonne j.

— Étape 2 : Calculer la matrice de corrélation de taille (N, N), notée : Xcorr.

$$\text{Xcorr} = \frac{1}{m}(XRC^T.XRC) \text{ (2)}$$

Dans la formule 2, $XRC^T.XRC$, désigne le produit matriciel entre le transposé de la matrice XRC et la matrice XRC.

— Étape 3 : Calculer les valeurs propres et le vecteur propre de la matrice Xcorr notés : $[\lambda, V]$

— Étape 4 : Trier le vecteur propre dans l'ordre décroissant des valeurs propres et prendre les k premières colonnes de V ($k < V$)

— Étape 5 : Projeter la matrice X sur le vecteur V : U = X.V

— Étape 6 : utiliser la nouvelle matrice U de taille (m, k) pour la visualisation de l'image hyperspectrale

5.3.2 Version distribuée et parallèle de l'algorithme : ACP

5.3.2.1 Travaux connexes

Il existe actuellement deux bibliothèques populaires qui fournissent une implémentation distribuée parallèle pour l'algorithme ACP : **MLlib** [131] sur spark et **Mahout** basé sur MapReduce [52]. Dans la bibliothèque **MLlib** de Spark, nous trouvons une implémentation pour l'ACP distribué parallèle, mais cette implémentation est faite avec les deux langages : Scala et Java. Aucune implémentation n'est faite pour le langage Python.

Dans [129], Tarek et al. ont montré que ces deux bibliothèques ne permettent pas une analyse parfaite d'une grande masse de données et ont proposé une nouvelle implémentation de l'ACP, appelée **sPCA**. L'algorithme sPCA a une meilleure précision et mise à l'échelle, que ses concurrents.

Dans [126], Zebin et al. ont proposé une nouvelle implémentation parallèle distribuée pour l'algorithme ACP. L'implémentation est faite à l'aide de la plateforme Spark et les résultats obtenus sont comparés avec une implémentation en série sur Matlab et avec une implémentation parallèle sur Hadoop. La comparaison montre l'efficacité de l'implémentation proposée en termes de précision et de temps de calcul.

Dans la suite, nous allons proposer une nouvelle implémentation pour l'algorithme ACP distribué parallèle, basé sur la plate-forme Apache Spark, en utilisant le langage de programmation **Python** et les matrices distribuées de la bibliothèque **Mllib** .

5.3.2.2 L'implémentation proposée :

Puisque l'image hyperspectrale est une représentation de la même scène avec plusieurs bandes spectrales, nous pouvons décomposer l'image hyperspectrale en plusieurs images, chaque image pour un spectre donné (voir Fig 5.3)

L'algorithme ACP classique nécessite un calcul intensif en raison de la grande taille de l'image hyperspectrale. Nous présenterons dans cette partie une implémentation parallèle distribuée de l'algorithme ACP et en utilisant la plate-forme Spark.

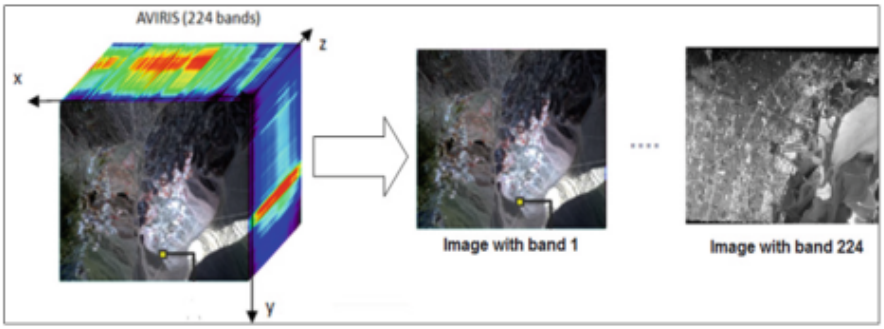


FIGURE 5.3 – Représentation de l'image hyperspectrale par plusieurs images

Premièrement, nous avons commencé par transformer la matrice X (voir Fig 5.4 5a) utilisée pour représenter l'image hyperspectrale dans l'ACP classique en un vecteur d'images M, où chaque colonne de X est représentée par une image en M (voir Fig 5.4 5b).

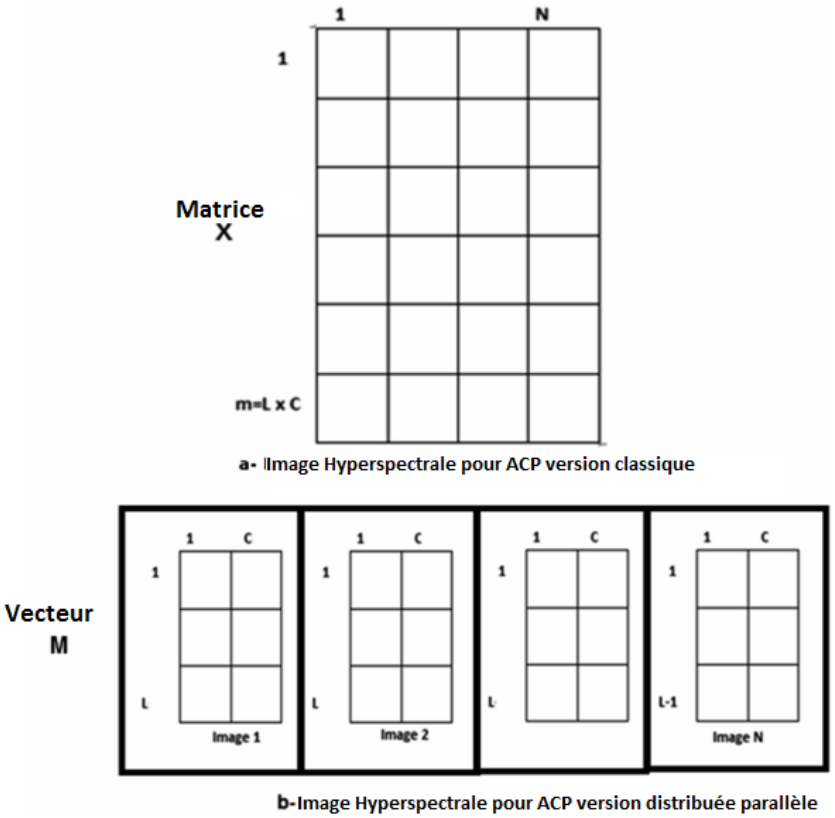


FIGURE 5.4 – Représentation de l'image hyperspectrale pour ACP classique et ACP distribué

Maintenant, chaque image t de M notée M_t est une matrice dans notre implémentation (Représente un RDD [132] dans la programmation distribuée parallèle de Spark).

Pour faire une implémentation parallèle distribuée de l'ACP, nous avons utilisé le paradigme Mapreduce de Spark. L'algorithme proposé procède comme suit :

— Étape 1 : Calculer la matrice centrée réduite de M :
Comme nous avons vu précédemment, la matrice M contient plusieurs images et chaque image M_t est représentée par une matrice (un RDD dans la notation Spark) de taille (L, C) où L est le nombre de lignes dans l'image et où C est le nombre de colonnes. Par conséquent, pour calculer la matrice centrée réduite de M, notée MCR, un calcul distribué parallèle est effectué pour chaque image M_t (Voir la description graphique de l'algorithme à la Fig. 5.5).

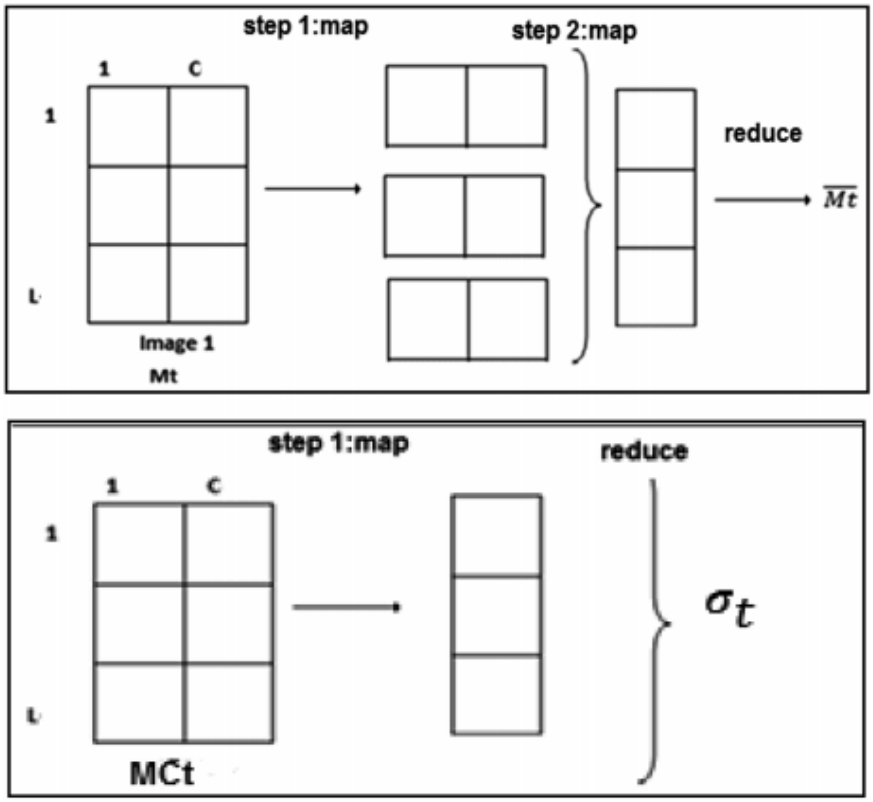


FIGURE 5.5 – Calcul de la moyenne de M_t et clacul de σ_t de MC_t avec Spark

- Calculer la matrice réduite de M notée MC :

$$MC_tij = M_tij - \bar{M}t \text{ Pour i allant de 1..L et Pour j allant 1..C (3)}$$

avec

$$\bar{M}_t = \sum_{i=1}^{L}(\sum_{j=1}^{C}(\frac{1}{LxC}xM_tij))$$

Dans la formule 3, M_t, indique la moyenne de l'image M_t

- Calculer la matrice centrée réduite de M, notée MCR :

$$MCR_tij = \frac{MC_tij}{\sigma_t} \text{ Pour i allant de 1..L et Pour j allant 1..C (4)}$$

$$\text{avec } \sigma_t^2 = \frac{1}{LxC}\sum_{i=1}^{L}(\sum_{j=1}^{C}(M_tij - \bar{M}_t)^2)$$

$$\text{ou } \sigma_t^2 = \frac{1}{LxC}\sum_{i=1}^{L}(\sum_{j=1}^{C}(MC_tij))$$

Dans la formule 4, σ_t, indique l'écart type de l'image t

— Étape 2 : Calculer la matrice de corrélation de MCR de taille (N, N), notée : **Mcorr**

Selon l'étape 1, le MCR est un vecteur d'images de taille N. Chaque image représente une matrice centrée réduite.

L'étape suivante consiste à calculer la matrice de corrélation de taille (N, N), en faisant le produit matriciel entre le vecteur MCR^T et le vecteur **MCR**, en utilisant le calcul parallèle distribué de Spark (Voir la description graphique de l'algorithme à la fig 5.6)

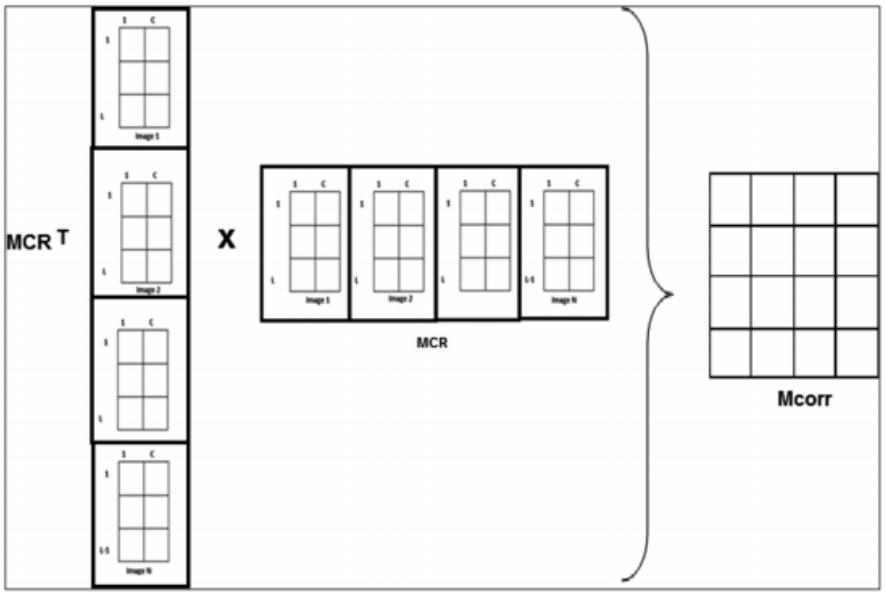


FIGURE 5.6 – Calcul de la matrice de corrélation avec Spark

$$Mcorr = \frac{1}{LxC}(MCR^T.MCR) \quad (5)$$

$$Mcorr_{t,k} = \frac{1}{LxC}(MCR_t.MCR_k) \quad (6)$$

Pour t allant de 1..N et Pour k allant 1..N

avec

$$MCR_t.MCR_k = \sum_{i=1}^{L}(\sum_{j=1}^{C}(MCR_tij.MCR_kij))$$

Pour calculer la valeur de chaque $Mcorr_{t,k}$ (Formule 6), l'image MCR_t est multipliée par l'image MCR_k pixel par pixel. Nous calculons ensuite la moyenne du résultat (voir la description graphique de l'algorithme à la fig. 5.7).

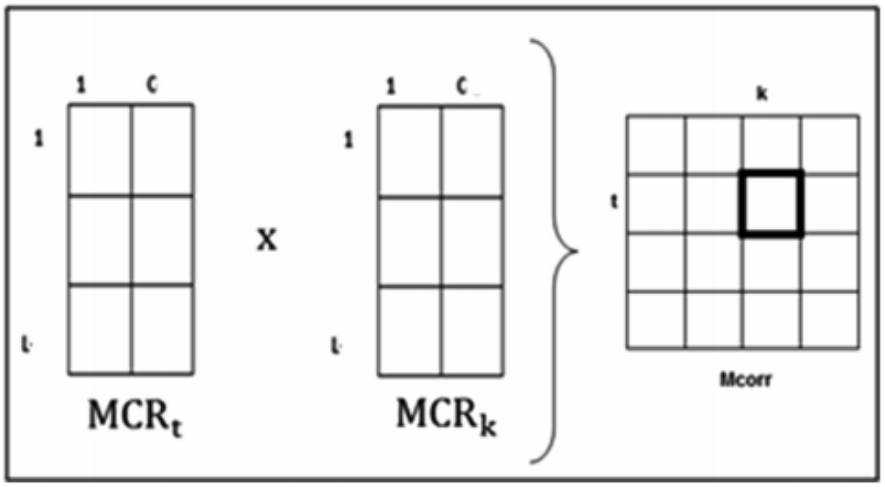


FIGURE 5.7 – Multiplication de deux images avec Spark

— Étape 3 : Calculer les valeurs propres et le vecteur propre de la matrice Mcorr : $[\lambda, V]$

— Étape 4 : Trier le vecteur propre dans l'ordre décroissant des valeurs propres et prendre les k premières colonnes de V ($k < V$)

— Étape 5 : Projeter la matrice X sur le vecteur V : U = X.V

— Étape 6 : utiliser la nouvelle matrice U de taille (m, k) pour la visualisation de l'image hyperspectrale

5.4 Expérimentations et calculs

Pour tester la validité de l'algorithme proposé sur les images hyperspectrales en utilisant la plateforme Apache Spark, nous avons choisi un ensemble d'images hyperspectrales ouvertes de différentes tailles (voir 5.1) et nous avons testé notre algorithme (version classique d'ACP), ACP classique de la bibliothèque Sklearn de Python et l'ACP parallèle distribué proposé sur ces ensembles de données.

TABLE 5.1 – Source de données Hyperspectrales

	Nom	**Dimension Spatiale**	**Nombre de bandes spectrales**	**Taille**
Dataset1	Moffett Field	500 x 500	3	5.3 MB
Dataset2	Moffett Field	500 x 500	10	17.5 MB
Dataset3	Moffett Field	1924 x 753	224	2.3 GB

L'image hyperspectrale utilisée dans nos expériences pour l'algorithme ACP classique ou pour l'algorithme proposé ACP distribué est l'image **Moffett Field AVIRIS** (Airborne Visible Infra-Red Imaging Spectromete) avec 224 bandes spectrales dans l'intervalle [2,5 nm à 400 nm]. L'image a été acquise le 20 août 1992 [85].

Les trois algorithmes utilisés dans la partie expérimentation sont implémentés en Python 3 et sont exécutés sur plusieurs configurations (voir 5.2) et les résultats d'application de l'ACP sur les images hyperspectrales sont donnés dans le tableau 5.3.

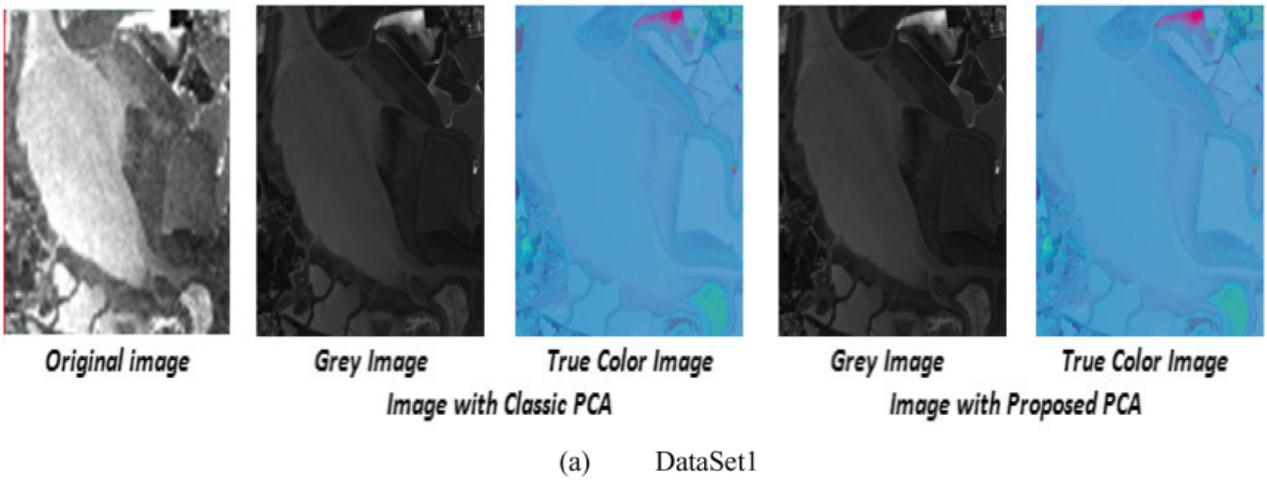
TABLE 5.2 – Paramètres de configuration

version classique : ACP	**version distribuée : ACP**
	Cluster Spark :
	Master Node : 1
- CPU :Intel Core I5, 3.3 GHZ	Slave Nodes : 4
- RAM : 4G	CPU of each node :Intel Core I5, 3.3 GHZ
- OS :Ubuntu 16.04 LTS	RAM of each node : 4G
	Network speed : 100 MB/s
	OS :Ubuntu 16.04 LTS

TABLE 5.3 – Les trois valeurs propres les plus significatives de l'ACP

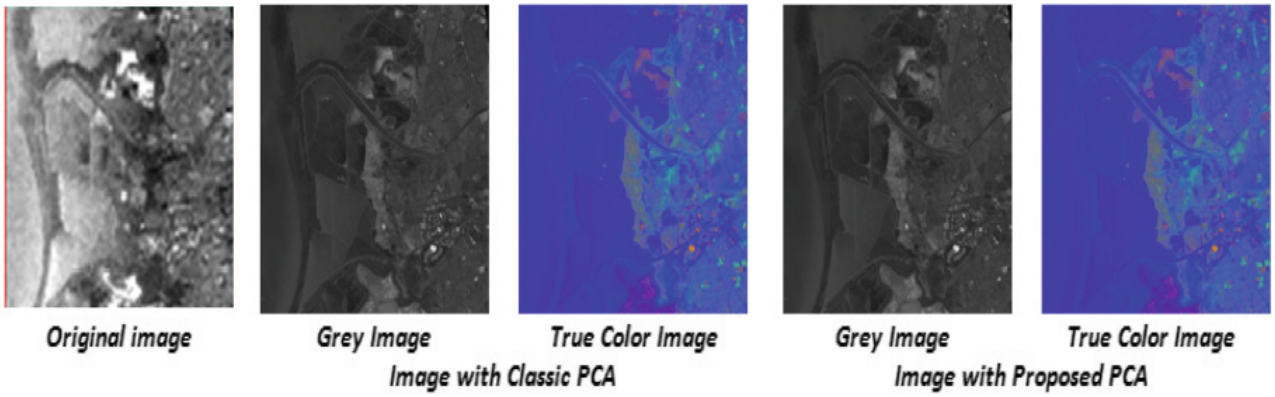
	ACP classique	**ACP de Sklearn**	**ACP distribué**
Dataset1	1.9371026343	1.9371026343	1.9371026343
	0.913755533084	0.913755533084	0.913755533084
	0.149141832615	0.149141832615	0.149141832615
Dataset2	8.12709201824	8.12709201825	8.12709201825
	0.880534232525	0.880534232525	0.880534232525
	0.797956006441	0.797956006442	0.797956006442
Dataset3	160.638762642	160.638762642	160.638762642
	28.0031335174	28.0031335174	28.0031335174
	14.5639033111	14.5639033111	14.5639033111

La visualisation de l'image hyperspectrale après l'application de l'algorithme ACP classique ou de l'algorithme proposé ACP distribué est donnée à la Fig 5.8, Fig5.9 et la Fig 5.10



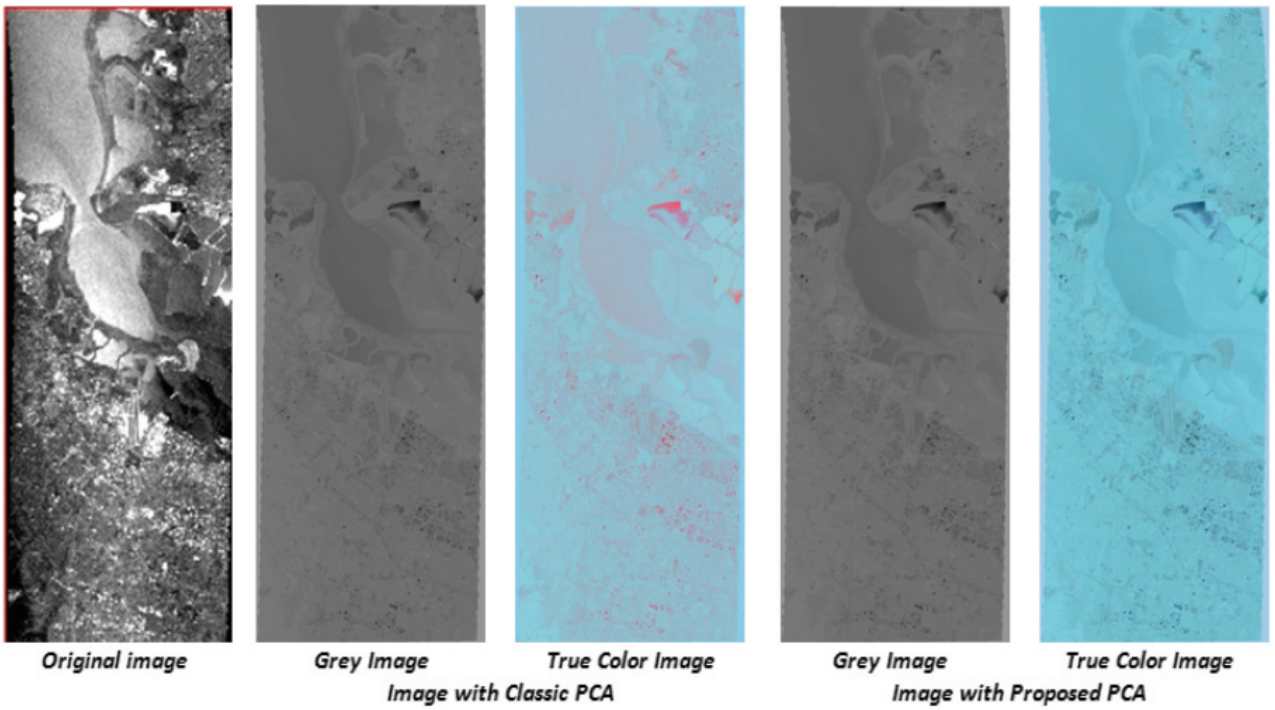
(a) DataSet1

FIGURE 5.8 – Visualisation de l'image Hyperspectrale (DataSet1), avant et après l'application de l'ACP classique (ACP de la bibliothèque Sklearn) et l'ACP distribué proposé



(b) DataSet2

FIGURE 5.9 – Visualisation de l'image Hyperspectrale (DataSet2), avant et après l'application de l'ACP classique (ACP de la bibliothèque Sklearn) et l'ACP distribué proposé



(c) DataSet3

FIGURE 5.10 – Visualisation de l'image Hyperspectrale (DataSet3), avant et après l'application de l'ACP classique (ACP de la bibliothèque Sklearn) et l'ACP distribué proposé

5.5 Conclusion

Dans ce travail, nous avons proposé un algorithme parallèle et distribué pour la réduction de la dimensionnalité appelé ACP. L'algorithme est développé en Python 3 et testé sur des images hyperspectrales en utilisant la plateforme Spark. Les résultats coïncident avec les résultats de l'ACP classique et la visualisation des images après l'application de notre algorithme de réduction confirme la validité de notre algorithme.

Dans la contribution suivante, on va confirmer la performance de l'algorithme proposé **ACP distribué**, par rapport à l'algorithme classique, par la comparaison du temps d'exécution de chaque méthode.

Chapitre 6

Visualisation de données : Application aux images Hyperspectrales

Sommaire

6.1 Introduction

Actuellement, les dispositifs d'affichage numérique produisent une image couleur pour l'œil humain en utilisant une combinaison de trois couleurs primitives. Ainsi, une image couleur RVB (en anglais RGB) classique est une combinaison de trois couches (ou bandes) : Rouge, vert et bleu. Une image couleur TSL classique (en anglais HSL) est une combinaison de 3 couches : Teinte, Saturation et Luminosité. Au contraire, on trouve des images hyperspectrales composées de centaines de couches (bandes) (voir chapitre 3). L'imagerie hyperspectrale est souvent utilisée dans le domaine de la télédétection et de l'imagerie médicale multimodale. En astronomie, par exemple, l'imagerie hyperspectrale est utilisée pour archiver les observations du sol et de l'espace. En imagerie médicale, l'imagerie hyperspectrale est utilisée pour la détection de maladies telles que le Cancer [133]. Maintenant, comment visualiser un cube hyperspectral et donner à l'utilisateur, généralement pas spécialiste, une vue synthétique des données contenues dans l'image avec le minimum de perte possible, et faciliter l'interprétation de l'image ? Parmi les premières solutions proposées c'est de visualiser le cube sous la forme d'une séquence vidéo, chaque couche du cube est représentée par une image. Cependant, lorsque nous travaillons dans le plan et avec beaucoup d'images hyperspectrales de grandes dimensions spectrales, cette solution reste difficile à mettre en pratique. Donc pour visualiser une image hyperspectrale en couleur et dans le plan avec le nombre de bandes spectrales qui dépasse trois bandes, il est souvent nécessaire d'obtenir, à partir de l'image originale, une image composite qui se compose seulement de trois bandes.

Plusieurs méthodes de visualisation d'images hyperspectrales existent : des méthodes basées sur la sélection de bandes spectrales ([134], [135], [136]), des méthodes basées sur la pondération ([137]), des méthodes basées sur l'optimisation ([138]) et des méthodes basées sur la projection ([139])

Pour contourner le problème de calcul posé par le traitement du grand cube hyperspectral, nous avons utilisé une plateforme open source nommée Apache Spark [140], qui distribue le stockage de données en mémoire vive (RAM) et qui traite les données en parallèle. Ce choix nous a donné un gain considérable dans le temps de visualisation d'une image hyperspectrale.

6.2 Travaux liés aux méthodes de visualisation d'images hyperspectrales

Dans la littérature, nous avons trouvé quatre méthodes utilisées pour la visualisation d'une image hyperspectrale [141]

— **Méthode basée sur la sélection des bandes** ([134], [135], [136]) :
Pour visualiser une image hyperspectrale dans un système de représentation RVB, trois bandes spectrales doivent être sélectionnées à partir de l'image hypespectrale originale composée de centaines de bandes. Ensuite, chaque bande sera affectée à une couleur : rouge, vert et bleu. Ce type de méthode de visualisation est utilisé dans le navigateur AVIRIS ("AVIRIS - Airborne Visible / Infrared Imaging Spectrometer "). La visualisation avec cette méthode est rapide, mais nous prenons, juste les données existantes dans les trois bandes sélectionnées. Les données des autres bandes seront ignorées. Ainsi, une grande quantité d'informations existantes dans l'image est perdue

— **Méthode basée sur la pondération** ([137]) :
Cette méthode fournit une image résultante d'une combinaison linéaire des bandes d'image d'entrée. Dans cette méthode, on trouve deux types : la méthode basée sur les CMFs étirés (Color Matching Functions) et la méthode basée sur le filtrage bilatéral. L'avantage de cette méthode est l'utilisation de toutes les bandes de l'image, mais le problème se pose dans le choix du même poids qui sera attribué aux pixels de l'image en ignorant la variété de pixels.

— **Méthode basée sur l'optimisation** ([138]) :
Dans cette méthode, certaines fonctions sont appliquées à l'image selon un critère optimisé. Nous trouvons : la méthode basée sur le champ aléatoire de Markov et la méthode basée sur des objectifs de maximisation multiples. Le grand défi pour cette méthode est de trouver la bonne fonction pour l'appliquer à l'image

— **Méthode basée sur la transformation** :
Avec cette méthode, nous pouvons visualiser une image hyperspectrale en projetant l'image originale sur une dimension plus petite (trois par exemple).
Au cours des dernières années, plusieurs techniques de réduction de la dimensionnalité ont été mises au point pour réduire les données hyperspectrales à un espace de dimension inférieure, parmi les exemples importants on trouve : ISOMAP, LLE, Laplacian eigenmap embedding, Hessian eigenmap embedding, cartes de conformité, cartes de diffusion, analyse indépendante des composantes (ICA) et analyse des composantes principales (ACP) ([128]).

Dans cette contribution, nous avons utilisé l'algorithme ACP de la dernière méthode de visualisation. L'ACP fait partie des algorithmes de réduction des dimensions qui peuvent être implémentés efficacement et qui sont utilisés avec succès dans les applications commerciales de télédétection ([142]). Puisque nous visualisons une grande image hyperspectrale, PCA prend beaucoup de temps de calcul. Donc, pour résoudre ce problème, nous avons utilisé un calcul distribué et parallèle (voir chapitre 5) .

6.3 Détails expérimentaux

Pour tester l'algorithme proposé, l'image gratuite « Moffett » d'AVIRIS a été utilisée avec 224 bandes spectrales dans l'intervalle de 2,5 nanomètres à 400 nanomètres. Sur cette image hyperspectrale, nous avons prélevé des échantillons de différentes tailles (voir la table 6.1).

TABLE 6.1 – DATASETS

	Nom	**dimensions spatiales**	***bandes hyperspectrales***
Dataset1	Moffett Field	500 x 500	3
Dataset2	Moffett Field	500 x 500	10
Dataset3	Moffett Field	1924 x 753	3
Dataset4	Moffett Field	1924 x 753	10
Dataset5	Moffett Field	1924 x 753	15
Dataset6	Moffett Field	1924 x 753	20
Dataset7	Moffett Field	1924 x 753	25
Dataset8	Moffett Field	1924 x 753	50
Dataset9	Moffett Field	1924 x 753	75
Dataset10	Moffett Field	1924 x 753	100
Dataset11	Moffett Field	1924 x 753	150
Dataset12	Moffett Field	1924 x 753	224

Sur chaque échantillon obtenu, nous avons testé l'algorithme parallèle distribué proposé et une implémentation en série de l'ACP classique à partir de la bibliothèque scikit-learn de Python. Nous avons collecté les trois valeurs propres les plus significatives (voir la table 6.2) et le temps d'exécution de chaque algorithme. (voir la table 6.3).

TABLE 6.2 – Les trois valeurs propres les plus significatives de l'ACP

	Sklearn ACP	***ACP Proposé***
Dataset1	1.9371026343 0.913755533084 0.149141832615	1.93710263 0.9137555 0.14914183
Dataset2	8.12709201825 0.880534232525 0.797956006442	8.12709202e+00, 8.80534233e-01, 7.97956006e-01
Dataset3	2.83836278 0.15547085 0.00616637	2.83836278, 0.15547085, 0.00616637
Dataset4	7.95653886 1.32405628 0.52304295	7.95653886, 1.32405628, 0.52304295
Dataset5	10.06478855 4.01771578 0.5480712	10.06478855, 4.01771578, 0.5480712
Dataset6	13.03722313 4.41080037 0.89920242	13.03722313, 4.41080037, 0.89920242
Dataset7	17.3548134106 4.52921603811 1.9900193197	17.3548134106, 4.52921603811, 1.9900193197
Dataset8	37.556473304 5.65857334764 2.76144687169	37.5564733 , 5.65857335, 2.76144687
Dataset9	51.24006993 19.21195598 2.92742233	51.24006993, 19.21195598, 2.92742233
Dataset10	70.07730225 24.23284534 2.99508837	70.07730225, 24.23284534, 2.99508837
Dataset11	105.36058484 25.29413561 6.79161627	105.36058484, 25.29413561, 6.79161627
Dataset12	160.63876264 28.00313352 14.56390331	160.63876264, 28.00313352 , 14.56390331

TABLE 6.3 – Le temps d'exécution de l'ACP en (s)

	ACP Sklearn	**ACP proposé**
Dataset1	0.0873646736	0.421477079
Dataset2	0.532052755	0.695195913
Dataset3	0.840823889	1.762059927

L'ACP classique de la bibliothèque scikit-learn est testé sur un ordinateur équipé de : CPU : Intel® Core ™ i7-2820QM CPU @ 2.30GHz 8, RAM : 8G, OS : Ubuntu 16.04 LTS. L'algorithme parallèle distribué proposé a été testé sur le Cloud Databricks [143] de la configuration : (voir la table 6.4). Les deux algorithmes sont programmés avec le langage Python.

TABLE 6.4 – Paramètres de configuration pour le cluster spark dans le cloud databricks

Driver	Nodes
Driver type : 36 GB Memory, 8 Cores	Number of Worker Nodes : 6 For each worker :8.0 GB Memory, 2 Core

La comparaison de temps d'exécution des deux algorithmes montre la rapidité de l'ACP sklearn pour les petites images, mais si l'image a un nombre élevé de bandes, plus de 10 bandes spectrales, notre ACP est plus rapide (voir Fig. 6.1, Fig. 6.2 et Fig.6.3).

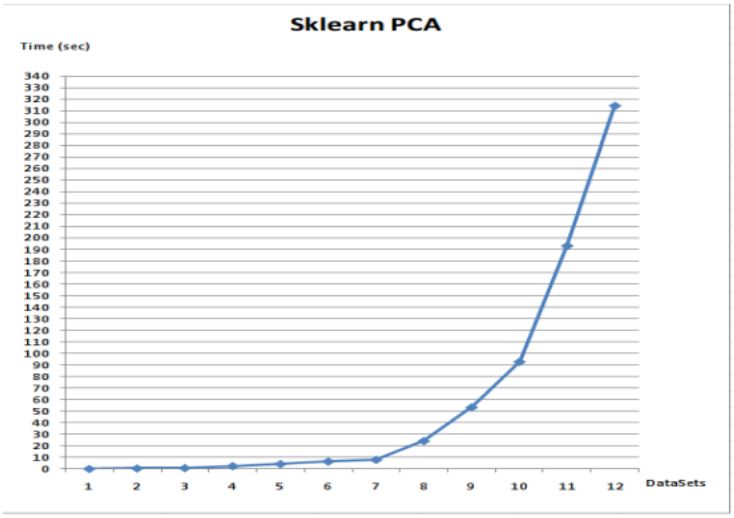


FIGURE 6.1 – Le temps d'exécution pour l'ACP Sklearn

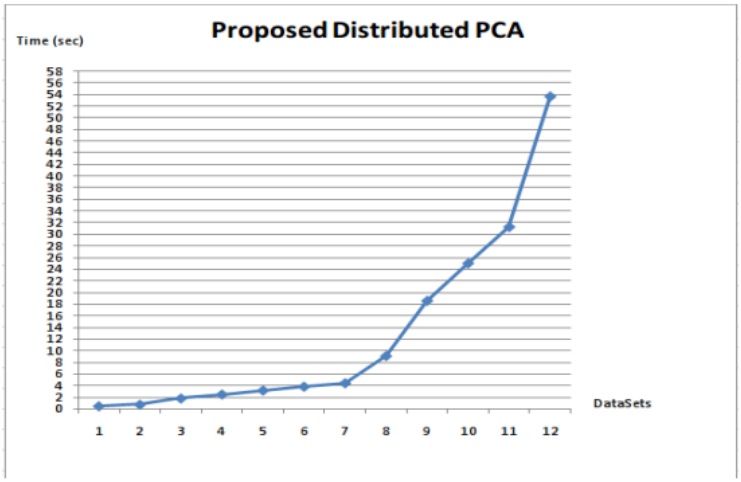


FIGURE 6.2 – Le temps d'exécution pour l'ACP proposé

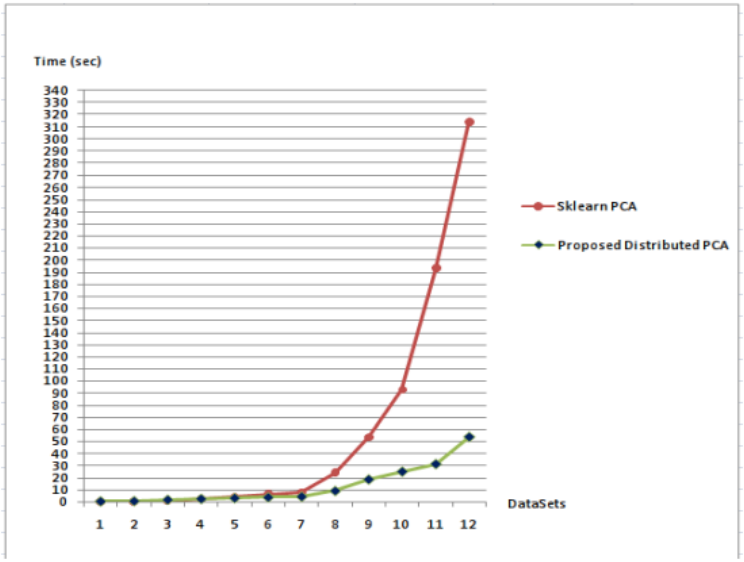


FIGURE 6.3 – La comparaison du temps d'exécution entre lACP Sklearn et le PCA proposé

La visualisation de l'image hyperspectrale après l'application de l'algorithme classique ACP ou de l'algorithme distribué ACP proposé est donnée à la figure 6.

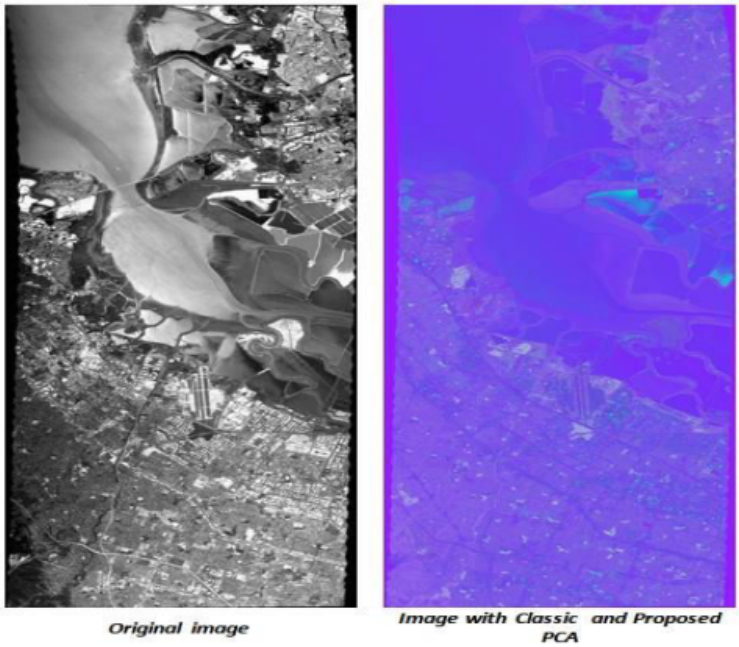


FIGURE 6.4 – Visualisation de l'image Dataset12, avant et après l'application de l'ACP classique et de l'ACP proposé

6.4 Conclusion

Dans ce travail, une méthode de visualisation d'une image hyperspectrale a été proposée sur la base de la réduction de la dimensionnalité de l'image dans un environnement parallèle distribué. L'algorithme est développé en Python 3 et testé sur des images hyperspectrales en utilisant la plateforme Spark. Les résultats coïncident avec les résultats de PCA classique et la visualisation des images après l'application de notre algorithme de réduction confirme la validité de notre algorithme. Le temps d'exécution des deux algorithmes montre que le PCA proposé est plus rapide pour les grandes images.

Chapitre 7

Classification par les réseaux de neurones convolutifs :Application aux images Hyperspectrales

Sommaire

7.1 Introduction

Parmi les méthodes qui permettent aux utilisateurs de rendre les données de l'image hyperspectrale utilisables, et d'extraire le maximum d'informations utiles, on trouve la classification. La classification est une opération qui divise un ensemble d'individus en plusieurs classes, et chaque classe regroupe les individus qui partagent la même similarité. Il existe deux familles d'algorithmes de classification : classification non supervisée (USVC) et classification supervisée (SVC).

Dans l'USVC, nous avons des éléments non classifiés et des classes inconnues, et nous essayons de regrouper les éléments qui ont une certaine similarité entre eux pour construire un ensemble de classes.

Dans le SVC, les classes sont connues à l'avance, nous avons des exemples sur chaque classe et nous essayons d'attribuer de nouveaux éléments à ces classes. Exemples d'algorithmes SVC : Naïve Bayes [144], Support Vector Machine (SVM), Deep learning [145] (Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM)).

Ces dernières années, le domaine du SVC et surtout de l'apprentissage profond a connu une grande évolution. En outre, la classification des images hyperspectrales par des algorithmes supervisés [146] a donné une précision supérieure à celle des algorithmes USVC.

Ce chapitre étudie un algorithme de classification de type "apprentissage profond" appelé CNN pour classifier le contenu d'un ensemble d'images hyperspectrales à l'aide d'un seul entrainement. L'objectif de la classification est de regrouper dans chaque classe les pixels qui ont une certaine similarité (propriétés communes) : eau, végétation, sable.

7.2 Réseau de neurones Convolutif

Dans cette section, on va décrire le CNN [147], le réseau d'apprentissage en profondeur supervisé le plus populaire, et qui a montré sa puissance dans l'extraction des caractéristiques, dans les applications de vision par ordinateur.

7.2.1 Les opérations standard dans un CNN

Le réseau de neurones convolutif (**CNN** ou **ConvNet**) est un type particulier et important des réseaux de neurones **feed-forward** (l'information se propage de couche en couche, sans possibilité de retourner en arrière). Il est inspiré par les processus biologiques qui se produisent dans le cortex visuel dans le cerveau des êtres vivants. Les modèles CNN sont construits sur le même modèle que les perceptrons multicouches dont nous trouvons : une couche d'entrée, plusieurs couches intermédiaires cachées (selon la profondeur du modèle) et une couche de sortie. CNN est utilisé pour résoudre plusieurs problèmes de vision informatique dans l'intelligence artificielle, par exemple : voitures autonomes, traitement vidéo et classification des images. Les opérations de base dans un réseau CNN standard sont les suivantes :

— **Opération de Convolution :**
L'opération de convolution est l'opération de base dans la construction d'un réseau CNN. Elle permet de faire glisser, pas à pas, une fenêtre nommée noyau (en anglais kernel) sur l'image entière, et pour chaque étape, on multiplie les pixels du noyau par

les pixels de la région sur laquelle elle glisse. Puis nous prenons la somme du résultat (voir Fig.7.1).

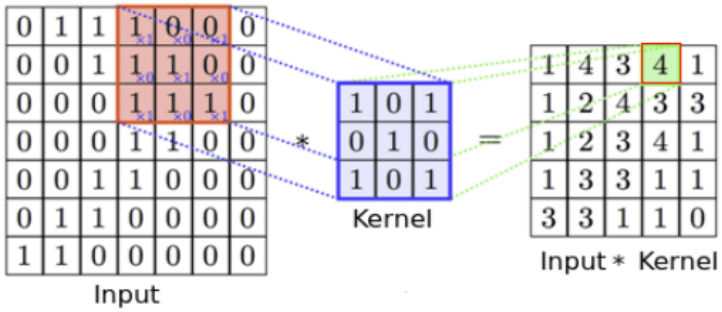


FIGURE 7.1 – Description de l'opération de convolution

— **Opération Max-Pooling :**
L'opération Max-pooling permet de glisser, pas à pas, une fenêtre, généralement de taille 2x2, sur l'image entière, et prend à chaque étape, la valeur maximale de la fenêtre. C'est une opération optionnelle dans la conception du réseau. En général, dans les architectures classiques de CNN, elle est mise après chaque opération de convolution et vise à réduire le nombre d'échantillons ou de neurones. Si la taille de la fenêtre est grande, nous risquons de perdre les informations de l'image (voir Fig. 7.2).

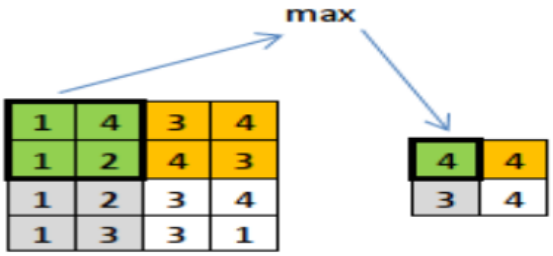


FIGURE 7.2 – Description de l'opération de Max-Pooling

— **Fonctions d'activation :**
Il s'agit de fonctions de correction qui jouent un rôle important dans les algorithmes d'apprentissage en profondeur. La fonction d'activation prend en entrée une valeur x et retourne la sortie f(x). Les fonctions d'activation sont généralement utilisées après chaque opération de convolution. Les célèbres fonctions d'activation sont :

— **Identity** :

$$f(x) = x$$

— **Binary Step** :

$$f(x) = 0 \quad if \quad x < 0; \quad else \quad f(x) = 1$$

— **Logistic or sigmoid** :

$$f(x) = \frac{1}{1 + e^{-}x}$$

— **Tanh** :

$$f(x) = tanh(x)$$

— **Rectified Linear Unit (ReLU)** :

$$f(x) = 0 \quad if \quad x < 0; \quad else \quad f(x) = x$$

— **Dropout :**
L'opération Dropout, permet de désactiver aléatoirement les sorties de certains neurones avec une probabilité prédéfinie (0,5 par exemple). Et ceci pour simuler la fonctionnalité réelle des neurones, qui peuvent dans une itération de la phase d'apprentissage, être inactifs.

— **Fully connected (FC) :**
Après plusieurs opérations de Convolution et de Max-Pooling, on trouve ces opérations pour connecter tous les neurones de la couche précédente (quel que soit leur type), avec les neurones de la couche suivante. Il n'est pas obligatoire d'avoir des FC dans un CNN, mais souvent on trouve deux couches FC consécutives comme couches finales dans le réseau.

7.2.2 Travaux connexes

Le premier modèle de classification CNN est nommé LeNet-5 [**lecun1998gradient**]] Il a été proposé par LeCun et al en 1998 pour classifier les nombres écrits à la main. Le modèle est composé de 7 couches (sans compter la couche d'entrée). D'autres modèles de classification sont apparus et contiennent une variété de nombres de couches : AlexNet [148] en 2012 avec 9 couches, ZFNet [148] en 2013 avec 8 couches, GoogleNet [149] en 2014 avec 22 couches, VGGNet [148] en 2014 avec 19 couches et ResNet [148] en 2015 avec jusqu'à 269 couches.

Dans la classification des images hyperspectrales à l'aide de CNN, nous trouvons : des modèles de classification spectrale [150], [151], des modèles de classification spatiale 2D [151], des modèles de classification spatiale 3D [152], [151] et des modèles de classification hybrides [**chen2016deep**]].

La plupart de ces modèles de classification mesurent la performance par la précision globale (en anglais Overall Accuracy (AO)), et ces AO sont presque égaux. Certains travaux utilisent la vitesse d'apprentissage, qui est un critère crucial pour choisir un modèle lorsqu'on travaille sur deux modèles qui ont presque la même précision. Parmi ces travaux, on peut parenciter :

— Dans [150], les auteurs ont proposé un algorithme de classification CNN basé sur les caractéristiques spectrales de l'image hyperspectrale, et qui contient 5 couches. Les résultats (Précision, Temps d'entraînement et le temps de teste) sont comparés avec LeNet-5, DNN et RBF-SVM.

— Dans [153], les auteurs ont proposé un modèle de classification CNN, basé sur deux canaux : le premier canal 1D pour extraire les caractéristiques spectrales et le second canal 2D pour extraire les caractéristiques spatiales. Les résultats des deux canaux sont combinés par le classificateur Softmax. Le temps d'entrainement du modèle est comparé à celui d'un autre modèle de l'état de l'art (SSDCNN [154], SSDL [155]).

— Dans [146], les auteurs ont proposé un modèle de classification 3D de 5 couches, qui utilise en même temps les caractéristiques spectrales et spatiales de l'image. Le modèle est implémenté à l'aide des unités de traitement graphique (GPU) [151]. Les résultats

(Précision et le temps d'entrainement) sont comparés avec le modèle MLP classique et un modèle CNN de l'état de l'art.

7.3 Architecture du modèle proposé

Les algorithmes de classification des images hyperspectrales de l'état de l'art [153]-[156]-[157], fonctionnent selon le principe suivant :

Objectif : classifier les pixels d'une image hyperspectrale X, selon un certain nombre de classes C.

- Étape 1 : Diviser l'image X en deux groupes de données : X_train, pour entrainer le modèle et X_test pour valider le modèle. Créer ensuite un modèle de classification basé sur les paramètres de l'image X (nombre de lignes, nombre de colonnes et nombre de bandes spectrales).
- Etape 2 : Entrainer le modèle ainsi créé, sur les données X_train, et enregistrer le temps pris dans cette étape (noté t_train).
- Etape 3 : Tester la validité du modèle créé, sur les données X_test.
- Etape 4 : Faire la classification de l'image entière X avec le modèle créé, et enregistrer le temps passé dans cette étape (noté t_pred). Nous constatons que le temps d'entrainement du modèle (t_train) est beaucoup plus long que le temps de prédiction (t_pred).
- Étape 5 : Visualiser le résultat.

Bien que ce principe de classification soit utilisé dans presque tous les algorithmes de classification des images hyperspectrales basés sur CNN, il présente plusieurs défauts : si on veut classifier deux nouvelles images Y et Z, nous devons répéter les mêmes étapes de 1 à 5 pour l'image Y et aussi pour l'image Z.

Cette méthode classique de classification prend beaucoup de temps [152], [156], causée par la répétition de l'étape d'entrainement pour chaque image, surtout lorsqu'on travaille avec un grand nombre d'images.

Dans cette contribution, nous avons proposé un algorithme de classification spectrale d'une image hyperspectrale composée de plusieurs HSI, basé sur le CNN, et qui utilise un seul entrainement.

L'algorithme proposé de classification procède comme suit :

Objectif : Classifier les pixels d'une image hyperspectrale X1 selon un certain nombre de classes, noté C1.

- Étape 1 : Prendre k images hyperspectrales de différentes tailles : $X1(H1, W1, N1, C1)$, $X2(H2, W2, N2, C2)$,. ,$Xk(Hk, Wk, Nk, Ck)$, avec Hi, Wi, Ni, Ci représente la hauteur, la largeur, le nombre de bandes spectrales et le nombre de classes pour l'image i (respectivement).
- Étape 2 : Choisir le nombre minimum de bandes entre les k images : $N = min(N1, N2, \ldots, Nk)$.
- Étape 3 : Appliquer l'algorithme de réduction de la dimensionnalité ACP, sur chaque image i de nombre de bandes Ni> N
- Étape 4 : Verticalement, concaténer les images obtenues, pour avoir une seule image X de caractéristiques suivantes :

- Le nombre de pixels : $m = \sum_{i=1}^{k}(Hi.Wi)$
- Le nombre de bandes : $N = min(N1, N2, \ldots, Nk)$
- Le nombre de classes : $C = \sum_{i=1}^{k}(Ci)$

— Étape 5 : Diviser l'image X en deux groupes de données : X_train pour l'entrainement et X_test pour la prédiction. Créer ensuite un modèle de classification spectrale basé sur les paramètres de l'image X.

— Étape 6 : Entrainer le modèle ainsi créé sur les données X_train, et noter le temps (t_train) pris dans cette étape.

— Étape 7 : Tester la validité du modèle créé sur les données X_test.

— Étape 8 : Maintenant, nous pouvons utiliser le modèle crée pour faire la prédiction sur chaque image Xi : $X1, X2 \ldots, Xk$ séparément, et nous notons le temps (t_pred) pris dans cette étape.

— Étape 9 : Visualiser l'image classifiée Xi

7.3.1 Réduction de la dimension avec ACP

La première étape de l'algorithme proposé est de prendre des images de tailles différentes, chaque image Xi de taille (Hi, Wi, Ni), sera convertie vers le format matriciel noté Mi, de taille (le nombre de lignes $Li = HixWi$ et le nombre de colonnes Ni). Chaque colonne j $()1 \leq j \leq Ni$) de la matrice Mi contient les pixels de l'image Xi pour la longueur d'onde j et chaque ligne k ($1 \leq k \leq Li$)de l'image Mi représente les valeurs d'un pixel k de Xi pour toutes les longueurs d'onde (voir Fig. 7.3).

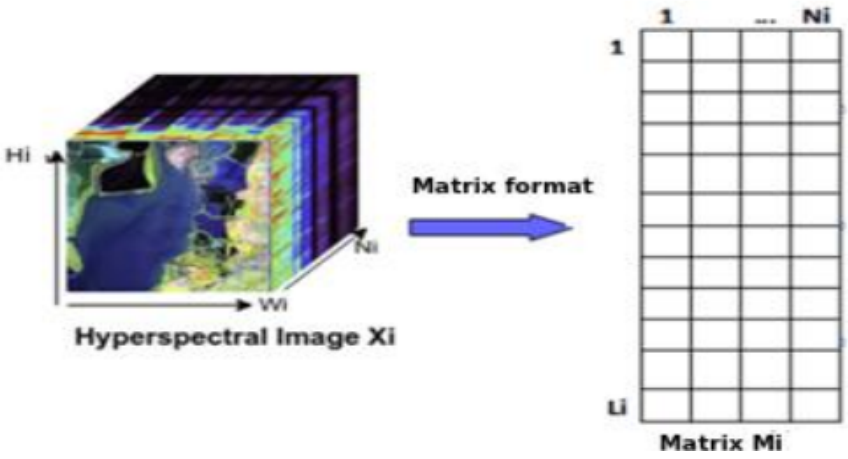


FIGURE 7.3 – Représentation matricielle de l'image hyperspectrale Xi

D'après les étapes 2 et 3, nous devons calculer le minimum de bandes entre les images hyperspectrales que nous utiliserons : $N = min(N1, N2, \ldots, Nk)$, avec k le nombre d'images. L'algorithme de réduction ACP [158] est ensuite appliqué à chaque image Mi, et les images réduites sont concaténées pour obtenir l'image M. Par exemple, à la Fig. 7.4, on trouve une illustration de l'algorithme sur les deux images hyperspectrales (k = 2) : Pavia University et Salinas.

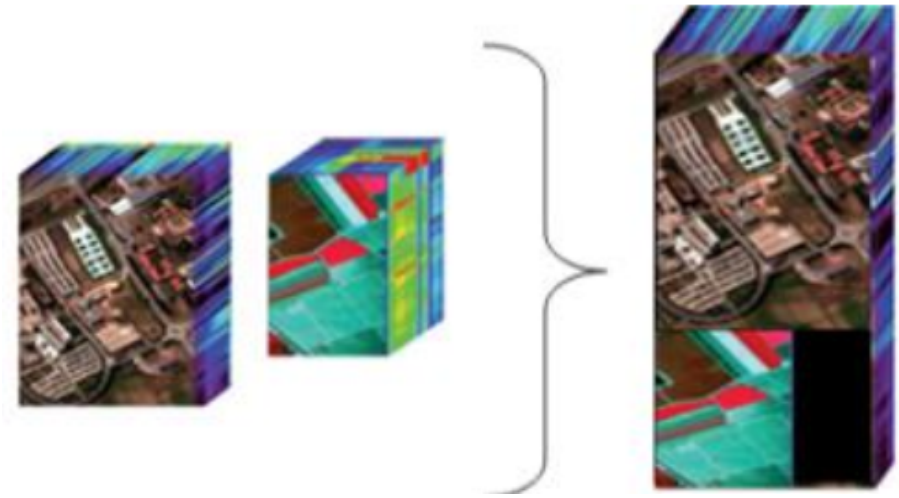


FIGURE 7.4 – Réduction de la dimensionnalité des images et concaténation pour obtenir une seule image M.

Par la suite, nous proposerons un algorithme de classification spectrale CNN, inspiré du papier [150] et qui sera utilisé pour la classification des images séparées (comme l'état de l'art) et aussi pour tester l'algorithme proposé.

7.3.2 Classification avec le CNN spectral

Pour classifier les pixels d'un HSI, nous avons proposé un modèle composé de 10 couches : 1 couche d'entrée, 3 couches de convolution, 3 couches de Max-Pooling, une couche Dropout, une couche FC et une couche de sortie, avec la configuration suivante (voir Fig. 7.5).

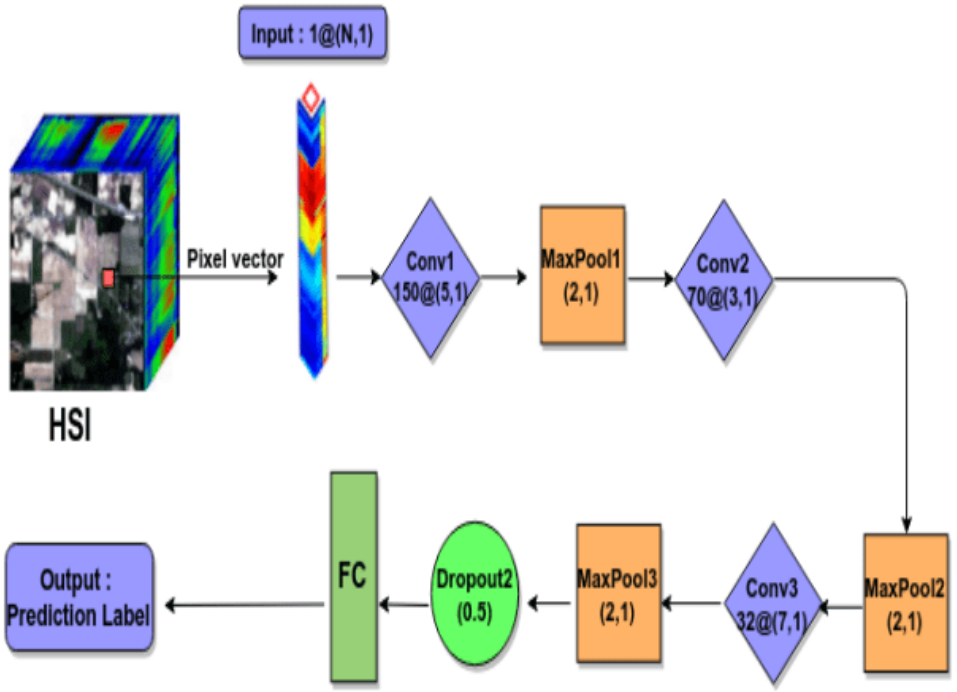


FIGURE 7.5 – Architecture du modèle proposé de classification CNN

Le modèle prend comme entrée un vecteur de pixel de taille N (nombre de bandes), nous avons appliqué sur le vecteur de pixel, diverses opérations : convolution, Max-Pooling, Dropout et Fully Connected Layer selon les paramètres suivants : (voir la table 7.1).

TABLE 7.1 – Paramètres du modèle proposé

	Conv1	Conv2	**Conv3**
Number of filters	150	70	32
Kernel size	5 x 1	3 x 1	7 x 1
	Max-Pool 1	**Max-Pool 2**	**Max-Pool 3**
Kernel size	2 x 1	2 x 1	2 x 1

7.4 Détails expérimentaux

7.4.1 Jeux de données :(DataSets)

Pour classifier les images hyperspectrales à l'aide de l'algorithme proposé, nous avons utilisé deux ensembles de données libres : Pavia University et Salinas. Pour les deux ensembles de données, nous avons pris 70% de pixels pour entrainer le modèle et 30% pour tester le modèle de classification.

— **Pavia :**

Il existe deux types de données Pavia :"Pavia Center" et "Pavia University". Dans cette expérience, nous avons utilisé "corrected Pavia University" [159] , qui représente la scène de Pavia, dans le nord de l'Italie, capturée par le capteur ROSIS (Reflective Optics System Imaging Spectrometer) en 2001. La scène a une dimension spatiale de 610 x 340 pixels avec 103 bandes de réflectance spectrale dans la plage d'onde de 0,43 à 0,86 μm. La scène contient 9 classes.

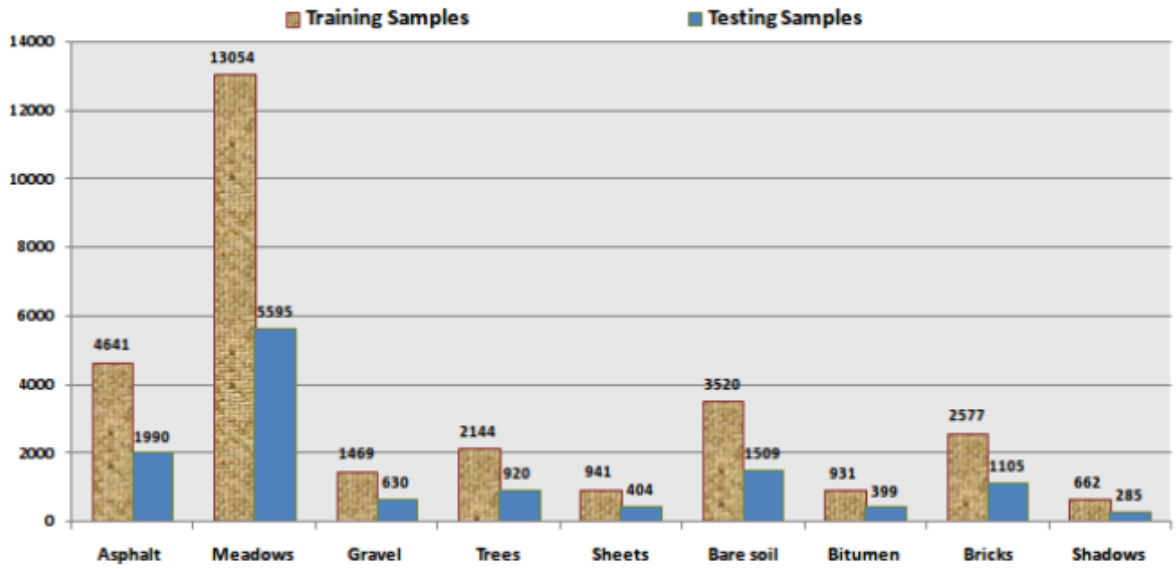


FIGURE 7.6 – Jeu de données Pavaia University

— **Salinas scène :**

Le deuxième jeu de données [160], est capturé par le capteur AVIRIS sur la vallée Salinas-California, on trouve dans cette scène 512 x 217 pixels avec 224 bandes et qui contient 16 classes. Dans cette scène, 20 bandes ont été supprimées : (108-112 ; 154-167 ; 224) qui représentent les bandes d'absorption d'eau.

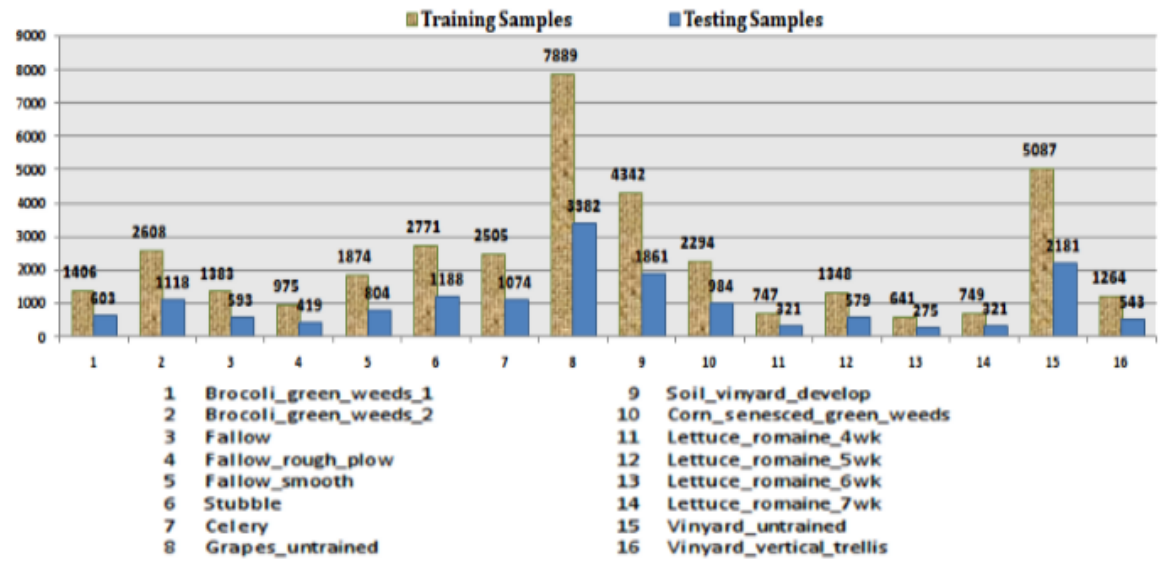


FIGURE 7.7 – Jeu de données Salinas

7.4.2 Détails et résultats

Premièrement, nous avons commencé par appliquer le modèle de classification proposé sur deux images séparément et nous avons noté pour chaque image, la précision (OA) et le temps passé à l'étape de l'entrainement. Ensuite, nous avons construit une seule image composite à partir des deux ensembles de données : Pavia University et Salinas. Nous avons appliqué le modèle de classification à cette image composite et nous avons noté la précision et le temps de d'entrainement sur l'image composite.

Les expériences sont effectuées sur un ordinateur équipé d'un processeur Intel® Core i7-2820QM CPU @ 2,30 Ghz 8, 16 Go de RAM. Le modèle de classification est implémenté en langage Python à l'aide de la bibliothèque d'apprentissage profond nommée : Keras. La table 7.2 contient les valeurs de l'expérience :

TABLE 7.2 – Expérimentation du modèle proposé sur plusieurs images

	Pavia U		**Salinas**			
	OA (%)	**Time (s)**	**OA (%)**	**Time (s)**	**OA_avg (%)**	**T (s)**
[11]	92.56	420	92.60	3180	92.58	3600
Proposed model (1)	92.59	117.34	92.8	578.48	92.69	695.82
Proposed model (2)	94.2	408.3	95.21	3176.07	94.71	3584.37

A partir du papier [150], nous avons pris la précision (OA) et le temps d'entrainement du modèle sur les deux images : Pavia University et Salinas. Ensuite, deux valeurs ont été

calculées : la précision moyenne (OA_avg) et le temps total d'entraînement (T) du modèle sur les deux images :

— $T = Temps\ d'entrainement(PaviaU) + Temps\ d'entrainement(Salinas)$

— $OA\_avg = \frac{OA(PaviaU)+OA(Salinas)}{2}$

Pour tester l'efficacité et la rapidité du modèle proposé, nous avons fait deux expérimentations : d'abord, nous avons entrainé le modèle proposé jusqu'à l'obtention de la précision du papier [150], et nous avons noté le temps d'entrainement effectué sur chaque image : PaviaU et Slainas. Le temps total d'entrainement obtenu sur les deux images du modèle proposé (695,82 s) est plus réduit que le temps total d'entrainement du modèle de papier [150] (3600 s).

Deuxièmement, nous avons entrainé notre modèle jusqu'à l'obtention du temps d'entrainement du papier [150], et nous avons noté la précision sur chaque image : PaviaU et Slainas. La précision moyenne du modèle proposé (94,71 %) est supérieure à la précision moyenne du papier [11] (92,58 %)

On remarque que l'algorithme proposé, en comparaison avec l'algorithme du papier [150], donne une meilleure précision sur les deux images, avec moins de temps d'entrainement. Le graphique de fa figure 7.8) donne l'évolution de la précision, en fonction du temps d'entrainement pour l'algorithme proposé.

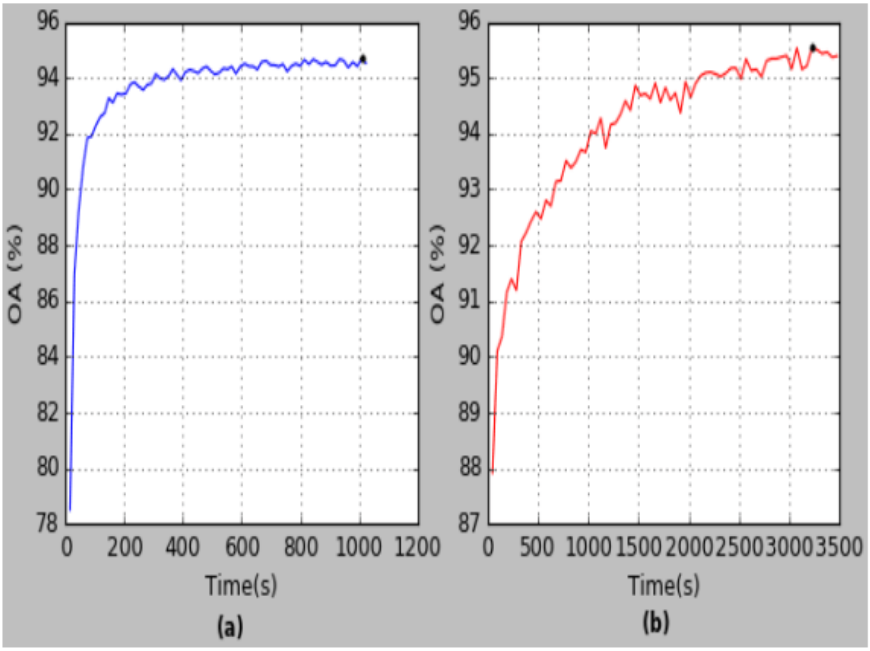


FIGURE 7.8 – La précision en fonction du temps d'entrainement pour Pavia University (a) et Salinas (b).

En conclusion, le modèle proposé est compétitif avec le modèle de classification de l'état de l'art. Nous avons utilisé notre modèle pour valider l'approche de classification d'une image composée de plusieurs images, à l'aide d'un seul entrainement.

La table 7.3 donne : La valeur de la précision, le temps d'entrainement de l'algorithme proposé sur une seule image composite de plusieurs images HSI et OA_avg, le temps total d'entrainement lorsque le modèle a été appliqué aux images séparées.

TABLE 7.3 – Expérimentation du modèle proposé sur une seule image, composée de 2 HSI

	Le modèle proposé sur 2 images HSI séparées		**Le modèle proposé sur une seule image composée de 2 images HSI**	
	OA_avg	T	OA	Training time
Test : 1	92.69 %	695.82 s	92.76 %	573.8 s
Test : 2	94.71 %	3584.37s	94.75 %	2869.41 s

Selon la table 7.3, nous notons que l'application du modèle de classification proposé sur une seule image composée de plusieurs HSI, donne une meilleure valeur de précision que l'application du modèle sur des images séparées et dans un temps d'entrainement plus court (voir Fig. 7.9).

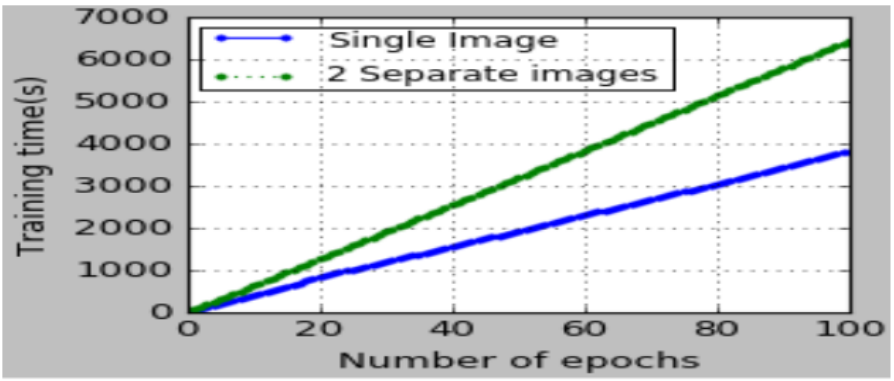


FIGURE 7.9 – La variation du temps d'entrainement du modèle proposé, sur les images séparées et sur une seule image composée, en fonction du nombre d'itérations.

Les résultats visuels de la prédiction sont présentés à la fig 7.10.

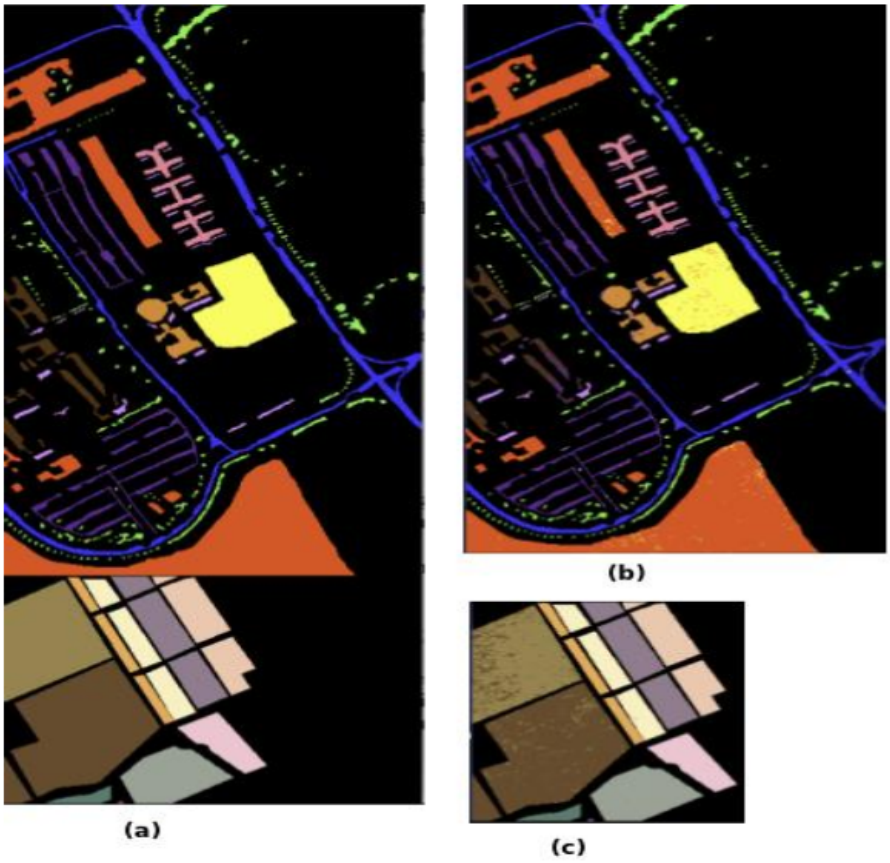


FIGURE 7.10 – Image HSI composée (a), Résultats de la classification du modèle proposé pour : PaviaU (b), Salinas (c)

7.5 Conclusion

Dans cette contribution, un nouveau modèle de classification d'une image hyperspectrale composée de plusieurs HSI et qui utilise un seul entrainement a été proposé. Les résultats de la comparaison de l'algorithme proposé avec un modèle de l'état de l'art [150] et même avec l'application de cet algorithme sur des images avec plusieurs étapes d'entrainement, montre la rapidité et la performance de l'algorithme proposé.

Chapitre 8

Prédiction par les CNNs :Application aux TALN

Sommaire

8.1 Introduction

Dans cette section, nous traitons de l'utilisation des modèles linguistiques basés sur l'architecture CNN appliqués à la langue Arabe dans l'objectif de prédire le texte manquant dans les documents arabes. Le processus de prédiction est le principal défi soulevé car il dépend d'une grande échelle d'opérations élémentaires telles que la segmentation de texte, la détection de l'incorporation de mots, et la récupération de sens.

La motivation retenue par la prédiction de texte est qu'elle s'ouvre à plusieurs formes d'exploitation de documents, mais ne se limite pas à l'analyse sémantique, à la détection de la période historique des manuscrits non datés, et à l'analyse du style de rédaction. D'un autre côté, le traitement de l'Arabe a mis le point sur certaines caractéristiques de cette langue riche sur le plan morphologique, telles que la signification des schémas de mots, les unités d'écriture (lettre, mot et phrase), les différentes formes de lettres, l'absence de vocalisation, et le faible usage des signes de ponctuation.

Notre idée repose sur l'habileté humaine à extraire les significations d'un texte ou d'une partie du discours qui implique de comprendre le sens d'un mot (une phrase ou une partie de texte) dans son contexte d'utilisation. Le modèle CNN proposé prend un texte arabe en entrée, s'entraîne dessus (apprend) et prédit du texte en fonction de son entraînement et de son processus d'apprentissage. L'utilisation de CNN en profondeur avait été motivée par le succès des modèles CNN confrontés à de nombreux problèmes dans plusieurs domaines, notamment l'identification du script [161], la classification du texte [162], la reconnaissance du texte [163], et la reconnaissance des caractères [164, 165]. Le succès des modèles CNN a été attribué à leur capacité à apprendre les fonctionnalités de grandes quantités de données de manière complète.

8.2 Expériences et résultats

8.2.1 Préparations de données

Au début de chaque travail portant sur une quantité énorme de données, la préparation des données reste une tâche fastidieuse mais surtout nécessaire. Nous avons d'abord téléchargé gratuitement plusieurs documents texte au format de document portable (pdf) à partir de trois sources sur le Web : Arab World Book (AWB) A , Bibliothèque Shamela (ShL) A et l'Organisation Hindawi (HND) A. Nous avons rassemblé plusieurs romans et poèmes de certains auteurs arabes. La taille globale de nos sources de données était d'environ 130 Méga octets de texte, répartis sur 144 fichiers texte de plus de 4000000 de mots. La figure 8.1 donne un aperçu de notre source de données.

Source de données	nombre de documents	nombre de mots
AWB	38	1009365
ShL	67	2133442
HND	39	1082727
Total	**144**	**4225534**

FIGURE 8.1 – Les documents utilisés dans la recherche

Tous les documents pdf provenant de ces sources ont été convertis au format texte à l'aide de l'outil "Free to PDF Converter" librement disponible sur Internet [1].

La figure 8.2 répertorie certains de ces documents que nous avons utilisés dans nos expériences et leurs auteurs, respectivement. Nous avons assigné un ID à chaque document pour le désigner lors de la phase d'implémentation.

Titre du document	Nom d'auteur	ID
Les jours	Taha Hussein	*HND_TH_1*
Larme et sourire	Jabran Khalil Jabran	*HND_JKJ_7*
La patrie	Mahmoud Darweesh	*HND_MD_1*
Diwan	Maarof Rosafi	*AWB_MR_2*
Le retour de la vague	May Ziayda	*AWB_MZ_5*
Les avares	Al Jahid	*ShL_JHD_1*
Kalila wa dimna	Ibn Almoqafaa	*ShL_MQF_1*

FIGURE 8.2 – Des documents et auteurs utilisés dans cette recherche

Après une exploration de ces fichiers texte, il a été constaté que le texte devait être nettoyé de certains aspects, tels que la succession d'espaces multiples et l'apparition de caractères indésirables tels que le point d'interrogation "?" et le carré. Cela est dû à deux problèmes : le codage des caractères arabes et la correspondance entre le codage des lettres arabes et les formes. D'une part, pour faire face au problème d'encodage, nous avons choisi l'encodage utf-8, et d'autre part, des caractères indésirables apparaissent lors de l'utilisation de polices d'écriture différentes dans des environnements différents, nous procédons par unifier les fonts de polices de l'intégralité des documents textes sur lesquels nous avons travaillé. Une fois les données nettoyées, nous procédons par les diviser en trois sous-ensembles : ensemble d'entraînement (Training Data - TrD), ensemble de validation (Validation Data - VD) et ensemble de test (Test Data - TsD).

Le processus d'entraînement est une opération qui consiste à apprendre au modèle CNN comment écrire le texte arabe, les catégories de mots en arabe, les particularités de l'arabe (en particulier celles prises en compte dans les travaux de cette thèse), la morphologie, la grammaire, la conjugaison, et la sémantique. Cela étant dit, le modèle apprend en parcourant une multitude de documents écrits en arabe tout en rappelant, entre autres, l'ordre des mots et la composition des phrases.

A la fin de l'opération d'entraînement, le modèle dispose de suffisamment de bagage pour pouvoir générer du texte arabe ou le prédire. Nous procédons ensuite à l'opération de validation, qui consiste à évaluer l'apprentissage du modèle en lui donnant des documents déjà traités mais cette fois-ci avec du texte manquant. Le modèle doit donc prédire le texte manquant. Nous comparons avec le texte d'origine et calculons la précision des résultats. L'étape de test vient ensuite en alimentant le modèle par de nouveaux documents (pour la première fois) avec du texte manquant. Ces documents n'ont pas du tout été traités par le modèle. Le modèle CNN tente de prédire le texte en fonction de son apprentissage.

Comme dans la plupart des cas de l'état de l'art de la préparation des données, TrD prenait environ 70% des données, soit 94 fichiers de documents sur 144. VD et TsD utilisaient chacun

1. **http://www.01.01.com /telecharger/windows/Multimedia/scanner_ocr/.fiches/115026.html**.

environ 15% des données, soit 25 fichiers chacun. La figure 8.3 montre la distribution des documents et des mots par source de données pour chacune de nos trois sources.

Source de données	TrD		VD		TsD	
	ND	*NM*	*ND*	*NM*	*ND*	*NM*
AWB	24	762910	7	178235	7	158220
ShL	45	1544197	11	289077	11	300168
HND	25	810873	7	163448	7	108406
Total	**94**	**3117980**	**25**	**630760**	**25**	**566794**

FIGURE 8.3 – La distribution du nombre de documents ND et du nombre de mots NM par source de données

8.2.2 Entrées du modèle CNN

Le texte (déjà préparé pour être comme entrée du modèle CNN) avait été transformé en codes numériques, car l'architecture CNN nécessite des données numériques en entrée. La transformation avait été effectuée selon les étapes suivantes :

— 1. Division du texte en unités d'écriture (Writing Units - WU).

— 2. Attribuer un ID numérique unique à chaque WU.

— 3. Pour chaque ID, nous avons calculé son code équivalent binaire. Nous l'avons appelé code d'unité d'écriture binaire (Binary Writing Unit Code - BWUC).

— 4. Créer un dictionnaire associant les BWUC, uniques, à leur WU respectives.

Une autre étape de paramétrage consiste à représenter chaque BWUC dans un vecteur (v) de taille fixe k, où (2^k = taille_du_vocabulaire). Les éléments du vecteur d'entité en entrée ($iv = wu_1, wu_2, wu_3, wu_4, ..., wu_N$) sont les BWUC associés de WU successives (wu_i) dans un document texte (D). La succession de N wu_i dans un texte génère nécessairement une WU unique (ou avec au moins une précision supérieure, avec une autre WU avec une précision inférieure) qui améliorera le processus de prédiction. Le iv est introduit dans le modèle CNN et, à la sortie, la prochaine WU est donnée, c'est-à-dire un vecteur (v) de N éléments ($wu_1, wu_2, wu_3, wu_4, ..., wu_N$) conduit à la prédiction de la prochaine WU qui sera $wu_N + 1$.

Pour réduire la navigation documentaire et gagner en temps d'exécution, nous ne nous sommes plus limités à l'utilisation de vecteurs, nous avons plutôt opté pour des matrices. Nous avons créé une matrice M contenant un nombre N de BWUC dans ses colonnes. N est déterminé en fonction de la performance des résultats de prédiction (nous avons évalué N = 3, N = 4 et N = 5). L'ordre des éléments d'une ligne dans M est le même que l'apparence de WU dans le texte. La matrice M associée à l'extrait du texte <alwatan> du document HND_MD_1 est illustrée à la figure 8.4.

$M[i,4]$	$M[i,3]$	$M[i,2]$	$M[i,1]$	$M[i,0]$	
بين	وطنه	في	المرء	يعيش	$M[0,j]$
أهله	بين	وطنه	في	المرء	$M[1,j]$
و	أهله	بين	وطنه	في	$M[2,j]$
أصدقائه	و	أهله	بين	وطنه	$M[3,j]$

FIGURE 8.4 – La matrice M associée à un extrait du document HND_MD_1, avec une variation de N = 5

Le nombre de lignes (nr) de M est déterminé par : nr $= \frac{nWU}{N}$ où nWU est le nombre total de WU dans tout le texte. Chaque ligne de M contient des colonnes N et chaque colonne est un élément (M_{ij}, où i correspond à la $(i+1)^{eme}$ ligne, et j correspond à la $(j+1)^{eme}$ colonne dans M) faisant référence à un WU. Nous avons ajusté les longueurs de tous les WU en complétant ces longueurs par le caractère vide pour avoir la même longueur pour tous les WU.

La prochaine étape consiste à créer un vecteur colonne Y contenant un élément par ligne. Y a le même nombre de lignes que M. La correspondance entre M et Y est qu'après chaque groupe d'éléments d'une ligne de M, on trouve nécessairement l'élément de la même ligne de Y, c'est-à-dire : après chaque groupe de N WU dans le texte, nous avons le WU qui est référé par le élément en Y. La figure 8.5 montre le vecteur Y correspondant à la matrice M présenté dans la figure 8.4. Nous avons ensuite créé M_codes, une matrice équivalente à M et Y_codes codant un vecteur équivalent à Y. M_codes et Y_codes contiennent BWUC de WU dans le texte, et elle vont servir comme entrées pour le modèle CNN

أهله	Y[0,0]
و	Y[1,0]
أصدقائه	Y[2,0]
.	Y[3,0]

FIGURE 8.5 – Le vecteur Y associé à la matrice M représentée dans la figure 8.4

8.2.3 Le modèle CNN proposé

La structure de notre CNN est illustrée par la figure 8.6. Elle se compose d'une couche d'entrée, de deux couches de convolution avec une fonction d'activation non linéaire, de deux couches de Max-Pooling et d'une couche entièrement connectée. La dernière couche est la couche de sortie.

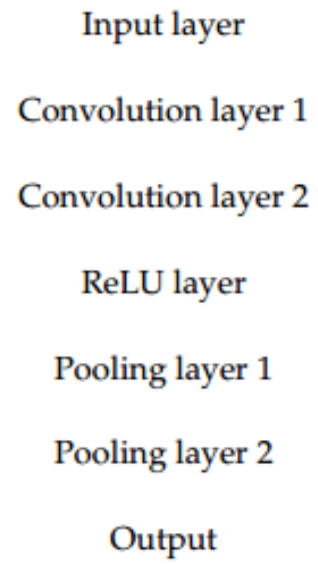


FIGURE 8.6 – Le modèle proposé de CNN

8.2.4 Résultats et discussion

Nos expériences sont basées sur trois étapes essentielles : entraînement, validation et test. Les opérations d'entraînement, de validation et de test ont été effectuées sur le niveau de trois expériences selon trois processus différents afin d'analyser et d'interpréter les résultats de la prédiction du texte manquant, tout en visant à améliorer la précision globale de cette prédiction.

Première expérience :

Nous avons d'abord commencé l'entraînement sur les documents d'un seul auteur, nous avons répété le processus pour chaque auteur à part, nous avons validé les documents déjà traités pendant l'entraînement, puis nous avons testé nos résultats sur des documents qui seront fournis au modèle CNN pour la première fois.

Deuxième expérience :

Elle permet de fournir des documents d'auteurs différents mais de la même source, car chacune des trois sources (AWB, ShL et HND) dispose de ses propres priorités en matière de classification des documents, de sélection de documents, de choix de sujets, d'affichage de sujets et de choix d'auteurs évidemment. Nous effectuons le même processus en matière d'entraînement, de validation et de test.

Troisième expérience :

Elle consiste à fournir des documents en arabe au modèle CNN sans prendre en compte ni l'auteur ni la source de données dans le but de prédire du texte en arabe.

La prédiction des textes manquants sur les documents traités avait été effectuée sur la base d'une approche statistique et d'une approche probabiliste. Le modèle calcule le taux d'une apparence d'un WU après N autres WU dans un texte. Dans le même texte, même du même auteur, on peut retrouver après 5 WU par exemple l'apparition de plusieurs WU en fonction du contexte. Le modèle calcule ensuite la probabilité de chaque apparition d'un WU (PWUA) et prédit le texte manquant en fonction de la probabilité la plus élevée. La figure 8.7 donne une illustration de ce concept.

PWUA	Y	M				
89.3 %	أهله	بين	وطنه	في	المرء	يعيش
2.1 %	عائلته	بين	وطنه	في	المرء	يعيش
4.3 %	أقاربه	بين	وطنه	في	المرء	يعيش
2.3 %	أصدقائه	بين	وطنه	في	المرء	يعيش
2.0 %	other	بين	وطنه	في	المرء	يعيش

FIGURE 8.7 – Le calcul de la probabilité de prédiction

Dans la première expérience, les résultats ont montré un haut niveau de précision globale. Pour le test, nous proposons à notre modèle CNN des documents avec le texte manquant, toujours du même auteur, mais cette fois-ci, ces documents n'ont jamais été traités par le modèle CNN, ni au stade de l'entraînement ni à celui de la validation. Le modèle a répondu de manière satisfaisante puisque les résultats prévus atteignaient un maximum de 92,8% de précision globale. Les résultats sont représentés dans la figure 8.8.

Auteur	**TrD**	**VD**		**TsD**	
	ND	ND	POA(%)	ND	POA(%)
Taha Hussein	22	5	93.0	5	92.8
Jabran Khalil Jabran	14	4	86.9	4	83.7
Ghassan Kanafani	8	2	80.0	2	78.4
May ziyada	9	3	80.1	2	80.0
Mahmoud Darweesh	19	4	90.3	4	89.4
Najeeb Mahfod	17	4	88.1	4	87.9
Maarof Rosafi	3	2	65.4	1	60.3
Al Sulayti	2	1	77.2	1	62.8
Total	94	25	82.62	25	79.41

FIGURE 8.8 – La précision globale de la prédiction par auteur

A titre d'observation préliminaire, la précision de la prédiction semble avoir une relation proportionnelle avec la quantité de texte entraîné et testé. La figure 8.8 montre que plus la quantité de texte est élevée (à la fois dans la phase d'entraînement ou dans la phase de validation), plus la précision globale de prédiction (POA) est élevée (dans les phases de validation et de test). Cela prouve que l'apprentissage par le modèle CNN est plus intéressant lorsque la quantité de données est plus importante.

Dans la deuxième expérience, Nous avons essayé, d'évaluer la prédiction du texte manquant appartenant à la même source de données, sans prendre en compte l'auteur du texte, dans le but de supposer si la prédiction est réalisée en suivant les mêmes critères que dans la première étape (prédiction par domaine, style d'écriture et conservation du formatage). Le modèle a été alimenté par des textes de la même source mais de tout auteur confondu. Nous avons effectué l'entraînement, la validation et le test de la même manière que pour la première expérience. Le modèle avait été entraîné sur les données issues de TrD de AWB, puis la validation a été réalisée en se basant encore sur des textes de VB de AWB, et puis après il avait été testé par des textes de TsD de AWB. Les mêmes opérations ont été répétées pour les textes de ShL à part et les textes de HND à part. Les résultats de la prévision de cette expérience sont décrits dans la figure 8.9où le nombre de documents utilisés est présenté par ND.

Source de données	TrD	VD		TsD	
	ND	ND	POA(%)	ND	POA(%)
AWB	24	7	97.3	7	94.8
ShL	45	11	98.1	11	96.9
HND	25	7	98.4	7	95.6
Taux moyen	94	25	97.93	25	95.77

FIGURE 8.9 – La précision globale de la prédiction par source de données

Certes, à cette expérience, nous avons obtenu des résultats satisfaisants en termes de POA. Il est clair et évident que le POA, à l'étape de validation (98,4 au maximum), est supérieur au POA calculé à l'étape de test (96.9), car les textes de VD ont déjà été traités par le modèle, tandis que les textes de TSD sont traités par le modèle pour la première fois. Le POA à ce stade est encore plus élevé par rapport au premier stade, sachant que la variante dont il est question à ce niveau est la quantité de texte fournie dans chacune des deux expériences. Nous concluons partiellement que la quantité de texte fournie au cours des étapes d'apprentissage et de test est proportionnelle au POA calculé lors de l'étape de validation.

La dernière expérience a été réalisée en prenant des documents d'auteurs différents provenant des trois sources de données. L'objectif était de disposer des documents en arabe, sans aucune condition préalable et sans restriction sur la nature des textes en entrée, et de pouvoir prédire le texte manquant de ces documents. Les résultats de ce processus sont rapportés dans la figure 8.10.

Source de données	TrD	VD		TsD	
Taux moyen	94	25	99.6	25	97.8

FIGURE 8.10 – La précision globale de la prédiction

A ce stade, nous avons calculé le POA par une méthode cumulative, c'est-à-dire que nous avons fourni au modèle CNN une quantité de texte et que nous avons calculé le POA, puis nous avons paramétré les données d'entraînement (données de validation et données de test) en augmentant la quantité de données pour améliorer les performances. Nous avons également réussi à alimenter le modèle CNN par des textes et à calculer le POA jusqu'à ce que nous fournissions tout le texte. Nous avons expérimenté 4 étapes de transformations (par exemple, un auteur, une source, une combinaison d'auteurs par source, une combinaison de sources par auteur), chaque transformation ayant été appliquée deux fois tout en augmentant la quantité des données d'une première à la seconde. La figure 8.11 montre les meilleures performances de POA tout en traitant de l'augmentation des données.

Source de données	pourcentage de représentation	$N = 3$	$N = 4$	$N = 5$
Mahmoud Darweesh	18.75 %	45.9	66.8	89.4
Taha Hussein	22.00 %	52.3	66.3	92.8
Najeeb Mahfod	17.36 %	50.0	63.2	87.9
AWB	26.39 %	64.0	76.2	94.8
HND	27.08 %	58.9	77.3	95.6
ShL	45.53 %	59.2	78.1	96.9
AWB + HND	53.47 %	58.4	74.2	95.6
HND + ShL	72.61 %	54.0	72.4	94.2
ShL + AWB	71.92 %	59.0	75.1	95.7
AWB + HND + ShL	100 %	60.0	77.1	97.8

FIGURE 8.11 – Meilleure performance pour l'augmentation de données

8.3 Conclusion

Dans cette contribution, nous avons utilisé les réseaux de neurones convolutifs pour proposer un modèle de prédiction du texte manquant dans les documents arabes. Dans la phase d'entrainement et de test, nous avons utilisé plusieurs jeux de données arabes. Les résultats des expériences montrent la performance de notre modèle, avec une précision de 97,8% dans le meilleur des cas.

Troisième partie

Conclusion générale

Chapitre 9

Conclusion et perspectives

9.1 Conclusion

L'objectif principal de cette thèse était d'implémenter des solutions pour le domaine de Big Data, afin d'analyser et d'extraire des informations pertinentes dans des données massives. Premièrement, nous avons commencé par le choix de la donnée massive à utiliser. Dans ce point, nous avons pris deux types de jeux de données : les Images Hyperspectrales et le corpus d'un texte arabe. Nous avons proposé de nouvelles approches pour chaque type de jeu de données. En outre, une attention particulière a été accordée à la précision et à la vitesse des approches proposées.

Comme principales remarques finales, les points suivants peuvent être mentionnés :

Pour le premier jeu de données nous avons remarqué que, pour faire des traitements sur les images hyperspectrales, on doit commencer à résoudre le problème de la grande dimension de HSI. Dans la première contribution, nous avons proposé une version distribuée parallèle de l'algorithme de réduction de dimension ACP. L'implémentation est faite dans un environnement parallèle distribué nommé Apache Spark. Le modèle proposé nous a permis d'obtenir les mêmes performances que l'ACP classique mais avec une vitesse élevée. Dans la deuxième contribution, nous avons proposé une méthode de visualisation rapide des HSI en se basant sur l'algorithme de réduction parallèle et distribuée et toujours dans l'environnement Apache Spark. Dans la troisième contribution, nous avons proposé un modèle de classification spectrale des HSI en se basant sur les réseaux de neurones convolutifs. Le modèle proposé combine plusieurs images hyperspectrales en une seule. L'entrainement du modèle est fait sur l'image obtenue. Les résultats de l'expérimentation montrent bien la performance en terme de précision et de rapidité du modèle proposé.

Pour le deuxième jeu de données, nous avons proposé un modèle de prédiction pour les textes manquants dans les documents arabes. Le modèle proposé est basé sur les réseaux de neurones convolutifs et implémenté en langage de programmation Python en utilisant la bibliothèque keras. Les résultats de l'expérimentation indiquent que le modèle proposé donne de meilleures performances pour des jeux de données de grandes tailles.

9.2 Perspectives

Le travail mené dans le cadre de cette thèse ouvre certainement la voie à plusieurs pistes de recherche, on peut citer :

— Dans la classification des images hyperspectrales, on propose d'augmenter le nombre de HSI qui composent l'image à classifier, puis créer un modèle de classification spectrale-spatiale 3D en se basant sur le CNN.

— Utiliser l'apprentissage en profondeur distribué (Distributed Deep Learning) pour entrainer un large modèle de classification des HSI avec des millions ou des millards de paramètres. Par exemple en utilisant des structures logicielles comme : Distributed Keras , SparkNet, BigDL, ...

— Utiliser les GPU pour la classification des HSI dans l'objectif d'accélérer le temps d'entrainement.

— Un autre sujet méritant des recherches futures est la compression des images hyperspectrales.

Quatrième partie

Annexes

Annexe A

jeux de données : Textes Arabes

Dans cette section, on va présenter les jeux de données arabes utilisés dans la contribution 8

AWB

AWB [1] est un club culturel et une librairie arabe qui visent à promouvoir la pensée arabe. Il fournit un service public aux écrivains et aux intellectuels et exploite le vaste potentiel d'Internet pour ouvrir une fenêtre dans laquelle le monde se tourne vers la pensée arabe, pour identifier ses créateurs et ses penseurs et dans le but de réaliser une communication intellectuelle entre les peuples des patries arabes .

ShL

ShL [2] est un grand programme gratuit qui vise à être complet pour les besoins des chercheurs : livres et recherches. La bibliothèque travaille actuellement sur un système approprié pour recevoir des fichiers de divers textes et les organiser dans un cadre avec une possibilité de recherche.

HND

HND [3] est une fondation à but non lucratif qui cherche à avoir un impact significatif sur le monde de la connaissance. La fondation travaille également à la création de la plus grande bibliothèque arabe contenant les livres les plus importants du patrimoine arabe moderne, après reproduction, à conserver de l'extinction.

1. **https ://www.arabworldbooks.com/**.
2. **http ://shamela.ws/index.php/main**.
3. **https ://www.hindawi.org/**.

Annexe B

Évaluation de la précision

Dans cette annexe, on va présenter les différentes méthodes utilisées pour l'évaluation de la précision des résultats de la classification.

Notations :

— N_c représente le nombre de classes.

— C_i désigne la classe i

— C_{ij} représente le nombre de pixels mal assignés à la classe j, référencés sous la classe i.

Matrice de confusion :

Une matrice de confusion est considérée comme un outil de visualisation, généralement utilisé dans l'apprentissage supervisé. Dans cette matrice, chaque colonne déduit les instances dans une classe prédite, tandis que chaque ligne représente les instances dans une classe réelle. Cette matrice est également en mesure d'indiquer où la technique de classification conduit à la confusion. Le tableau suivant (voir Fig B.1) montre une matrice de confusion pour un problème de classification avec 3 classes.

Percentage	Classification data				
Reference data	C_1	C_2	C_3	Row total	Producer's accuracy
C_1	C_{11}	C_{12}	C_{13}	$\sum_i^{N_c} C_{1i}$	$\frac{C_{11}}{\sum_i^{N_c} C_{1i}}$
C_2	C_{21}	C_{22}	C_{23}	$\sum_i^{N_c} C_{2i}$	$\frac{C_{22}}{\sum_i^{N_c} C_{2i}}$
C_3	C_{31}	C_{32}	C_{33}	$\sum_i^{N_c} C_{3i}$	$\frac{C_{33}}{\sum_i^{N_c} C_{3i}}$
Column total	$\sum_i^{N_c} C_{i1}$	$\sum_i^{N_c} C_{i2}$	$\sum_i^{N_c} C_{i3}$	N	
User's accuracy	$\frac{C_{11}}{\sum_i^{N_c} C_{i1}}$	$\frac{C_{22}}{\sum_i^{N_c} C_{i2}}$	$\frac{C_{33}}{\sum_i^{N_c} C_{i3}}$		

FIGURE B.1 – Matrice de confusion pour un problème de classification à 3 classes

Précision globale (OA) :

OA est le pourcentage de pixels correctement classifiés :

$$OA = \frac{\sum_i^{N_c}(C_{ii})}{\sum_i^{N_c}(C_{ij})} x100$$

Précision de classe (CA) :

Le CA (ou la précision du producteur) est considéré comme le pourcentage de pixels correctement classifiés pour chaque classe.

$$CA_i = \frac{C_{ii}}{\sum_j^{N_c}(C_{ij})} x100$$

Précision moyenne (AA) :

L'AA est la moyenne des précisions de classe pour toutes les classes :

$$AA = \frac{C_{ii}}{\sum_j^{N_c}(CA_i)} x100$$

Coefficient Kappa (k) :

Cette métrique est une mesure statistique de l'accord entre la carte de classification finale et la carte de référence. Il s'agit du pourcentage d'accord corrigé du degré d'accord auquel on pouvait s'attendre du seul fait du hasard. On pense généralement que cette mesure est plus robuste que le simple calcul du pourcentage d'accord, car k tient compte de l'accord survenu par hasard.

$$AA = \frac{P_o - P_e}{1 - P_e}$$

avec

$$P_o = OA$$

$$P_e = \frac{1}{N^2} \sum_i^{N_c}(C_{i+} C_{+i})$$

$$C_{i+} = \sum_j^{N_c} C_{ij}$$

$$C_{+i} = \sum_j^{N_c} C_{ji}$$

Annexe C

Outils et corpus pour le TALN Arabe

Dans cette annexe, on va présenter quelques ressources créées et de systèmes construits pour le traitement automatique de la langue arabe.

Corpus Arabe :

Corpus de parole :

— Arabic Broadcast News : audio - transcriptions

— Appen's Gulf Arabic Conversational Telephone Speech : audio - transcriptions

— Levantine Arabic QT Training DataSet 5 :audio - transcription, Une combinaison de quatre jeux de données d'entrainement, totalisant 250 heures de conversation téléphonique en arabe Levantin

— OrienTel Morocco Modern Colloquial Arabic – ELRA Catalog S0183

Corpus de la reconnaissance de l'écriture arabe et de l'évaluation :

— LDC Ressources pour la reconnaissance de l'écriture arabe

— Jeu de données Applied Media Analysis pour l'arabe manuscrit

— OpenHaRT 2010 The 2010 NIST Open Handwriting Recognition and Translation Evaluation (OpenHaRT 2010)

Corpus de texte :

— Arabic Gigaword

— ArabiCorpus

— Quranic Arabic Corpus (Annotations for POS tags and Syntax)

— ISI Arabic-English Automatically Extracted Parallel Text (newswire)

— Arabic English Parallel News Part 1

— Arabic Wikipedia with many terms paired with other languages (not strictly parallel)

— GALE Phase 1 Distillation Training data

— enn Arabic Treebank (LDC) Part 1 v 3.0 Part 2 v 2.0 Part 3 v 2.0

— Arabic Proposition Bank (Propbank)

Bases de données lexicales :

— Al-Baheth Al-Arabi – Online search of a collection of classic Arabic dictionaries, such as Lisan Al-Arab and Al-Qamus Al-Muheet (in Arabic)
— Google Online dictionary (multilingual)
— Unified Medical Dictionary of the World Health Organization
— UN Bibliographical Information System Thesaurus

Ontologie sémantiques :

— Arabic Wordnet

Outils :

Outils de traduction automatique :

— Google Translate – bidirectional translation for over 50 languages including Arabic
— Microsoft's Bing Translator – bidirectional translation for over 30 languages including Arabic
— Sakhr's Tarjim (Arabic-English and English-Arabic)
— Almisbar Arabic-English translation
— Statistical MT public resources : Giza alignment, Pharaoh and Moses decoders, etc.

Saisie de texte :

— Yamli.com
— Google's Ta3reeb
— Mircorosft's Maren

Reconnaissance d'entité nommée :

— Yassine Benajiba's ANER (Arabic Named Entity Recognition) system
— BBN's Identifinder (English, Arabic, Chinese)

Lexicographie :

— aConCorde : A concordance generation program for Arabi

Parseurs :

— The Stanford Parser
— The Bikel Parser
— MALTParser
— Mohammed Attia's Rule-based Parser for MSA

Annexe D

Les implémentations en langage Python

Dans cette annexe, on va présenter, quelques programmes implémentés en langage de programmation Python durant cette thèse.

```
from __future__ import print_function

import numpy as np
import scipy as sp
import matplotlib.pyplot as plt
from time import time
import sys
from scipy import linalg as la
from PIL import Image
from  pylab import *
import scipy.io as sio
from random import sample


#------------------------------keras----------------------------

import keras
#from keras.datasets import mnist
from keras.models import Sequential   #a linear stack of layers
from keras.layers import Dense, Dropout, Flatten,Activation
from keras.layers import Conv2D, MaxPooling2D,Conv3D, MaxPooling3D
from keras import backend as K   #la librairie de deep learning (tensorflow or theano)
from keras.utils import plot_model
from keras.models import model_from_json

import my_callbacks


path_data="/home/pc/spark_tst/data/"   #chemin
my_callbacks.num_file="tst_spect_"


#-----------------------------------------
#---------------------------------------lecture à partir d'un ichier matlab
def fichierMat2matrice(nom_data):
    '''
    A partir d'un fichier matlab (matrice de taille lxcxN) on retourne une matrice M (individus x variables=taille (lxc)xN)
    et une liste de labels(taille:lxc) qui a la même taille que le nombre de ligne de M
    '''
    mat = sio.loadmat(path_data+nom_data+".mat");
    mat_gt = sio.loadmat(path_data+nom_data+"_gt.mat");
    l,c,N=len(mat[nom_data]),len(mat[nom_data][0]),len(mat[nom_data][0][0])
    print("Taille Image : ",l,c,N)
    M=np.zeros((l*c,N))
    Y=np.ones(l*c)
    k=0
    for i in range(l):
        for j in range(c):
            M[k,:]=mat[nom_data][i][j]
            Y[k]=mat_gt[nom_data+"_gt"][i][j]
            k+=1

    nbr=N
```

```
    return M,Y
#------------------------------------------------------------------------
def fichierMatGroundTruth2matrice(path_data,nom_data):
    '''
    A partir d'un fichier matlab on retourne une matrice (individus x variables)
    '''
    mat = sio.loadmat(path_data+nom_data+".mat");
    print(mat)
    l,c,N=len(mat[nom_data]),len(mat[nom_data][0]),1
    M=mat[nom_data].reshape(l*c,1)

    return M
#choix du dataset------------------------------------------------------
def variables(num_data):
    p=open(path_data+"pile_execution.txt","r")
    dataset=p.readlines()
    L=dataset[num_data].split(",")
    print(L,end="\t")
    p.close()
    nomImage,l,c,N,num_classes=L[1],int(L[2]),int(L[3]),int(L[4]),int(L[5])

    return nomImage,l,c,N,num_classes
#------------------------------------------------------------------------
def PCA_sklearn(data,N):
    from sklearn.decomposition import PCA
    pca = PCA(n_components=N)
    pca.fit(data)
    evals=pca.explained_variance_   #valeurs propres
    y=pca.fit_transform(data)
    return y

#------------------------------------------------------------------------
def reduireIndian8classes(Y,num_classes,lnew_classes):
    new_classes={}
    k=1
    for i in range(num_classes):
        if i in lnew_classes:
            new_classes[i]=k
            k+=1
        else:new_classes[i]=0

    for i in range(len(Y)):
        Y[i]=new_classes[Y[i]]
    new_num_classes=num_classes-len(lnew_classes)
    return new_num_classes
#------------------------------------------------------------------------
def alldata2One(liste_images):
    #les tableaux contenantles data de chaque image
    M=[0]*len(liste_images) #les iamges
    Y=[0]*len(liste_images) #les labels
    num_classes=[0]*len(liste_images) #les classes
    l=[0]*len(liste_images) #les nbr de lignes
    c=[0]*len(liste_images) #les nbr de colonnes
    N=[0]*len(liste_images)  #les nbr de bandes
```

```
    nomImages=[0]*len(liste_images) #les noms des images

    taille=0 #taille de l'image composée
    for i in range(len(liste_images)):
        nomImages[i],l[i],c[i],N[i],num_classes[i]=variables(liste_images[i])
        M[i],Y[i]=fichierMat2matrice(nomImages[i])
        taille+=l[i]*c[i]
        #garder que les classes [2,3,5,8 ,10,11,12,14] de indian pines
        if nomImages[i]=="indian_pines":num_classes[i]=reduireIndian8classes(Y[i],num_classes[i],[2,3,5,8
,10,11,12,14])

        if len(liste_images)>1: #enregistrer les statistiques
            my_callbacks.nom_image="all"
        else:
            my_callbacks.nom_image=nomImages[i]

    M_all=[0]*taille
    k=0
    for i in range(len(M)):
        N[i]=min(N)
        M[i]=PCA_sklearn(M[i],min(N))


        M_all[k:k+len(M[i])]=M[i][:]
        k+=len(M[i])

    print("image composée",len(M_all),len(M_all[0]),max(N))

    #préparer les labels composés
    Y_all=[0]*taille
    k=0
    ajout_classe=0
    for i in range(len(Y)):
        for j in range(len(Y[i])):
            if Y[i][j]!=0:
                Y_all[k]=Y[i][j]+ajout_classe
                Y[i][j]= Y_all[k] #???? je change le label de l'image orig
            else:
                Y_all[k]=0
            k+=1

        ajout_classe+=num_classes[i]-1
    print("label composée",len(Y_all),np.unique(np.array(Y_all)))


    return np.array(M_all),np.array(Y_all),M,Y,l,c,N,num_classes,nomImages

#--------------------------------------------------------------------------
#--------------------------------------------------------------------------
def data_CNN(M_all,Y_all,num_classesGlobal):
    """ prépare  train et test data et data pour vérifier"""
    x_test,y_test,x_train,y_train=[],[],[],[]
    index_classes=[[] for i in range(num_classesGlobal)]
    for i in range(len(Y_all)):
```

```
        index_classes[int(Y_all[i])].append(i) #index_classes[Y[i]] contient les indices des pixels contenant Y[i]
    ll=[]
    index_classes_sample=[ [] for i in range(len(index_classes))]
    for i in range(len(index_classes)):
        nbr=len(index_classes[i])

        if nbr>=200:
            index_classes_sample[i]=sample(index_classes[i],int(nbr*0.70))
            print(i,nbr)
        else:
            index_classes_sample[i]=sample(index_classes[i],nbr//2)
            print(i,nbr)


    for i in range(1,len(index_classes)):
        for j in index_classes[i]:
            if j in index_classes_sample[i]:
                x_train.append(M_all[j])
                y_train.append(Y_all[j])
            else:
                x_test.append(M_all[j])
                y_test.append(Y_all[j])


    M_verif=M_all.copy()
    return (np.array(x_train), np.array(y_train)), (np.array(x_test), np.array(y_test)),M_verif

#----------------------------------------Préparer les données :train & test
def train_test_data(M_all,Y_all,num_classesGlobal,img_rows, img_cols):
    (x_train, y_train), (x_test, y_test) ,M_verif= data_CNN(M_all,Y_all,num_classesGlobal)
    x_train = x_train.reshape((x_train.shape[0], img_rows, img_cols))
    x_test = x_test.reshape((x_test.shape[0], img_rows, img_cols))
    M_verif=M_verif.reshape((M_verif.shape[0], img_rows, img_cols))

    if K.image_data_format() == 'channels_first':
        x_train = x_train.reshape(x_train.shape[0], 1, img_rows, img_cols)
        x_test = x_test.reshape(x_test.shape[0], 1, img_rows, img_cols)
        M_verif = M_verif.reshape(M_verif.shape[0], 1, img_rows, img_cols)

        input_shape = (1, img_rows, img_cols)
    else:
        x_train = x_train.reshape(x_train.shape[0], img_rows, img_cols, 1)
        x_test = x_test.reshape(x_test.shape[0], img_rows, img_cols, 1)
        M_verif = M_verif.reshape(M_verif.shape[0], img_rows, img_cols,1)

        input_shape = (img_rows, img_cols, 1)

    x_train = x_train.astype('float32')
    x_test = x_test.astype('float32')
    M_verif = M_verif.astype('float32')
    x_train /= 255
    x_test /= 255
    M_verif /=255
    print('M_verif shape:', M_verif.shape)
```

```
    print(x_train.shape[0], 'train samples',x_train.shape)
    print(x_test.shape[0], 'test samples',x_test.shape)
    print(y_train.shape, 'train labels')
    print(y_test.shape, 'test labels')

    y_train = keras.utils.to_categorical(y_train, num_classesGlobal)
    y_test = keras.utils.to_categorical(y_test, num_classesGlobal)
    return (x_train, y_train), (x_test, y_test) ,M_verif,input_shape

#------------------------------------------------------------------------#-----------------------------------------------------------------------------
def createModel(input_shape,num_classesGlobal):


    model = Sequential() #a linear stack of layers
    model.add(Conv2D(150, kernel_size=(5, 1),input_shape=input_shape))

    model.add(MaxPooling2D(pool_size=(2, 1)))
    model.add(Conv2D(70, kernel_size=(3, 1),input_shape=input_shape))
    model.add(MaxPooling2D(pool_size=(2, 1)))

    model.add(Conv2D(32, kernel_size=(7, 1),input_shape=input_shape))

    model.add(MaxPooling2D(pool_size=(2, 1)))
    model.add(Dropout(0.5))

    model.add(Flatten())
    model.add(Dense(num_classesGlobal))

    model.add(Activation('softmax'))
    return model
#------------------------------------------------------------------------
def save_model(model):
    # serialize model to JSON
    model_json = model.to_json()
    with open(path_data+"model.json", "w") as json_file:
        json_file.write(model_json)
    # serialize weights to HDF5
    model.save_weights(path_data+"model.h5")
    print("Saved model to disk")
#------------------------------------------------------------------------
def load_model(model_name="model.json"):
    # load json and create model
    json_file = open(path_data+model_name, 'r')
    loaded_model_json = json_file.read()
    json_file.close()
    loaded_model = model_from_json(loaded_model_json)
    # load weights into new model
    loaded_model.load_weights(path_data+"model.h5")
    print("Loaded model from disk")
    return loaded_model
#------------------------------------------------------------------------
def plot_GoundTruth(I_Y,nomImage,couleurs,I_lignes,I_colonnes):
    #concatener les images verticalement
    I=sum(I_lignes)
```

```
    c=max(l_colonnes)

    M_plot=np.zeros((l,c))

    t=0
    for k in range(len(l_Y)): #l_Y[k]:image k
        M_plot[t:t+l_lignes[k],:l_colonnes[k]]=np.array(l_Y[k]).reshape(l_lignes[k],l_colonnes[k])
        t+=l_lignes[k]


    M_Couleur=[]

    for i in range(l):
        for j in range(c):
            M_Couleur.append(couleurs[M_plot[i,j]])


    imNew=Image.new("RGB",(c,l))
    imNew.putdata(M_Couleur)
    imNew.save(path_data+"/img/Orig_"+nomImage+".jpg")
    imNew.show()


def plot_pred(classes,Y,l,c,nomImage,couleurs):
    classes=classes.argmax(axis=1)
    classes=classes.reshape(l,c)



    MACPCouleur=[]
    k=0
    for i in range(l):
        for j in range(c):
            if Y[k]==0:
                MACPCouleur.append(couleurs[0])
            else:
                MACPCouleur.append(couleurs[classes[i,j]])
            k+=1



    imNew=Image.new("RGB",(c,l))
    imNew.putdata(MACPCouleur)
    imNew.save(path_data+"/img/Pred_"+nomImage+".jpg")
    imNew.show()

def plot_pred_all(classes,l_Y,Y_all,nomImage,couleurs,l_lignes,l_colonnes):
     #concatener les images verticalement
    l=sum(l_lignes)
    c=max(l_colonnes)

    classes=classes.argmax(axis=1)
    for i in range(len(classes)):
        if Y_all[i]==0: classes[i]=0
```

```
    M_plot=np.zeros((l,c))

    t1=0
    t2=0
    for k in range(len(l_Y)): #l_Y[k]:image k

M_plot[t1:t1+l_lignes[k],:l_colonnes[k]]=classes[t2:t2+l_lignes[k]*l_colonnes[k]].reshape(l_lignes[k],l_colonnes[k]
)
        t1+=l_lignes[k]
        t2+=l_lignes[k]*l_colonnes[k]

    M_Couleur=[]
    k=0
    for i in range(l):
        for j in range(c):
            M_Couleur.append(couleurs[M_plot[i,j]])
            k+=1



    imNew=Image.new("RGB",(c,l))
    imNew.putdata(M_Couleur)
    imNew.save(path_data+"/img/Pred_"+nomImage+".jpg")
    imNew.show()
#-------------------------------------------------------------------------
def bar_with_classes(P_orig,P,num_classes,couleurs):
    X=np.array([i for i in range(0,num_classes)])

    for i in range(1,num_classes):
        plt.bar([X[i]],[P_orig[i]],1,color=np.array(couleurs[i])/255)
    plt.bar(X[1:]+0.2,P[1:],0.6,color="black",label="nbr effectifs préd")
    plt.xticks(np.arange(1,num_classes))


    plt.legend()
    plt.show()

#-------------------------------------------------------------------------
def pourcentage_pred(classes,Y_label,num_classes,couleurs):
    Y_label=list(Y_label)
    P=[0 for i in range(num_classes)] #pred correcte
    P_orig=[Y_label.count(i) for i in range(num_classes)] #avant pred
    classes=classes.argmax(axis=1)
    for i in range(len(classes)):
        if Y_label[i]!=0 and classes[i]==Y_label[i]:
            ind=int(Y_label[i])
            P[ind]+=1
    p_tt=(sum(P[1:])/sum(P_orig[1:]))*100
    bar_with_classes(P_orig,P,num_classes,couleurs)

    return P,P_orig,p_tt #effectifs correct,effectifs orig,pourcentage pred tt

def generate_colors(nbr_c):
```

```
    Lcouleurs=[(0,0,0),(0,0,255),(255, 0, 0),(88, 41, 0),(0, 255, 0),(253, 108, 158),(255, 255, 0),(237, 127,
16),(102, 0, 153),(223, 115, 255),
              (64, 130, 109),(149, 165, 149),(254, 191, 210),(254, 163, 71),(255, 240, 188),(153, 122, 144),(254,
195, 172)]
    while len(Lcouleurs)<nbr_c:
        c=(randint(1,255),randint(1,255),randint(1,255))
        if c not in Lcouleurs:
            Lcouleurs.append(c)
        else:print(c,"existe")

    couleurs={ i: Lcouleurs[i] for i in range(len(Lcouleurs))}


    return couleurs

def Produit_tensoriel(A,nbr_ajout):
    MT=np.zeros((len(A),len(A[0])+nbr_ajout))
    print(MT.shape)
    k=0
    i=0
    while k<len(MT[0]):
        if i<nbr_ajout:
            MT[:,k]=A[:,i%len(A[0])]
            if k+1<len(MT[0]): MT[:,k+1]=A[:,i%len(A[0])]
            i+=1
            k+=2
        else:
            MT[:,k]=A[:,i%len(A[0])]
            i+=1
            k+=1

    return MT



#-------------------------------------------------------------------
#-------------------------------------------------------------------
#----Programme Principal-------------------------------------------
#-------------------------------------------------------------------
batch_size = 128
batch_size = 128
L=[[10],[15],[18],[10,15,18]]
L=[[15,18]]
epochs = int(input("nbr epoche:"))

for liste_images in L:

    M_all,Y_all,M,Y,l,c,N,num_classes,nomImages=alldata2One(liste_images)
    num_classesGlobal=sum([e-1 for e in  num_classes])+1
    couleurs=generate_colors(num_classesGlobal)



    img_rows, img_cols=max(N),1    # input image dimensions :Pixel
```

```
    print("taille M_all orig: ",M_all.shape,"Labels all orig:", Y_all.shape,"classes all:",num_classesGlobal)

    (x_train, y_train), (x_test, y_test)
,M_verif,input_shape=train_test_data(M_all,Y_all,num_classesGlobal,img_rows, img_cols)
    #------------------------------------------------------------------

    type_op=1#int(input("New model or load lodel (1 or 0) :"))
    if type_op==1: #creation d'un nouveau modèle

        model=createModel(input_shape,num_classesGlobal)
    else:
        model=load_model(model_name=path_data+my_callbacks.nom_image+"_model.json")

    #---------------------------------------configure the learning process
    model.compile(loss=keras.losses.categorical_crossentropy,
                  optimizer=keras.optimizers.Adadelta(),
                  metrics=['accuracy'])
    # prepare callback
    histories = my_callbacks.Histories()


    #---------------------------------------You can now iterate on your training data in batches:
    #on a 4 valeurs pour chaque epoche :loss: - acc: train - val_loss: validation - val_acc: validation

    if type_op==1: #creation d'un nouveau modèle
        model.fit(x_train, y_train,
                  batch_size=batch_size,
                  epochs=epochs,
                  verbose=0,
                  validation_data=(x_test, y_test),callbacks=[histories])

        save_model(model)
```

Bibliographie

[1] Amir GANDOMI et Murtaza HAIDER. « Beyond the hype : Big data concepts, methods, and analytics ». In : *International journal of information management* 35.2 (2015), p. 137-144.

[2] Jeff REED. *Data Analytics : Applicable Data Analysis to Advance Any Business Using the Power of Data Driven Analytics*. 2017.

[3] Sridhar ALLA. *Big Data Analytics with Hadoop 3 : Build highly effective analytics solutions to gain valuable insight into your big data*. 2018.

[4] Doug LANEY. « 3d data management : Controlling data volume ». In : *Velocity and Variety* (2001).

[5] Hsinchun CHEN, Roger HL CHIANG et Veda C STOREY. « Business intelligence and analytics : From big data to big impact. » In : *MIS quarterly* 36.4 (2012).

[6] Ohbyung KWON, Namyeon LEE et Bongsik SHIN. « Data quality management, data usage experience and acquisition intention of big data analytics ». In : *International journal of information management* 34.3 (2014), p. 387-394.

[7] In LEE. « Big data : Dimensions, evolution, impacts, and challenges ». In : *Business Horizons* 60.3 (2017), p. 293-303.

[8] Richard K LOMOTEY et Ralph DETERS. « Towards knowledge discovery in big data ». In : *2014 IEEE 8th International Symposium on Service Oriented System Engineering*. IEEE. 2014, p. 181-191.

[9] Hafidha AL-BARASHDI et Rahma AL-KAROUSI. « Big Data in academic libraries : literature review and future research directions ». In : *Journal of Information Studies & Technology (JIS&T)* 2018.2 (2019), p. 13.

[10] Marco POSPIECH et Carsten FELDEN. « Big data–a state-of-the-art ». In : (2012).

[11] Ryan R CURTIN et al. « MLPACK : A scalable C++ machine learning library ». In : *Journal of Machine Learning Research* 14.Mar (2013), p. 801-805.

[12] *Official apache hadoop*. URL : http://hadoop.apache.org/.

[13] Dilpreet SINGH et Chandan K REDDY. « A survey on platforms for big data analytics ». In : *Journal of big data* 2.1 (2015), p. 8.

[14] Sara LANDSET et al. « A survey of open source tools for machine learning with big data in the Hadoop ecosystem ». In : *Journal of Big Data* 2.1 (2015), p. 24.

[15] Xiufeng LIU, Nadeem IFTIKHAR et Xike XIE. « Survey of real-time processing systems for big data ». In : *Proceedings of the 18th International Database Engineering & Applications Symposium*. ACM. 2014, p. 356-361.

[16] Jeffrey DEAN et Sanjay GHEMAWAT. « MapReduce : simplified data processing on large clusters ». In : *Communications of the ACM* 51.1 (2008), p. 107-113.

[17] Sanjay GHEMAWAT, Howard GOBIOFF et Shun-Tak LEUNG. « The Google file system ». In : (2003).

[18] Tom WHITE. *Hadoop : The definitive guide*. " O'Reilly Media, Inc.", 2012.

[19] Matei ZAHARIA et al. « Spark : Cluster computing with working sets. » In : *HotCloud* 10.10-10 (2010), p. 95.

[20] Craig CHAMBERS et al. « FlumeJava : easy, efficient data-parallel pipelines ». In : *ACM Sigplan Notices*. T. 45. 6. ACM. 2010, p. 363-375.

[21] Nishant GARG. *Apache Kafka*. Packt Publishing Ltd, 2013.

[22] Michael ARMBRUST et al. « Spark sql : Relational data processing in spark ». In : *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. ACM. 2015, p. 1383-1394.

[23] Reynold S XIN et al. « Graphx : A resilient distributed graph system on spark ». In : *First International Workshop on Graph Data Management Experiences and Systems*. ACM. 2013, p. 2.

[24] Grzegorz MALEWICZ et al. « Pregel : a system for large-scale graph processing ». In : *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM. 2010, p. 135-146.

[25] Wei FAN et Albert BIFET. « Mining big data : current status, and forecast to the future ». In : *ACM sIGKDD Explorations Newsletter* 14.2 (2013), p. 1-5.

[26] Francis X DIEBOLD. « On the Origin (s) and Development of the Term'Big Data' ». In : (2012).

[27] Sholom M WEISS et Nitin INDURKHYA. *Predictive data mining : a practical guide*. Morgan Kaufmann, 1998.

[28] Ming-Chao CHIANG, Chun-Wei TSAI et Chu-Sing YANG. « A time-efficient pattern reduction algorithm for k-means clustering ». In : *Information Sciences* 181.4 (2011), p. 716-731.

[29] Adil FAHAD et al. « A survey of clustering algorithms for big data : Taxonomy and empirical analysis ». In : *IEEE transactions on emerging topics in computing* 2.3 (2014), p. 267-279.

[30] Rui XU et Donald C WUNSCH. « II, Clustering. Hoboken ». In : *NJ : Wiley/IEEE Press* 6 (2009), p. 583-617.

[31] Ali Seyed SHIRKHORSHIDI et al. « Big data clustering : a review ». In : *International conference on computational science and its applications*. Springer. 2014, p. 707-720.

[32] Tian ZHANG, Raghu RAMAKRISHNAN et Miron LIVNY. « BIRCH : an efficient data clustering method for very large databases ». In : *ACM Sigmod Record*. T. 25. 2. ACM. 1996, p. 103-114.

[33] Huiqi XU et al. « Cloudvista : interactive and economical visual cluster analysis for big data in the cloud ». In : *Proceedings of the VLDB Endowment* 5.12 (2012), p. 1886-1889.

[34] Xiaohui CUI, Jinzhu GAO et Thomas E POTOK. « A flocking based algorithm for document clustering analysis ». In : *Journal of systems architecture* 52.8-9 (2006), p. 505-515.

[35] Xiaohui CUI, Jesse St CHARLES et Thomas POTOK. « GPU enhanced parallel computing for large scale data clustering ». In : *Future Generation Computer Systems* 29.7 (2013), p. 1736-1741.

[36] Dan FELDMAN, Melanie SCHMIDT et Christian SOHLER. « Turning big data into tiny data : Constant-size coresets for k-means, pca and projective clustering ». In : *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial et Applied Mathematics. 2013, p. 1434-1453.

[37] Cem TEKIN et Mihaela VAN DER SCHAAR. « Distributed online big data classification using context information ». In : *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE. 2013, p. 1435-1442.

[38] Patrick REBENTROST, Masoud MOHSENI et Seth LLOYD. « Quantum support vector machine for big data classification ». In : *Physical review letters* 113.13 (2014), p. 130503.

[39] Jiawei HAN, Jian PEI et Yiwen YIN. « Mining frequent patterns without candidate generation ». In : *ACM sigmod record*. T. 29. 2. ACM. 2000, p. 1-12.

[40] Carson Kai-Sang LEUNG, Richard Kyle MACKINNON et Fan JIANG. « Reducing the search space for big data mining for interesting patterns from uncertain data ». In : *2014 IEEE International Congress on Big Data*. IEEE. 2014, p. 315-322.

[41] Ming-Yen LIN, Pei-Yu LEE et Sue-Chen HSUEH. « Apriori-based frequent itemset mining algorithms on MapReduce ». In : *Proceedings of the 6th international conference on ubiquitous information management and communication*. ACM. 2012, p. 76.

[42] Jen-Wei HUANG, Su-Chen LIN et Ming-Syan CHEN. « DPSP : Distributed progressive sequential pattern mining on the cloud ». In : *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2010, p. 27-34.

[43] Lai YANG et al. « DH-TRIE frequent pattern mining on Hadoop using JPA ». In : *2011 IEEE International Conference on Granular Computing*. IEEE. 2011, p. 875-878.

[44] Sushmita MITRA, Sankar K PAL et Pabitra MITRA. « Data mining in soft computing framework : a survey ». In : *IEEE transactions on neural networks* 13.1 (2002), p. 3-14.

[45] Ian H WITTEN et al. *Data Mining : Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[46] K KRISHNA et Narasimha M MURTY. « Genetic K-means algorithm ». In : *IEEE Transactions on Systems Man And Cybernetics-Part B : Cybernetics* 29.3 (1999), p. 433-439.

[47] Mehmet KAYA et Reda ALHAJJ. « Genetic algorithm based framework for mining fuzzy association rules ». In : *Fuzzy sets and systems* 152.3 (2005), p. 587-601.

[48] Chuang MA, Hao Helen ZHANG et Xiangfeng WANG. « Machine learning for big data analytics in plants ». In : *Trends in plant science* 19.12 (2014), p. 798-808.

[49] Erick CANTÚ-PAZ. « A survey of parallel genetic algorithms ». In : *Calculateurs paralleles, reseaux et systems repartis* 10.2 (1998), p. 141-171.

[50] Chun-Wei TSAI et al. « Big data analytics : a survey ». In : *Journal of Big data* 2.1 (2015), p. 21.

[51] Yingyi BU et al. « Scaling datalog for machine learning on big data ». In : *arXiv preprint arXiv :1203.0160* (2012).

[52] *Mahout machine learning library*. URL : http://mahout.apache.org/.

[53] Ku Ruhana KU-MAHAMUD. « Big data clustering using grid computing and ant-based algorithm ». In : *Proceedings of the Inter-national Conference on Computing and Informatics*. 2013, p. 6-14.

[54] Jean-Louis DENEUBOURG et al. « The dynamics of collective sorting robot-like ants and ant-like robots ». In : *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*. 1991, p. 356-363.

[55] Shafaatunnur HASAN, Siti Mariyam SHAMSUDDIN et Noel LOPES. « Soft computing methods for big data problems ». In : *GPU Computing and Applications*. Springer, 2015, p. 235-247.

[56] Ahmed OUSSOUS et al. « Big Data technologies : A survey ». In : *Journal of King Saud University-Computer and Information Sciences* 30.4 (2018), p. 431-448.

[57] CL Philip CHEN et Chun-Yang ZHANG. « Data-intensive applications, challenges, techniques and technologies : A survey on Big Data ». In : *Information sciences* 275 (2014), p. 314-347.

[58] Anwaar ALI et al. « Big data for development : applications and techniques ». In : *Big Data Analytics* 1.1 (2016), p. 2.

[59] Maryam M NAJAFABADI et al. « Deep learning applications and challenges in big data analytics ». In : *Journal of Big Data* 2.1 (2015), p. 1.

[60] Nawsher KHAN et al. « Big data : survey, technologies, opportunities, and challenges ». In : *The Scientific World Journal* 2014 (2014).

[61] Junfei QIU et al. « A survey of machine learning for big data processing ». In : *EURASIP Journal on Advances in Signal Processing* 2016.1 (2016), p. 67.

[62] Wullianallur RAGHUPATHI et Viju RAGHUPATHI. « Big data analytics in healthcare : promise and potential ». In : *Health information science and systems* 2.1 (2014), p. 3.

[63] Hai WANG et al. « Towards felicitous decision making : An overview on challenges and trends of Big Data ». In : *Information Sciences* 367 (2016), p. 747-765.

[64] Wendy Arianne GÜNTHER et al. « Debating big data : A literature review on realizing value from big data ». In : *The Journal of Strategic Information Systems* 26.3 (2017), p. 191-209.

[65] Hamid EKBIA et al. « Big data, bigger dilemmas : A critical review ». In : *Journal of the Association for Information Science and Technology* 66.8 (2015), p. 1523-1545.

[66] Anna KARPOVSKY et Robert D GALLIERS. « Aligning in practice : from current cases to a new agenda ». In : *Journal of Information Technology* 30.2 (2015), p. 136-160.

[67] HV JAGADISH et al. « Big data and its technical challenges ». In : *Communications of the ACM* 57.7 (2014), p. 86-94.

[68] Roger CLARKE. « Big data, big risks ». In : *Information Systems Journal* 26.1 (2016), p. 77-90.

[69] *Earth Science Data Systems (ESDS) Program*. URL : https://earthdata.nasa.gov/esds.

[70] *2019 Conference on Big Data from Space (BiDS'19) Turning Data into Insights - ISSN :1831-9424*. URL : https://www.bigdatafromspace2019.org/

QuickEventWebsitePortal/2019-conference-on-big-data-from-space-bids19/bids-2019.

[71] *Partnership of more than 100 national governments*. URL : https://www.earthobservations.org/index.php/.

[72] *Explore geospatial data*. URL : https://www.digitalglobe.com/.

[73] *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. URL : https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=4609443.

[74] *Journal of Applied Remote Sensing*. URL : https://www.spiedigitallibrary.org/journals/journal-of-applied-remote-sensing?SSO=1.

[75] *IEEE Geoscience and Remote Sensing Magazine*. URL : https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=6245518.

[76] *An International Journal on Advances of Computer Science for Geographic Information Systems*. URL : https://link.springer.com/journal/10707.

[77] *Environmental Remote Sensing*. URL : https://www.mdpi.com/journal/remotesensing/sections/environmental_remote_sensing.

[78] *Remote Sensing — Open Access Journal*. URL : https://www.mdpi.com/journal/remotesensing.

[79] *International Journal of Digital Earth*. URL : https://www.tandfonline.com/loi/tjde20.

[80] Hua WU et Zhao-Liang LI. « Scale issues in remote sensing : A review on analysis, processing and modeling ». In : *Sensors* 9.3 (2009), p. 1768-1793.

[81] Sona SALEHIYAN, Hossein AREFI et Reza SHAH-HOSEINI. « Fusion of UAV-SAR and Quickbird data for Urban Growth Detection ». In : (2019).

[82] Christian SCHWATKE, Daniel SCHERER et Denise DETTMERING. « AWAX : A new Approach for Automated Extraction of Consistent Time-Variable Water Surfaces of Lakes and Reservoirs using Landsat and Sentinel-2 ». In : *IUGG General Assembly 2019, Montreal, Canada*. 2019.

[83] Saysongkham SAYAVONG et al. « Mapping rubber stand ages in Luangnamtha district (Northern Laos) using NDVI and LSWI from Landsat images ». In : *Asia-Pacific Journal of Science and Technology* 24.2 (2019).

[84] L FANG et al. « Measuring surface water salinity of Pearl River Estuary by MODIS 250-m imageries ». In : *Journal of Environmental Biology* 40.3 (2019), p. 472-485.

[85] *AVIRIS - Airborne Visible/Infrared Imaging Spectrometer - Data*. URL : http://aviris.jpl.nasa.gov/data/image_cube.html.

[86] Muhammad Jaleed KHAN et al. « Modern trends in hyperspectral image analysis : a review ». In : *IEEE Access* 6 (2018), p. 14118-14129.

[87] Richard BELLMAN et Robert E KALABA. *Dynamic programming and modern control theory*. T. 81. Citeseer, 1965.

[88] PJ ZARCO-TEJADA et al. « Understanding the temporal dimension of the red-edge spectral region for forest decline detection using high-resolution hyperspectral and Sentinel-2a imagery ». In : *ISPRS journal of photogrammetry and remote sensing* 137 (2018), p. 134-148.

[89] David TILMAN et al. « Global food demand and the sustainable intensification of agriculture ». In : *Proceedings of the National Academy of Sciences* 108.50 (2011), p. 20260-20264.

[90] Kenneth G CASSMAN. « Ecological intensification of cereal production systems : yield potential, soil quality, and precision agriculture ». In : *Proceedings of the National Academy of Sciences* 96.11 (1999), p. 5952-5959.

[91] Deepak K RAY et al. « Recent patterns of crop yield growth and stagnation ». In : *Nature communications* 3 (2012), p. 1293.

[92] H Charles J GODFRAY et al. « Food security : the challenge of feeding 9 billion people ». In : *science* 327.5967 (2010), p. 812-818.

[93] Jonathan A FOLEY et al. « Solutions for a cultivated planet ». In : *Nature* 478.7369 (2011), p. 337.

[94] Prabhu L PINGALI. « Green revolution : impacts, limits, and the path ahead ». In : *Proceedings of the National Academy of Sciences* 109.31 (2012), p. 12302-12308.

[95] Liangpei ZHANG, Lefei ZHANG et Bo DU. « Deep learning for remote sensing data : A technical tutorial on the state of the art ». In : *IEEE Geoscience and Remote Sensing Magazine* 4.2 (2016), p. 22-40.

[96] R COLOMBO et al. « Estimation of leaf and canopy water content in poplar plantations by means of hyperspectral indices and inverse modeling ». In : *Remote Sensing of Environment* 112.4 (2008), p. 1820-1834.

[97] Uwe RASCHER et al. « Monitoring spatio-temporal dynamics of photosynthesis with a portable hyperspectral imaging system ». In : *Photogrammetric Engineering & Remote Sensing* 73.1 (2007), p. 45-56.

[98] Micol ROSSINI et al. « Assessing canopy PRI from airborne imagery to map water stress in maize ». In : *ISPRS Journal of Photogrammetry and Remote Sensing* 86 (2013), p. 168-177.

[99] Andrew C SCHUERGER et al. « Comparison of two hyperspectral imaging and two laser-induced fluorescence instruments for the detection of zinc stress and chlorophyll concentration in bahia grass (Paspalum notatum Flugge.) » In : *Remote sensing of environment* 84.4 (2003), p. 572-588.

[100] LY LIU et al. « Improving winter wheat yield prediction by novel spectral index ». In : *Trans. CSAE* 20 (2004), p. 172-175.

[101] Richard SOCHER et al. « Recursive deep models for semantic compositionality over a sentiment treebank ». In : *Proceedings of the 2013 conference on empirical methods in natural language processing*. 2013, p. 1631-1642.

[102] Yoon KIM. « Convolutional neural networks for sentence classification ». In : *arXiv preprint arXiv :1408.5882* (2014).

[103] Joonatas WEHRMANN et al. « A character-based convolutional neural network for language-agnostic Twitter sentiment analysis ». In : *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2017, p. 2384-2391.

[104] Dzmitry BAHDANAU, Kyunghyun CHO et Yoshua BENGIO. « Neural machine translation by jointly learning to align and translate ». In : *arXiv preprint arXiv :1409.0473* (2014).

[105] Kyunghyun CHO et al. « Learning phrase representations using RNN encoder-decoder for statistical machine translation ». In : *arXiv preprint arXiv :1406.1078* (2014).

[106] Yonghui WU et al. « Google's neural machine translation system : Bridging the gap between human and machine translation ». In : *arXiv preprint arXiv :1609.08144* (2016).

[107] Richard SOCHER et al. « Dynamic pooling and unfolding recursive autoencoders for paraphrase detection ». In : *Advances in neural information processing systems*. 2011, p. 801-809.

[108] Wenpeng YIN et al. « Abcnn : Attention-based convolutional neural network for modeling sentence pairs ». In : *Transactions of the Association for Computational Linguistics* 4 (2016), p. 259-272.

[109] Mikael KÅGEBÄCK et al. « Extractive summarization using continuous vector space models ». In : *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*. 2014, p. 31-39.

[110] Li DONG et al. « Question answering over freebase with multi-column convolutional neural networks ». In : *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*. 2015, p. 260-269.

[111] Minwei FENG et al. « Applying deep learning to answer selection : A study and an open task ». In : *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE. 2015, p. 813-820.

[112] Igor MOZETIČ, Miha GRČAR et Jasmina SMAILOVIĆ. « Multilingual Twitter sentiment classification : The role of human annotators ». In : *PloS one* 11.5 (2016), e0155036.

[113] Ibrahim BADR, Rabih ZBIB et James GLASS. « Segmentation for English-to-Arabic statistical machine translation ». In : *Proceedings of ACL-08 : HLT, Short Papers*. 2008, p. 153-156.

[114] Ibrahim BADR, Rabih ZBIB et James GLASS. « Syntactic phrase reordering for English-to-Arabic statistical machine translation ». In : *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. 2009, p. 86-93.

[115] Ahmed EL KHOLY et Nizar HABASH. « Techniques for Arabic morphological detokenization and orthographic denormalization ». In : *Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC), Valletta, Malta*. 2010.

[116] Hassan AL-HAJ et Alon LAVIE. « The impact of Arabic morphological segmentation on broad-coverage English-to-Arabic statistical machine translation ». In : *Machine translation* 26.1-2 (2012), p. 3-24.

[117] Nizar HABASH et Jun HU. « Improving Arabic-Chinese statistical machine translation using English as pivot language ». In : *Proceedings of the Fourth Workshop on Statistical Machine Translation*. Association for Computational Linguistics. 2009, p. 173-181.

[118] Mossab AL-HUNAITY, Bente MAEGAARD et Dorte HANSEN. « Using English as a pivot language to enhance Danish-Arabic statistical machine translation ». In : *Editors & Workshop Chairs*. 2010, p. 108.

[119] Kavita GANESAN, ChengXiang ZHAI et Jiawei HAN. « Opinosis : A graph based approach to abstractive summarization of highly redundant opinions ». In : *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. 2010, p. 340-348.

[120] R.M DUWAIRI. « Arabic text categorization ». In : *Int. Arab J. Inf. Technol 4(2), 125-132* (2007).

[121] M Ali FAUZI, Agus Zainal ARIFIN et Anny YUNIARTI. « Arabic Book Retrieval using Class and Book Index Based Term Weighting ». In : *International Journal of Electrical and Computer Engineering (IJECE)* 7.6 (2017), p. 3705-3710.

[122] Xiang ZHANG et Yann LECUN. « Byte-Level Recursive Convolutional Auto-Encoder for Text ». In : *arXiv preprint arXiv :1802.01817* (2018).

[123] Ángel MORERA et al. « Gender and handedness prediction from offline handwriting using convolutional neural networks ». In : *Complexity* 2018 (2018).

[124] Saad Bin AHMED et al. « Sub-sampling Approach for Unconstrained Arabic Scene Text Analysis by Implicit Segmentation based Deep Learning Classifier ». In : *Global Journal of Computer Science and Technology* (2019).

[125] Ludovic MERCIER. « Système d'analyse et de visualisation d'images hyperspectrales appliqué aux sciences planétaires ». Mém. de mast. Avr. 2011, p. 101. URL : https://dumas.ccsd.cnrs.fr/dumas-00592170.

[126] Zebin WU et al. « Parallel and distributed dimensionality reduction of hyperspectral data on cloud computing architectures ». In : *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9.6 (2016), p. 2270-2278.

[127] Sam T ROWEIS et Lawrence K SAUL. « Nonlinear dimensionality reduction by locally linear embedding ». In : *science* 290.5500 (2000), p. 2323-2326.

[128] Laurens VAN DER MAATEN, Eric POSTMA et Jaap VAN DEN HERIK. « Dimensionality reduction : a comparative ». In : *J Mach Learn Res* 10.66-71 (2009), p. 13.

[129] Tarek ELGAMAL et al. « spca : Scalable principal component analysis for big data on distributed platforms ». In : *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM. 2015, p. 79-91.

[130] Jonathon SHLENS. « A tutorial on principal component analysis ». In : *arXiv preprint arXiv :1404.1100* (2014).

[131] *MLlib machine learning library*. URL : https://spark.apache.org/mllib.

[132] Matei ZAHARIA et al. « Resilient distributed datasets : A fault-tolerant abstraction for in-memory cluster computing ». In : *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*. USENIX Association. 2012, p. 2-2.

[133] Yasser KHOUJ et al. « Hyperspectral imaging and K-means classification for histologic evaluation of ductal carcinoma in situ ». In : *Frontiers in oncology* 8 (2018), p. 17.

[134] Hongjun SU, Qian DU et Peijun DU. « Hyperspectral image visualization using band selection ». In : *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7.6 (2013), p. 2647-2658.

[135] Yuan YUAN, Guokang ZHU et Qi WANG. « Hyperspectral band selection by multitask sparsity pursuit ». In : *IEEE Transactions on Geoscience and Remote Sensing* 53.2 (2014), p. 631-644.

[136] Guokang ZHU et al. « Unsupervised hyperspectral band selection by dominant set extraction ». In : *IEEE Transactions on Geoscience and Remote Sensing* 54.1 (2015), p. 227-239.

[137] Ketan KOTWAL et Subhasis CHAUDHURI. « Visualization of hyperspectral images using bilateral filtering ». In : *IEEE Transactions on Geoscience and Remote Sensing* 48.5 (2010), p. 2308-2316.

[138] Max MIGNOTTE. « A bicriteria-optimization-approach-based dimensionality-reduction model for the color display of hyperspectral images ». In : *IEEE Transactions on Geoscience and Remote Sensing* 50.2 (2011), p. 501-513.

[139] Ch THEOHARATOS et al. « Hyperspectral image fusion using 2-D principal component analysis ». In : *2011 2nd International Conference on Space Technology*. IEEE. 2011, p. 1-4.

[140] *Apache Spark - Lightning-Fast Cluster Computing*. URL : http://spark.apache.org.

[141] Hongwen LIN, Anqing ZHANG et Shaoqing YANG. « Comparison of Several Hyperspectral Image Fusion Methods for Visualization ». In : (2015).

[142] Hongqin ZHANG, David W MESSINGER et Ethan D MONTAG. « Perceptual display strategies of hyperspectral imagery based on PCA and ICA ». In : *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XII*. T. 6233. International Society for Optics et Photonics. 2006, p. 62330X.

[143] *Making Big Data Simple*. URL : https://databricks.com/.

[144] Biernat E LUTZ M. « Data science : fondamentaux et études de cas : Machine learning avec Python et R - Oct 2015 ». In : Eyrolles, 2015.

[145] Muhammad Imran RAZZAK, Saeeda NAZ et Ahmad ZAIB. « Deep learning for medical image processing : Overview, challenges and the future ». In : *Classification in BioApps*. Springer, 2018, p. 323-350.

[146] ME PAOLETTI et al. « A new deep convolutional neural network for fast hyperspectral image classification ». In : *ISPRS journal of photogrammetry and remote sensing* 145 (2018), p. 120-147.

[147] Zilong ZHONG et al. « Deep residual networks for hyperspectral image classification ». In : *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE. 2017, p. 1824-1827.

[148] Liang LUO et al. « Motivating in-network aggregation for distributed deep neural network training ». In : *Workshop on Approximate Computing Across the Stack*. 2017.

[149] Christian SZEGEDY et al. « Going deeper with convolutions ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, p. 1-9.

[150] Wei HU et al. « Deep convolutional neural networks for hyperspectral image classification ». In : *Journal of Sensors* 2015 (2015).

[151] Yushi CHEN et al. « Deep feature extraction and classification of hyperspectral images based on convolutional neural networks ». In : *IEEE Transactions on Geoscience and Remote Sensing* 54.10 (2016), p. 6232-6251.

[152] Shaohui MEI et al. « Hyperspectral image spatial super-resolution via 3D full convolutional neural network ». In : *Remote Sensing* 9.11 (2017), p. 1139.

[153] Haokui ZHANG et al. « Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network ». In : *Remote Sensing Letters* 8.5 (2017), p. 438-447.

[154] Jun YUE et al. « Spectral–spatial classification of hyperspectral images using deep convolutional neural networks ». In : *Remote Sensing Letters* 6.6 (2015), p. 468-477.

[155] Jun YUE, Shanjun MAO et Mei LI. « A deep learning framework for hyperspectral image classification using spatial pyramid pooling ». In : *Remote Sensing Letters* 7.9 (2016), p. 875-884.

[156] Hyungtae LEE et Heesung KWON. « Going deeper with contextual CNN for hyperspectral image classification ». In : *IEEE Transactions on Image Processing* 26.10 (2017), p. 4843-4855.

[157] Jingxiang YANG, Yong-Qiang ZHAO et Jonathan Cheung-Wai CHAN. « Learning and transferring deep joint spectral–spatial features for hyperspectral classification ». In : *IEEE Transactions on Geoscience and Remote Sensing* 55.8 (2017), p. 4729-4742.

[158] Abdelali ZBAKH et al. « Proposition of a Parallel and Distributed Algorithm for the Dimensionality Reduction with Apache Spark ». In : *Proceedings of the Mediterranean Symposium on Smart City Applications*. Springer. 2017, p. 490-501.

[159] URL : www.ehu.eus/ccwintco/uploads/e/e3/Pavia.mat;.

[160] URL : www.ehu.eus/ccwintco/uploads/f/f1/Salinas.mat.

[161] Wenzhe SHI et al. « Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 1874-1883.

[162] Wang LING et al. « Finding function in form : Compositional character models for open vocabulary word representation ». In : *arXiv preprint arXiv :1508.02096* (2015).

[163] Tao WANG et al. « End-to-end text recognition with convolutional neural networks ». In : *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE. 2012, p. 3304-3308.

[164] Patrice Y SIMARD et al. « Using character recognition and segmentation to tell computer from humans ». In : *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* IEEE. 2003, p. 418-423.

[165] Adam COATES et al. « Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning. » In : *ICDAR*. T. 11. 2011, p. 440-445.

[166] Abdelwadood MOH'D A MESLEH. « Chi square feature extraction based svms arabic language text categorization system ». In : *Journal of Computer Science* 3.6 (2007), p. 430-435.

[167] Aurangzeb KHAN et al. « A review of machine learning algorithms for text-documents classification ». In : *Journal of advances in information technology* 1.1 (2010), p. 4-20.

[168] Jafar ABABNEH et al. « Vector space models to classify Arabic text ». In : *International Journal of Computer Trends and Technology (IJCTT)* 7.4 (2014), p. 219-223.

[169] Mohamed EL KOURDI, Amine BENSAID et Tajje-eddine RACHIDI. « Automatic Arabic document categorization based on the Naive Bayes algorithm ». In : *proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*. Association for Computational Linguistics. 2004, p. 51-58.

[170] Aya AL-ZOGHBY et al. « Mining Arabic text using soft-matching association rules ». In : *2007 International Conference on Computer Engineering & Systems*. IEEE. 2007, p. 421-426.

[171] S AL-HARBI et al. « Automatic Arabic text classification ». In : (2008).

[172] Yailé CABALLERO et al. « Two new feature selection algorithms with Rough Sets Theory ». In : *IFIP International Conference on Artificial Intelligence in Theory and Practice*. Springer. 2006, p. 209-216.

[173] Sasa HASAN, Anas EL ISBIHANI et Hermann NEY. « Creating a Large-Scale Arabic to French Statistical MachineTranslation System. » In : *LREC*. 2006, p. 855-858.

[174] *NVIDIA-CUDA*. URL : https://developer.nvidia.com/.

[175] *Apache Storm*. URL : http://storm.apache.org/.

[176] Wendy Arianne GÜNTHER et al. « Debating big data : A literature review on realizing value from big data ». In : *The Journal of Strategic Information Systems* 26.3 (2017), p. 191-209.

[177] Ren LI et al. « MapReduce parallel programming model : a state-of-the-art survey ». In : *International Journal of Parallel Programming* 44.4 (2016), p. 832-866.

[178] C. H. CHEN. « Information Processing for Remote Sensing ». In : *World Scientific Publishing Company* (2000).

[179] Yann LECUN et al. « Gradient-based learning applied to document recognition ». In : *Proceedings of the IEEE* 86.11 (1998), p. 2278-2324.

[180] AUTHOR. *Ibn Taymiya. Book of Al Iman. Fifth Edition. 1996.*

[181] Wenzhe SHI et al. « Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 1874-1883.

[182] Shasha WANG, Liangxiao JIANG et Chaoqun LI. « Adapting naive Bayes tree for text classification ». In : *Knowledge and Information Systems* 44.1 (2015), p. 77-89.

[183] Tao WANG et al. « End-to-end text recognition with convolutional neural networks ». In : *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE. 2012, p. 3304-3308.

[184] Adam COATES et al. « Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning. » In : *ICDAR*. T. 11. 2011, p. 440-445.

[185] Patrice Y SIMARD et al. « Using character recognition and segmentation to tell computer from humans ». In : *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* IEEE. 2003, p. 418-423.

[186] Adnan SOURI et al. « Arabic Text Generation Using Recurrent Neural Networks ». In : *International Conference on Big Data, Cloud and Applications*. Springer. 2018, p. 523-533.

[187] Zizhao ZHANG, Yuanpu XIE et Lin YANG. « Photographic text-to-image synthesis with a hierarchically-nested adversarial network ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, p. 6199-6208.

[188] Ángel MORERA et al. « Gender and handedness prediction from offline handwriting using convolutional neural networks ». In : *Complexity* 2018 (2018).

[189] Saad Bin AHMED et al. « Sub-sampling Approach for Unconstrained Arabic Scene Text Analysis by Implicit Segmentation based Deep Learning Classifier ». In : *Global Journal of Computer Science and Technology* (2019).

[190] Adnan SOURI, Mohammed AL ACHHAB et Badr Eddine EL MOUHAJIR. « A proposed approach for Arabic language segmentation ». In : *2015 First International Conference on Arabic Computational Linguistics (ACLing)*. IEEE. 2015, p. 43-48.

[191] *A study towards building an Arabic corpus*.

[192] Chris TENSMEYER et Tony MARTINEZ. « Analysis of convolutional neural networks for document image classification ». In : *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. T. 1. IEEE. 2017, p. 388-393.

[193] Y LECUN, Y BENGIO et G HINTON. « Deep learning. nature 521 ». In : (2015).

[194] Ross GIRSHICK et al. « Rich feature hierarchies for accurate object detection and semantic segmentation ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, p. 580-587.

[195] Christian SZEGEDY et al. « Going deeper with convolutions ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, p. 1-9.

[196] Karen SIMONYAN et Andrew ZISSERMAN. « Very deep convolutional networks for large-scale image recognition ». In : *arXiv preprint arXiv :1409.1556* (2014).

[197] URL : https://www.un.org/ar/.

[198] URL : https://www.arabworldbooks.com/.

[199] URL : http://shamela.ws/index.php/main.

[200] URL : https://www.hindawi.org/.

[201] Jean-Baptiste COURBOT. « Traitement statistique d'images hyperspectrales pour la détection d'objets diffus : application aux données astronomiques du spectro-imageur MUSE ». Thèse de doct. 2017.