UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTÉ DES SCIENCES ET TECHNIQUES FÈS
DÉPARTEMENT D'INFORMATIQUE

PROJET DE FIN D'ETUDES

MASTER SCIENCES ET TECHNIQUES
SYSTÈMES INTELLIGENTS & RÉSEAUX

DÉVELOPPEMENT D'UN FRAMEWORK DE CLASSIFICATION DES IMAGES A BASE DE DEEP LEARNING

LIEU DE STAGE : LA FONDATION MASCIR, RABAT

RÉALISÉ PAR : RAGHIB LOTFI

SOUTENU LE : **15/06/2017**

ENCADRÉ PAR :

PR. ZENKOUAR KHALID.
PR. NAJAH SAID
M. ZENNAYI YAHYA

DEVANT LE JURY COMPOSÉ DE :

PR. ZENKOUAR KHALID.
PR. NAJAH SAID
M. ZENNAYI YAHYA
PR. CHAOUKI ABOUNAIMA MOHAMED
PR. BEGDOURI AHLAME
PR. BENABBOU ABDERRAHIM

ANNÉE UNIVERSITAIRE 2016-2017

Dédicaces

Je dédie ce modeste travail à :

A ma très chère mère Affable, honorable, aimable : Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi. Ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études.

Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis ma naissance, durant mon enfance et même à l'âge adulte.

Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études. Je te dédie ce travail en témoignage de mon profond amour. Puisse Dieu, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.

A mon très cher père, Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

A tous les membres de ma famille, petits et grands Veuillez trouver dans ce modeste travail l'expression de mon affection

A mes chères ami(e)s

A mes chers collègues

Remerciements

Avant de présenter ce rapport, nous tenons à remercier tous ceux qui nous ont aidés à faire émerger ce projet, nous voudrons en particulier remercier :

• ***Mme. Charaibi Nawal**, la directrice générale de MASciR, de m'avoir accordé ce stage et m'avoir ainsi offert l'occasion de connaître cette nouvelle expérience professionnelle.*

• ***M. François Bourzeix**, le manager de l'équipe « Système Embarqué » au sein de la société MASciR, de m'avoir accueilli au sein de son équipe pendant cette période de stage et d'avoir créé les conditions nécessaires à l'adaptation et à l'évolution dans le cadre de mon travail de réalisation de mon projet de fin d'études.*

• ***M. Yahya ZENNAYI,** mon encadrant de stage pour l'encadrement, l'aide, l'encouragement et la sympathie qu'il nous a donné, Grâce à ses conseils j'ai pu terminer ce travail.*

• ***Pr. Khalid Zankouar et Pr. Said Najah,** mes professeurs encadrants, pour leurs conseils, leurs orientations, et leur disponibilité tout au long de la période de stage.*

• les membres du jury pour avoir accepté d'évaluer mon travail, et pour l'honneur qu'ils me font en jugeant ce travail.

***J**e remercie mes professeurs de la Fst de Fés qui ont fait beaucoup d'efforts pour nous transmettre leurs connaissances. Vos compétences incontestables ainsi que vos qualités humaines vous valent l'admiration et le respect de tous. Je vous adresse mes sincères remerciements pour votre patience et votre encadrement durant toutes ces années.*

Résumé

Les réseaux de neurones profonds (DNN) ont réussi à se faire une place de choix dans des domaines tels que le traitement d'images, le traitement de signal vocal et le traitement des langues naturelles. Des résultats remarquables ont été obtenus, par exemple en détection d'objets, et en classification des images. Dans ce cadre s'inscrit mon stage de fin d'études au sein de la fondation MASciR. Il s'agit du développement et de l'implémentation d'un Framework de classification des images et particulièrement les images hyperspectrales, basée sur les méthodes de Deep Learning.

Ce travail peut être résumé en trois étapes. La première consiste à faire un état de l'art sur les méthodes de Deep Learning, ainsi que sur les images hyperspectrales, et une étude benchmarking de différentes solutions existantes, puis la deuxième étape consiste à utiliser les connaissances acquises pour faire une conception du Framework, pour finalement entamer le développement et la réalisation de ce dernier.

Mots clés : Apprentissage profond, Réseaux de neurones, classification des images, Tensorflow, imagerie hyperspectrale,

Abstract

Deep Neural Networks (DNN) have gained prominence in areas such as image processing, voice signal processing and natural language processing. Remarkable results have been obtained, for example in the detection of objects, and in the classification of images. Within this framework is my internship of end of studies within the MASciR foundation. This involves the development and implementation of a framework for classifying images and especially hyperspectral images based on Deep Learning methods.

This work can be summarized in three steps. The first is to make a state of the art on Deep Learning methods, as well as hyperspectral images, and a benchmarking study of different existing solutions, then the second step is to use the knowledge acquired to make a conception of the Framework, to finally begin the development and realization of the latter.

Key-words: Deep Learning, Neural Network, image classification, Tensorflow, hyperspectral image.

Table des Matières

Introduction Générale 1
Chapitre.I Contexte général du projet 3
I.1. Présentation de l'organisme d'accueil 3
I.1.1. Présentation de MASciR 3
I.1.2. MAScIR Microélectronique 4
I.1.3. Partenaire de la fondation 4
I.2. Contexte du Projet 5
I.2.1. Charte du projet 6
I.2.2. Planification du projet 7
I.3. Cahier de charge 8
I.4. Conclusion 10
Chapitre.II les réseaux de neurones profonds 11
II.1. Origine 11
II.2. L'apprentissage profond 14
II.2.1. Les réseaux de neurones profonds 14
II.2.2. Les réseaux de neurones à convolution 15
II.2.3. Réseaux de croyance profonde 17
II.2.4. Deep autoencoders 18
II.3. Éviter le sur-apprentissage 19
II.4. Conclusion 20
Chapitre.III Les images hyperspectrales 21
III.1. Image Hyperspectrale 21
III.1.1. Définition et notions 21
III.1.2. Domaine d'application 22
III.2. Classification des images Hyperspectrale à base de Deep Learning. 26
III.2.1. Enjeux 26
III.2.2. Réduction de dimension 27
III.2.3. Classification basé sur les caractéristiques spectrales 28
III.2.4. Classification basé sur les caractéristiques spatiales 29
III.2.5. Coopération spatio-spectral 31
III.3. Conclusion 32

Chapitre.IV Analyse et conception 33
IV.1. Diagramme de cas d'utilisateurs 33
IV.2. Architecture 34
IV.2.1. la partie model 35
IV.2.2. la partie Vue 37
IV.2.3. la partie Controller 39
IV.3. Conclusion 43
Chapitre.V Réalisation et démonstration du Framework 44
V.1. Environnement de travail 44
V.1.1. Environnement Matériels 44
V.1.2. Environnement logiciel et APIs utilisées 45
V.2. IHM de l'application 47
V.2.1. Partie d'apprentissage 48
V.2.2. Phase de test 54
V.2.3. La partie accueil 56
V.3. Conclusion 57
Conclusion et perspectives 58
Annexe 62

Liste des Figures

FIGURE 1. DIAGRAMME DU GANT DE LA PHASE DE L'ETAT DE L'ART 7
FIGURE 2.DIAGRAMME DU GANT DE LA PHASE DE CONCEPTION 8
FIGURE 3.DIAGRAMME DU GANT DE LA PHASE DU DEVELOPPEMENT 8
FIGURE 4. REPRESENTATION DU PERCEPTRON 12
FIGURE 5.REPRESENTATION DU PERCEPTRON MULTI-COUCHES 13
FIGURE 6. L'ARCHITECTURE GENERALE DU RESEAU DE NEURONES PROFOND 15
FIGURE 7. L'ARCHITECTURE D'UN RESEAU DE NEURONES A CONVOLUTION 16
FIGURE 8. PRINCIPE DE LA COUCHE A CONVOLUTION 16
FIGURE 9.PRINCIPE DE LA COUCHE POOLING 17
FIGURE 10.MACHINE BOLTZMANN RESTREINTE 17
FIGURE 11. STRUCTURE GENERALE D'UN DBN 18
FIGURE 12. STRUCTURE GENERALE D'UN AUTOENCODER 19
FIGURE 13. PRINCIPE DE LA METHODE DE DROPOUT 20
FIGURE 14. EXEMPLE D'IMAGE HYPERSPECTRALE 22
FIGURE 15. IMAGE HYPERSPECTRAL PRESENTE L'AZOTE ET LE STRESS HYDRIQUE DANS LE BLE 23
FIGURE 16.HYPERSPECTRAL IMAGE POUR LA DETECTION DES MINERAUX 24
FIGURE 17.EXEMPLE D'IMAGE HYPERSPECTRALE D'UNE ZONE URBAIN 24
FIGURE 18. EXTRACTION DES CARACTERISTIQUES SPECTRALES A BASE DE DBN 28
FIGURE 19.EXTRACTION DES CARACTERISTIQUES SPECTRALES BASE SUR L'AUTOENCODERS 29
FIGURE 20.CLASSIFICATION ET EXTRACTION DES CARACTERISTIQUES SPATIALES BASEE SUR SAE 30
FIGURE 21.EXTRACTION DES CARACTERISTIQUES SPATIALES BASEE SUR LE CNN 30
FIGURE 22. ARCHITECTURE PROPOSE PAR [27], QUI COMBINE LES CARACTERISTIQUES SPECTRALES ET SPATIALE 31
FIGURE 23. CLASSIFICATION DES IMAGES HYPERSPECTRALES BASEE SUR LES CARACTERISTIQUES SPATIO-SPECTRALES EN UTILISANT LE DBN 31
FIGURE 24. DIAGRAMME DE CAS D'UTILISATEUR DE NOTRE FRAMEWORK 34
FIGURE 25 ARCHITECTURE MVC 35
FIGURE 26.LE DIAGRAMME DE CLASSE POUR LA PARTIE MODEL DE NOTRE FRAMEWORK 36
FIGURE 27. DIAGRAMME PRESENTE L'ARBORESCENCE DES PAGES DU FRAMEWORK 37
FIGURE 28.DIAGRAMME DE SEQUENCE PRESENTE LE SCENARIO DE LA PHASE D'APPRENTISSAGE 40
FIGURE 29. DIAGRAMME DE SEQUENCE PRESENTE LE SCENARIO DE LA PHASE DE TEST 41
FIGURE 30. DIAGRAMME DE CLASSE DE LA PARTIE CONTROLLER 42
FIGURE 31.CLAQUE GENERALE DU GUI DE NOTRE FRAMEWORK 48
FIGURE 32. L'INTERFACE GRAPHIQUE DE CHARGEMENT DES DONNEES POUR LA PHASE D'APPRENTISSAGE 49
FIGURE 33. CONFIGURATION DES POUR L'EXTRACTION DES CARACTERISTIQUES SPATIALES (GUI) 50
FIGURE 34.CONFIGURATION DE LA COUCHE DROPOUT 50
FIGURE 35.CONFIGURATION DE LA COUCHE TOTALEMENT CONNECTEE 50
FIGURE 36.CONFIGURATION DE COUCHE A CONVOLUTION 50
FIGURE 37.CONFIGURATION DE LA COUCHE DE POOLING 50
FIGURE 38.EMPILEMENT DES COUCHES POUR L'EXTRACTION DES CARACTERISTIQUES SPECTRALES (GUI) 51
FIGURE 39.CONCATENATION DES CARACTERISTIQUES ET CLASSIFICATION (GUI) 52
FIGURE 40. CONFIGURATION DES PARAMETRES D'APPRENTISSAGE 53
FIGURE 41. RESULTAT D'APPRENTISSAGE 54
FIGURE 42.CHARGEMENT DES DONNEES DE TEST 55
FIGURE 43.RESULTAT DE TEST DE MODEL 55
FIGURE 44. LA PAGE PERMET DE CHARGER UN MODELE GENERE DANS LA PHASE D'APPRENTISSAGE 56
FIGURE 45.CLASSIFICATION D'UNE IMAGE 57

Liste des Tableaux

TABLEAU 1. DESCRIPTION DES CLASSE DE LA PARTIE MODELE 36
TABLEAU 2. TABLEAU DECRIT LA FONCTIONNALITE DE CHAQUE PAGE DE LA PARTIE VUE 38
TABLEAU 3 . TABLEAU DECRIT LES CLASSES DE LA PARTIE CONTROLLER 42
TABLEAU 4.OUTILS ET APIS UTILISES PENDANT LE DEVELOPPEMENT DU FRAMEWORK 46
TABLEAU 5. STRUCTURE GENERALE DU GUI DU FRAMEWORK 48
TABLEAU 6. RESUME DE L'ETUDE BENCHMARKING SUR LES OUTILS DE DEEP LEARNING 62
TABLEAU 7. RESUME DE L'ETUDE BENCHMARKING DES OUTILS DE TRAITEMENT DES IMAGES HYPERSPECTRALES 65

Liste des abréviations

API	Application Programming Interface
CNN	Convolutional Neural Network
DAFE	Discriminant Analysis Feature Extraction
DBFE	Decision Boundary Feature Extraction
DBN	Deep belief network
DNN	Deep Neural Network
EMP	Extended Morphological Profile
FC	Fully Connected
GLCM	Gray-Level co-occurrence Matrix
GUI	Graphical User Interface
HSI	Hyperspectral Imaging
IHM	Interface Homme Machine
LDA	Linear discriminant analysis
MAScIR	Moroccan Foundation for Advanced Science, Innovation & Reserarch
MCRD	Minimum Change Rate Déviation
MPCA	Morphological Principal Component Analysis
MLP	Multi-Layer Perceptron
MVC	Model View Controller
PCA	Principal Component Analysis
PFE	Projet de fin d'études
RBM	Restricted Boltzmann machine
RGB	Rouge Vert Blue
RN	Réseau de Neurones
SVM	Support Vector Machien
SAE	stacked autoencoder

Introduction Générale

Les chercheurs et les inventeurs ont longtemps rêvé de créer des machines qui pensent, réfléchissent et qui prennent des décisions d'une façon autonome, Lorsque les ordinateurs ont été créés pour la première fois, les gens se demandaient s'ils deviendraient intelligents. Aujourd'hui, l'intelligence artificielle (AI) vient pour répondre à ces questions en donnant plus d'intelligence aux machines. Les entreprises se tournent vers l'intelligence artificielle pour résoudre des problèmes difficiles, comme la reconnaissance vocale, la reconnaissance d'objets et la traduction automatique, et la fondation MASciR ne fait pas l'exception .En effet, dans le département systèmes embarqués, la thématique dominante est la vision par ordinateur (computer vision), basée sur des méthodes d'intelligence artificielle et principalement les méthodes de Deep Learning.

Comme il est bien connu l'utilisation de la vision par ordinateur (computer vision) pour analyser, traiter, comprendre, détecter et classifier un ou plusieurs objets physiques dans une image est une tâche difficile. Généralement, dans la littérature, et pour les problèmes de classification, l'image à trois bandes de couleur (Rouge, Vert, Blue) est la plus utilisée, puisqu'elle est la plus proche de l'interaction visuelle humaine, mais pour des problèmes complexes, l'utilisation des images à trois bandes ne permet pas une bonne distinction des différents composants physiques. En fait, ce type d'images ne constitue qu'une petite plage du spectre électromagnétique, d'où vient l'importance de l'utilisation des images hyperspectrales qui sont des images permettant la représentation d'une scène suivant un grand nombre de bandes spectrales, ce qui donne plus d'informations sur la scène observée.

Dans ce cadre, la fondation, MASciR a pris l'initiative de mettre en place un Framework de classification des images, et principalement les images hyperspectrales, basé sur les méthodes de Deep Learning. Le Framework va permettre à l'utilisateur à l'aide d'une interface graphique de configurer l'architecture du réseau de neurones profonds ainsi que ses paramètres d'apprentissage. Ensuite, les valeurs de ses paramètres seront optimisées à l'aide d'une base de données d'apprentissage. Finalement, une base de données de test permettra d'évaluer les performances de notre modèle de classification généré.

Le présent rapport sera organisé comme suit : dans un premier chapitre nous allons présenter, la fondation MASciR, ses activités ainsi que le cadre général de projet réalisé durant ce stage. Le deuxième chapitre sera consacré au réseau de neurones profond. Nous allons présenter le concept de « Deep Learning », l'architecture générale, le principe de fonctionnement ainsi que les types les plus utilisés dans la littérature : les réseaux de neurones à convolution (Convolutional Neural Networks,

CNN) et les réseaux de croyance profonde (Deep belief network, DBN) ainsi que les Autoencoders. Dans le troisième chapitre, nous allons introduire dans premier temps, les différentes notions relatives aux images hyperspectrales ainsi que les différents domaines d'application de ce type des images. Dans un deuxième temps, nous allons décrire la classification des images hyperspectrales en se basant sur des réseaux de neurones profonds. Le quatrième chapitre fait l'objet d'une présentation détaillée du cahier de charges du projet ainsi que sa spécification software. Dans le cinquième chapitre, nous allons présenter l'environnement matériel et logiciels utilisés pour la réalisation du Framework de classification des images, ainsi que les fonctionnalités générales et les différentes interfaces graphique qui constituent ce Framework. Pour clore ce projet de fin d'études, nous allons inclure une conclusion générale ainsi que les perspectives de ce travail.

Chapitre.I Contexte général du projet

Introduction

Le présent chapitre permet de situer le projet dans son contexte général. Dans un premier temps, il s'agit de présenter l'organisme d'accueil : La fondation MAScIR, sa structure, ses domaines d'activités, ses partenaires et ses réalisations. Par la suite, nous allons introduire, les objectifs de notre projet et décrire la démarche que nous avons adoptée dans la conduite du projet. Finalement nous allons présenter le cahier des charges qu'on doit satisfaire.

I.1. Présentation de l'organisme d'accueil

I.1.1. Présentation de MASciR

MASciR est un organisme de recherche à caractère scientifique et technologique, dédié à la recherche en nanotechnologie, en biotechnologie, en technologie numérique, en microélectronique, en énergie et en environnement.

Rassemblant d'éminents chercheurs des quatre coins du monde, MASciR, regroupe des équipes scientifiques œuvrant dans des domaines innovants et complémentaires et met à leur disposition des instruments scientifiques de pointe.

Initialement fondée en 2007 par le Gouvernement Marocain en tant que fondation à but non lucratif. Depuis sa création jusqu'à aujourd'hui, la nanotechnologie, la biomédecine, et les microélectroniques représentent les institutions de recherche de cette fondation :

- **MASciR MicroElectronics** : créé vers la fin de l'année 2008, a pour objectif de devenir un centre de Recherche et Développement dans le domaine de la microélectronique.
- **MASciR BioTechnology** : deuxième centre inscrit dans MASciR œuvrant dans le domaine de la Biotechnologie : recherche et développement des médicaments ou des biocides.

- ***Nanomaterials et NanoTechnology*** : qui a pour mission de mener des recherches appliquées et innovantes dans le domaine des nanomatériaux et des nanotechnologies.

I.1.2. MAScIR Microélectronique

MAScIR MicroElectronics est un centre d'innovation et de développement des technologies dans le domaine microélectronique. Il se concentre sur le micro packaging, l'ingénierie, les tests de simulation, le design, la qualification, le prototypage de produits micro-électroniques et les systèmes embarqués. Parmi les thématiques traitées par MAScIR MicroElectronics, nous citons :

- Design et Micro packaging CSP et PILR,
- Tests de fiabilité sur les packages
- Circuits embarqués sur une application wafer level camera fabriqué au Maroc par Nemotek Technologie.
- Produits embarqués médicaux et agricultures à cout réduit.
- Produits embarqués pour la gestion du trafic routier.
- Projets énergétiques.
- Traitement d'image
- etc

I.1.3. Partenaire de la fondation

Le modèle de management de MAScIR ainsi que le savoir-faire et le professionnalisme de ses équipes scientifiques ont permis à la Fondation d'assoir sa notoriété auprès des industriels et renforcer son éventail de partenaires industriels tant à l'échelle nationale qu'internationale.

Les principaux partenaires de la fondation MASCIR MicroElectronics sont :

- **Lear Corporation** : L'un des principaux fournisseurs mondiaux de sièges automobiles et les systèmes de gestion de l''énergie électrique.
- **Thales** : figure parmi les leaders européens de la fabrication et de la commercialisation d'équipements et de systèmes électroniques destinés aux secteurs de l'aérospatial, du transport, de la défense et de la sécurité.

- **OCP** : est un acteur incontournable sur le marché des phosphates et de ses produits dérivés. Présent sur toute la chaine de valeur, il est le premier exportateur de cette matière dans le monde.
- **STERIMED** : est une société spécialisée dans le domaine de l'eau et des technologies de l'environnement. Son objectif est d'accompagner les entreprises et collectivités dans la résolution des problématiques liées à l'eau, l'environnement.
- **COSUMAR** : est un groupe marocain, filiale de la Société nationale d'investissement, spécialisé dans l'extraction, le raffinage et le conditionnement du sucre sous différentes formes. Il est devenu l'unique opérateur sucrier marocain après l'acquisition de SUTA, SUCRAFOR, SUNABEL et SURAC en 2005.

I.2. Contexte du Projet

Le Deep Learning est un domaine de recherche récent et très répandu qui a réussi à captiver l'intérêt des chercheurs et des inventeurs. Aussi, les grandes entreprises se tournent vers le Deep Learning pour résoudre des problèmes qui demandent l'intelligence (ex : Google, IBM, Microsoft, Amazon, Adobe, Yandex...), MAScIR ne fait pas l'exception, la fondation a commencé à investir dans ce domaine en raison du besoin progressif de l'équipe des systèmes embarqués.

La thématique dominante dans l'équipe est le traitement d'images. Dans ce cadre, l'équipe travaille sur plusieurs projets visant la réalisation de systèmes d'inspection visuelles, citons par exemple :

- Projet Dari développé pour Dari Couspate, dont la fonctionnalité fondamentale est la classification et le comptage des graines de couscous non conformes, ainsi que les corps étrangers.
- Projet développé pour les centres techniques de contrôle de véhicule pour faire la reconnaissance des plaques de véhicule, et tester sa conformité.
- Projet développé pour l'OCP (*Office chérifien des phosphates*), pour la distinction du phosphates, de l'Astrid dans les zones miniers qui sont des matériaux visuellement semblables, d'où la nécessité d'utiliser des images hyperspectrales
- Projet développé pour ADM (Autoroutes du Maroc).l'objectif de ce projet est le développement d'un logiciel de comptage, classification, reconnaissance des véhicules dans un flux vidéos.

On remarque que la plupart de ces projets ont une tâche commune, la classification d'images ; que ça soit la classification des grains de couscous (conforme, non conforme) ; ou bien la

classification de véhicules (voiture, camion, vélo), etc. Cela nous a motivé a développé une solution générique et efficace pour la classification des images basé sur les méthodes de Deep Learning.

I.2.1. Charte du projet

Lors de la planification d'un projet dans le milieu industriel on utilise un outil nommé 'charte du projet ' permettant de définir les objectifs et les contraintes principales du projet. La charte du présent projet est détaillée ci-dessous.

Pourquoi lancer le projet

- Le besoin d'un outil qui permet de classifier des images pour un ensemble de projets au sein de MASciR

Problème à résoudre

- Le Framework doit être générique, c'est à dire il doit supporter la majorité des types d'images ainsi que les images hyperspectrales.

Objectifs du projet

- Développement d'un Framework à base de Deep Learning qui permet la classification des images

Dates

- Début : 13 Février
- Fin : 13 Août

Livrables

- Cahier de charge du projet
- Spécification Software du projet
- Framework permet la classification des images.
- Rapport de stage
- Support finale PPT

Critère de fin fonctionnelle

- Garantie le bon fonctionnement de Framework

I.2.2. Planification du projet

La planification d'un projet est un outil incontournable pour le management de projet. Elle permet de définir les travaux à réaliser, fixer les objectifs, coordonner les actions, maîtriser les moyens, diminuer les risques, suivre les actions en cours, et de rendre compte de l'état d'avancement du projet.

Le diagramme de Gantt est un outil permettant de planifier le projet et de rendre plus simple le suivi de son avancement. Le présent planning est réajusté au fur et à mesure de l'avancement du projet.

Notre Projet est divisé en trois phases principales ;

- Etat de l'art et étude bibliographique

Durant cette phase on a commencé à se familiariser avec le concept des images hyperspectrales, ses enjeux, les outils existants de traitement des images hyperspectrales et ses caractéristiques, puis on est passé à une étude sur les méthodes de Deep Learning, et la façon d'appliquer ces méthodes pour résoudre des problèmes de classification des images, et particulièrement les images hyperspectrales.

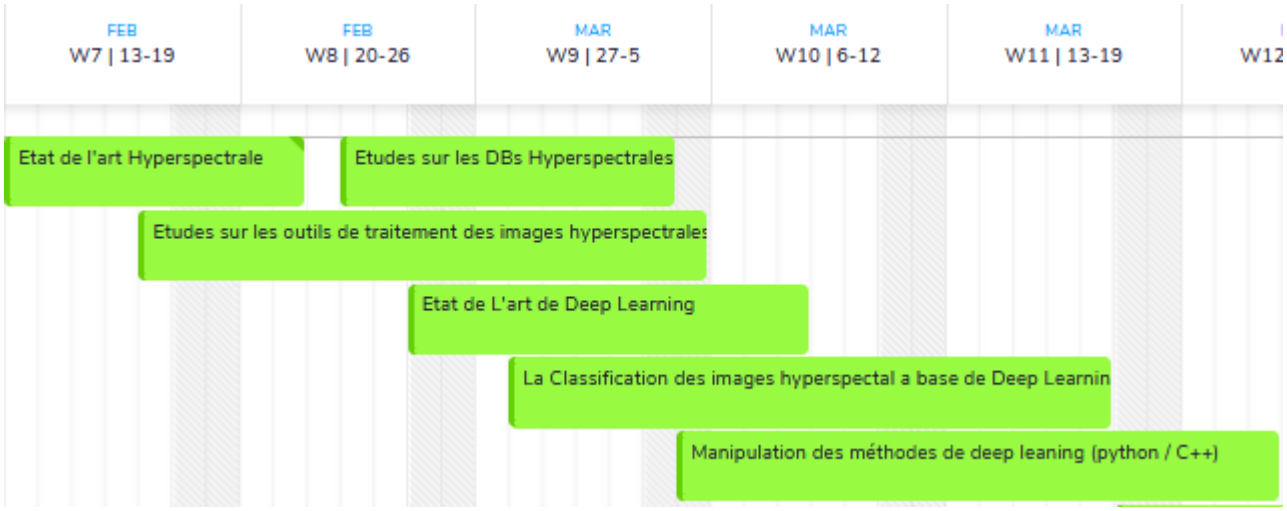


Figure 1. Diagramme du Gant de la phase de l'état de l'art

- Conception du Framework

Durant cette phase, on a rédigé le cahier de charge de notre Framework qui englobe toutes les exigences techniques et fonctionnelles qu'on doit satisfaire lors de la réalisation du Framework, puis on a entamé la réalisation du Document de spécification software qui détaille l'architecture globale Du Framework

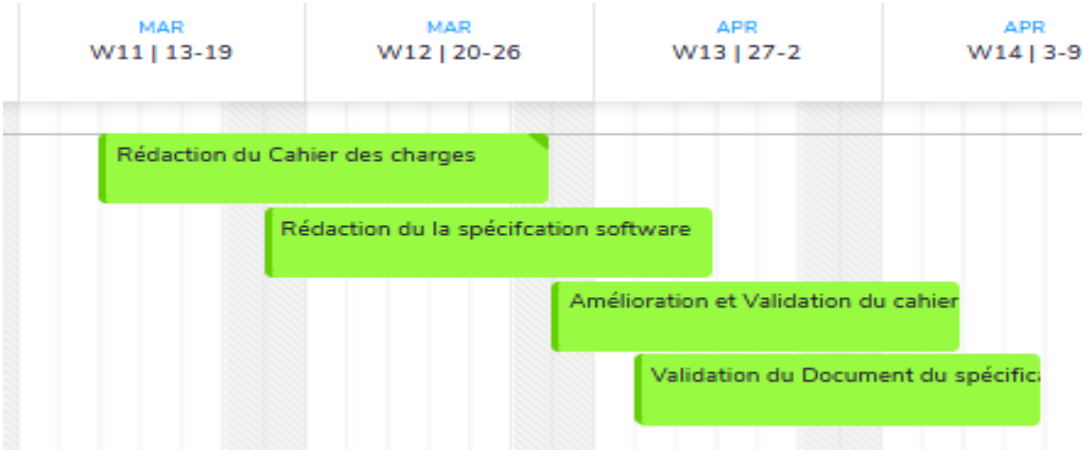


Figure 2.Diagramme du Gant de la phase de conception

- Développement et réalisation

Cette phase est consacrée au développement du Framework, qui se base sur les résultats de la phase précédente, on a commencé par le développement de l'interface graphique du Framework, puis la partie modèle et finalement la partie Contrôleur de l'architecture MVC qu'on va détailler par la suite.

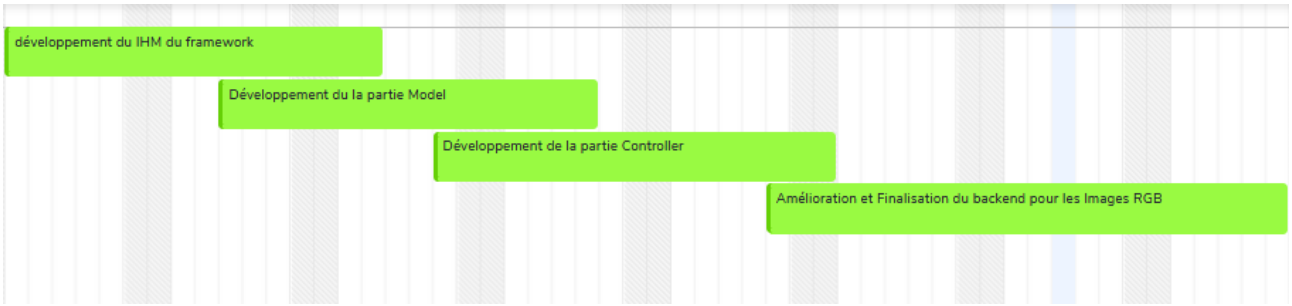


Figure 3.Diagramme du Gant de la phase du développement

I.3. Cahier de charge

Le cahier de charges de notre Framework définie un ensemble des exigences techniques et fonctionnelles pour le développement, ces exigences sont regroupées en plusieurs catégories :

- Exigences Globales

Exigences globales	Le Framework doit supporter plusieurs formats d'images hyperspectrales ainsi que les formats standards des images (PNG, JPEG,BSQ, BIP, BIL, HDR,.img, LAN, GIS, etc.)
	Le Framework doit être générique d'un point de vue de traitement, c'est à dire, il permet d'appliquer le traitement sur les images hyperspectrales ou non (RGB, Niveau de gris, Multi-Spectrale...)
	Le Framework doit supporter plusieurs plateformes d'exécution, notamment Windows, Linux et le Mac

- Exigences relatives à la phase d'apprentissage

Phase d'apprentissage	Le Framework permet à l'utilisateur de configurer l'architecture globale d'apprentissage sur plusieurs niveaux : prétraitement, configuration des couches, les paramétres d'apprentissage.
	Le Framework permet de faire une estimation de temps d'apprentissage, après la configuration de l'architecture d'apprentissage.
	Le Framework permet d'enregistrer la configuration du réseaux en format XML, pour la réutliser en cas de besion
	Le Framework permet de générer un modèle après la phase d'apprentissage qui va permettre de classifier les images
	Le Framework permet de stocker les modèles générés après la phase d'apprentissage.
	Le Framework permet de visualiser et stocker un ensemble des statistiques et informations relatives à chaque modèle.
	Le Framework permet à l'utilisateur d'importer les modéles générés aprés la phase d'apprentissage pour les évaluer .

- Exigences relatives au système embarqué

Système embarqué	Le modèle généré aprés la phase d'apprentissage doit être facilement intégrable dans les système embarqués
	Le Framework permet de générer un code C qui permet l'intérgation du modéle dans le systémes embarqué

I.4. Conclusion

Au cours de ce chapitre, nous avons défini le contexte général du projet et la démarche de développement adoptée, puis une présentation d'un diagramme de Gantt décrivant le déroulement du PFE, pour finir avec une présentation de cahier des charges du projet. Dans le chapitre suivant, nous verrons le concept de Deep Learning, son origine, le principe de fonctionnement ainsi que les types les plus utilisés dans la littérature : les réseaux de neurones à convolution (Convolutional Neural Networks, CNN) et les réseaux de croyance profonde (Deep belief network, DBN) ainsi que les Autoencoders.

Chapitre.II les réseaux de neurones profonds

Introduction

Le présent chapitre présent une étude générale sur les réseaux de neurones artificiels et précisément les réseaux de neurones profonds. Nous allons dans un premier temps introduire l'origine biologique des réseaux de neurones, puis nous expliquons le principe du premier réseau de neurones artificiels appelée perceptron, et comment le perceptron multicouche a pallié aux problèmes de ce perceptron, puis nous passons à la notion de "profond" pour les réseaux de neurones et finalement nous détaillons quelques méthodes d'apprentissage profonds (DNN, CNN, Deep Autoencoders, DBN).

II.1. Origine

La connaissance du fait que le cerveau fonctionne de manière entièrement différente de celle d'un ordinateur conventionnel a joué un rôle très important dans le développement des réseaux de neurones artificiels. Les travaux effectués pour essayer de comprendre le comportement du cerveau humain ont menés à représenter le cerveau par un ensemble de composants structurels appelés neurones. Le cerveau humain en contiendrait des centaines de milliards de neurones, et chacun de eux serait, en moyenne, connecté à dix mille autres. Le cerveau est capable d'organiser ces neurones, selon un assemblage complexe, non-linéaire et extrêmement parallèle, de manière à pouvoir accomplir des tâches très élaborées. Par exemple, n'importe qui est capable de reconnaître des visages, alors que c'est là une tâche quasiment difficile pour un ordinateur. C'est la tentative de donner à l'ordinateur les qualités de perception du cerveau humain qui a conduit à une modélisation algorithmique qui permet l'introduction de concept des réseaux de neurones artificiels.

La première étude systématique du neurone artificiel a été faite par le neuropsychiatre *McCulloch* et au logicien *Pitts* [4], qui, s'inspirant de leurs travaux sur les neurones biologiques. Des années après et exactement en 1957 *Frank Rosenblatt* a inventé le perceptron [5] qui peut être vu comme le type de réseau de neurones le plus simple. C'est un classificateur linéaire monocouche qui n'a qu'une sortie à laquelle toutes les entrées sont connectées (Figure 4).

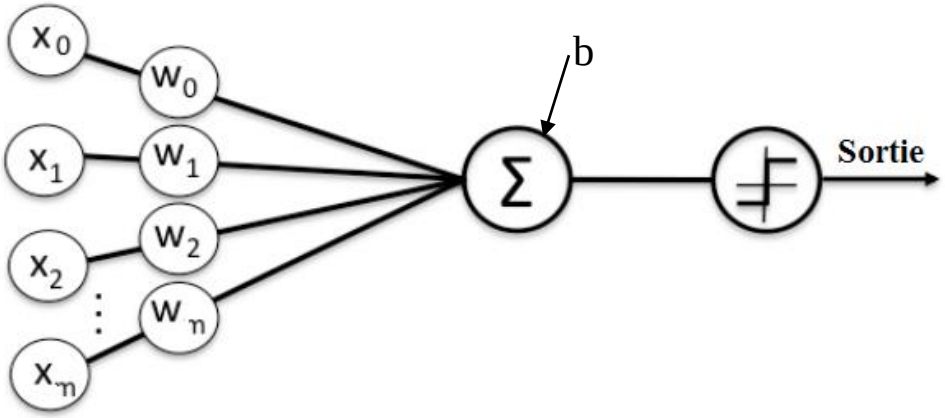


Figure 4. Représentation du perceptron

Le fonctionnement du perceptron est simple : c'est une somme pondérée d'entrées à laquelle on applique une fonction d'activation, afin de trouver la sortie du modèle. Mathématiquement parlant la sortie est calculée selon la relation suivante :

$$Sortie = \begin{cases} 0 & si\ W.X + b \leq 0 \\ 1 & si\ W.X + b > 0 \end{cases}$$

Avec $W = (w_0, w_1, w_2\ \cdots, w_n)$, $X = (X_0, X_1, X_2 \cdots, X_n)$ et b le Bais

En effet, appliqué un seuil sur un vecteur pondéré revient à découper linéairement l'espace des vecteurs d'entrée en deux avec un hyperplan, d'où le perceptron ne permet de représenter que des fonctions linéairement séparables. Un nouveau type de réseau permettant de dépasser cette limitation a été proposée dans les années 80, c'est le perceptron multicouche [6].

L'architecture d'un perceptron multicouche (qu'on notera MLP pour Multi-Layer Perceptron) est une généralisation du perceptron simple. Où chaque couche contient un ensemble des neurones. Les neurones de chaque couche sont connectés à l'ensemble des neurones de la couche qui suit (Figure 5).

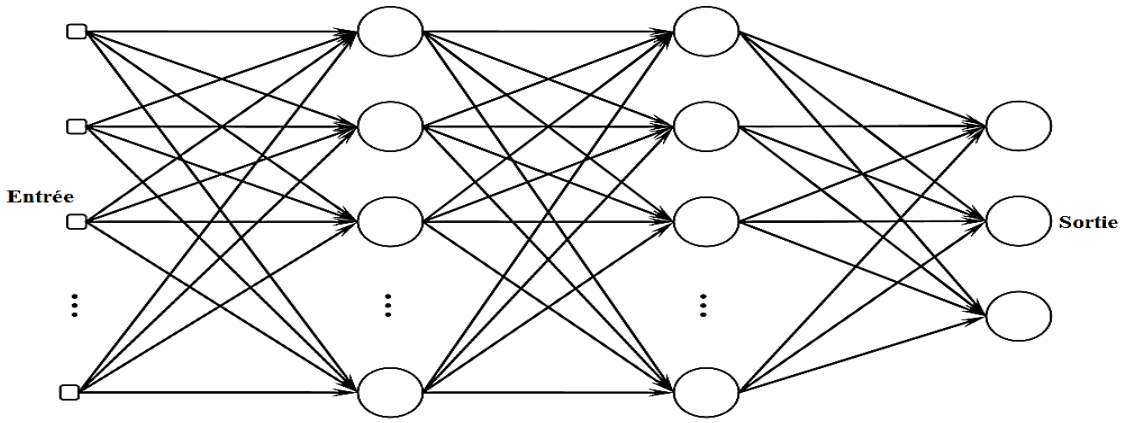


Figure 5.représentation du perceptron multi-couches

L'apprentissage du réseau se déroule en deux étapes:

1. **Propagation :** cette étape consiste à calculer le vecteur de sortie qui se fait en calculant les sorties successives des différentes couches, la sortie d'une couche servant d'entrée à la couche suivante. Ce vecteur est ensuite comparé au vecteur de sortie désiré : on calcule le vecteur d'erreur quadratique servant pour la phase de rétro propagation.
2. **Rétro propagation** : c'est à dire le calcul du gradient de la fonction d'erreur quadratique par rapport à chaque paramètre du réseau puis la mise à jour de ces paramètres, afin de minimiser l'erreur quadratique du réseau. Pour algorithme existe pour minimiser l'erreur du réseau :
    - SGD (stockastic gradient descent) [34].
    - L'algorithme d'ADAM [36].
    - Adaptif Gradient (AdapGrad) [35].
    - RMS propagation (Rmsprop).

L'algorithme de rétro-propagation souffre cependant de deux inconvénients majeurs :

1. **Temps d'apprentissage** : l'expérience montre que le temps d'apprentissage d'un RN croît rapidement lorsque le nombre de couches augmente. C'est d'ailleurs l'une des raisons pour lesquelles à partir des années 1990 les RN ont été remisés au profit d'autres algorithmes non-linéaires moins gourmand en ressources comme les SVM.
2. **sur-apprentissage** : est un problème pouvant survenir dans les méthodes d'apprentissage automatique. Quand on parle des réseaux des neurones, on parle d'un grand nombre de paramètres dans le réseau (poids, bias, etc.), ça donne une grande liberté dans le réseau, ce qui provoque le problème de sur-apprentissage, autrement dit le sur-apprentissage fait référence à un modèle qui donne des bonnes résultats sur les données d'apprentissage mais pas sur des nouvelles données.

II.2. L’apprentissage profond

Les réseaux de neurones fournissent actuellement des solutions à de nombreux problèmes de reconnaissance d'image, de reconnaissance vocale et de traitement du langage naturel, etc. mais ces solution reste toujours un peu loin de l’idéal, c’est pour cela l’apparition de Deep Learning (réseaux de neurones profonds), pour mettre fin aux problèmes des réseaux de neurones classiques.

Le Deep Learning ou l’apprentissage profond est un ensemble des méthodes d'apprentissage automatique focalisant sur les réseaux de neurones profonds (DNN pour Deep Neural Network) et tentant de modéliser, avec un haut niveau d’abstraction, des données grâce à des architectures articulées de différentes transformations non linéaires [6]. Ces techniques ont permis des progrès importants et rapides dans les domaines de l'analyse du signal sonore, la reconnaissance vocale, la vision par ordinateur ainsi que du le traitement des langues naturelles et autre.

Les idées de base du Deep Learning remontent à la fin des années 80, avec la naissance des premiers réseaux de neurones constituaient d’une seule couche caché, Pourtant, cette méthode vient seulement de connaître son heure de gloire, C’est en raison des moyens qui ne sont apparus que très récemment (La puissance des ordinateurs actuels et à la masse de données d’apprentissage).

II.2.1. Les réseaux de neurones profonds

Les réseaux de neurones profonds sont des réseaux de neurones qui font leur apprentissage d’une manière profonde, la partie cachée du réseau est constituée de plusieurs couches (Figure 6), chaque couche cachée joue deux rôles, elle est la fois la couche de sortie de la couche précédente et la couche d’entrée de la couche suivante.

L’idée des réseaux de neurones profonds c’est de donner le pouvoir à chaque couche cachée de contribuer dans la phase de la reconnaissance. Prenons le cas de la reconnaissance de forme, les neurones d’entrée vont être les pixels de l’image, ensuite les neurones de la première couche cachée peuvent reconnaitre les bords de la forme, les neurones de la deuxième couche cachée peuvent reconnaitre des formes plus complexes (Triangle, Rectangles …), etc.

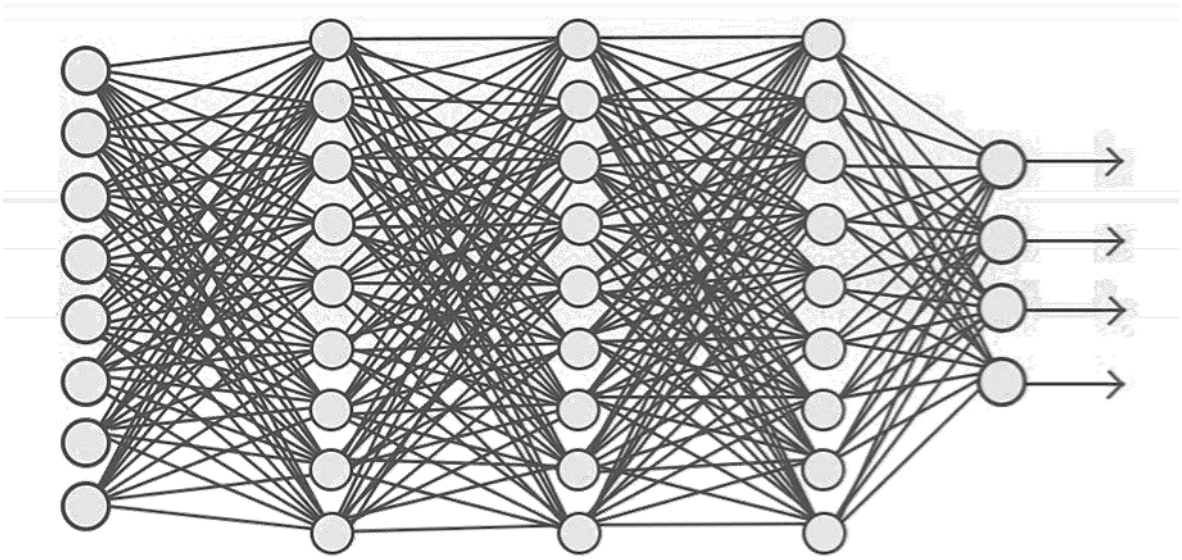


Figure 6. L'architecture générale du réseau de neurones profond

II.2.2. Les réseaux de neurones à convolution

Les réseaux de neurones à convolutions (qui seront désignés ci-après par CNN, pour Convolutional Neural Networks) peuvent être vus comme des réseaux de neurones MLPs particulièrement adaptés au traitement des signaux 2D. Ces réseaux ont été inspirés par les travaux de *Hubel* et *Wiesel* sur le cortex visuel chez les mammifères [8]

Les premiers CNNs datent des années 1980 avec les travaux sur le Necognitron de *K. Fukushima* [9], mais c'est dans les années 1990 que ces réseaux devenus popularisés avec les travaux de *Y. Le Cun* et al sur la reconnaissance de caractères [10].

Un CNN est un réseau de neurones dont les couches sont liées non pas par une opération matricielle mais par une convolution en deux dimensions. Les CNN permettent de traiter directement de grandes quantités de données que sont les images pour plusieurs raisons :

- Les images sont corrélées spatialement (les valeurs des pixels adjacents sont généralement très proches) et les couches de convolutions permettent de créer des liens entre ces données corrélées spatialement.
- Les couches de convolutions sont généralement suivies de couches de sous échantillonnage (pooling) qui diminuent grandement la taille des données.

Une architecture CNN est formée par un empilement de couches de traitement indépendantes (Figure 7):

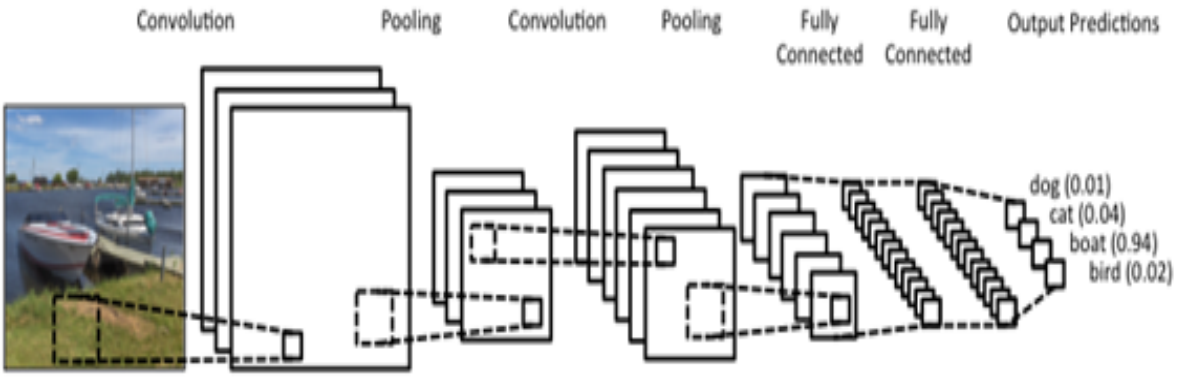


Figure 7. L'architecture d'un réseau de neurones à convolution

- **La couche de convolution** (Figure 8) qui a pour mission l'extraction des caractéristiques pertinentes de l'image, en la balayant avec un ensemble des filtres.
- **La couche de Pooling** (Figure 9) qui permet de compresser l'information en réduisant la taille de signal. On peut dire qui est une sorte de sous-échantillonnage. Généralement le type le plus utilisé est le « max-pooling » c'est à dire considérer seulement la valeur maximale des entrées mais il existe d'autre type comme « average-pooling » qui se base sur l'utilisation de la valeur moyenne des entrées.
- **Couche entièrement connectée (FC)** Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des multi-couches simples entièrement connectées.

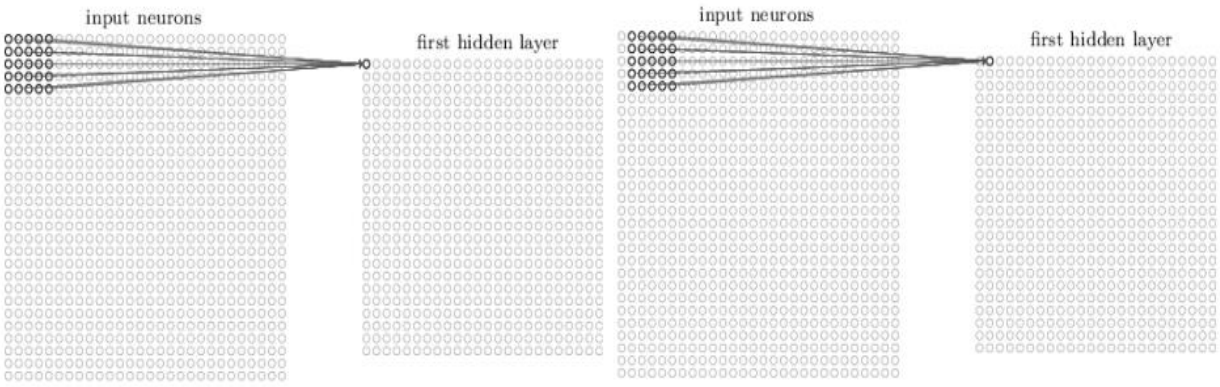


Figure 8. Principe de la couche à convolution

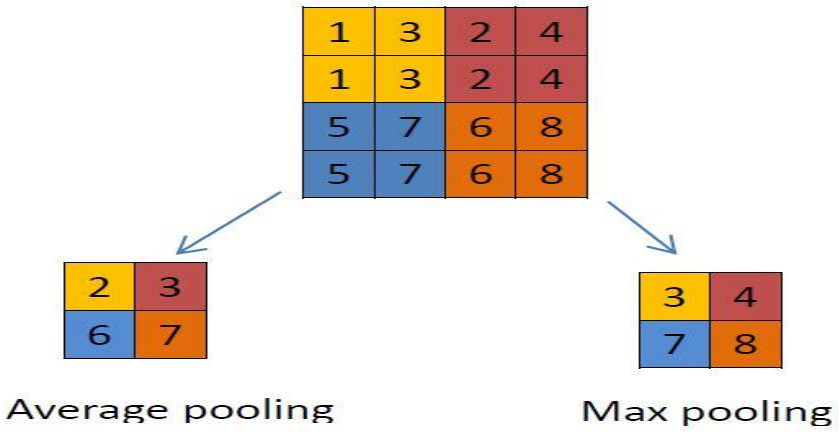


Figure 9.principe de la couche pooling

II.2.3. Réseaux de croyance profonde

Le Réseaux de croyance profonde (DBN pour Deep belief network) est un réseau de neurone non-supervisé, introduit en 2OO6 par Geoff Hinton avec la naissance de l'apprentissage automatique profonds [11].

Un DBN peut être défini comme un empilement d'un ensemble des machines Boltzmann restreintes (RBM) (Figure 11), dans laquelle chaque couche RBM communique avec les couches précédentes et suivantes.

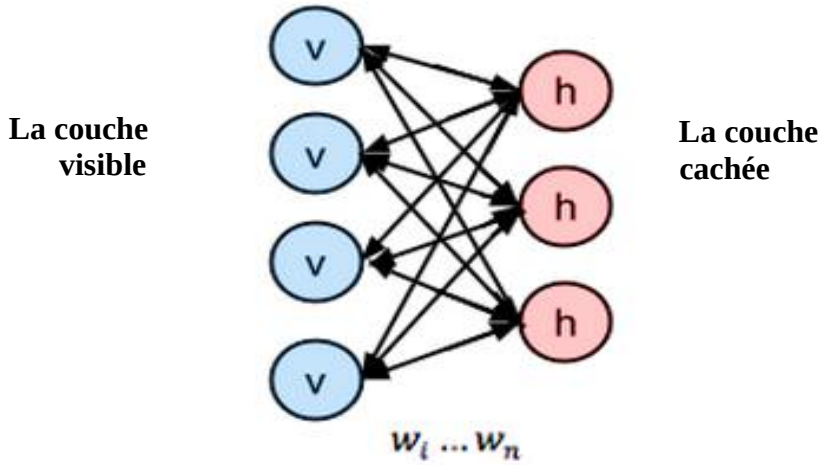


Figure 10.Machine Boltzmann restreinte

Les RBMs sont des réseaux des neurones non supervisé non orienté, à deux couches, La première couche est appelée couche visible, ou d'entrée, et la seconde est la couche cachée (Figure 10), une

fois les valeurs d'activation de la couche caché sont calculées, le RBM reconstituer les données d'une façon non-supervisé, cette phase est appelée phase de reconstruction, dont les activations de la couche cachée devient l'entrée dans une étape de passe en arrière.

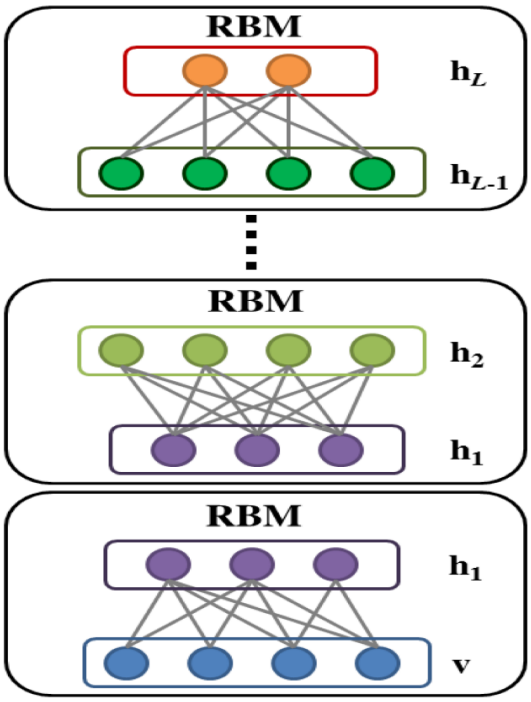


Figure 11. Structure générale d'un DBN

II.2.4. Deep autoencoders

Un réseau de neurone autoencoders est un type de réseaux de neurones non-supervisé qui est composé de deux DBN (Deep Belief network) symétriques qui ont typiquement quatre ou cinq couches, le premier DBN représente la partie « **encoder** » du réseau et le deuxième représente la partie « **Décoder** » [6] (Figure 12).

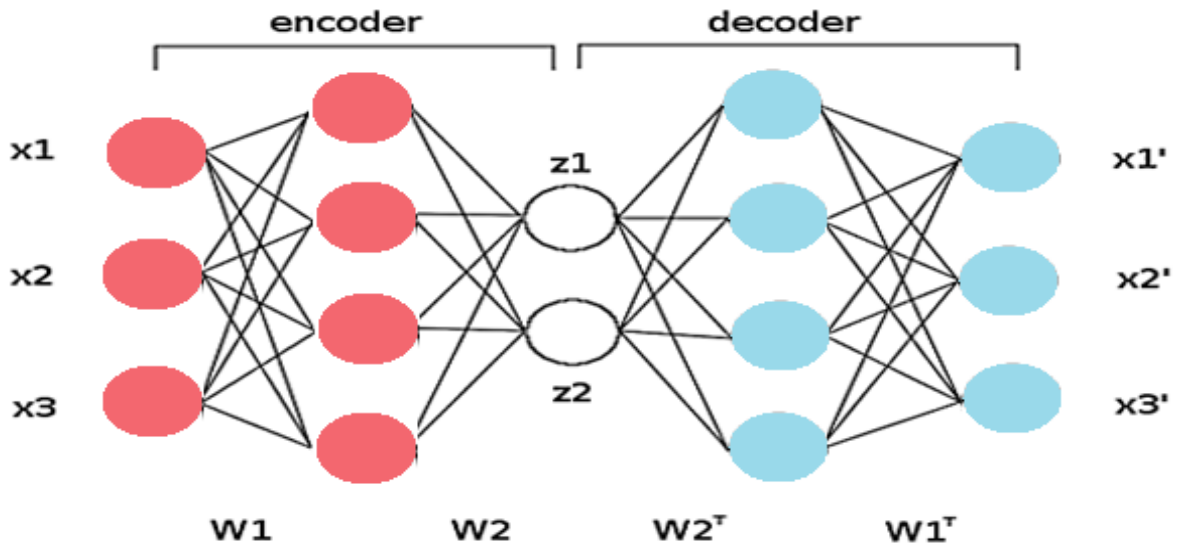


Figure 12. Structure générale d'un Autoencoder

Les données d'entrée sont codées le long du chemin vers la couche cachée z, et ces mêmes données sont décodées le long du chemin vers la couche de sortie.

Pour bien éclairer les choses prenant une application concrète pour les Deep autoencoders, le problème de la réduction de dimension. Par exemple, transformer une image de 256x256 pixels en une représentation avec seulement 30 valeurs (taille de la couche cachée). L'image peut alors être reconstruite avec les poids appropriés et les biais. Ce type de réseaux de neurones a montré sa robustesse pour la réduction de dimension, et il a surpassé plusieurs méthodes de réduction de dimension au niveau de performance comme l'analyse en composant principale [32].

II.3. Éviter le sur-apprentissage

Comme déjà décrit auparavant, le problème de sur-apprentissage survient lorsqu'on possède un grand nombre du paramètre dans le réseau, et c'est le cas pour un réseau de neurones profonds. Pour éviter, et détecter ce problème de sur-apprentissage, des méthodes ont été proposées :

- La validation croisée

Pour détecter un sur-apprentissage, on sépare les données en deux sous-ensembles : l'ensemble d'apprentissage et l'ensemble de validation. L'ensemble d'apprentissage comme son nom l'indique permet de faire évoluer les poids du réseau de neurones avec par exemple une rétropagation L'ensemble de validation n'est pas utilisé pour l'apprentissage mais permet de vérifier la pertinence du réseau avec des échantillons qu'il ne connait pas.

On peut vraisemblablement parler de sur-apprentissage si l'erreur de prédiction du réseau sur l'ensemble d'apprentissage diminue alors que l'erreur sur la validation augmente de manière significative. Cela signifie que le réseau continue à améliorer ses performances sur les échantillons d'apprentissage mais perd son pouvoir de prédiction sur ceux provenant de la validation.

- La régularisation

Les méthodes de régularisation sont des méthodes permettant d'éviter le sur-apprentissage. La méthode de régularisation la plus utilisé est le « Dropout », cette méthode consiste à éteindre et rallumer les neurones aléatoirement, dans le cadre d'entraînements successifs. Une fois les séries d'entraînements terminées, on rallume tous les neurones et on utilise le réseau comme d'habitude. Cette technique a montré non seulement un gain dans la vitesse d'apprentissage, mais en déconnectant les neurones, on a aussi limité des effets marginaux, rendant le réseau plus robuste et capable de mieux généraliser les concepts appris [33].

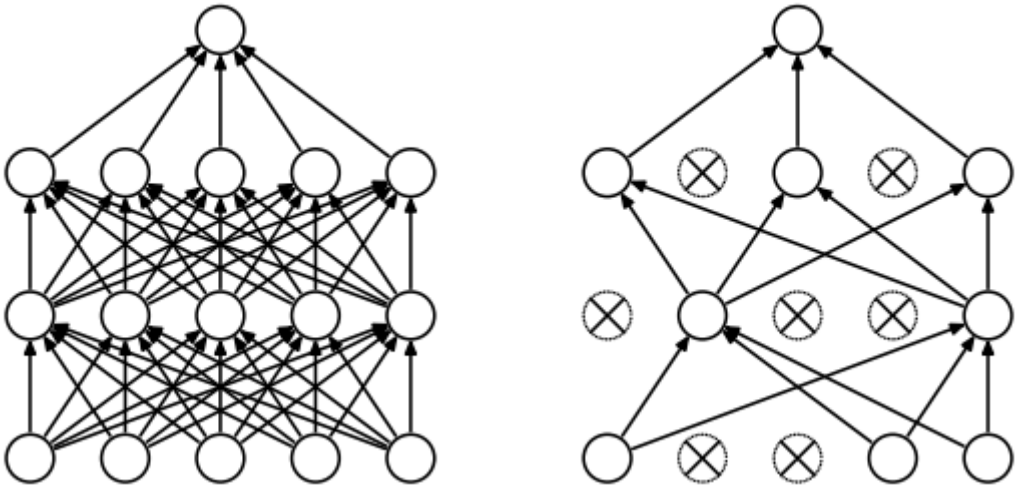


Figure 13. Principe de la méthode de Dropout

La technique du dropout est notamment utilisée dans les systèmes de reconnaissance d'image, de voix, le classement de documents et sur des problèmes de calculs en biologie.

II.4. Conclusion

Après avoir détaillé le concept de réseaux de neurones ainsi que les méthodes Deep Learning appliqué dans la vision par ordinateur, le prochain chapitre a pour but de présenter la notion des images hyperspectrales, ses domaines d'applications, ainsi que la classification de ses images à base de Deep Learning.

Chapitre.III Les images hyperspectrales

Introduction

Ce chapitre présente une vue générale sur le concept d'imagerie hyperspectrale, ses domaines d'application ses enjeux, ainsi que le processus de classification de ces image en se basant sur les méthodes de Deep Learning

III.1. Image Hyperspectrale

III.1.1. Définition et notions

Les récents progrès effectués dans le domaine des instruments optiques ont donné naissance aux spectro-imageurs qui produisent des images dites hyperspectrales (HSI), par opposition à l'imagerie à trois bandes (RGB), l'imagerie hyperspectrale est une technologie permettant la représentation d'une scène suivant un grand nombre de bandes spectrales (généralement plus d'une centaine).

Une définition formelle de l'IHS est donnée dans [1] : "L'imagerie hyperspectrale consiste à acquérir des spectres pour tous les pixels d'une image, où un spectre est une mesure contigüe d'une distribution de longueur d'onde, avec une résolution suffisante pour résoudre la variabilité naturelle du système d'intérêt", à partir cette définition on peut distinguer l'IHS de l'imagerie couleur et multispectrale par trois caractéristique principales :

- Les systèmes couleurs ou multispectrales enregistrent une image de scène dans 3 ou au plus 10 bandes spectrales alors que les systèmes IHS acquièrent des centaines de bandes contiguës ;
- Les systèmes multispectrales ont une résolution spectrale (longueur d'onde centrale divisée par la largeur de la bande spectrale) de l'ordre de 10, alors qu'elle est de l'ordre de 100 pour les systèmes IHS.
- les systèmes multispectrales acquièrent les images dans des bandes de longueur d'onde larges et irrégulièrement espacées, cependant que les systèmes IHS ont des bandes

spectrales contiguës et régulièrement espacées permettant d'obtenir un spectre quasi-continu pour chaque pixel.

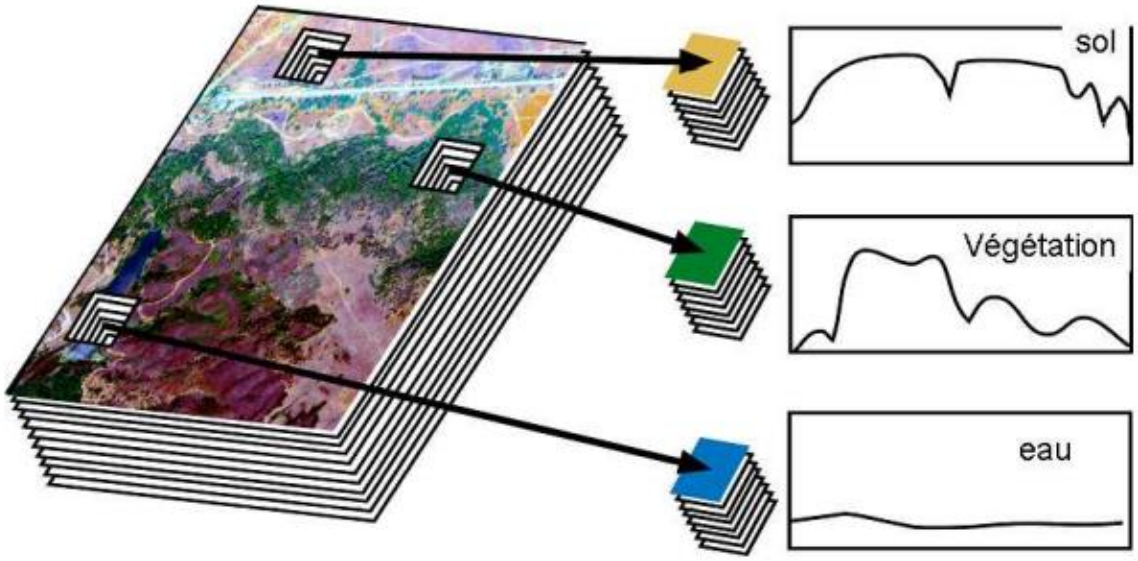


Figure 14. Exemple d'image hyperspectrale

Les images IHS fournissent donc une information plus détaillée des propriétés spectrales d'une scène et permettent de faire une identification et une discrimination plus précises des objets que les images RGB (Figure 13)

Sur le plan figuratif, les images hyperspectrales recueillent l'information sous la forme d'un ensemble d'images. Chaque image représente une plage de longueurs d'onde étroite du spectre électromagnétique, également appelée bande spectrale. Ces «images» sont combinées pour former un cube de données hyperspectrales tridimensionnel (x, y, λ) pour le traitement et l'analyse, où x et y représentent deux dimensions spatiales de la scène, et λ représente la dimension spectrale (comprenant une plage de Longueurs d'onde).

III.1.2. Domaine d'application

L'imagerie hyperspectrale a répondu aux besoins dans plusieurs domaines, elle est utilisée dans un large éventail d'applications, bien qu'elle soit développée à l'origine pour l'exploitation minière et la géologie [2] [3], mais elle est utilisée dans des domaines aussi répandus comme l'écologie et la surveillance, ainsi que l'agriculture etc.

- Agriculture et la végétation

Il s'agit de répondre à des nouveaux besoins sociétaux (agriculture raisonnée, biocarburants, protection des espaces naturels, évaluation des risques naturels, etc.) ou à des questions scientifiques émergentes (biodiversité, cartographie des écosystèmes terrestres, étude du changement climatique, etc.).

La crise alimentaire qui menace une partie du monde, les nouvelles directives sur la protection de l'environnement, les effets du réchauffement global sur les espèces végétales cultivées ou naturelles, et leur effet rétroactif, sont autant des raisons pour mieux connaître la végétation et sa dynamique. Dans ce sens, Des travaux ont été réalisé qui se basent sur l'imagerie hyperspectrale, afin de détecter la composition chimique des plantes [12], mettre au point des systèmes d'alerte précoce pour les éclosions de maladies dans des zones de végétation [13],

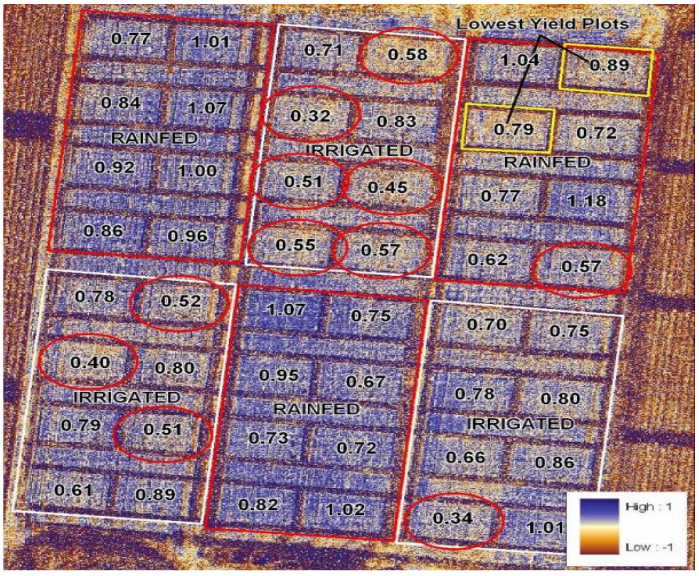


Figure 15. Image hyperspectral présente l'azote et le stress hydrique dans le blé

- Les Géosciences – Sciences de la Terre Solide.

L'analyse spectrale du rayonnement solaire réfléchi et du rayonnement émis par la surface terrestre constitue une importante source d'information sur les caractéristiques chimiques et minéralogiques des matériaux. L'intérêt de l'imagerie hyperspectrale pour les applications géologiques a été démontré dans les années 1980. Aujourd'hui, cette technique est en plein essor dans le domaine des Sciences de la Terre et de l'Environnement.

Les échantillons géologiques, tels que des carottes de forage, peuvent être rapidement cartographiés par imagerie hyperspectrale, plus que ça, l'imagerie hyperspectrale permet la détection d'un ensemble de minéraux car ils ont une signature spectrale très distinctive [14].

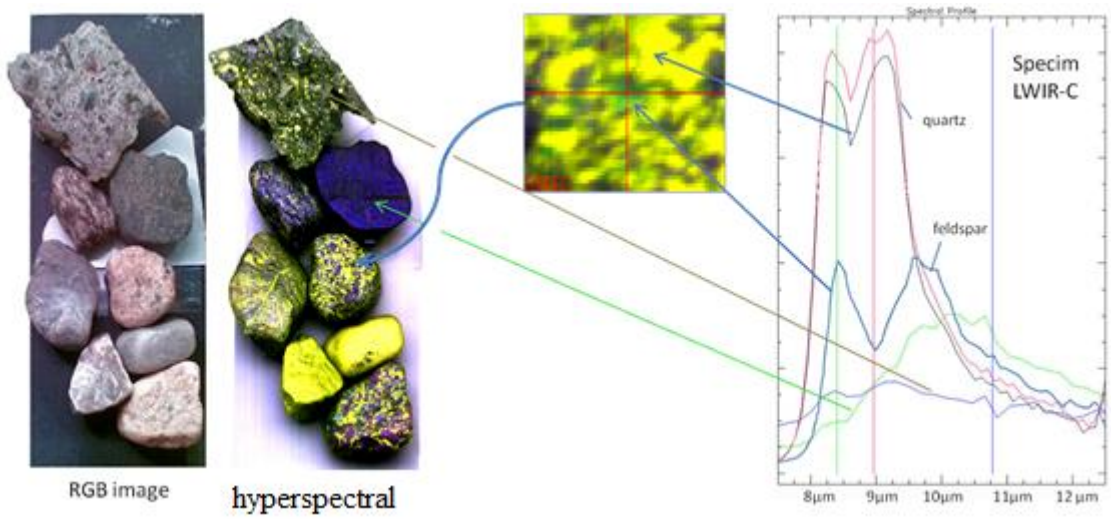


Figure 16.Hyperspectral image pour la détection des minéraux

- Le milieu urbain

La ville est un milieu complexe qu'est connu pour sa diversité à plusieurs points de vue. Premièrement par son aspect physique, c'est-à-dire ses caractéristiques géographiques et géométriques. Deuxièmement par sa dimension métaphysique, c'est-à-dire par les différentes dimensions sociale, culturelle, économique et politique de son fonctionnement. C'est pourquoi, la surveillance, l'aménagement et le développement du milieu urbain jouent un rôle important de nos jours.

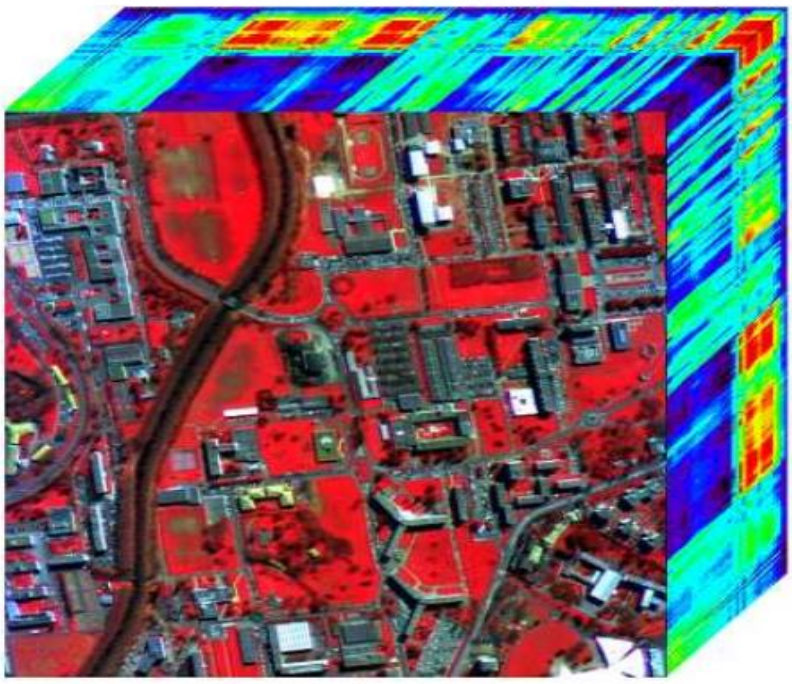


Figure 17.exemple d'image hyperspectrale d'une zone urbain

Les images hyperspectrales sont un moyen intéressant pour relever, étudier et accompagner l'évolution de ces éléments urbains. En effet, l'utilisation des images standard ne permet pas de discriminer des classes spectrales semblables (bâtiments et différentes classes de routes par exemple), alors que l'exploitation de données hyperspectrales le permet [15].

Dans ce contexte, l'imagerie spectrale apporte des réponses à plusieurs niveaux :

- Caractérisation et cartographie des matériaux urbains.
- Qualité de l'air (détection et caractérisation des aérosols).
- Caractérisation de la végétation dans les milieux urbains.
- Amélioration des systèmes d'information géographiques urbains

- Santé

Grâce à la technologie de l'imagerie hyperspectrale, il est possible de diagnostiquer une maladie, de sauver des vies et d'améliorer les thérapies. La médecine moderne dépend de l'imagerie hyperspectrale pour établir des diagnostics ou superviser en temps réel les interventions chirurgicales très complexes, par exemple, dans le diagnostic de la rétinopathie et de l'œdème maculaire, L'imagerie hyperspectrale détecte une baisse de la consommation d'oxygène dans la rétine, ce qui indique une maladie avant que l'œdème ne soit endommagé [18].

- Détection des composants chimiques

L'imagerie hyperspectrale est très efficace pour la détection des composants chimiques, c'est pour cela qu'elle est pratiquement utilisée pour les détecter et les classifier par exemple dans l'agriculture, les engrais et les produits chimiques sont couramment utilisés pour le sol et les cultures pour maintenir les cultures cultivées en bonne santé et leur fournir des éléments essentiels pour le développement des cultures. L'utilisation de l'imagerie hyperspectrale dans l'agriculture peut améliorer considérablement l'efficacité de l'utilisation des produits chimiques et réduire les dommages causés à l'environnement, ainsi elle fournit une solution complète pour la surveillance et le diagnostic à grande échelle des terres agricoles.

Très loin de l'agriculture, Les soldats peuvent être exposés à une grande variété de risques chimiques. Ces menaces sont pour la plupart invisibles mais détectables par la technologie d'imagerie hyperspectrale [19].

- L’industrie

Dans l'industrie de la transformation des aliments, l'imagerie hyperspectrale, combinée à un logiciel intelligent, permet aux trieurs numériques (également appelés trieurs optiques) d'identifier et de supprimer les défauts et les matières étrangères invisibles aux trieuses traditionnelles de caméras et de laser [20].

III.2. Classification des images Hyperspectrale à base de Deep Learning.

III.2.1. Enjeux

Plusieurs facteurs, ont été reconnus et qui influencent sur les résultats de classification des images hyperspectrales :

- la dimension élevée des informations spectrales (c'est-à-dire des centaines de bandes spectrales corrélées) produit un phénomène de Hughes [21] qui peut réduire significativement la précision de classification.
- plusieurs objets dans une image hyperspectrale peuvent partager des propriétés spectrales similaires (par exemple, dans une zone urbaine les parcs et les toits ont des propriétés presque similaires), ce qui rend même impossible de classer les HSI en utilisant seulement les informations spectrales.
- la précision de classification souffre beaucoup de la résolution spatiale qui est très élevée, en outre Il est essentiel d'introduire l'information structurelle et contextuelle dans le domaine spatial pour la classification des images.
- Le nombre limité des données hyperspectral étiquetées.

Sur la base de ces problèmes, il est nécessaire de suivre un processus bien définis afin d’assurer des bonnes résultats de classification qui commence par une phase de réduction de la dimension, puis une autre phase d’extraction des caractéristiques spatiales et spectrales, et finalement appliquer l’algorithme de classification.

III.2.2. Réduction de dimension

Lorsqu'on étudie simultanément un nombre important des variables quantitatives comme le cas des images hyperspectrales, le problème c'est que tous les individus étudiés ne sont pas représenté dans un plan. L'objectif de La réduction de dimension est de revenir à un espace de dimension inférieur au premier qui déformant le moins possible la réalité et en gardant uniquement le résumé le plus pertinent des données initiales.

On peut distinguer entre deux approches pour La réduction de dimension :

- Réduction de dimension avec sélection des caractéristiques (Features Selection).
- Réduction de dimension avec extraction des caractéristiques.

Les approches de sélection de caractéristiques consistent à faire une déduplication des informations présente dans les images sans compromettre le contenu d'origine des cubes d'images brutes, et garder seulement les données jugées pertinentes, l'avantage de ces approches est de permettre la préservation des informations d'origine, et permettre aussi de préserver l'espace de représentation. Dans la littérature, nombreux sont les méthodes de sélection des caractéristiques, par exemple les méthodes de filtres (Filter Methods), qui sont des méthodes heuristiques qui se base sur les caractéristiques globales des données pour évaluer la mérite de ces caractéristiques. ces méthodes sont généralement très rapide, et pratique pour la réduction de dimension, parmi ces méthodes on peut citer la méthode de FOCUS [22], RELIEF [23], ainsi que la méthode de LVF [23], aussi on peut utiliser des méthodes d'apprentissage pour la sélection des caractéristiques comme les arbres de décisions, C4.5 et ID3 (extensions des arbres de décisions)[23].

Pour les approches d'extraction de caractéristiques, une transformation linéaire ou non linéaire est appliquée aux caractéristiques d'origine pour extraire certaines nouvelles fonctionnalités, leurs objectif est de revenir à un espace de représentation inférieure au initial et qui déforme le moins possible les données initiales, et qui garde les informations pertinents, ces techniques sont divisées en deux catégories :

- supervisées, qui utilisent les informations étiqueter, comme l'analyse discriminante linéaire (LDA) et DAFE, DBFE [24].
- non supervisé, qui n'utilisent pas les informations d'étiquette de classe pour l'apprentissage L'analyse des composantes principales (PCA), Locally Linear Embedding [25], MPCA [23].

III.2.3. Classification basé sur les caractéristiques spectrales

La classification des images hyperspectrales basé sur les caractéristiques spectrale, nécessite une étape d'extraction de ces information spectrale qui consiste à identifier une structure c'est à dire un ensemble de spectres communs à tous les pixels de l'image et définissant un sous espace vectoriel dont la dimension est inférieure à celle de l'espace initial.

Généralement les méthodes d'extraction des caractéristiques spectrales fait toujours appel à des méthodes de réduction de dimension, par exemple PCA, Minimum Noise Fraction, LDA, Minimum Change Rate Déviation (MCRD) [26].

Puisque le Deep Learning a montré sa performance dans plusieurs domaines, les chercheurs ont inventé des modèles basés sur les méthodes du Deep Learning pour extraire les caractéristiques spectrales des HSI.

Dans [26] une méthode de classification a base des caractéristiques spectrales a était proposé, cette méthode est basé sur DBN (Deep belief network). La méthode proposé prend comme entré de DBN, une vecteur correspond au spectre du chaque pixel pour extraire ces caractéristique, après classificateur est appliquer pour classifier ces caractéristiques :

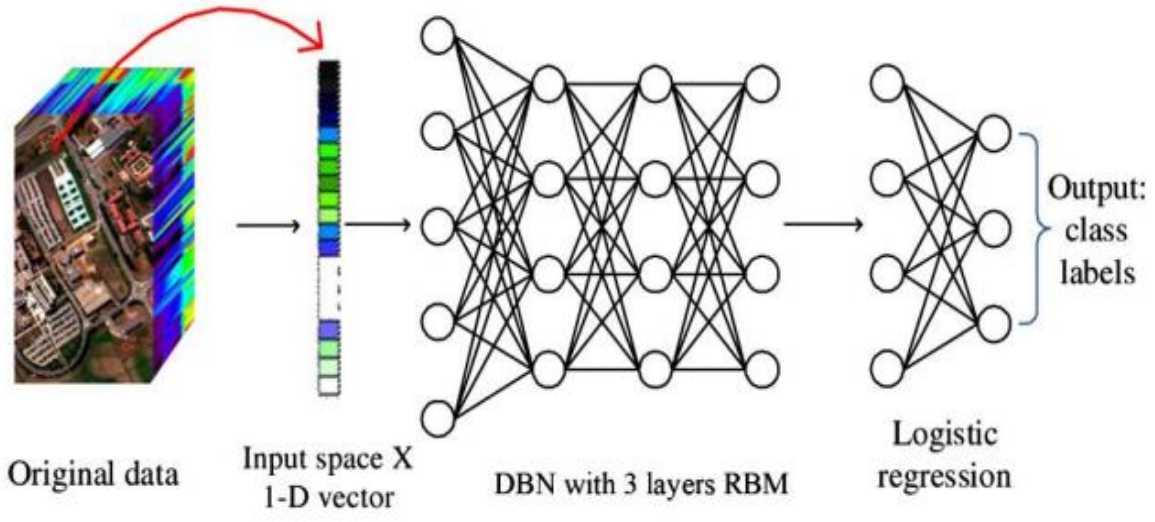


Figure 18. Extraction des caractéristiques spectrales à base de DBN

Toujours dans ce sens d'extraction des caractéristiques spectrales des données hyperspectrales, une méthode proposé dans [27] basée sur la méthode d'Autoencoders.

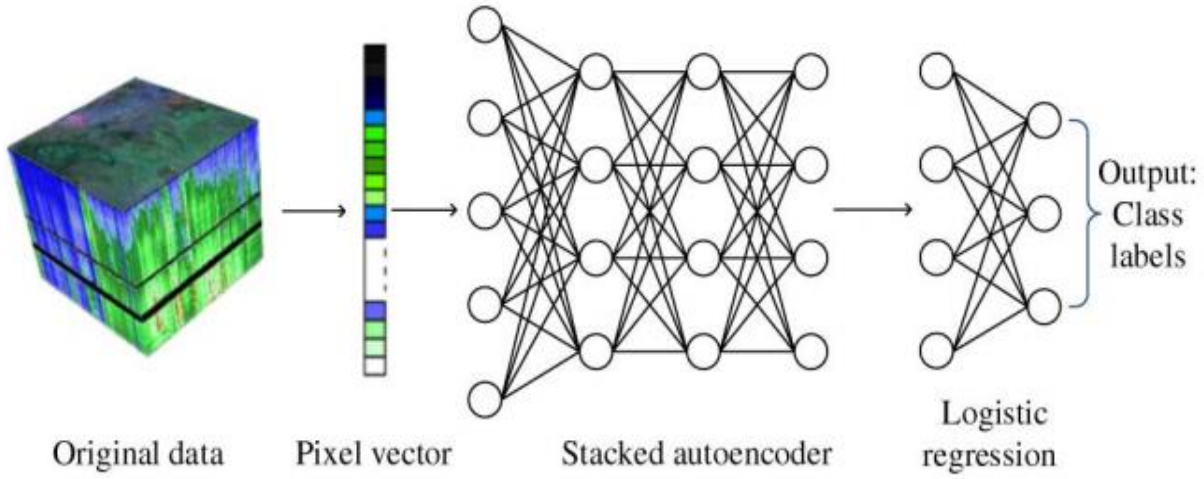


Figure 19.extraction des caractéristiques spectrales basé sur l'autoencoders

III.2.4. Classification basé sur les caractéristiques spatiales

Les caractéristiques spatiales sont très utiles pour améliorer la représentation des données des images hyperspectrales et pour améliorer la précision de classification. Ces dernières années, des nombreuses études ont été faites dans ce sens.

L'extraction des caractéristiques spatiales dans les images hyperspectrales nécessite généralement des filtres spatiaux prédéfinis qui sont déterminés en fonction de la connaissance du problème: Matrices de co-occurrence de niveau de gris (GLCM), Spectres de texture (wavelet texture), les Profils morphologiques (EMP) et les profils d'attributs [28].

Les caractéristiques spatiales sont spécifiques en termes de but, ce qui signifie que seul un type spécifique d'objets peut être détecté par chaque configuration, cependant, les propriétés spatiales des images hyperspectrales présentent une grande variété, ce qui rend impossible de décrire tous les types d'objets.

Récemment, le Deep Learning a montré sa capacité de générer des caractéristiques spatiales de haut niveau (Deep Features). Chen et al. [27], [29] ont présenté une méthode basée sur l'algorithme Stack AutoEncodeur (SAE) pour classifier les images hyperspectrales en se basant sur ces caractéristiques spatiales (Figure 20).

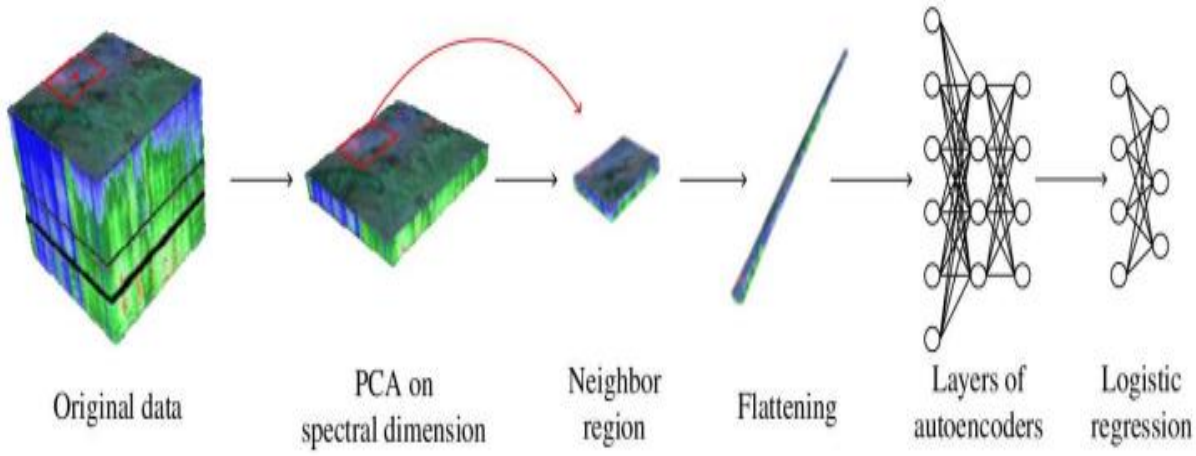


Figure 20.Classification et extraction des caractéristiques spatiales basée sur SAE

Cette architecture commence d'abord par une étape de réduction de dimension de l'image hyperspectrale via la méthode du PCA, la deuxième étape consiste à extraire une région spatiale de l'image réduit, puis transformer la région en vecteur, ce dernier va être entré du SAE, et finalement une étape de finalisation du modèle via une méthode de la régression.

Une autre méthode a était proposé par chen et al [30] pour d'extraction de caractéristique bidimensionnelle basée sur le réseau neuronal à convolution (CNN) pour détecter des véhicules à partir d'images à haute résolution spatiale. Cette architecture peut être divisée en deux parties, La première est l'extracteur des caractéristiques, en utilisant les couches convolutionnelles et les couches de max-pooling. La deuxième partie est un perceptron multicouche (MLP), qui classe les données en se basant sur les caractéristiques extraites (Figure 21).

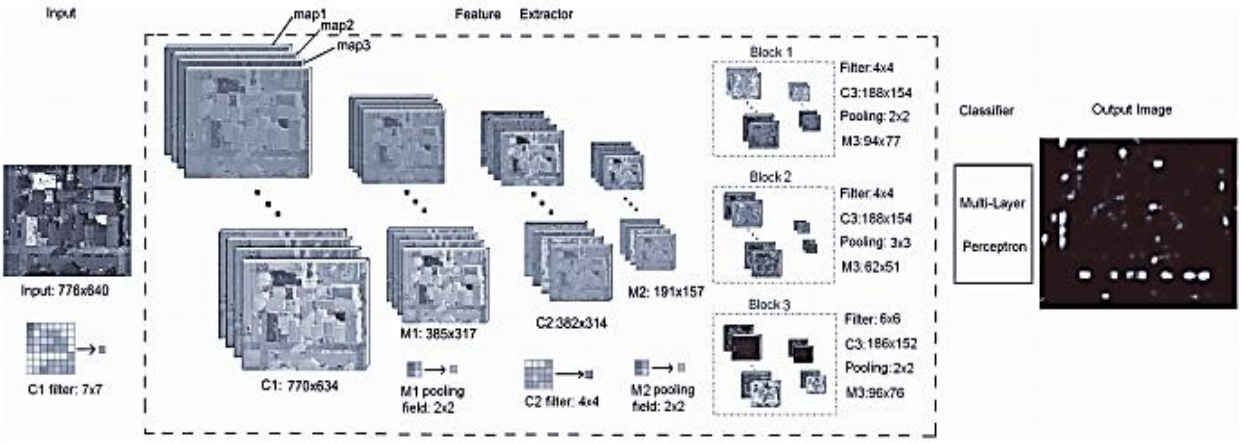


Figure 21.Extraction des caractéristiques spatiales basée sur le CNN

III.2.5. Coopération spatio-spectral

La prise en compte de l'information spectrale et spatial est très utile et importante dans le cas de problème de classification très difficiles où les objets à discriminer sont très proches spectralement comme le cas des HSI, c'est pour cela que la tendance dans ce domaine vise à faire une coopération entre les caractéristiques spatiales et spectrales.

Les auteurs de [27] ont proposé un modèle pour intégrer les caractéristiques spatiales et spectrales pour construire un extracteur des caractéristiques Spatio-Spectrales basé sur Stack Autoencoders (SAE).

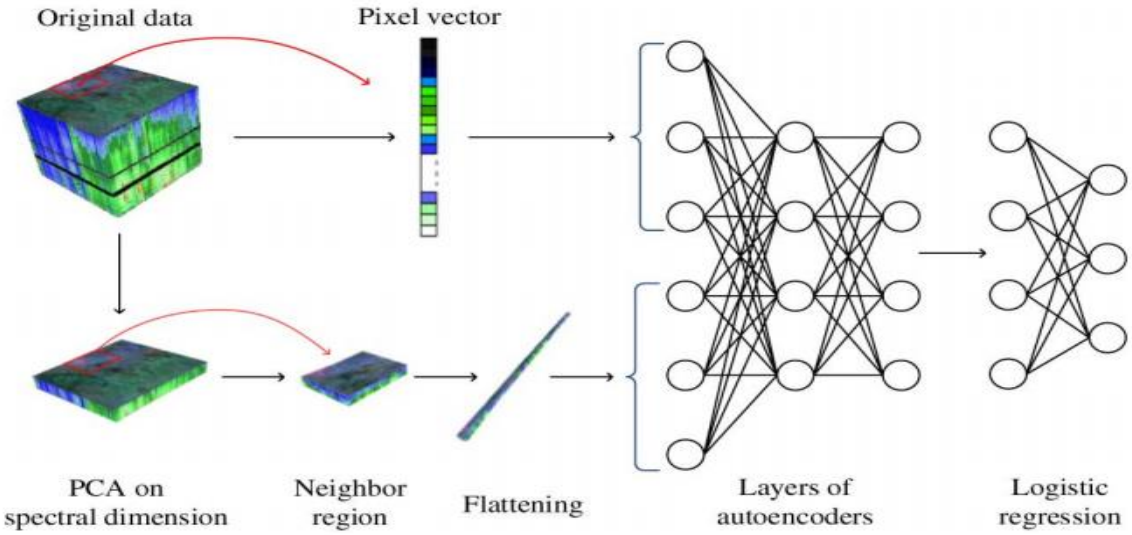


Figure 22. Architecture proposé par [27], qui combine les caractéristiques spectrales et spatiale

Une autre architecture basée sur le Deep Learning présenté dans [10], qui exploite la capacité de DBN pour extraire les caractéristiques spectrales et spatiales profondes.

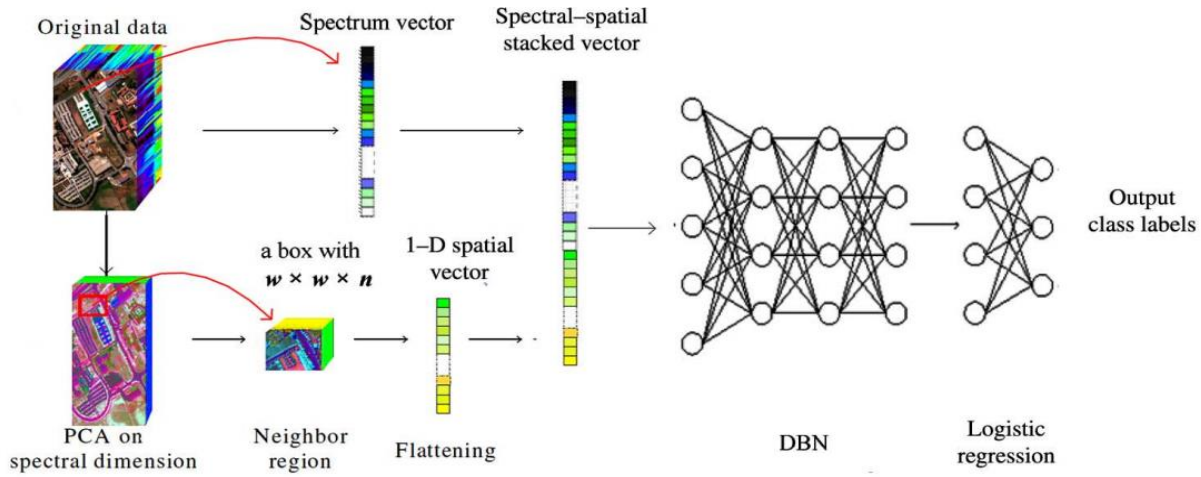


Figure 23. Classification des images hyperspectrales basée sur les caractéristiques spatio-spectrales en utilisant le DBN

III.3. Conclusion

Après avoir présenté le concept d'imagerie hyperspectrale, ses domaines d'application ses enjeux, ainsi que le processus de classification de ces image en se basant sur les méthodes de Deep Learning, le prochain chapitre va être l'objet de faire une analyse et conception du Framework, qui va nous faciliter le développement par la suite.

Chapitre.IV Analyse et conception

Introduction

Dans le cadre du projet, nous sommes amenés à développer un Framework pour la classification et la segmentation des images et particulièrement les images hyperspectrales à base des méthodes de Deep Learning.

Après avoir cité les exigences techniques et fonctionnelles de notre projet, ainsi que présenté l'état de l'art des méthodes de Deep Learning et des images hyperspectrales, dans ce chapitre, on va commencer une phase importante et indispensable dans le cycle de vie d'une application. Cette phase est l'analyse et la conception qui a pour but de faire une modélisation du Framework, en se basant sur les connaissances acquises durant les phases précédentes.

IV.1. Diagramme de cas d'utilisateurs

On se basant sur le cahier des charges présenté dans le premier chapitre, on a pu regrouper les fonctionnalités de notre Framework dans un diagramme de cas d'utilisation (Figure 24), qui permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.

Le Framework va permettre à l'utilisateur de :

- Configurer un réseau de neurones profonds
- Apprendre le réseau avec une base de données d'apprentissage.
- Après l'apprentissage l'utilisateur peut stocker le modèle appris.
- Par la suite, l'utilisateur charge le modèle pour l'évaluer avec une base de données de Test.
- Une fois que le modèle est validé après la phase de test, on peut l'utilisé pour classifier des images

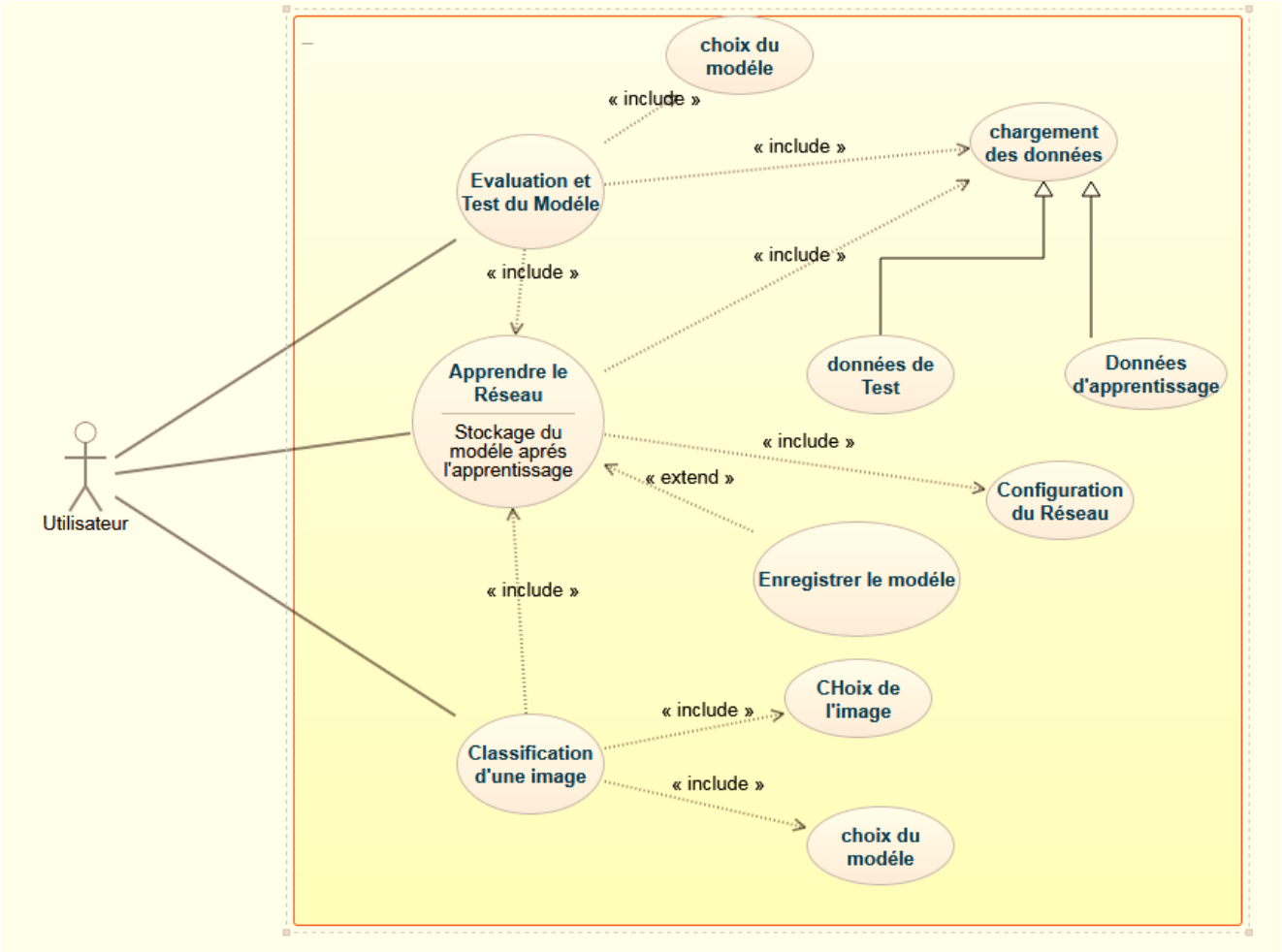


Figure 24. Diagramme de cas d'utilisateur de notre Framework

IV.2. Architecture

Il est primordiale à la conception de tout système informatique de choisir le modèle d'architecture qui lui sera adéquat, pouvant assurer un bon fonctionnement, des meilleurs performances ainsi que la réutilisation et l'interconnexion fiable de ce système avec d'autres.

C'est à cet effet que nous optons pour le modèle MVC (Model - View - Controller), cette architecture permet plus qu'organisé notre code, de simplifier la tâche de maintenance et d'amélioration sur projet (Figure 25).

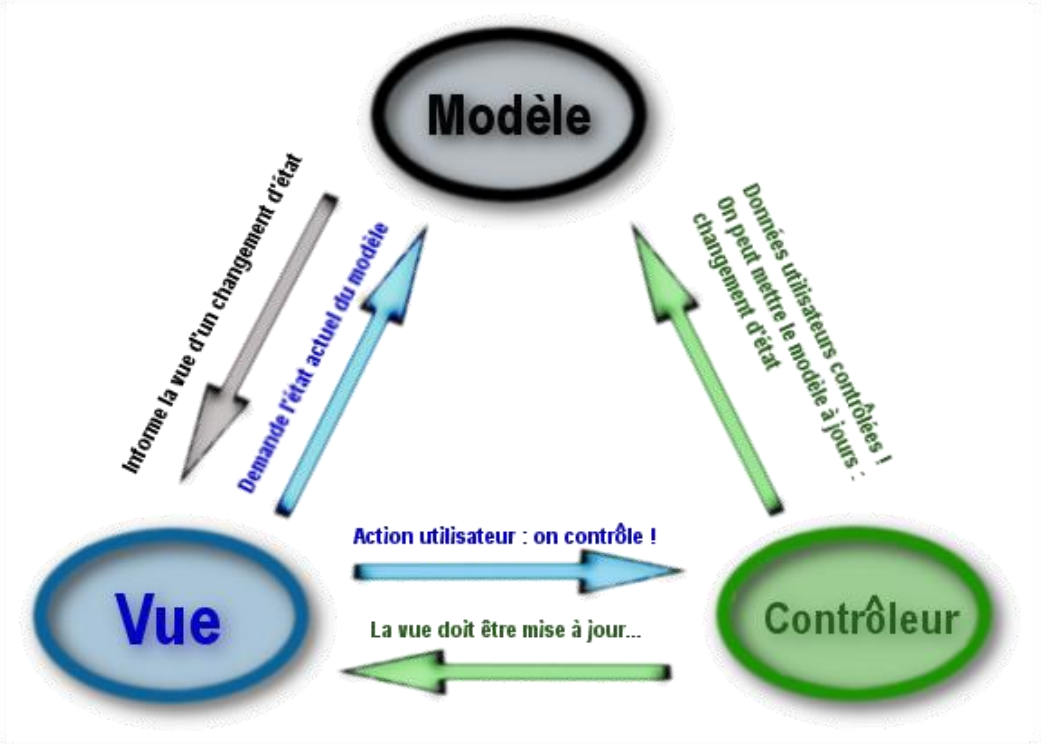


Figure 25 Architecture MVC

Dans l'architecture MVC, les rôles des trois entités sont les suivants :

- modèle : Manipulation des données (accès, mise à jour, stockage)
- vue : interface graphique GUI avec l'utilisateur (entrées et sorties).
- contrôleur : Traitement et gestion des événements et synchronisation

Les avantages apportés par l'architecture MVC sont :

- La séparation des données de la vue et du contrôleur (ce qui permet une conception claire et efficace de l'application)
- Une indépendance des données, de l'affichage et des actions (ce qui donne plus de souplesse pour la maintenabilité et l'évolutivité du système).
- Un gain de temps de maintenance et d'évolution de l'application.

IV.2.1. la partie model

Le modèle représente les structures de données. Typiquement, les classes modèles contiennent des fonctions qui aident à récupérer, insérer et mettre à jour des informations de la base de données.

Ci-dessous, Le diagramme de classes de la partie modèle :

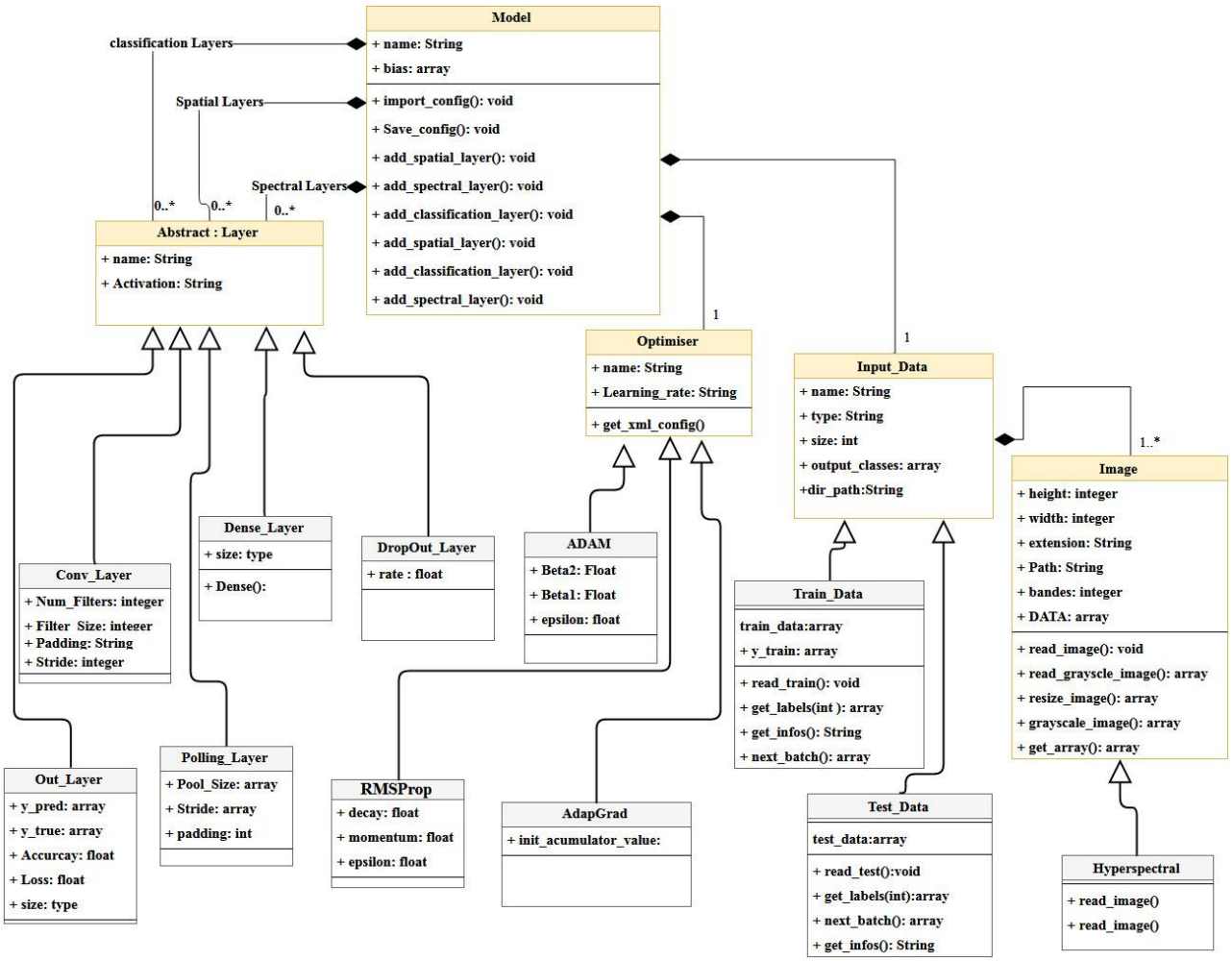


Figure 26.Le diagramme de classe pour la partie model de notre Framework

Le tableau ci-dessous décrit la fonctionnalité de chaque classe:

Tableau 1. Description des classe de la partie Modèle

Classe	Description
Model	La classe principale de la partie Modèle, cette classe a comme rôle de gérer toutes les variables de notre modèle d'apprentissage et d'offrir l'interface de connexion avec La partie Controller
Input_Data	Permet la gestion des données d'apprentissage (chargement, traitement…), elle est la classe mère de : • Train_Data : gestion des données d'apprentissage • Test_Data : gestion des données de test
Image	Permet de manipuler les images (chargement, traitement..), elle est la classe mère de : • Hyperspectral qui permet de manipuler les images hyperspectrales

Optimiser	Permet de gérer les paramètres des différents Optimiser : • SGD (Stocastic gradient descend) • Adam (Adam Optimiser) • AdapGrad • RMSprop
Layer	Permet de gérer les différents paramètres de différents couches du modèle, elle les classe Mère de : • Conv_layer : la couche convolutionnelle • Poolling_Layer : la couche de Pooling • DropOut_Layer : la couche de Dropout • Dense_Layer : la couche totalement connecté

IV.2.2. la partie Vue

Spécification de cette partie comprenant l'interface graphique GUI, cette partie représente la spécification des pages, les différentes opérations offertes pour chacune des pages, et les fonctionnalités de chacune des pages.

Chaque page occupe (1200x800) et comporte un ensemble d'objets graphiques selon le besoin.

La structure générale des pages est conçue se forme d'une hiérarchie du page

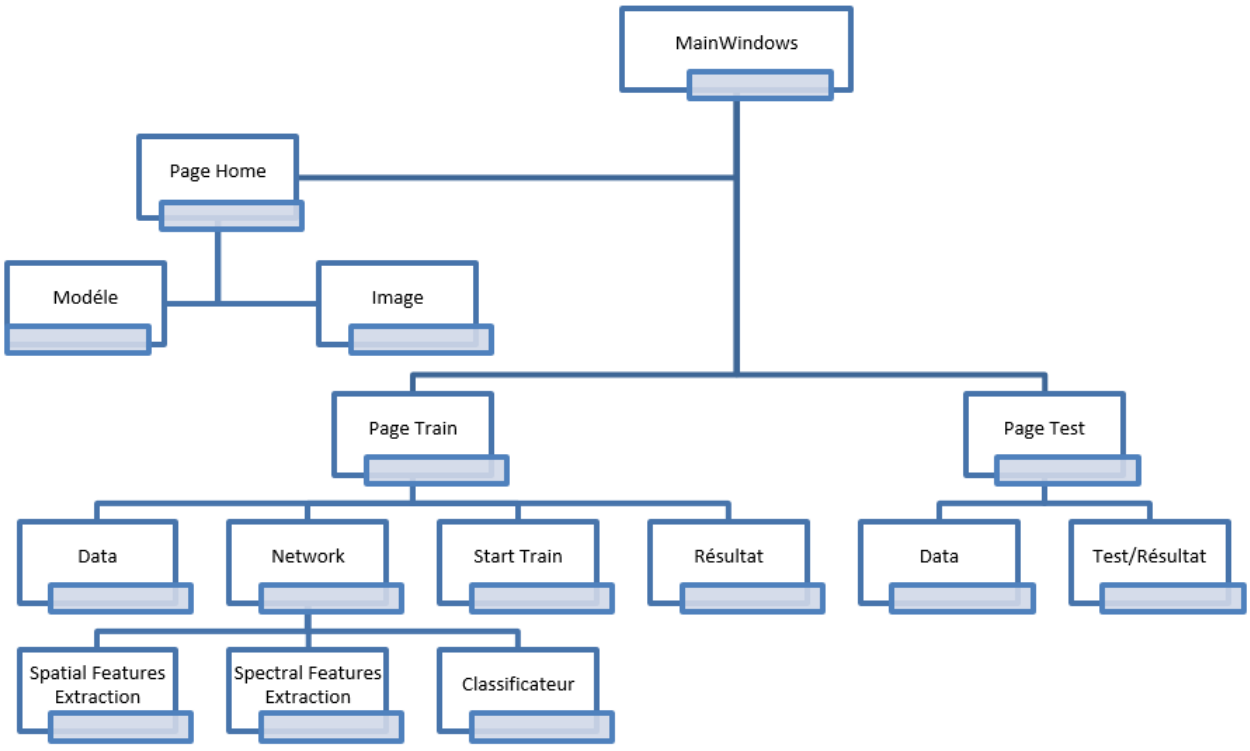


Figure 27. Diagramme présente l'arborescence des pages du Framework

Les pages de notre Framework permettent à utilisateur de faire une configuration de tous les paramètres des modèles au niveau d'apprentissage ainsi qu'au niveau du test, pour cela on 'a choisi de poser tous les pages sur un calque qui va fluidifier l'expérience de l'utilisateur, ci-dessous une description du rôle de chaque page :

Tableau 2. Tableau décrit la fonctionnalité de chaque page de la partie vue

Page	Description
La Page Home	La page Home permet à l'utilisateur d'importer un modèle enregistrer après la phase d'apprentissage, visualiser les informations relatives au modèle ainsi que choisir une image et la classifier en utilisant le modèle choisit. Cette page contient de sous page : • Modèle : Importer le Modèle. • Image : Importer l'image à classifier et lancer la classification.
La page Train	La page Train est divisé on 4 onglets : • Data : Permet d'importer les données d'apprentissage et visualiser ses informations • Network : Cette page permet de configurer l'empilement des couches de notre réseau de neurones, elle est divisée en trois sous-pages qui permettent de configurer le modèle d'apprentissage sur trois niveaux : o Spatial Features Extraction : la partie du modèle qui va extraire les caractéristiques spatiales de l'image o Spectral Features Extraction : la partie du modèle qui va extraire les caractéristiques spectrales de l'image o Classification : les couches de classification • Train : Cette page permet de configurer les paramètres d'apprentissage (taux d'apprentissage, batch_size, critère d'arrêt…) • Results : Cette page permet de Visualiser le résultat d'apprentissage (Durée d'apprentissage, Loss, la précision …)
La page Test	La page de Test à son tour est divisée en deux sous-pages : • Data : Cette page permet d'importer la base de Test • Results : Cette page permet de visualiser le pourcentage de Taux de reconnaissance pour la base de Test

IV.2.3. la partie Controller

Dans cette partie nous avons défini les opérations qui traitent les actions de l'utilisateur et qui modifie les données du modèle et de la vue, c'est-à-dire la partie qui s'occupe de l'interfaçage entre le modèle et le vue.

On peut diviser cette partie en trois sous scénarios importantes :

L'apprentissage

Après la configuration Globale du Modèle, l'utilisateur lance l'apprentissage, c'est l'opération la plus importante dans le Framework qui demande une connexion entre toutes les parties de l'architecture MVC et qui suit un processus bien défini :

- Importer les données d'apprentissage
- Configuration de Modèle
- Apprentissage
- Générer le Modèle

Le schéma ci-dessous représente le scénario, et la communication entre les parties de MVC pour apprendre le modèle :

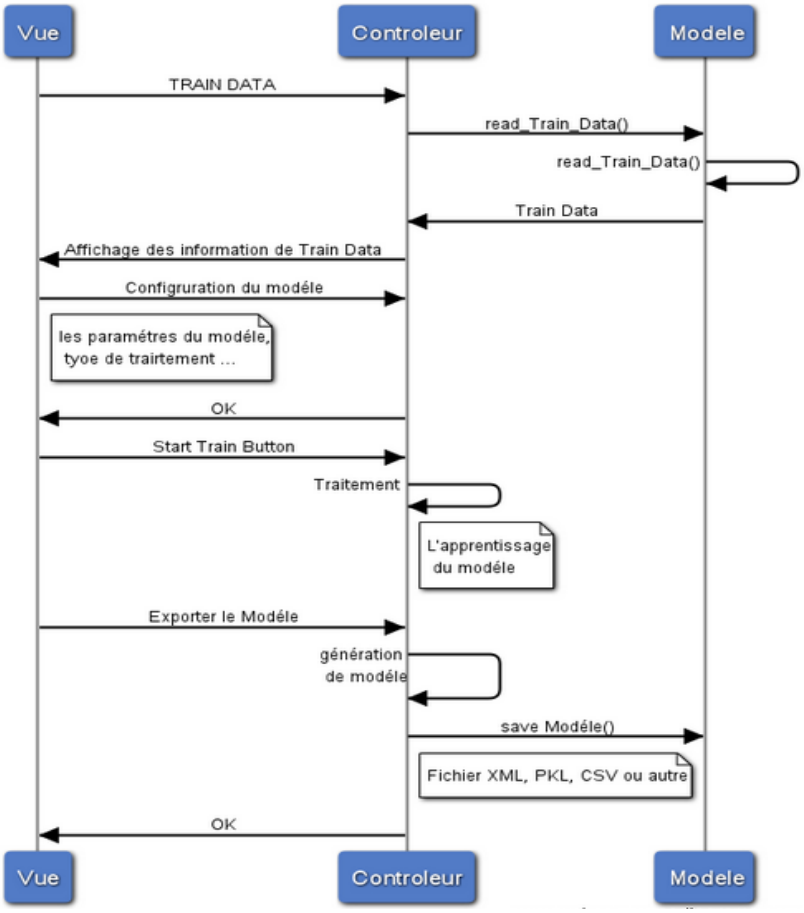


Figure 28.diagramme de séquence présente le scénario de la phase d'apprentissage

Test

Après la phase d'apprentissage et la génération du modèle, une phase de Test est nécessaire, pour évaluer la performance du modèle. Cette phase à sa part suit un processus bien définit :

- Importer les données d'apprentissage
- Importer le Modèle
- Lancer le Test

Le schéma ci-dessous représente le scénario de Test:

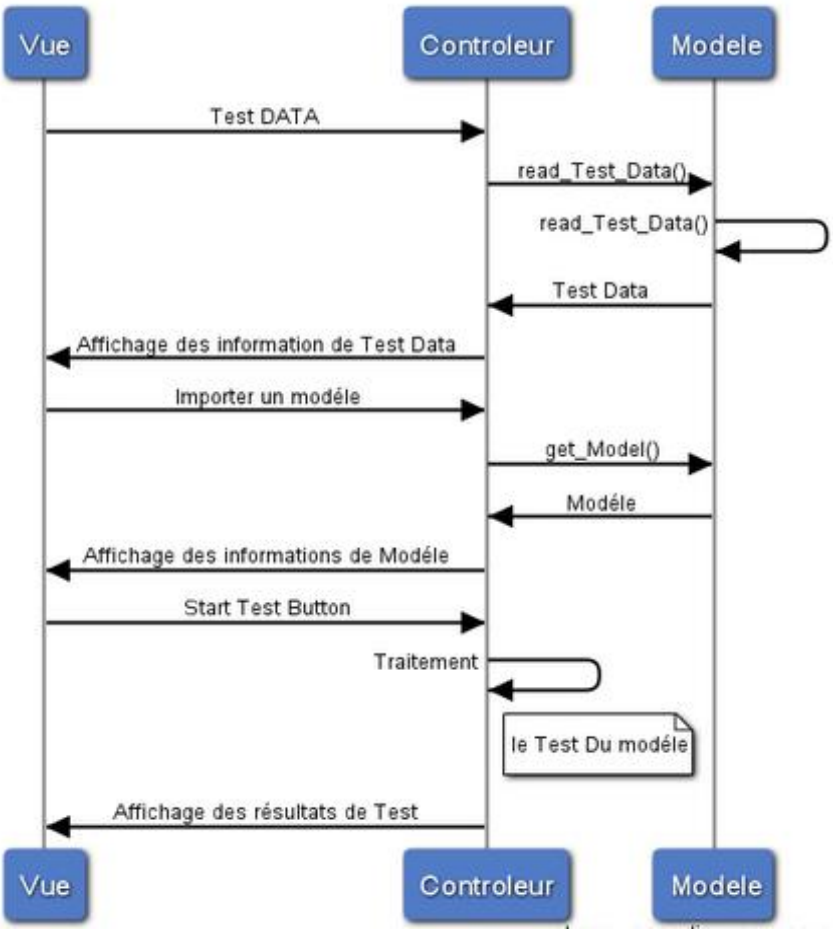


Figure 29. Diagramme de séquence présente le scénario de la phase de test

Classification

Après la génération du modèle et l'évaluation de sa performance, la phase de classification permet de classifier une image donnée par l'utilisateur, qui est à son tour suit un processus bien défini :

- Choix de l'image requête.
- Choix du modèle de classification
- Classification de l'image

La partie Controller est développée se forme de plusieurs classes basées sur le principe d'orienté objet, Ci-dessous une visualisation des classes présentent dans la partie Controller ainsi que la relation entre eux :

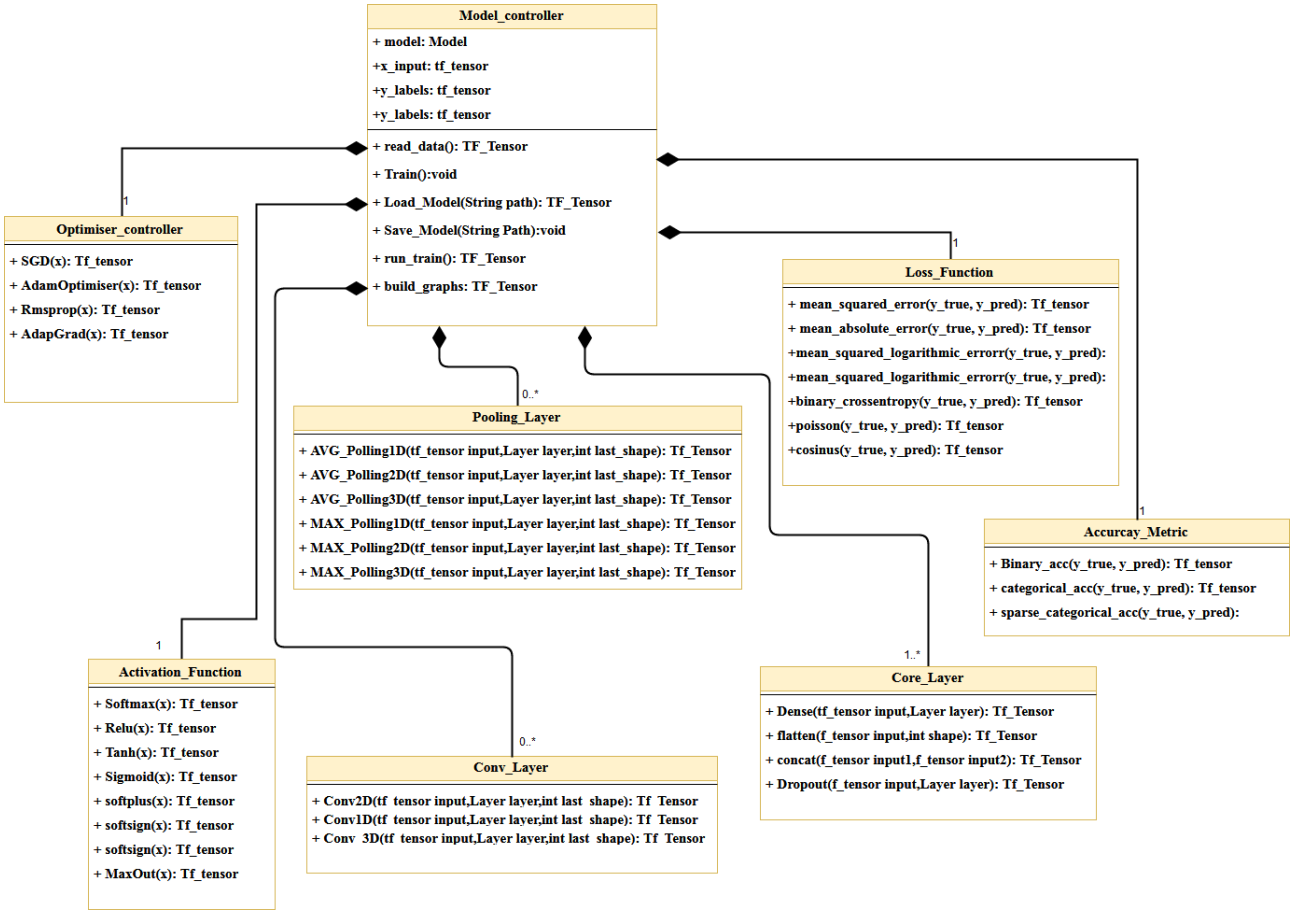


Figure 30. Diagramme de classe de la partie Controller

Tableau 3 . Tableau décrit les classes de la partie Controller

Page	Description
Model_controller	Cette classe permet • lancer l'apprentissage du modèle. • enregistrer le modèle après l'apprentissage. • charger le modèle. • évaluer le modèle.
Loss_Function	Cette classe contient ensemble de fonction permet de calculer la perte (Loss) du Modèle.
Accurcay_Metrics	Cette classe contient ensemble de fonction permet de calculer la précision (Accurcay) du Modèle.
Optimiser_controller	Cette classe contient ensemble d'Optimisateur (Optimizer) qui permet de minimiser l'erreur du modèle
Activation_Function	Cette classe contient ensemble de fonction d'activation :

Conv_Layer	Permet d'ajouter des couches convolutionnelles au modèle
Pooling_Layer	Permet d'ajouter des couches Pooling au modèle
Core_Layer	Permet d'ajouter un ensemble des couches au modèle (Dropout, totalement connecté…)

IV.3. Conclusion

Après avoir présenté la conception détaillée de notre système, nous sommes capables maintenant d'entamer la partie réalisation et démonstration du Framework qui sera l'objet du chapitre suivant.

Chapitre.V Réalisation et démonstration du Framework

Introduction

Dans ce chapitre nous allons exposer le travail achevé.

Nous présenterons en premier lieu l'environnement matériel et logiciel dans lesquels notre application a été développée en indiquant les technologies utilisées. Nous clôturons ce chapitre par lister les différentes fonctionnalités de notre Framework, avec quelques captures d'écran traduisant le déroulement de l'application.

V.1. Environnement de travail

V.1.1. Environnement Matériels

Nous avons élaboré ce travail sur ensemble de matériels dont les principales caractéristiques sont les suivantes:

- HP Elitebook 8460:
  - CPU : 8GB
  - GPU : intel(R) HD Graphics Family
  - Disque dure : 500 GB
  - Processeur : Intel(R) Core(TM) i5-2520M CPU @ 2,50GHZ (4CPUs).
- Dell Precision M6800:
  - CPU : 12GB
  - GPU : cartes graphiques AMD FirePro™
  - Disque dure : 750 GB
  - Processeur : Intel® Core™ de 4e génération, i7

V.1.2. Environnement logiciel et APIs utilisées

L'environnement logiciel employé lors le développement de Framework est :

- Système d'exploitation linux (Ubuntu 14.04.5 LTS).
- Pycharm comme IDE de développement Python
- QTCreator comme IDE de développement C++
- langage de programmation : Python (Python 3.4)
- langage de la programmation C++

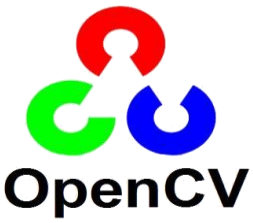
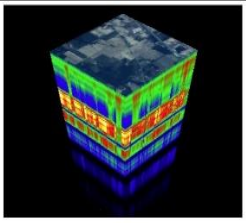
Lors de développement un ensemble d'outils et APIs ont était utilisés, le choix de ces outils n'était pas anodin, mais il vient après une étude benchmarking appropries à notre matérielle, la plateforme, et le langage avec lequel nous avons travaillé, sur toutes les outils et les solutions existants:

- **Annexe1** : présente une étude benchmarking sur les outils et les APIs de Deep Learning. Après cette étude, on a décidé de travailler avec l'API Tensorflow pour plusieurs raison :
  - Il est Open source.
  - Il Possède une documentation très riche.
  - sa communauté est très active.
  - Il Supporte l'exécution du traitement sur GPU et CPU.
  - Il Fournit une API en python ainsi qu'en C++.

- **Annexe2** : présente une étude benchmarking sur les outils et les APIs de Traitement des images hyperspectrales, on a choisi de travailler avec l'API GDAL et SPY, pour les raisons suivants :
  - Ils Supportent la majorité des formats d'images hyperspectrales.
  - Ils Possèdent une documentation riche.
  - Ils sont Open source.
  - GDAL fournit une API en plusieurs langages (Python, C++, R…).

Le tableau ci-dessous regroupe l'ensemble des Outils utilisés ainsi que l'utilité de chacun :

Tableau 4.Outils et APIs utilisés pendant le Développement du Framework

Outils/API	Logo	Description
git		Un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git.
PyQT		PyQt est un module libre qui permet de lier le langage Python avec la bibliothèque Qt distribué sous deux licences : une commerciale et la GNU GPL. Il permet ainsi de créer des interfaces graphiques en Python
Doxygen		Un générateur de documentation sous licence libre capable de produire une documentation logicielle à partir du code d'un programme. Pour cela, il tient compte de la grammaire du langage dans lequel est écrit le code source, ainsi que des commentaires s'ils sont écrits dans un format particulier.
Opencv		Une bibliothèque optimisée, de fonctions dédiées à la programmation pour la vision en temps réel. Cette librairie publiée et gratuite pour un usage scolaire ou commercial.
Tensorflow		TensorFlow est un outil open source d'apprentissage automatique et précisément pour l'apprentissage profonds développé par Google. En 2015, Google annonce la mise sous licence open source de Tensorflow afin de permettre aux experts dans ce domaine d'apporter leurs contributions à ce projet en vue d'accélérer son développement.
Spy		Spectral Python (SPy) est un module Python pur pour le traitement des données d'images hyperspectrales. Il a des fonctions pour lire, afficher, manipuler et classer des images hyperspectrales. Il peut être utilisé de

		manière interactive à partir de l'invite de commande Python ou via les scripts Python. SPy est un logiciel open source distribué sous licence GNU General Public License.
Gdal		**GDAL** (Geospatial Data Abstraction Library) est une bibliothèque libre permettant de lire et de traiter un très grand nombre de format d'images géographique, et hyperspectrales (notamment GeoTIFF et ECW) depuis des langages de programmation tels que C, C++, C sharp / .Net, Java, Ruby, VB6, Perl, Python, ou encore le langage statistique R.

V.2. IHM de l'application

Comme c'est indiqué auparavant, notre Framework utilise les méthodes de Deep Learning pour la classification des images.

L'interface graphique est structurée sous forme des pages respectant une certaine arborescence, globalement on peut diviser les interfaces graphiques du Framework on trois partie :

1. La partie Accueil : Classifier une image en chargeant un modèle déjà appris.
2. La partie Apprentissage : Configurer l'architecture du réseau des neurones profond, lancer l'apprentissage et enregistrer le modèle après l'apprentissage.
3. La partie Test : Evaluer le modèle en utilisant une base de données de test

Pour cela on 'a choisi de poser tous les pages sur un calque qui va faciliter l'interaction homme-machine :

Figure 31.claque générale du GUI de notre Framework

Tableau 5. Structure générale du GUI du Framework

Le nom	Remarque
1	Actionneur : permet d'accéder à la partie accueil
2	Actionneur : permet d'accéder à la partie d'apprentissage
3	Actionneur : permet d'accéder à la partie de test
	Zone d'affichage des pages de différentes parties

V.2.1. Partie d'apprentissage

Cette partie de l'interface graphique est consacré à la phase d'apprentissage (Figure 32), elle est diviser en quatre onglets :

1 **Data** : permet de charger les données d'apprentissage, et visualiser ses informations (ex : les Classes, nombre des images …) (Figure 32).

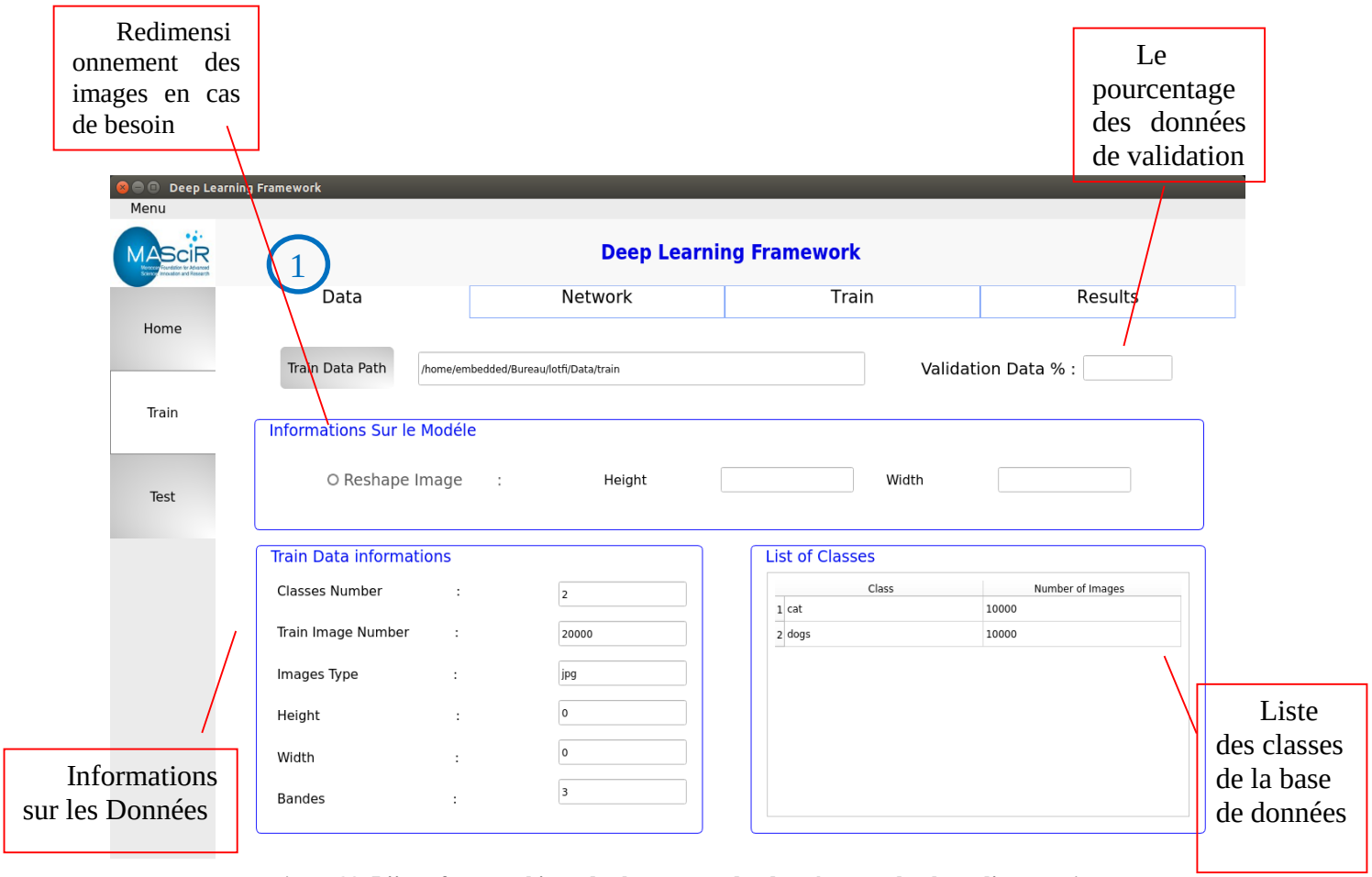


Figure 32. L'interface graphique de chargement des données pour la phase d'apprentissage

Ici la base de données choisit est la base de données « cats Vs dogs ». Cette base de données est largement utilisée par la communié de Deep Learning, parce qu'elle présente beaucoup d'ambiguïté. Elle est aussi utilisée dans plusieurs compétitions de Deep Learning, pour évaluer la performance des modèles.

2 **Network** : permet de configurer l'empilement des couches du réseau profond ainsi leurs que paramètres au niveau spatial (Figure 33) et spectrale (Figure 38).

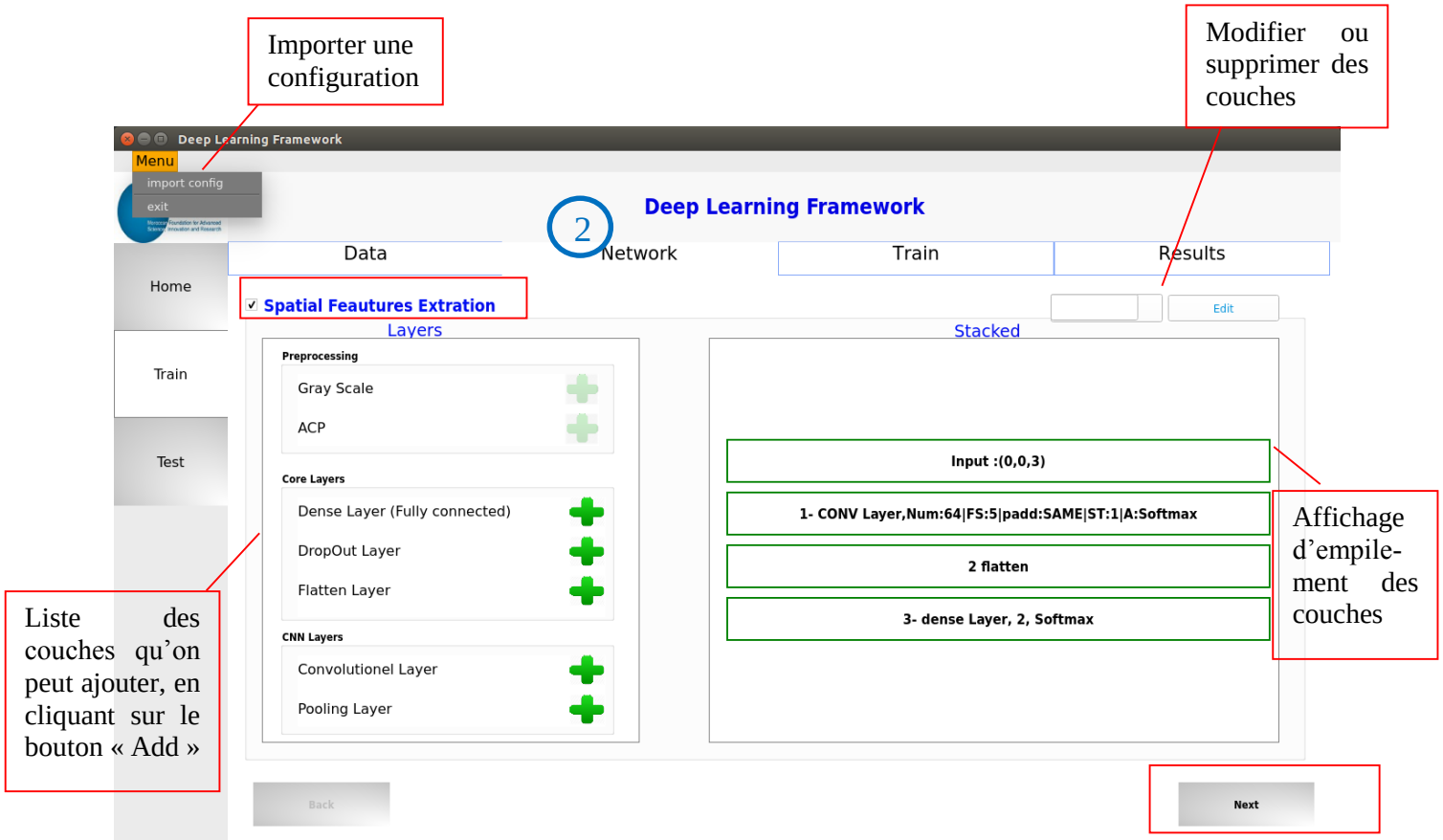


Figure 33. Configuration des pour l'extraction des caractéristiques spatiales (GUI)

Lorsqu'on clique sur le bouton « add » une fenêtre de dialogue s'affiche qui contient des zones de texte pour configurer les différents paramètres de la couche correspondante (Figure 34, Figure 35, Figure 36, Figure 37).

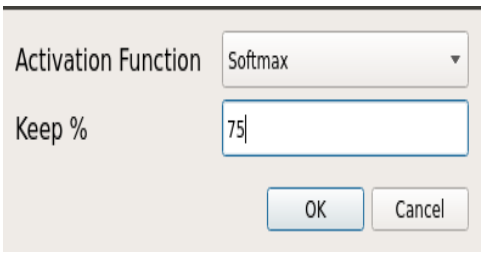


Figure 34.Configuration de La couche Dropout

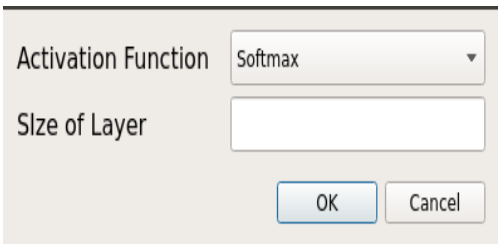


Figure 35.Configuration de la couche Totalement connectée

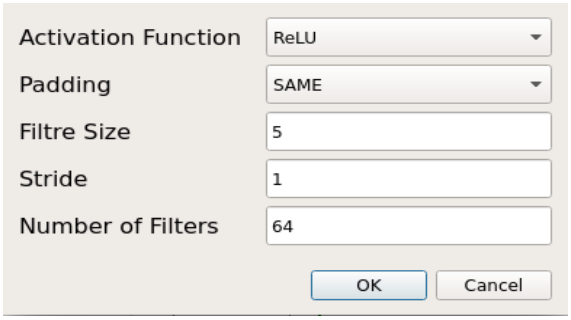


Figure 36.Configuration de couche à convolution

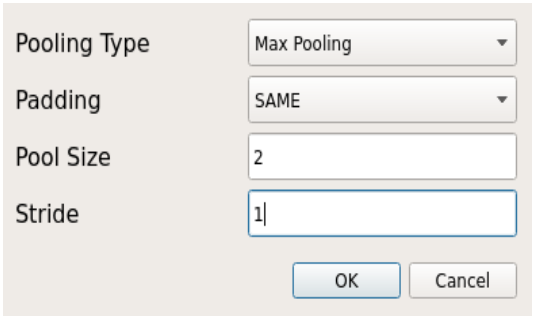


Figure 37.Configuration de la couche de Pooling

Lorsqu'on clique sur le bouton « next » on passe à la configuration des couches pour l'extraction des caractéristiques spectrales, cette zone est dédiée aux données hyperspectrales. Pour les données non-hyperspectrales cette zone est désactivée (Figure 38).

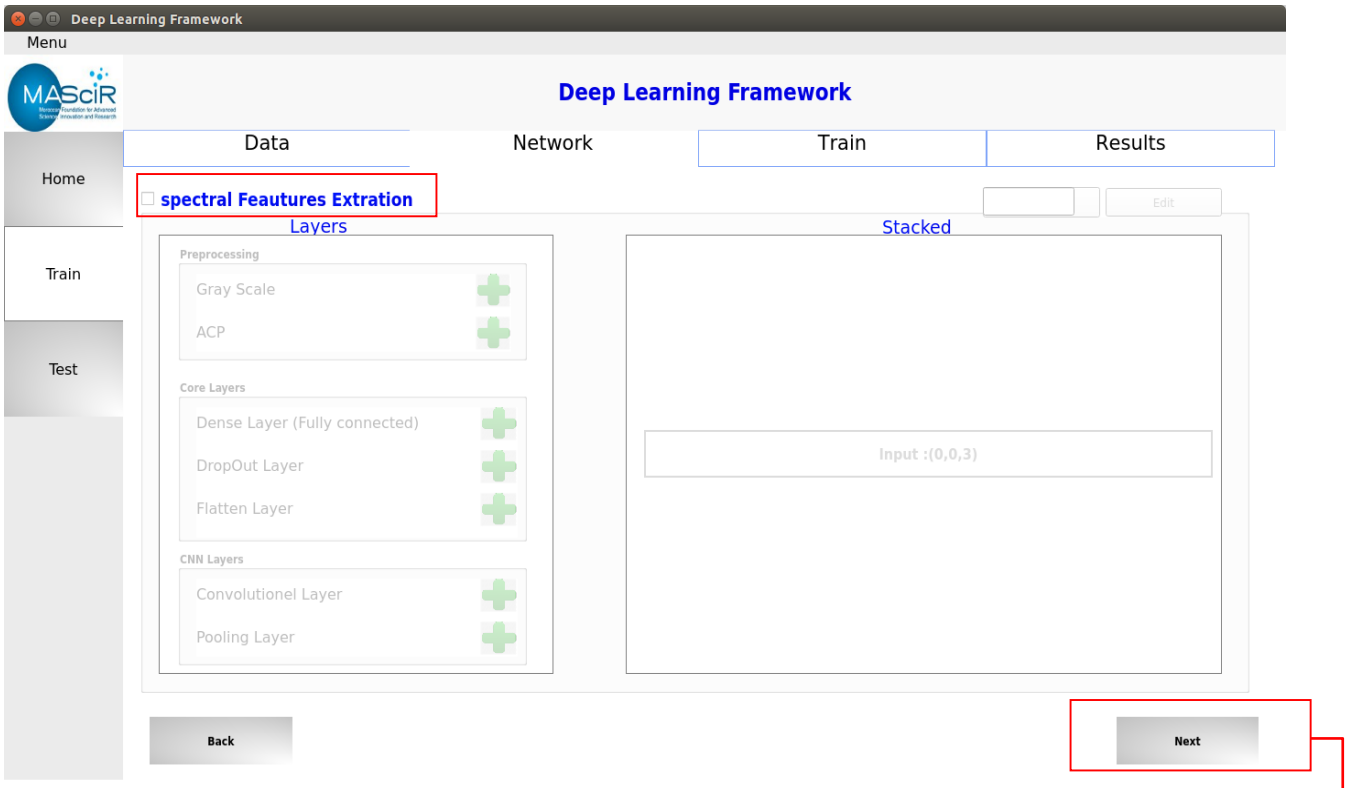


Figure 38.Empilement des couches pour l'extraction des caractéristiques spectrales (GUI)

Le bouton « next »nous permet d'aller à la dernière étape de configuration de réseau. Cette étape permet concaténer les caractéristiques spatiales et spectrales, ainsi que la finalisation de notre réseau (Figure 39):

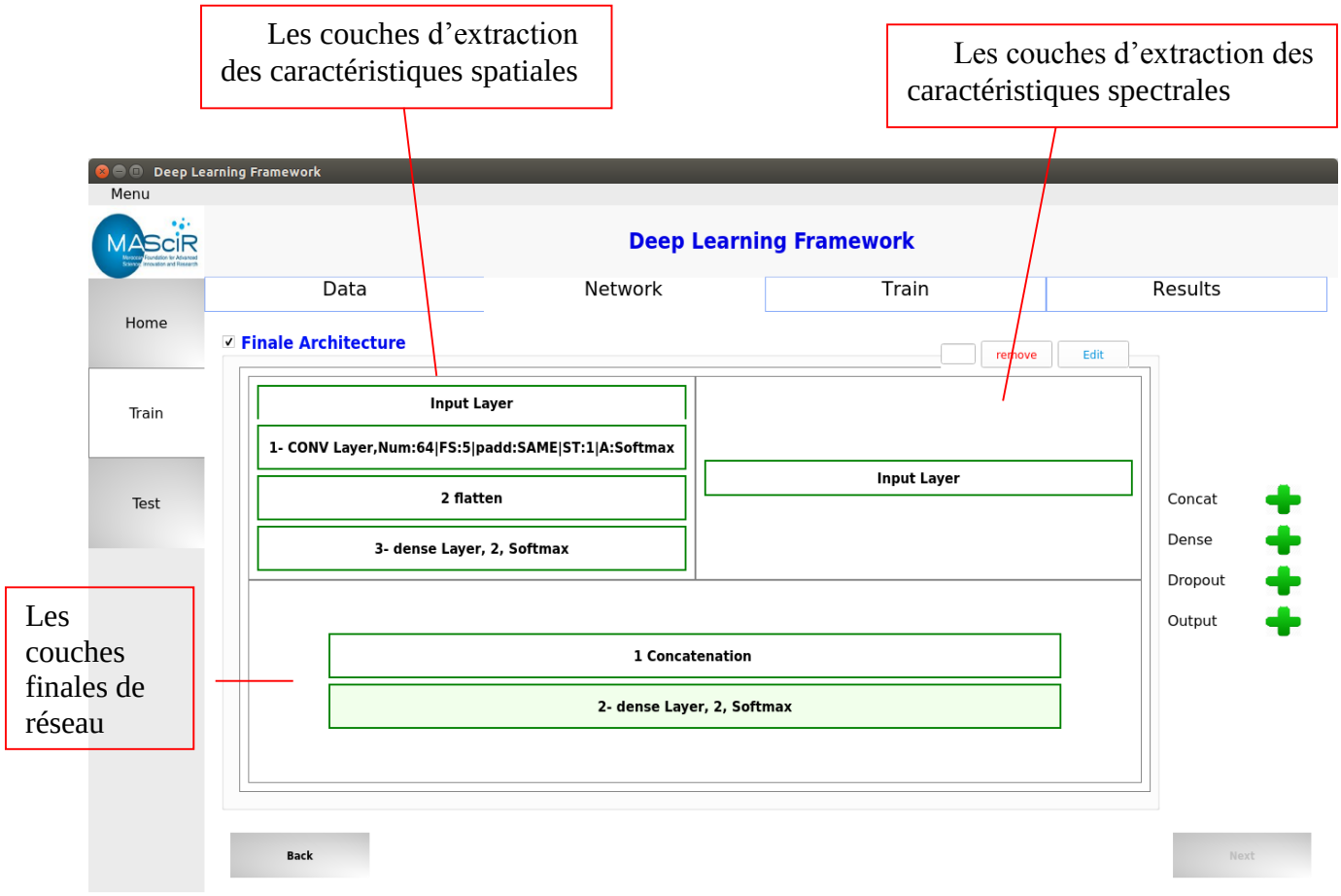


Figure 39.Concaténation des caractéristiques et classification (GUI)

3 **Train** : permet de configurer les différents paramètres d'apprentissage (Figure 40) :

1. Optimiser : l'algorithme utilisé pour minimiser l'erreur du modèle (Loss), nous proposons plusieurs algorithmes :
    - SGD (stockastic gradient descent).
    - L'algorithme d'ADAM
    - Adaptif Gradient (AdapGrad)
    - RMS propagation (Rmsprop)
2. La fonction Loss : la fonction permet de calculer l'erreur du modèle, nous proposons un ensemble de fonction :
    - Erreur quadratique moyen (Mean squar error).
    - Erreur quadratique absolut (absolut squar error).
    - Etc.
3. Batch : le nombre d'images propageant le réseau pour chaque itération.
4. le taux d'apprentissage (Learning rate).
5. La métrique pour calculer la précision du modèle (Accurcay Metrics).
6. le critère d'arrêts

Après la configuration de ces paramètres on peut exporter la configuration et de l'enregistre en format XML pour la réutiliser, en cliquant sur le bouton « export config » [Annexe3]. On peut aussi lancer l'apprentissage, en cliquant sur le button « start train », qui va nous diriger vers l'onglet « results » permet suivre la progression d'apprentissage.

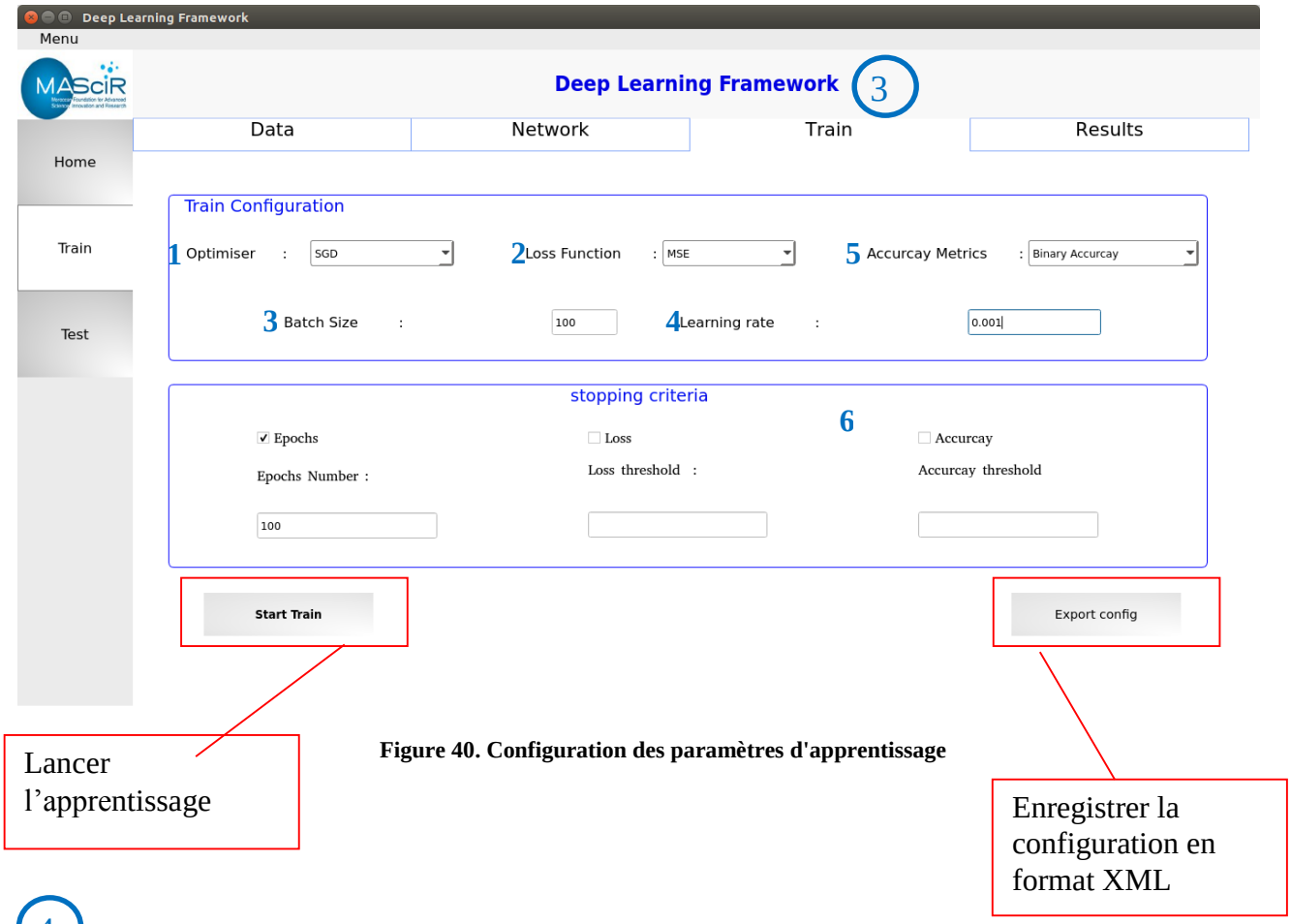


Figure 40. Configuration des paramètres d'apprentissage

4 **Results** : permet de suivre l'avancement, et les résultats durant l'apprentissage, pour les données d'apprentissage ainsi que de validation, puis stocker le modèle après, via le button « save model » (Figure 41).

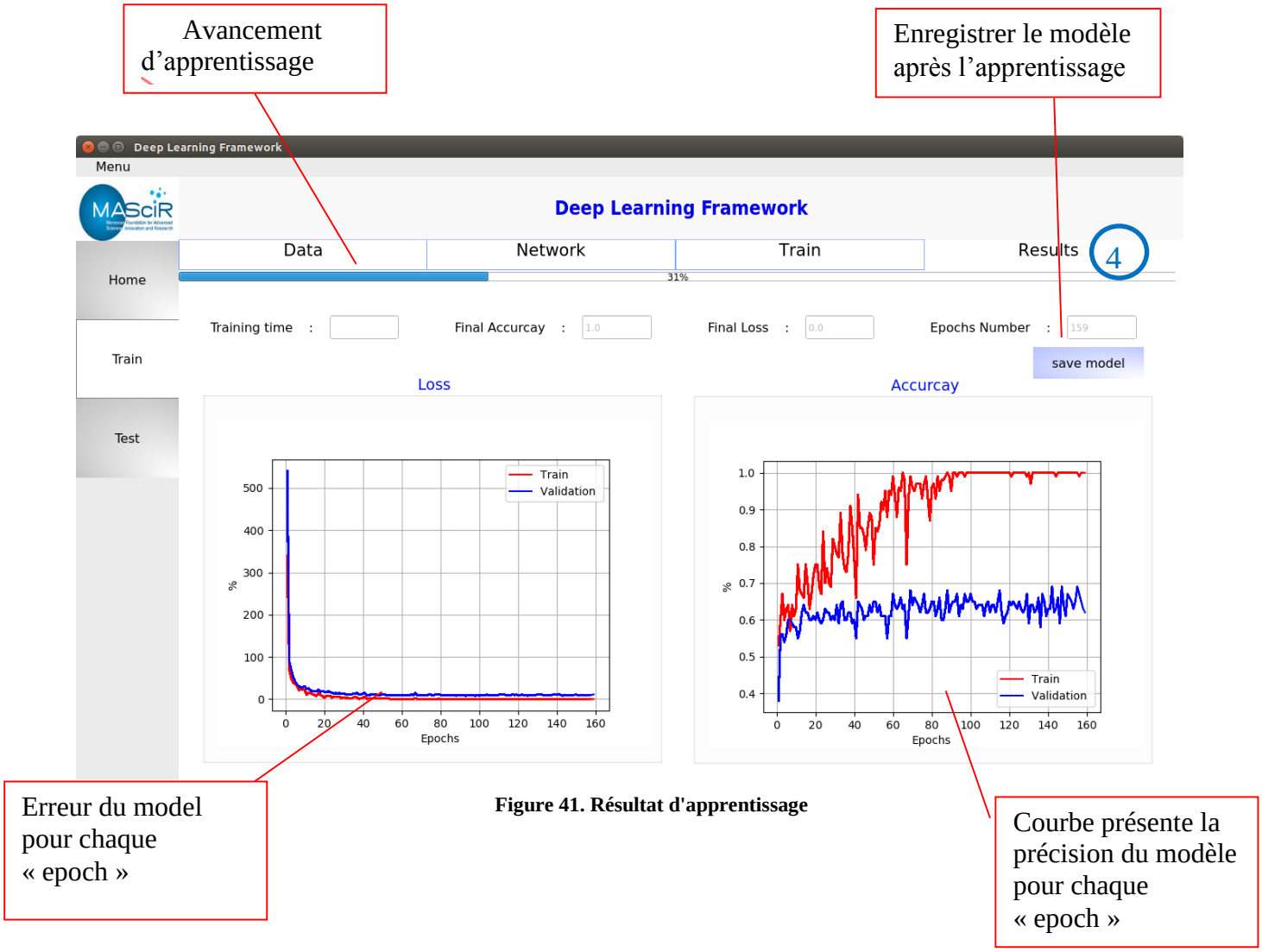


Figure 41. Résultat d'apprentissage

V.2.2. Phase de test

Cette partie de l'interface permet d'évaluer le modèle généré après la phase d'apprentissage, elle est divisée en deux onglets :

Data : permet de charger les données d'apprentissage, et visualiser ses informations (ex : les Classes, nombre des images …) (Figure 42).

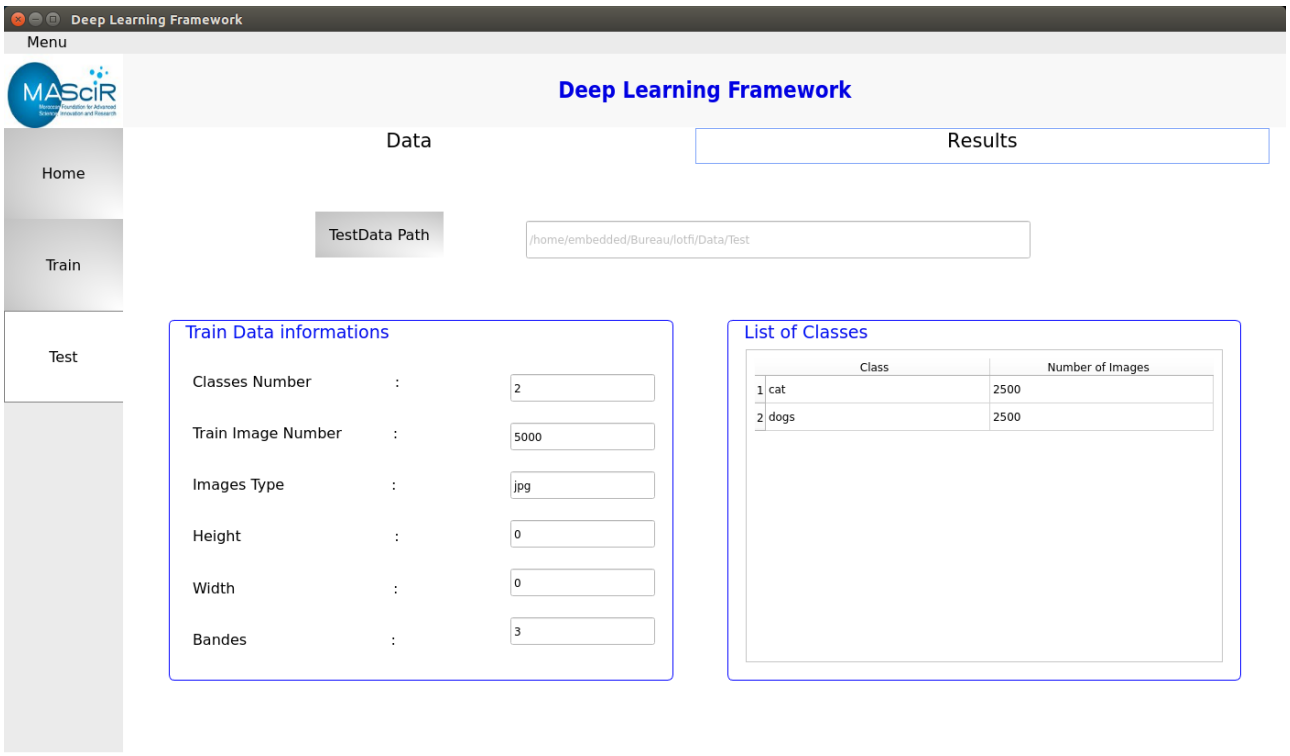


Figure 42.Chargement des données de test

Results : permet de charger un modèle déjà stocké après la phase d'apprentissage, puis lancer le test et visualiser ses résultats dans un graphe (Figure 43).

Figure 43.Resultat de test de modèl

V.2.3. La partie accueil

Cette partie de l'interface graphique permet de choisir un modèle généré après la phase d'apprentissage (Figure 44), puis choisir une image et la classifier en utilisant le modèle choisit. Cette partie de l'interface est divisée en deux onglets :

Model : permet de charger le modèle, et visualiser ces informations (architecture de réseaux, les classes, la base de données utilisée pour apprendre le modèle, la précision finale …) (Figure 42).

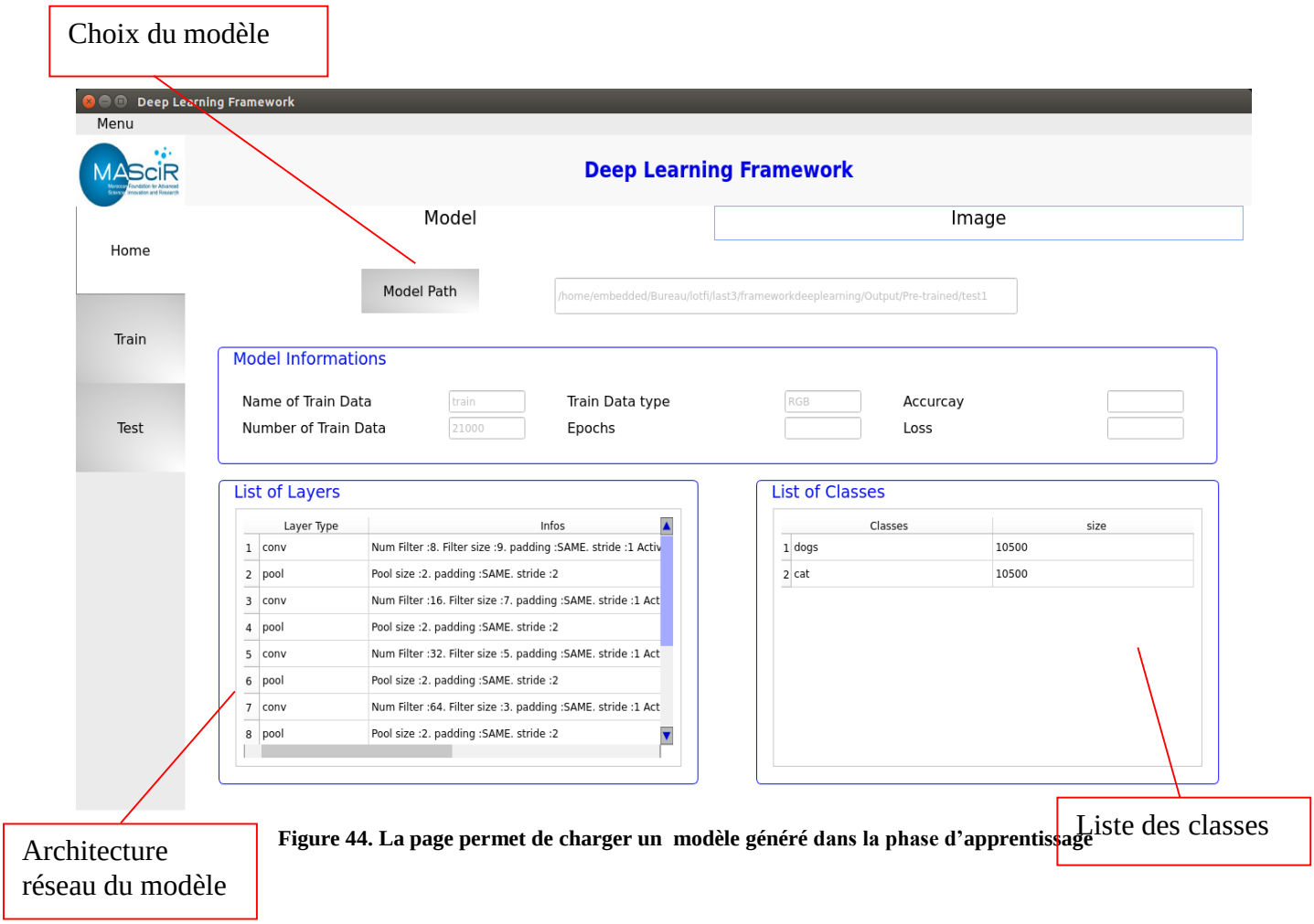


Figure 44. La page permet de charger un modèle généré dans la phase d'apprentissage

Image : permet de charger une image, la visualiser, puis lancer la classification à base du modèle choisit dans l'étape précédente, et afficher la probabilité d'appartenance à chaque classe (Figure 45).

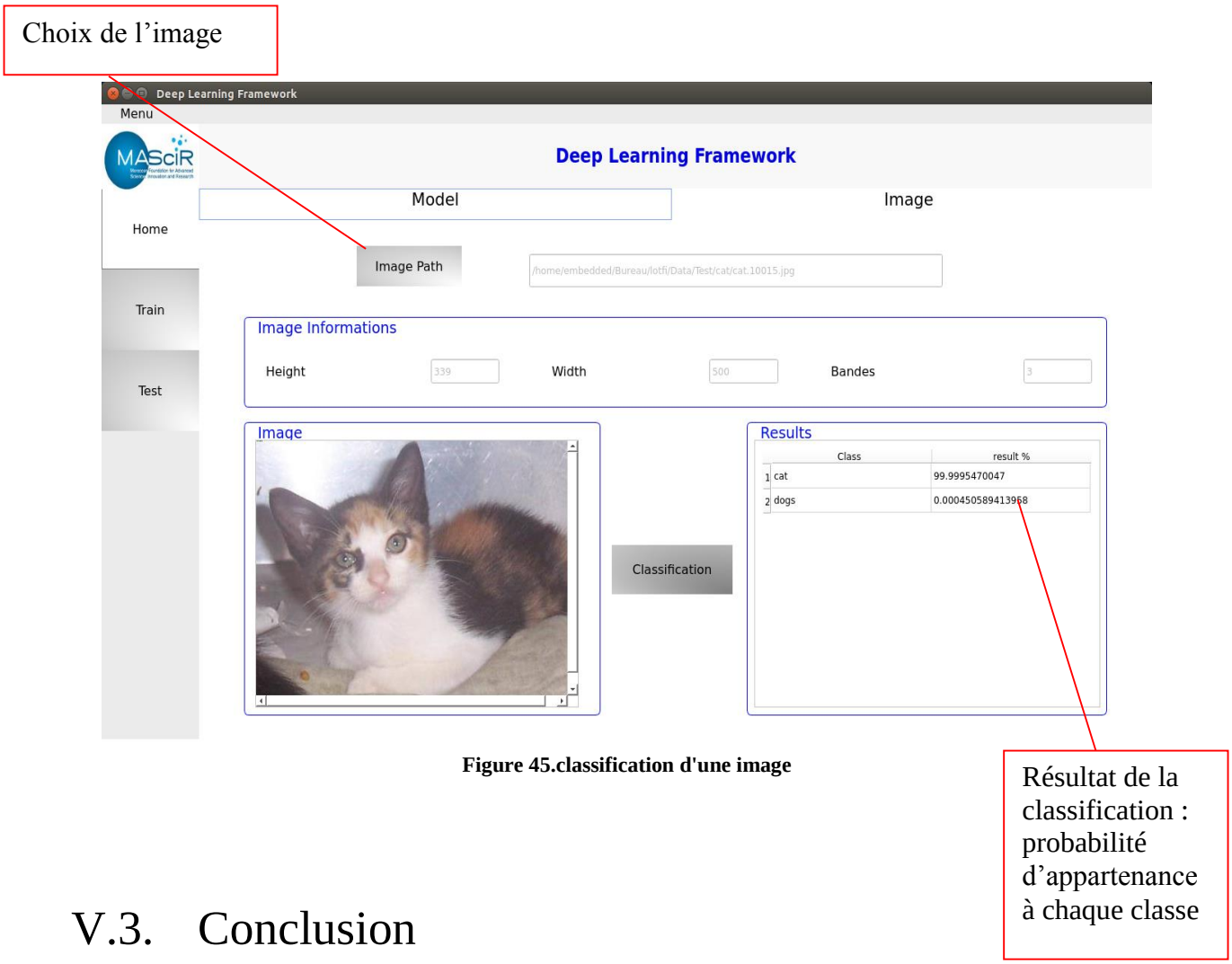


Figure 45.classification d'une image

V.3. Conclusion

Dans ce chapitre, nous avons présenté les plates-formes matérielles et logicielles sur lesquelles nous avons développé notre projet, ainsi que les technologies employées. Nous avons, par la suite, présenté les interfaces les plus significatives de notre application.

Conclusion et perspectives

Dans le cadre de Mon stage au sein de la fondation MASciR, nous étions amenés à développer un Framework pour la classification et la segmentation des images et particulièrement les images hyperspectrales à base des méthodes de Deep Learning.

Dans un premier temps nous avons fait un état de l'art sur les méthodes de Deep Learning utilisé pour la classification des images, nous avons présenté le concept de « Deep Learning », son origine d'apparition, le principe de fonctionnement ainsi que les types les plus utilisés dans la littérature : les réseaux de neurones à convolution (Convolutional Neural Networks, CNN) et les réseaux de croyance profonde (Deep belief network, DBN) ainsi que les Autoencoders. Par la suite, nous avons introduit la notion d'imagerie hyperspectrale, ses domaines d'application, ainsi que sa classification à base de Deep Learning. À la base des connaissances acquises durant la phase précédente on a pu mettre en place un cahier des charges regroupant un ensemble d'exigences et besoins que le produit final doit satisfaire. Le cahier des charges nous a permis de faire une conception technique du Framework, que nous a facilité la tâche de développement.

En perspective, la prochaine étape du stage consiste à finaliser cette première version du Framework, puis intégrer les modèles générés dans des systèmes embarquées pour classifier des objets, et finalement ajouter la fonctionnalité de classification des images hyperspectrales dans notre Framework.

Références

[1] Kruse, F. (2000). « Introduction to hyperspectral data analysis. » In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Honolulu, HI, USA.

[2] M Ellis, HH Davis, JA Zamudio « Exploring for onshore oil seeps with hyperspectral imaging » Published in Oil and Gas Journal, 10 Sept 2001, volume 99.37, p. 49 -58.

[3] Smith, R.B. (July 14, 2006) « *Introduction to hyperspectral imaging with TMIPS* », MicroImages Tutorial Web site

[4] W. S. McCulloch and W. Pitts. « A logical calculus of the idea immanent in nervous activity. » *Bulletin of Mathematical Biophysics*, 5 :115–133, 1943.

[5] F. Rosenblatt, « The perceptron: à probabilistic model for information storage and organization in the brain », *Psychol. Rev.*, vol. 65, no 6, p. 386-408, nov. 1958.

[6] Li Deng and Dong Yu (2014), « Deep Learning: Methods and Applications », Foundations and Trends® in Signal Processing: Vol. 7: No. 3–4, pp 197-387.

[7] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, et L. D. Jackel, « Backpropagation Applied to Handwritten Zip Code Recognition », *Neural Comput.*, vol. 1, no 4, p. 541-551, déc. 1989.

[8] D.H. Hubel, T.N. Wiesel « Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. » *The Journal of physiology*, 160(1):106, 1962

[9] K. Fukushima : Neocognitron « A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. » *Biological cybernetics*, 36(4):193–202, 1980

[10] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard et L.D. Jackel « Handwritten digit recognition with a backpropagation network. » *Advances in Neural Information Processing Systems*, pages 396–404, 1990.

[11] Geoffrey Hinton, « Deep Belief Nets », Department of Computer Science University of Toronto, Canadian Institute for Advanced Research.

[12] Ferwerda, J.G. (2005), « *Charting the quality of forage: measuring and mapping the variation of chemical components in foliage with hyperspectral remote sensing*, » Wageningen University, ITC Dissertation 126, 166p. ISBN 90-8504-209-7

[13] Lacar, F.M., et al., « *Use of hyperspectral imagery for mapping grape varieties in the Barossa Valley, South Australia*, Geoscience and remote sensing symposium (IGARSS'01) » – IEEE 2001 International, vol.6 2875-2877p.

[14] Camille Truche, « Caractérisation et quantification des minéraux argileux dans les sols expansifs par spectroscopie infrarouge aux échelles du laboratoire et du terrain. » Planète et Univers [physics]. Université Paul Sabatier - Toulouse III, 2010. Français.

[15] Saeid Homayouni, « *Caractérisation des Scènes Urbaines par Analyse des Images Hyperspectrales*. » domain other. Télécom ParisTech, 2005. English. <pastel-00002521>

[16] Matthieu PETREMAND, « détection des galaxies à faible brillance de surface, segmentation hyperspectrale dans le cadre de l'observatoire virtuel, » Université Louis Pasteur - Strasbourg I.

[17] Hassan Mortada, « Décomposition conjointe par approximation parcimonieuse en imagerie multispectrale astronomique. » École doctorale Mathématiques, sciences de l'information et de l'ingénieur (Strasbourg)

[18] *AM Shahidi; et al. (2013).* « Regional variation in human retinal vessel oxygen saturation » *Elsevier – Experimental Eye Research. 113: 143–147. doi:10.1016/j.exer.2013.06.001. PMID 23791637.*

[19] Farley, V., Chamberland, M., Lagueux, P., et al., « Chemical agent detection and identification with a hyperspectral imaging infrared sensor. » Proceedings of SPIE Vol. 6661, 66610L (2007)

[20] *Higgins, Kevin. «* Five New Technologies for Inspection. » *Food Processing. Retrieved 6 September 2013.*

[21] Hughes, G. (1968). « On the mean accuracy of statistical pattern recognizers. » *IEEE Transaction Information Theory*, 14(1):55–63

[22] Hussein Almuallim and Thomas G. ietterich, « Learning with Many Irrelevant Features. » Oregon State University, 1991.

[23] Oded Maimon · Lior Rokach « Data Mining and Knowledge Discovery Handbook. » Springer New York Dordrecht Heidelberg London.

[24] Jon Atli Benediktsson, « Classification and Feature Extraction for Remote Sensing Images from Urban Areas Based on Morphological Transformations. » IEEE transactions on geoscience and remote sensing, vol. 41, no. 9, September 2003.

[25] Y. Fang *et al.*, « Dimensionality reduction of hyperspectral images based on robust spatial information using locally linear embedding. » *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1712–1716, Oct. 2014

[26] Yushi Chen, Xing Zhao, Xiuping Jia, « Spectral–Spatial Classification of Hyperspectral Data Based on Deep Belief Network » *IEEE Geosci. Remote Sens.*

[27] Zhouhan Lin, Yushi Chen, Xing Zhao, Gang Wang « Spectral-Spatial Classification of Hyperspectral Image Using Autoencoders. » Nanyang Technological University, Harbin Institute of Technology

[28] Mauro Dalla Mura, « Morphological Attribute Profiles for the Analysis of Very High Resolution Images. » IEEE transactions on geoscience and remote sensing.

[29] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, « Deep learning-based classification of hyperspectral data. » *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[30] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, « Vehicle detection in satellite images by hybrid deep convolutional neural networks » *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, Oct. 2014.

[31] Yushi Chen, Xing Zhao, Xiuping Jia, « Spectral–Spatial Classification of Hyperspectral Data Based on Deep Belief Network » *IEEE Geosci. Remote Sens.*

[32] G. E. Hinton, R. R. Salakhutdinov, « Reducing the Dimensionality of Data with Neural Networks » 28 JULY 2006 VOL 313,

[33] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). « Dropout: A simple way to prevent neural networks from overfitting. » *The Journal of Machine Learning Research*, *15*(1), 1929-1958.

[34] Duchi, J., Hazan, E., & Singer, Y. (2011). « Adaptive subgradient methods for online learning and stochastic optimization. » *Journal of Machine Learning Research*, *12*(Jul), 2121-2159.

[35] Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., ... & Ng, A. Y. (2012). « Large scale distributed deep networks. » In *Advances in neural information processing systems* (pp. 1223-1231).

[36] Kingma, D., & Ba, J. (2014). « Adam: A method for stochastic optimization. » *arXiv preprint arXiv:1412.6980*.

Annexe

Annexe1 : solutions existantes pour le Deep Learning (Benchmark)

Tableau 6. Résumé de l'étude benchmarking sur les outils de Deep Learning

API / Propriété	TensorFlow	Theano	Torch	Caffe	MXNET	DeepLearning 4J	CNTK
plateforme	Linux, Mac OS X, Windows	multi-Platform	Linux, Mac OS X, Windows, Android iOS	Toutes les plateformes	Linux, ,Mac OS X, Windows, Docker, AWS, Android, IOS.	JVM	Windows, Linux
Core	C++	Python	Lua, C	C++	Small C++	C, C++	C++
Interface de programmation	-Python (stable)	Python	Lua, LuaJIT, C.	Python, Matlab	C++, Python, R, Scala, Julia, Matlab , Javascript, Raspberry Pi.	Java, Scala	Python, C++, BrainScript, . NET

	-C++, Java, Go (non stable)						
Exigences techniques	○ CUDA (pour le GPU)	○ SciPy ○ NumPy ○ nose ○ nose parameterized ○ Sphinx ○ pygments, ○ graphiz. ○ libgpuarray	○ LuaJIT ○ LuaRocks ○ OpenBlas	○ CUDA pour GPU ○ BLAS ○ Boost ○ protobuf, glog, gflags, hdf5.	○ C++ compiler ○ G++ ○ Clang ○ BLAS ○ Graphviz ○ Jupyter Notebook ○ CUDA (pour GPU) ○ OpenCV	○ Java ○ Apache Maven ○ IntelliJ IDEA ou Eclipse	-
GPU	✓	✓	✓	✓	✓	✓	✓
Multi-thread CPU	✓	✓	✓	✓	✓	✓	✓
Documentation	✓	✓	✓	✓	✓	✓	✓
Modèle pré entrainé ?	✓	✓	✓	✓	✓	✓	✓

Les modèles supportés	CNN RNN LSTM RBM, DBN	CNN RNN RBM, DBN	CNN, model, RNN DBN, RBM	CNN RNN LSTM RB M, DBN	Autoencoders LSTM. RNN CNN	CNN Autoencoders RBM LSTM RNN DBN	CNN RNN LSTM, Reinforceme nt learning

Annexe2 : solutions existantes de traitement d'images hyperspectrales (Benchmark) :

Tableau 7. Résume de l'étude benchmarking des outils de traitement des images hyperspectrales

Propriétés / APIs	Publier par	Platform	Code	Exigences techniques	Caractéristiques			
					open source	Visualisation De données	Traitement De données	Format D'image supporté
ENMAP -BOX	Université Humboldt -Berlin	indépendante	IDL	Environnement d'exécution IDL 8.4	✓	✓	○ Classification (SVM, Random Forests…) ○ régression (PLSR, SVR.) ○ analyse spectral (ASI, iterativeSMA …)	ENVI (.hdr, bsq, bil, bip)
HyperMix Tool	Research Group "Hyperspectral Computing", Université de Extremadura	Linux, Windows.	C++.	–	✓	✓	○ Estimation des composantes spectrales (HYSIME, VD). ○ Prétraitement d'image (PCA, SPP). ○ Extraction des composantes spectrales (N-FINDER, VCA, OSP…)	ENVI (.hdr, bsq, bil, bip)
Orfeo ToolBox (OTB)	CNES, CS Systèmes d'Information, OSGeo foundation.	Windows, Linux et Mac	C++.	–	✓	✓	○ Segmentation d'image. ○ Classification des images. ○ Prétraitement des données. ○ Extraction des caractéristiques. Traitement hyperspectral. ○ Traitement SAR (Synthetic aperture radar).	tiff, hrd, img, data providers specifics …)

Scyllarus	Scyllarus beyond Vision	indépendant	C++	–	×	Scyven (Scyllarus Visualization Environment)	- Traitement d'image d'image hyperspectrale. - clustreing / segmentation de matériaux à base de Spectre. -PCA…	HDR (ENVI) flat/raw file, BIL, BSQ, BIP, HSZ (Hyperspectral Zip
Spectral Python (Spy)	Thomas Boggs	indépendant	python	○ NumPy ○ Pillow or Python Imaging Library (PIL) ○ wxPython, matplotlib ○ IPython ○ PyOpenGL	✓	✓	○ Affichage / Visualisation ○ Réduction dimensionnelle Détection de cible Classification ○ Transformations spectrales	Lan , GIS, mat
Spectronon	Resonon	independant	C++.	–	✓	✓	○ Classification hyperspectral (SAM, Logistic Regression, etc. ○ Unmixing utilities ○ outils de visualisation. ○ support d'autres plugins	.bil, .bip, .bsq
HyperSpy	HyperSpy development team	Multiplatform	python	○ Python 3.0, ○ pour Python 2.7 install ○ HyperSpy 0.8.3. ○ NumPy, SciPy, matplotlib and scikit-learn.	✓	✓	○ l'analyse interactive de données ○ machine learning (PCA, ICA, RPCA). ○ Traitement d'image. ○ Traitement de signal.	HDF5
Opticks	Ball Aerospace pour the US Air Force	Windows, Linux, Solaris	C++, Python	–	✓	✓	○ Supporte des jeux de données de plus de quatre giga-octets. ○ Plusieurs algorithmes pour traitement des données. ○ prétraitement des images. ○ Calcule les fonctionnalités SIG	NITF, GeoTIFF, ENVI, ASPAM/PAR, CGM, DTED, Generic RAW, ESRI Shapefile, HDF5,

								AVI, MPEG, JPEG, GIF, PNG, BMP.
Gerbil	Pattern Recognition Lab, University of Erlangen-Nuremberg.	Windows, Linux, OS X	C++,	○ CMake ○ boost C++, modules thread, ○ intel *TBB* ○ Qt ○ OpenCV ○ GDALgeospatial data abstraction library	✓	✓	○ Réduction de donnée (SOM...) ○ prétraitement. ○ Extraction des caractéristiques. ○ Segmentation (Graph cuts, Power Watersheds). ○ Clustering (fast-adaptive mean shift (FAMS))	-ENVI .hdr -ESRI .hdr FITS (.fits) -.tif -ERDAS (.ecw, .img, .lan, .gis, …) -JPEG200 -PNG -Netpbm
GDAL	Open source Geospectral Foundation	Multi-platform	C, C++, Python, java, R		✓	×	○ Manipulation des images GEOspatial	Tous les formats des images Geospatial

Annexe3 : Exemple de fichier XML de configuration

```
−<model name="My Model">
  −<resize value="True">
      <xsize>100</xsize>
      <ysize>100</ysize>
    </resize>
  −<Layers>
    −<Spatial>
      −<convolotionnel>
          <num_filter>8</num_filter>
          <filter_size>9</filter_size>
          <padding>SAME</padding>
          <stride>1</stride>
          <activation_function>relu</activation_function>
        </convolotionnel>
      −<Pooling>
          <type>max</type>
          <pool_size>2</pool_size>
          <padding>SAME</padding>
          <stride>2</stride>
        </Pooling>
      −<convolotionnel>
          <num_filter>16</num_filter>
          <filter_size>7</filter_size>
          <padding>SAME</padding>
          <stride>1</stride>
          <activation_function>relu</activation_function>
        </convolotionnel>
      −<Pooling>
          <type>max</type>
          <pool_size>2</pool_size>
          <padding>SAME</padding>
          <stride>2</stride>
        </Pooling>
      −<convolotionnel>
          <num_filter>32</num_filter>
          <filter_size>5</filter_size>
          <padding>SAME</padding>
          <stride>1</stride>
          <activation_function>relu</activation_function>
        </convolotionnel>
```

```
      <flatten/>
    −<Dense>
        <size>125</size>
        <activation_function>relu</activation_function>
      </Dense>
    −<Dropout>
        <keep>50</keep>
        <activation_function>Softmax</activation_function>
      </Dropout>
    −<Dense>
        <size>2</size>
        <activation_function>Softmax</activation_function>
      </Dense>
    </Spatial>
  </Layers>
  <Loss_Function>MSE</Loss_Function>
  <accurcay_metric>Binary Accurcay</accurcay_metric>
−<optimiser>
    <type>SGD</type>
    <Learning_Rate>0.01</Learning_Rate>
  </optimiser>
  <batch_size>100</batch_size>
−<Stop_Criteria>
    <type>epochs</type>
    <thershold>1000</thershold>
  </Stop_Criteria>
−<Train>
    <name>train</name>
  −<classes>
      <cat>10000</cat>
      <dogs>10000</dogs>
    </classes>
    <size>20000</size>
    <type>RGB</type>
  </Train>
</model>
```