



**MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDELHAMID IBN BADIS MOSTAGANEM**

**Faculté des Sciences Exactes & de l'Informatique
Département de Mathématiques et d'Informatique
Filière Informatique**

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : Systèmes d'Information Géographiques

**Compression des Photos Aériennes par les
Réseaux de Neurones**

Présenté par :

**Mlle Moghtit Leila
Mlle Beklaouz Fatima**

Encadré par:

Mlle Bentaouza Chahinez Mérièm

Année Universitaire 2012/ 2013

Résumé

Ce projet est la réalisation d'une application de compression des **photos aériennes** par un algorithme issu de l'intelligence artificielle qui est le réseau de neurones. La **compression** consiste à minimiser la taille des photos aériennes connus par leur taille volumineuse dans un but d'archivage ou de transmission afin qu'elles prennent moins d'espace et moins de temps. Ainsi **le réseau de neurones** artificiel inspiré du fonctionnement des neurones biologiques sera appliqué pour la compression des photos aériennes à travers un apprentissage supervisé.

Mot clés : Apprentissage supervisé, Compression, Photos Aérienne, Réseau de neurones.

Abstract

Compression consists in minimizing the size of aerial photos known by their voluminous size in a purpose of archiving or transmission so that they take less space and less time. So the artificial neural network inspired by functioning of biological neurons will be applied for compression of aerial photos through learning.

Keywords: Supervised Learning (Apprenticeship), Compression, Aerial Photos, Neural network.

Remerciement

Tous d'abord nous remercions DIEU de nous avoir donné la force et le courage de terminer nos études.

Nous exprimons notre reconnaissance à Melle Bentaouza .C .M , qui a bien voulu accepter de nous prendre en charge pour réaliser ce modeste travail dont le mérite lui revient grâce à son aide, ses conseils précieux et son gratitude et leur gentillesse, et surtout pour la confiance qu'elle nous a témoigné et qu'elle nous a accordé.

Nous remercions vivement les membres de notre jury pour avoir accepté d'évaluer ce projet.

Nous adressons nos remerciements à l'ensemble des enseignants du département d'informatique chacun en son nom.

À tous les membres de la promotion SIG 2012-2013 pour leur soutien, leurs sourires et la chaleur avec laquelle ils nous 'ont accueilli au sein de notre période d'étude, et on les souhaite bonne chance.

Au terme de ce travail nous voudrions remercier tous ceux qui nous ont aidés de le réaliser.

Nous voudrions bien qu'il soit pour vous le témoignage de notre gratitude et de notre profond respect.

Qu'ALLAH nous bénisse tous.

Leila & Fatima



Dédicace

*J*e dédie ce modeste travail à

*C*elle qui m'a transmis la vie,

*L'*amour, et le courage, à toi chère

Maman toutes mes joies.

Leila



Dédicace

Avec l'aide du tout Puissant, j'ai pu réaliser ce modeste travail que je dédie :

A mes chers parents sur qui j'ai pu compter et me ressourcer d'affection et de bénédictions durant toute ma vie ; je ne les remercierai jamais assez pour ça, qu'ils trouvent dans ces modestes mots mon grand estime et ma profonde reconnaissance et que dieu les bénisses ;

A mon très cher frère Mohammed qui m'a soutenue durant toute ma vie et qui m'aide beaucoup pour être meilleur dans mes études je lui remercie infiniment.

A ceux que j'aime tant et dont je ne peux oublier le soutien, mes chères sœurs Assia et ses enfants, Fatiha et ses enfants, Kenza, Djamila et mes adorables frères : Hamid et nasrdinne.

une mention spéciale à Mlle Moghtit Leila avec qui j'ai partagé ce travail.

A mon encadreur Mlle BENTAOUZA Meriem Chahinez qui n'a pas cessé de m'encourager sagement aux moments de détresse ;

A ma meilleur amie : Rahma qui je garderai les meilleurs souvenirs

A tous ceux qui n'ont pas trouvé leurs noms dans cette dédicace, sachez bien que vous avez une place très grande dans mon cœur.

A tous les professeurs et enseignants universitaires qui m'ont permis, par leurs efforts, d'atteindre cette réussite.

Je dédie ce mémoire

Fatima



Sommaire

Introduction Générale.....	11
Chapitre I : L'image numérique.....	12
I.1. Introduction	13
I.2. Image numérique.....	13
I.3. Caractéristiques des images Raster	13
I.4. Définition de l'image numérique	14
I.5. Formats des fichiers images	15
I.6. Acquisition d'images	15
I.7. Image Satellite	16
I.7.1. Image panchromatique.....	16
I.7.2. Images couleurs, multi spectrales et hyper spectrales	17
I.8. Photo aérienne	18
I.8.1. Les types de photographie.....	18
I.8.1.1 Les photographies verticales	18
I.8.1.3. Les photographies obliques	19
II.8.1.3. Les photographies panoramiques	19
I.8.2. Caractéristiques de photo aérienne	20
1.8.2.1. Echelles et restitutions.....	20
I.8.2.2. Émulsions et longueurs d'onde	20
I.8.2.3. Spectre visible	20
I.8.2.4. Spectre invisible.....	21
I.8.3. Caractéristiques des vols	21
I.8.4. Voir en trois dimensions.....	21
I.8.5. Ortho photographie numérique.....	22
I.8.6. Etablissement de l'ortho photo	22
I.9. Conclusion.....	23
Chapitre II : La Compression	24
II.1. Introduction	25
II.2. Définition de compression	25

II.3.	Historique du codage des images	25
II.4.	Intérêt	26
II.5.	Classification des algorithmes de compression.....	26
II.5.1.	Compression symétrique / asymétrique.....	26
II.5.2.	Compression physique / logique	27
II.5.3.	Compression statistique / numérique.....	27
II.5.4.	Compression sans / avec perte	28
II.5.4.1.	Compression sans perte (compactage).....	28
II.5.4.2.	Compression avec perte (irréversible ou non conservative).....	28
II.6.	Taux de compression.....	29
II.7.	Méthodes d'intelligence artificielle pour la compression	29
II.8.	Conclusion.....	30
Chapitre III : Les réseaux de neurones		31
III.1.	Introduction	32
III.2.	Neurone biologique.....	32
III.3.	Neurone formel	33
III.3.1.	Synoptique	33
III.3.2.	Fonctionnement.....	34
III.3.3.	Réseaux de neurones non bouclés.....	35
III.3.4.	Réseaux de neurones bouclés	36
III.4.	Propriétés d'un réseau de neurones	37
III.5.	Apprentissage.....	38
III.5.1.	Apprentissage supervisé	38
III.5.2.	Apprentissage non supervisé.....	38
III.6.	Modèles des réseaux de neurones.....	39
III.6.1.	Modèle de Hopfield	39
III.6.2.	Modèle Kohonen	39
III.6.3.	Les Perceptrons Multicouches (PMC)	40
III.6.4.	Le modèle Adaline.....	41
III.7.	Règle d'apprentissage du Perceptron	42
III.8.	Description et principe	42
III.9.	Classification par le MLP	42
III.9.1.	Architecture.....	42

III.9.2. Phase d'apprentissage	43
III.10. Conclusion	43
Chapitre IV : L'application et les résultats	44
IV.1. Introduction	45
IV.2. Environnement de développement	45
IV.2.1. Les ressources physiques	45
IV.2.2. Les ressources logicielles.....	45
IV.3. Description de l'application.....	46
IV. 4. Description du travail réalisé.....	48
IV.5. Phase des tests & résultats	51
IV.5.1. Prétraitement.....	51
IV.5.2 Base Image	51
IV.5.3 Résultats.....	52
IV.5.3.1. Test N°1	53
IV.5.3.2. Test N° 2	54
IV.5.3.3. Test N° 3	55
IV.5.4. Discussion.....	56
IV.6 Conclusion	57
Conclusion générale	58
Bibliographié.....	59

Liste des figures

Figure I.1. Détail d'une image binaire.....	14
Figure I .2. Détail d'une image en niveaux de gris.....	14
Figure I.3. Image de Toulouse acquise par le mode panchromatique de SPOT.....	16
Figure.I.4.Image couleurs simulée PLEIADES à la résolution de 70 cm par pixel.....	17
Figure I.5. Prise de vue photographie verticale.....	18
Figure I.6. Prise de vue photographie oblique	19
Figure I.7. Prise de vue photographie panoramique.....	19
Figure I.8. Spectre visible.....	21
Figure I.9. Recouvrement latéral et Recouvrement longitudinal.....	22
Figure I.10. L'obtention de l'ortho photo.....	23
Figure.II.1.Evaluation en temps de calcul et en volume de la compression symétrique..	27
Figure. II .2. Image initiale et image compressée	29
Figure. II .3.Schéma de compression d'image par les réseaux de neurones	30
Figure. III .1.Le neurone biologique.....	32
Figure III.2.Fonctionnement du neurone formel	33
Figure III.3.- Neurone Formel	34
Figure III.4.fonction Heaviside	35
Figure.III.5.fonction signe	34
Figure III.6.fonction sigmoïde	35
Figure III.7.un perceptron multicouche	36
Figure III.8.Un réseau de neurones bouclé	37
Figure III.9.Apprentissage supervisé.....	38
Figure III.10.Apprentissage non supervisé	39
Figure III.11.Le modèle de Kohonen	40
Figure III.12.Le modèle du perceptron	41
Figure III.13.La séparation linéaire entre la classe A et B	41
Figure III.14.Présentation schématique d'un ML	42
Figure IV.1 Schéma général de l'application	47
Figure IV.2 Fenêtre principale	48
Figure IV.3 Fenêtre des actions N° 1	49
Figure IV.4 Fenêtre des actions N° 2	50
Figure IV.5 Fenêtre des actions N° 3	50

Figure.IV.6. Dimension 180x180 pixels	51
Figure.IV.7 Dimension 90x90pixels	51
Figure IV.8. Dimension 45x45pixels	51
Figure IV .9. Décomposition de la photo aérienne (région Mostaganem)	52
Figure IV.10 Taux de compression dans les trois tests	56

Lite des tables

Table IV.1 Résultats optimaux.....	51
Table IV.2 Résultat obtenu pour l'image 180x180 pixels de test N°1.....	53
Table IV.3 Résultat obtenu pour l'image 90x90 pixels de test N°1.....	53
Table IV.4 Résultat obtenu pour l'image 45x45 pixels de test N°1.....	53
Table IV.5 Résultat obtenu pour l'image 180x180 pixels de test N°2	54
Table IV.6 Résultat obtenu pour l'image 90x90 pixels de test N°2	54
Table IV.7 Résultat obtenu pour l'image 45x45 pixels de test N°2	54
Table IV.8 Résultat obtenu pour l'image 180x180 pixels de test N°3	55
Table IV.9 Résultat obtenu pour l'image 180x180 pixels de test N°3	55
Table IV.10 Résultat obtenu pour l'image 45x45 pixels de test N°3	55

Liste des abréviations

2D : Deux Dimensions.

3D: Trois Dimensions.

ANNs : Artificiel Neural Networks.

CCD : Dispositif à Transfert de Charge (français), Charge Coupled Device (anglais).

CR: compression rate.

DIB : Device Independent Bitmap.

EO-1: Earth Observing-1.

HSV : Hue/Teinte, Saturation, Valeur.

MLP : Multi Layer Perceptron.

MNT : Modèle Numérique de Terrain.

PMC : Perceptrons Multicouches

RVB : Rouge, Vert, Bleu.

YCBCR : Luminance, Chrominance Bleue, Chrominance Rouge.

Introduction Générale

L'utilisation des photos aériennes, images satellites, ... pose un grand problème de stockage et d'archivage. En plus du problème de stockage, si de telles images doivent être transmises via un réseau, la durée de la transmission est souvent trop longue. Pour pallier à tous ces problèmes, la compression des images numériques volumiques apparaît donc incontournable, elle consiste à minimiser le nombre de bits nécessaire à une représentation fidèle de l'image originale et d'accéder uniquement à l'information requise, allégeant ainsi les transferts et autorisant un accès à distance aux données. On distingue deux types de compression, la compression sans perte qui préserve l'intégrité des données, et la compression avec perte qui génère des dégradations mais offre des performances en termes de réduction de l'information bien plus grandes que celles issues de la compression sans perte.

Dont le but de réaliser de telle compression, plusieurs approches et méthodes sont mise à la place et à la disposition, chacune a ces propres caractéristiques ainsi que son taux de compression, dans ce travail on va faire appel aux réseaux de neurones qui sont l'une des méthodes issus de l'intelligence artificielle.

Dans cette optique, ce mémoire va porter sur le développement d'une application de compression des photos aériennes basée sur les réseaux de neurones, et il est organisé comme suit :

Le premier chapitre englobe les différentes caractéristiques de l'image numérique, ses outils d'acquisition et ses formats.

Dans le chapitre deux nous exposons quelques différentes méthodes de compression qui existent, ainsi que leurs types.

Le troisième chapitre présente les principes de la méthode de compression qui est le réseau de neurones.

Le dernier chapitre sera consacré à notre logiciel ainsi que les différents résultats obtenus lors de leur application sur différentes images.

Et, nous terminons ce mémoire par une conclusion et quelques perspectives pour les travaux futurs.

Chapitre I : L'image numérique

I.1.	Introduction	13
I.2.	Image numérique	13
I.3.	Caractéristiques des images Raster	13
I.4.	Définition de l'image numérique	14
I.5.	Formats des fichiers images.....	15
I.6.	Acquisition d'images	15
I.7.	Image Satellite.....	16
I.7.1.	Image panchromatique.....	16
I.7.2.	Images couleurs, multi spectrales et hyper spectrales.....	17
I.8.	Photo aérienne.....	18
I.8.1.	Les types de photographie	18
I.8.1.1	Les photographies verticales	18
I.8.1.3.	Les photographies obliques	19
II.8.1.3.	Les photographies panoramiques	19
I.8.2.	Caractéristiques de photo aérienne	20
1.8.2.1.	Echelles et restitutions	20
I.8.2.2.	Émulsions et longueurs d'onde.....	20
I.8.2.3.	Spectre visible	20
I.8.2.4.	Spectre invisible	21
I.8.3.	Caractéristiques des vols	21
I.8.4.	Voir en trois dimensions	21
I.8.5.	Ortho photographie numérique.....	22
I.8.6.	Etablissement de l'ortho photo	22
I.9.	Conclusion.....	23

I.1. Introduction

Depuis de nombreuses années, on assiste à une croissance constante des besoins en images numériques (fixes ou animées), dans de multiples domaines [1]. L'image est un des supports physiques les plus privilégiés pour transmettre un message à notre cerveau. Avant de nous intéresser à l'image numérique en tant que telle, nous allons tout d'abord aborder de façon très succincte l'image dans son aspect le plus perceptif, c'est à dire du point de vue de notre perception visuelle et du décodage mental que nous y associons.

I.2. Image numérique

Avant toute chose, mettons-nous d'accord sur le terme " image " ; en effet, le traitement d'images fait appel non pas à des images optiques classiques (telles que notre œil les perçoit), mais à des images numériques. Les logiciels de traitement d'images travaillent sur des données chiffrées contenues dans l'image, modifient ces données qui sont ensuite utilisées pour construire une seconde image " transformée " visualisable. La représentation numérique des images permet de combiner des images obtenues sous différents modes de macro ou de microscopie, de représenter une organisation spatiale (3D) à partir d'un ensemble de clichés successifs (2D), d'archiver les images et de les transférer par les réseaux informatiques [2].

Les calculs effectués sur ces données numériques, par un logiciel de traitement d'images ou par un logiciel d'analyse d'images, vont permettre de modifier l'aspect visuel de l'image et d'extraire des données quantifiées sur cette image [1].

I.3. Caractéristiques des images Raster

Il existe 2 grandes classes d'images : les images vectorielles et les images rasters. Une image raster est transformée en une image vecteur par une vectorisation. L'opération inverse est une rastérisation.

❖ **Les images vectorielles** : utilisées principalement dans le monde du graphisme et de la conception assistée par ordinateur,

❖ **Les images rasters**: utilisées dans le domaine du traitement et de l'analyse d'images ; ce sont celles qui seront décrites dans ce support [2].

I.4. Définition de l'image numérique

L'image numérique correspond à une matrice (ensemble ordonné à deux ou trois dimensions) de données numériques. Nous nous intéresserons uniquement aux images en deux dimensions. On peut concevoir ces images en deux dimensions comme un tableau de valeurs auxquelles on fait correspondre une position sur un plan (x, y) et une couleur pour visualiser l'image sur l'écran d'un ordinateur [3].

-Exemple d'un détail d'une image binaire dont la couleur est codée en 0 ou en 1 :

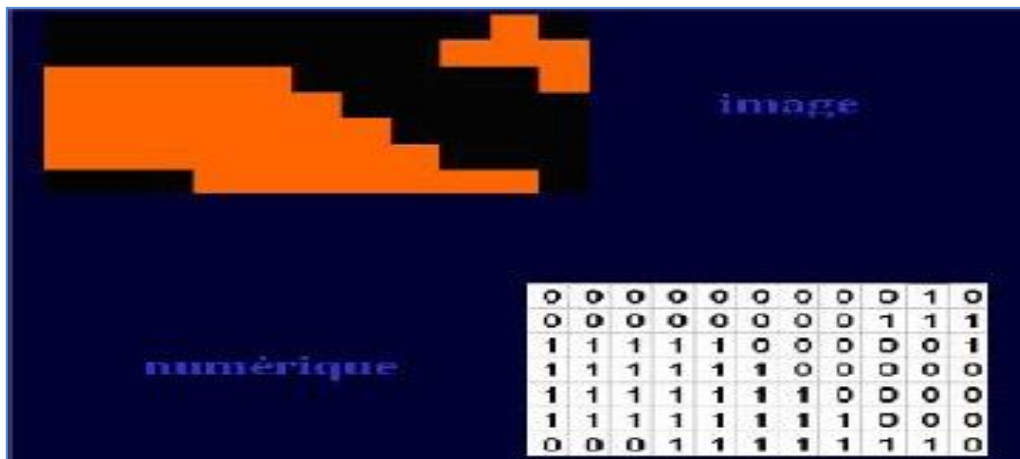


Figure I.1. Détail d'une image binaire.

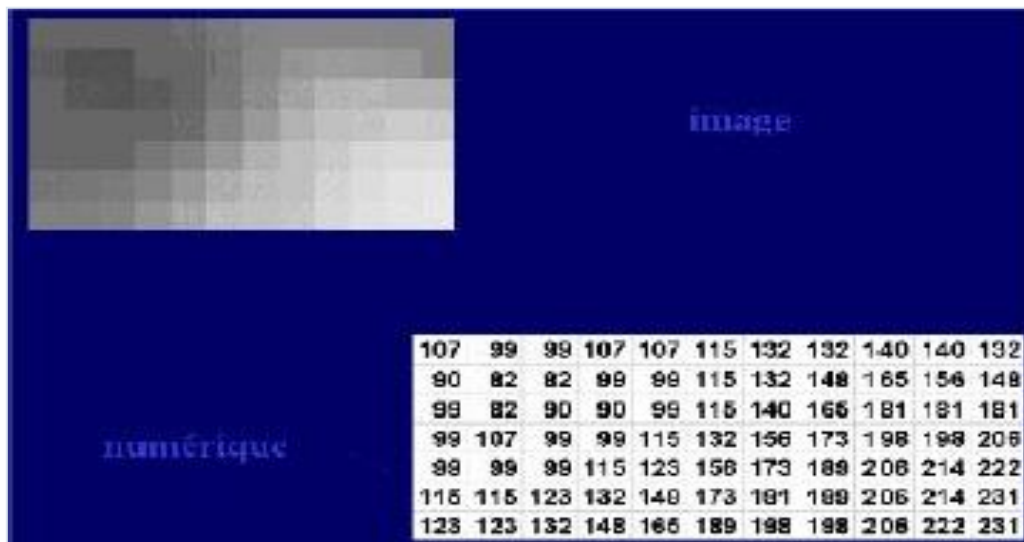


Figure I.2. Détail d'une image en niveaux de gris.

Une image numérique 2D est donc composée d'unités élémentaires appelées pixels représente chacune, une portion de l'image, codée par des valeurs numériques (des explications complémentaires sur les pixels seront données dans la partie résolution spatiale

d'une image, située plus loin dans ce mémoire. Une image est définie par le nombre de pixels qui la composent en largeur et en hauteur (qui peut varier théoriquement presque à l'infini) et l'étendue des teintes de gris ou des couleurs que peut prendre chaque pixel (on parle de dynamique de l'image). Toutes les données correspondant aux informations chiffrées contenues dans l'image sont structurées, afin de permettre leur stockage [3].

Une image numérique en elle-même est en fait un concept tout à fait abstrait (des données chiffrées qui ne trouvent une signification à nos yeux qu'à la visualisation, lorsque l'on utilise un logiciel de représentation adéquat [2]).

I.5. Formats des fichiers images

Il n'existe pas de format idéal. Selon le cas, certains formats peuvent être préférables. Le terme bitmap employé ci-dessous fait référence aux formats de fichier qui stockent les pixels sous forme de tableau de points, avec une gestion des couleurs soit en couleur vraie soit par le biais d'une palette indexée.

- **Format RAW** (.raw) : Format de stockage des données brutes (pas d'en-tête).
- **Format GIF** (.gif) : Format de fichier graphique bitmap (raster) relativement dépassé. Il est utilisé pour des images codées sur 8 bits au plus, c'est à dire des images de basse qualité : 2 à 256 pixels.
- **Format TGA** : Il est très utilisé pour stocker les images en vraie couleur codées sur 24 ou 32 bits.
- **Format JPEG** (.jpg) et **JPEG 2000** (.jp2) : Le plus connu et le plus employé pour ses modes de compression variés. Son niveau varie de 0 à 10. Le niveau 8 est souvent conseillé.
- **Format TIFF** (.tif) : Ancien format de fichier graphique bitmap (raster) très utilisé.
- **Format PCX** : Ce format bitmap. Il permet d'encoder des images dont la dimension peut aller jusqu'à 65536 par 65536 et codées sur 1 bit, 4 bit, 8 bit ou 24 bit.
- **Format BMP** : C'est un des formats bitmaps les plus simples. Il a été conçu pour être indépendant du périphérique d'affichage (DIB, Device independent bitmap) [4].

I.6. Acquisition d'images

Divers types de capteurs permettent d'acquérir des images raster numériques. Les plus usuels sont :

- ❖ les appareils photo et caméras numériques monocanal et multicanal, L'acquisition d'images dans les domaines spectraux du bleu, du vert et du rouge permet de reconstituer la vision humaine.
- ❖ les scanners médicaux : Ils donnent des images 3-D sous la forme de séries d'images 2-D.
- ❖ les scanners et micro densitomètres : Ils numérisent des images ou négatifs sous forme analogique.
- ❖ les microscopes : En modifiant leur focale, ils permettent d'observer la nature 3-D des objets étudiés.
- ❖ les capteurs de télédétection : Systèmes rigides (e.g., barrette de 6000 CCD pour SPOT) ou à balayage (e.g., Thematic Mapper), avec un nombre très variable de bandes spectrales[4].

I.7. Image Satellite

I.7.1. Image panchromatique

A chaque pixel est associée une valeur correspondant à une intensité lumineuse, ces intensités sont en général représentées par une gamme de gris allant du noir pour la plus faible intensité au blanc pour la plus forte. En imagerie satellite, on parle d'image panchromatique lorsque l'acquisition est effectuée sur une bande spectrale large exemple 480 nm à 710 nm [6].



Figure I.3. Image de Toulouse acquise par le mode panchromatique de SPOT 5

I.7.2. Images couleurs, multi spectrales et hyper spectrales

Pour une image en couleurs, ce n'est pas une valeur qui est associée à chaque pixel mais un vecteur de trois valeurs. Ce vecteur est couramment représenté dans la base RVB (Rouge, Vert, Bleu) mais il existe aussi d'autres bases, YCBCR (Luminance, Chrominance bleue, Chrominance rouge) ou HSV (Hue/Teinte, Saturation, Valeur). On parle d'images multi spectrales lorsque le vecteur de valeurs associé à un pixel est au moins de dimension deux. Par exemple, les satellites d'observation de la Terre PLEIADES permettront l'acquisition d'images multi spectrales avec les 4 bandes spectrales B0 à B3 dans le bleu, le vert, le rouge et le proche infrarouge. L'image en couleurs représentée sur la figure 4 a été créée à partir des bandes B0, B1 et B2. La multiplication du nombre de bandes spectrales permet de différencier la nature des objets observés. Par exemple, la bande verte et la bande proche infrarouge sont sensibles à la chlorophylle alors que la bande moyen infrarouge est sensible à l'amidon, à la lignine, un composé du bois, et permet de séparer les nuages de la neige et de la glace. On parle généralement d'images hyper spectrales lorsque le nombre de bandes spectrales est supérieur à la centaine, Hypérion est le premier capteur hyper spectral lancé à bord du satellite EO-1 (Earth Observing-1) de la NASA en 2000. Il est capable d'acquérir 220 bandes spectrales entre 400nm et 2500nm [6].



Figure I.4. Image couleurs simulée PLEIADES à la résolution de 70 cm par pixel

I.8. Photo aérienne

Les photos aériennes numériques sont obtenues par scénarisation à partir de films analogiques grand format (généralement 23x23 cm). Elles sont prises par avion, ou bien elles sont prises directement en mode numérique, avec des caméras numériques aéroportées. Ces images numériques aéroportées sont prises, à la manière des images satellite. Par ailleurs, ces caméras prennent simultanément des images panchromatiques (en noir et blanc) et des images couleur qu'on dit aussi multi spectrales (en rouge, vert et bleu) [7].

I.8.1. Les types de photographie

Elles peuvent être classées en fonctions de différents critères : l'émulsion, la longueur D'onde observée et les angles de prise de vue.

I.8.1.1 Les photographies verticales

Les plus intéressantes pour la cartographie et l'interprétation. Lecture facile des détails et des Mesures. Permet de restituer les lignes ou les points ; la vision stéréoscopique. L'axe optique de ce type de photo est perpendiculaire par rapport au sol. Cette perpendicularité n'est jamais parfaite. Elle est considérée comme verticale si l'angle d'inclinaison de l'axe optique n'excède pas 5 grades.



Figure I.5. Prise de vue photographie verticale

I.8.1.3. Les photographies obliques

Elles présentent un aspect perspectif du terrain. Elles couvrent une grande superficie que les verticales pour une même altitude de prise de vue.



Figure I.6. Prise de vue photographie oblique

II.8.1.3. Les photographies panoramiques

C'est une observation similaire à celle que l'on pourrait prendre à partir d'un poste terrestre. L'horizon doit être apparent.



Figure I.7. Prise de vue photographie panoramique

I.8.2. Caractéristiques de photo aérienne

Les renseignements principaux de l'image se trouvent dans les marges de la photographie [8].

1.8.2.1. Echelles et restitutions

Les échelles varient en fonction de l'altitude de prise de vue. Le choix des échelles varie en fonction du domaine d'application.

Échelles	Domain d'application
1/500	Urbanisme : Opérations de rénovation.
Du 1/ 2 000 au 1/ 5 000	Urbanisme : Étude de faubourg, tissu urbain.
Du 1/5 000 au 1/ 10 000	Pédologie.
Du 1/5 000 au 1/ 25 000	Biogéographie : Botanique Études humaines : Urbaines, Rurales.
Du 1/15 000 au 1/ 40 000	Géomorphologie
Du 1/25 000 au 1/ 1 000 000	Géologie : une bonne vision des structures.
Échelles très petites (1/ 1 000 000 par ex.)	Structures continentales
Du 1/ 1 000 000 au 1/ 5 000 000	Météorologie

Table I.1. Échelle en fonction de domaine étutilisé.

I.8.2.2. Émulsions et longueurs d'onde

Les photographies aériennes et les images permettent d'explorer une partie du spectre électromagnétique plus large que la seule fraction visible à l'œil humain.

I.8.2.3. Spectre visible

Un rayon de lumière blanche peut être décomposé (par un prisme de cristal) et on obtient une bande de couleur allant du violet au rouge. Ces couleurs nous sont familières car elles correspondent à l'arc en ciel visible lors de la décomposition de la lumière du jour par les gouttes de pluie. Les longueurs d'ondes deviennent d'autant plus grandes que l'on progresse

dans le spectre du bleu au rouge. Le spectre visible comprend les ondes de fréquence allant d'environ 400 nm à 700 nm (millionième partie du millimètre)

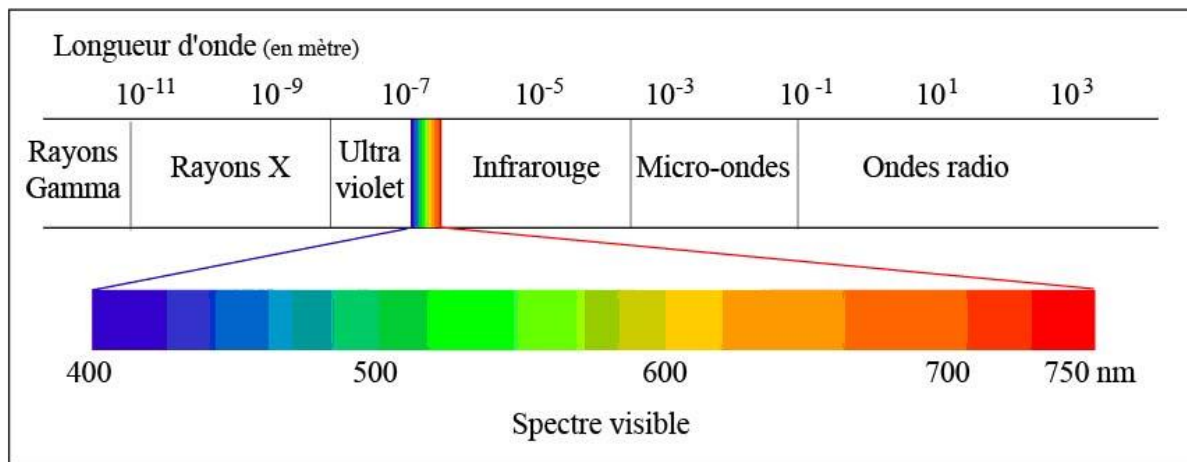


Figure I.8. Spectre visible

I.8.2.4. Spectre invisible

Certaines radiations ne sont pas visibles à l'œil humain. Elles se situent à chaque extrémité du spectre, l'ultraviolet (fréquence courte) et l'infrarouge (fréquences supérieure à celle du rouge).

I.8.3. Caractéristiques des vols

L'altitude varie en fonction du vecteur utilisé :

- de 100 m à 1 km pour un hélicoptère
- de 5 à 15 km pour les avions
- de 30 à 35 km pour les ballons
- de 250 à 1000 km pour les stellites [7]

I.8.4. Voir en trois dimensions

Les prises de vues aériennes sont toujours effectuées de façon à ce que deux photographies aient un même **recouvrement** longitudinal et latéral. Ces recouvrements permettent d'observer le territoire en trois dimensions, à l'aide d'un appareil spécialisé. Le phénomène optique qui permet de voir en trois dimensions s'appelle la **stéréoscopie** [7].

Afin d'obtenir une vision en trois dimensions du territoire couvert, une partie du sol représentée sur une photographie doit également être présente sur la suivante. C'est ce qu'on appelle le recouvrement. Lorsque l'avion se déplace, la zone photographiée qui est commune

sur les deux photographies aériennes fournit une vue du territoire prise sous deux angles différents.

En regardant cette partie à l'aide d'un stéréoscope, il est possible d'obtenir un aperçu du relief puisqu'il apparaît en trois dimensions.

-recouvrement longitudinal de 60 % pour la vue stéréoscopique;

-recouvrement latéral de 30 % pour assurer une couverture complète du territoire [7].

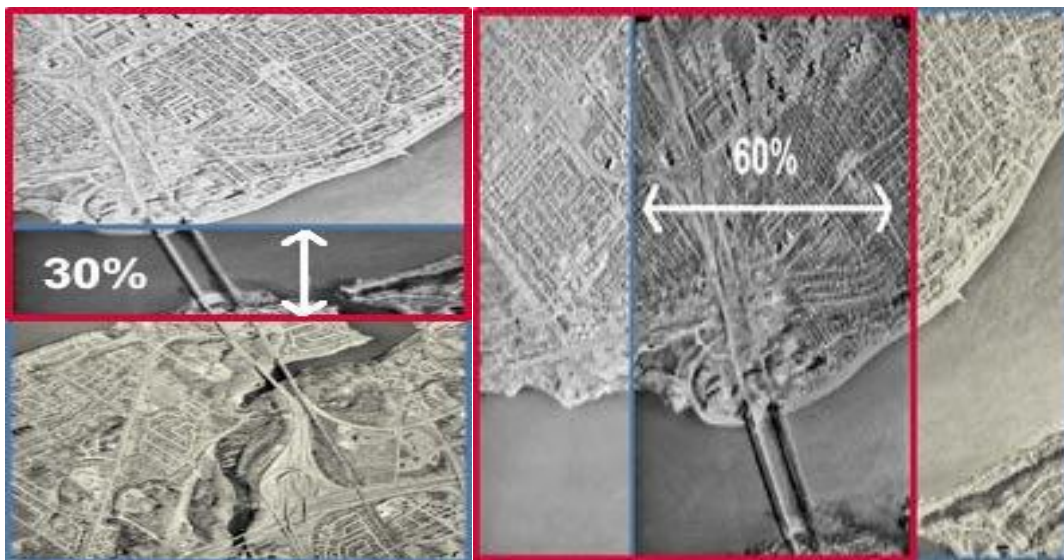


Figure I.10. Recouvrement latéral et Recouvrement longitudinal

I.8.5. Ortho photographie numérique

La photogrammétrie numérique est une source non contournable pour les systèmes d'information géographique, extraction de données géo référencées pour leur alimentation et superposition du vecteur et du raster pour la mise à jour. Elle est flexible en termes de rendement du fait que son produit sera destiné pour différentes applications.

L'ortho photo numérique, en particulier, reste une représentation visuelle combinant raster (richesse de l'image) et vecteurs (mesures fiables et homogènes) [8].

I.8.6. Etablissement de l'ortho photo

L'ortho photo est produit à l'aide d'une photographie (image) dans laquelle les déplacements de l'image causés par le relief et l'attitude du capteur ont été éliminés.

Pour cela chaque élément de l'image doit subir une transformation géométrique (rectification) suivie d'une interpolation des niveaux de gris des pixels transformés (ré échantillonnage) [8].

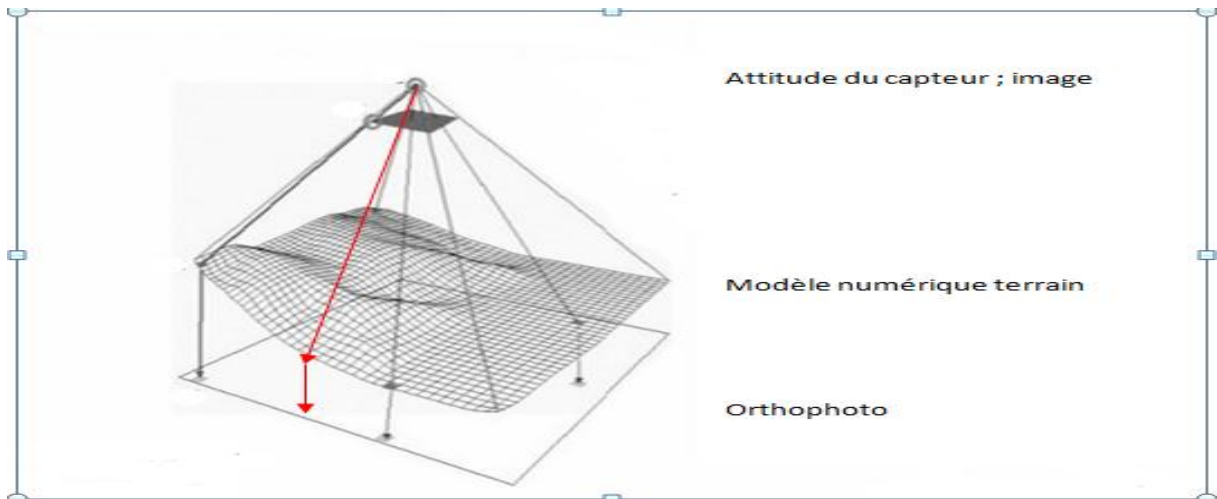


Figure I.11. L'obtention de l'ortho photo

I.9. Conclusion

Les photos aériennes vont par série (mission aérienne). Elles sont normalement observées dans des stéréoscopes à miroirs. Ce sont des photos à haute résolution dont seule une grande moitié est en commun avec la suivante. Mais l'utilisation des photos aériennes pose un grand problème de stockage et d'archivage. En plus du problème de stockage, si de telles images doivent être transmises via un réseau, la durée de la transmission est souvent trop longue. Pour pallier à tous ces problèmes, la compression de ces images devient une opération nécessaire et impérative.

Chapitre II : La Compression

II.1.	Introduction	25
II.2.	Définition de compression.....	25
II.3.	Historique du codage des images	25
II.4.	Intérêt	26
II.5.	Classification des algorithmes de compression	26
II.5.1.	Compression symétrique / asymétrique.....	26
II.5.2.	Compression physique / logique	27
II.5.3.	Compression statistique / numérique.....	27
II.5.4.	Compression sans / avec perte.....	28
II.5.4.1.	Compression sans perte (compactage)	28
II.5.4.2.	Compression avec perte (irréversible ou non conservative).....	28
II.6.	Taux de compression.....	29
II.7.	Méthodes d'intelligence artificielle pour la compression.....	29
II.8.	Conclusion.....	30

II.1. Introduction

L'objectif de ce chapitre est de présenter la compression avec les réseaux des neurones artificiels basés sur l'apprentissage supervisé. Après une brève définition, nous présentons quelques repères historiques pour le codage des signaux des images numériques. Ensuite nous décrivons leur intérêt. Puis nous présentons la classification selon les méthodes et les algorithmes. Et nous terminons par une conclusion.

II.2. Définition de compression

La compression consiste à réduire la taille physique de blocs d'informations. Elle est très utile pour plusieurs applications informatiques. Un compresseur utilise un algorithme qui sert à optimiser les données en fonction du type de données à compresser ; un décompresseur est donc nécessaire pour reconstruire les données grâce à l'algorithme dual de celui utilisé pour la compression.

La méthode de compression dépend du type de données à compresser car une image ou un fichier audio ne représentent pas le même type de données [9].

II.3. Historique du codage des images

On présente ici quelques repères historiques pour cette question du codage des signaux d'image, avec cependant un biais certain en faveur de la littérature « image ». On peut sans doute fixer le début des travaux concernant la compression des images aux années 1950, qui voient le dépôt d'un brevet sur le codage prédictif [10]. En 1952, une publication de Harrison [11] sur l'application de la prédiction linéaire à la télévision la même année, et les travaux de Huffman [12] sur la construction de codes à longueur variable efficaces. Il convient aussi de mentionner les travaux antérieurs de Gabor [13] qui ont posé les fondements de l'analyse temps (espace)/fréquence qui auront comme conséquence l'intérêt porté aux ondelettes dans les années 80, ainsi que l'œuvre de Shannon [14] qui a posé les bases théoriques pour le codage efficace de l'information en général.

Les années 60 ont vu de nombreux travaux sur le codage par transformation (Huang et Schultheiss, Andrews et Pratt) [15]. Un autre intérêt est aussi apparu concernant l'étude des relations entre les propriétés visuelles d'un observateur et des algorithmes de codage adaptés, à travers des papiers de Limb , Pearson , et Schreiber [19]. Pendant les années 70 et début 80,

les développements concernant le codage par transformation ont explosé. Parallèlement, les travaux sur la quantification vectorielle gagnaient en maturité (Kang et Coulter [20].

Et passaient au domaine de l'image, grâce à Gersho et Ramamurthi [21] pour se poser en concurrent de l'approche par transformation.

Au cours des années 80, on a aussi assisté à l'éclosion des techniques de compression d'images par approches fractales, suite aux travaux pionniers de Barnsley [22], avec des réalisations intéressantes. Mais c'est sans doute dans le domaine de vidéo (l'image animée) que se sont produits les changements les plus importants, avec de nombreux travaux sur l'estimation du mouvement et son utilisation par compensation dans les schémas de codage. Les études de Netravali et de son équipe (voir par exemple Netravali et Robbins) [23].

II.4. Intérêt

De nos jours, la puissance des processeurs augmente plus vite que les capacités de stockage, Et énormément plus vite que la bande passante des réseaux (car cela imposerait d'énormes changements dans les infrastructures de télécommunication).

Il y a donc un déséquilibre entre le volume des données qu'il est possible de traiter, de stocker, et de transférer. Par conséquent, il faut donc réduire la taille des données. Pour cela, il faut exploiter la puissance des processeurs, pour pallier aux insuffisances des capacités de stockage en mémoire et des vitesses de transmission sur les réseaux [9].

II.5. Classification des algorithmes de compression

La compression des données peut être classée de différentes manières:

II.5.1. Compression symétrique / asymétrique

Dans le cas de la compression symétrique, la même méthode est utilisée pour compresser et décompresser l'information, il faut donc la même quantité de travail pour chacune de ces opérations. C'est ce type de compression qui est généralement utilisée dans les transmissions de données [24].

La compression asymétrique demande plus de travail pour l'une des deux opérations, la plupart des algorithmes requiert plus de temps de traitement pour la compression que pour la décompression. Des algorithmes plus rapides en compression qu'en décompression peuvent être nécessaires lorsque l'on archive des données auxquelles on accède peu souvent (pour des raisons de sécurité par exemple) [2].

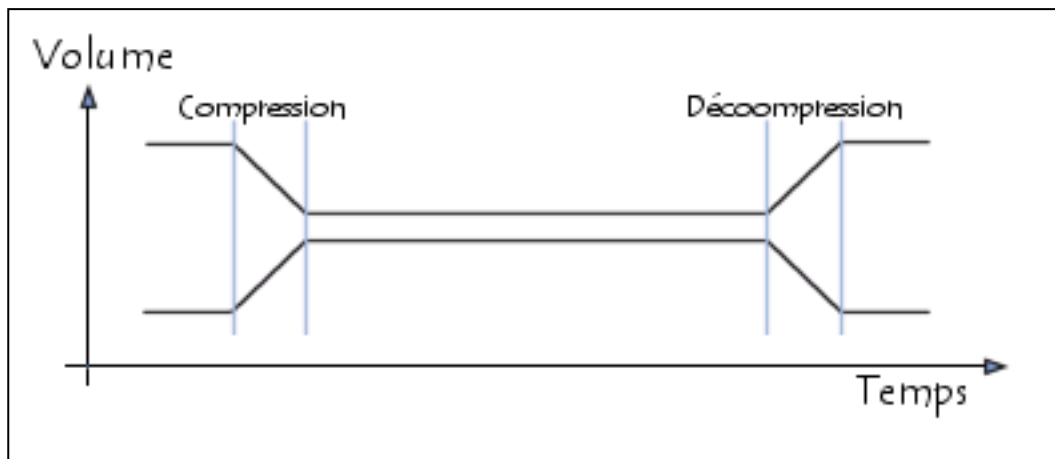


Figure. II .1.Evaluation en temps de calcul et en volume de la compression symétrique.

II.5.2. Compression physique / logique

On considère généralement la compression comme un algorithme capable de compresser des données dans un minimum de place (compression physique), mais on peut également adopter une autre approche et considérer qu'en premier lieu un algorithme de compression a pour but de recoder les données dans une représentation différente plus compacte contenant la même information (compression logique)[25].

La distinction entre compression physique et logique est faite sur la base de comment les données sont compressées ou plus précisément comment est-ce que les données sont réarrangées dans une forme plus compacte.

La compression physique est exécutée exclusivement sur les informations contenues dans les données. Cette méthode produit typiquement des résultats incompréhensibles qui apparemment n'ont aucun sens. Le résultat d'un bloc de données compressées est plus petit que l'original car l'algorithme de compression physique a retiré la redondance qui existait entre les données elles-mêmes. Toutes les méthodes de compression dont nous allons traiter sont des méthodes physiques [25].

II.5.3. Compression statistique / numérique

On peut encore distinguer les algorithmes qui travaillent au niveau statistique et ceux qui opèrent au niveau numérique.

Pour les premiers, la valeur des motifs ne compte pas. Ce sont les probabilités qui comptent, et le résultat est inchangé par substitution des motifs tandis que pour les seconds, les valeurs des motifs influent sur la compression (par exemple JPEG), et les substitutions sont interdites [9].

II.5.4. Compression sans / avec perte

On distingue à ce jour deux types de compression. La première catégorie est la compression sans perte, conduisant à une reconstruction parfaite. Ce type de compression est notamment utilisé dans des domaines où l'on souhaite compresser des informations, sans toutefois introduire une quelconque erreur sur des données hautement sensibles. Ainsi les efforts actuels portent plus spécifiquement sur l'élaboration de nouvelles techniques en compression d'images avec pertes [26].

II.5.4.1. Compression sans perte (compactage)

Les algorithmes de compression sans perte (connu aussi sous le nom de non destructible, réversible, ou conservative) sont des techniques permettant une reconstitution exacte de l'information après le cycle de compression / décompression. Il existe 3 types d'algorithmes de compression sans perte :


1. **Codage statistique** : le but est de :
 - ✓ Réduire le nombre de bits utilisés pour le codage des caractères fréquents.
 - ✓ Augmenter ce nombre pour des caractères plus rares.
2. **Substitution de séquences** : le but est de :
 - ✓ Comprise les séquences de caractères identiques.
3. **Utilisation d'un dictionnaire** : le but est de :
 - ✓ Réduire le nombre de bits utilisés pour le codage des mots fréquents.
 - ✓ Augmenter ce nombre pour des mots plus rares [9].

II.5.4.2. Compression avec perte (irréversible ou non conservative)

Ce type de compression est utilisé pour des données de type images, sons ou vidéo. En effet, étant donné que l'être humain a une acuité visuelle et auditive limitée, il ne peut, au-delà d'un certain seuil, faire la différence entre deux couleurs qui ont des nuances proches ni même entendre des sons au-delà de certaines fréquences.

Dans ce type de compression, les logiciels et plus précisément les algorithmes vont s'attacher à définir ces seuils et, soit supprimer les informations associées (cas d'une donnée audio), soit les grouper par pixels contiguës ayant des nuances voisines le tout codé sous une même couleur. Cette méthode est considérée comme irréversible car une fois supprimées ou recodées, on ne peut retrouver les informations d'origines. On retrouve ce type de compression sous des fichiers de type MP3 pour les sons; JPEG, JPEG2000 pour les images ou encore MJPEG, MJPEG2000 pour les vidéos [9].

227	229	225
225	225	227
229	227	227



225	225	225
225	225	225
225	225	225

Figure. II .2. Image initiale & Image compressée

La différence majeure, en comparaison avec la première catégorie (compression sans perte), est que l'on ne cherche plus à obtenir une reconstruction identique à la source. On tolère ainsi une approximation qui n'induit pas de dégradations visuellement perceptibles. Bien évidemment, tout le savoir-faire réside dans la maîtrise de ces pertes [26].

II.6. Taux de compression

On mesure l'efficacité de la compression par le taux de compression. Le taux de compression T est relié au rapport entre la taille b du fichier comprimé B et la taille a du fichier initial A . Le taux de compression est généralement exprimé en pourcentage. Un taux de 50 % signifie que la taille b du fichier comprimé B est la moitié de a . La formule pour calculer ce taux est : $T = 100 - ((b/a) * 100)$ [9].

Exemple: $a : 550 \text{ Mo}$, $b : 250 \text{ Mo}$; $T = 100 - ((250/550)*100) = 54 \%$.

II.7. Méthodes d'intelligence artificielle pour la compression

En générale on peut définir deux classes pour y attribuer les algorithmes de compression : des algorithmes statistiques, et des algorithmes issus de l'intelligence artificielle, on va situer quelques algorithmes statistiques et détaillées d'autres de l'intelligence artificielle et terminerons par expliquer notre approche de compression issus de l'intelligence artificielle.

Parmi les algorithmes de l'intelligence artificielle pour compression des images numériques nous avons la compression par réseaux de neurones.

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau [27].

Nous allons citer les différentes étapes nécessaires pour appliquer une méthode de compression d'une image proposée par [28] :

- 1- Acquisition des images
- 2- Prétraitement de l'image : la binarisation est l'une des prétraitements nécessaires avant la compression
- 3- Segmentation de l'image : consiste à diviser l'image en régions (images plus petites) elle a comme objectif de minimiser le nombre des entrées au ANNs.
- 4- Préparation des formations paires : après la décomposition de telle image en petites autres images, il est indispensable d'interpréter chacune d'eux sous format acceptable par les ANNs. Donc, les images vont être représentées par une paire de vecteurs, un vecteur d'entrée et l'autre de sortie
- 5- Exclusion des paires similaires
- 6- Compression des images par les ANNs
- 7- Implémentation du programme de reconstruction de l'image.

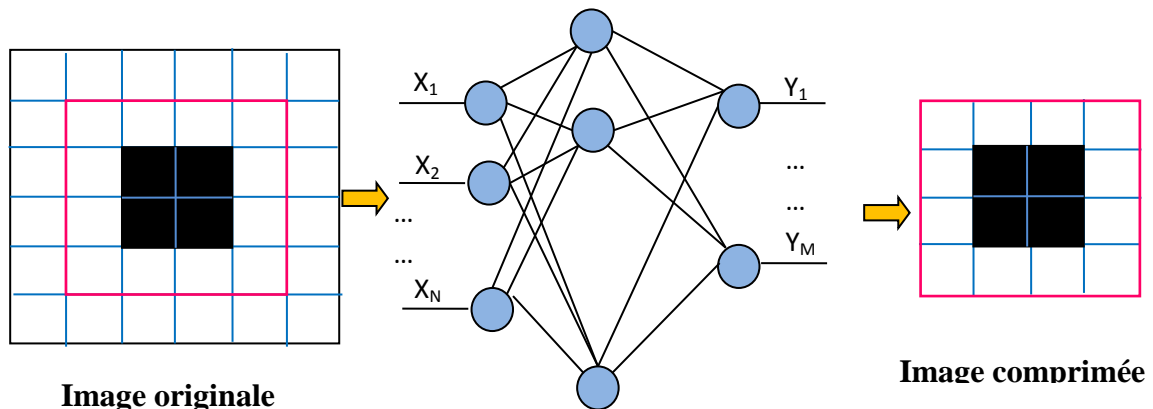


Figure. II .3.Schéma de compression d'image par les réseaux de neurones

II.8. Conclusion

La compression des données est appelée à prendre un rôle encore plus important en raison du développement des réseaux et du multimédia. Son importance est surtout due au décalage qui existe entre les possibilités matérielles des dispositifs que nous utilisons (débits sur Internet...) et les besoins qu'expriment les utilisateurs (vidéo plein écran, transfert de quantités d'information toujours plus importantes dans des délais toujours plus brefs) [2].

Chapitre III : Les réseaux de neurones

III.1.	Introduction	32
III.2.	Neurone biologique	32
III.3.	Neurone formel.....	33
III.3.1.	Synoptique.....	33
III.3.2.	Fonctionnement	34
III.3.3.	Réseaux de neurones non bouclés	35
III.3.4.	Réseaux de neurones bouclés	36
III.4.	Propriétés d'un réseau de neurones.....	37
III.5.	Apprentissage	38
III.5.1.	Apprentissage supervisé	38
III.5.2.	Apprentissage non supervisé	38
III.6.	Modèles des réseaux de neurones.....	39
III.6.1.	Modèle de Hopfield.....	39
III.6.2.	Modèle Kohonen	39
III.6.3.	Les Perceptrons Multicouches (PMC).....	40
III.6.4.	Le modèle Adaline	41
III.7.	Règle d'apprentissage du Perceptron	42
III.8.	Description et principe	42
III.9.	Classification par le MLP.....	42
III.9.1.	Architecture	42
III.9.2.	Phase d'apprentissage.....	43
III.10.	Conclusion.....	43

III.1. Introduction

Aujourd'hui de nombreux termes sont utilisés dans la littérature pour désigner le domaine des réseaux de neurones artificiels, comme connexionnisme ou neuromimétique. Pour notre part, il nous semble qu'il faut associer à chacun de ces noms une sémantique précise. Ainsi, les réseaux de neurones artificiels ne désignent que les modèles manipulés ; Connexionnisme et neuromimétique sont tous deux des domaines de recherche à part entière, qui manipulent chacun des modèles de réseaux de neurones artificiels, mais avec des objectifs différents [29].

III.2. Neurone biologique

C'est l'organe élémentaire de base qui forme le système nerveux des êtres vivants, il sert à la mise au point des premiers neurones formels.

Cette partie permet de présenter rapidement les éléments de base du cerveau que sont les neurones et d'avoir un bref aperçu sur leur fonctionnement.

Le neurone biologique est composé des éléments suivants : le corps cellulaire, l'axone, les dendrites et les synapses [30].

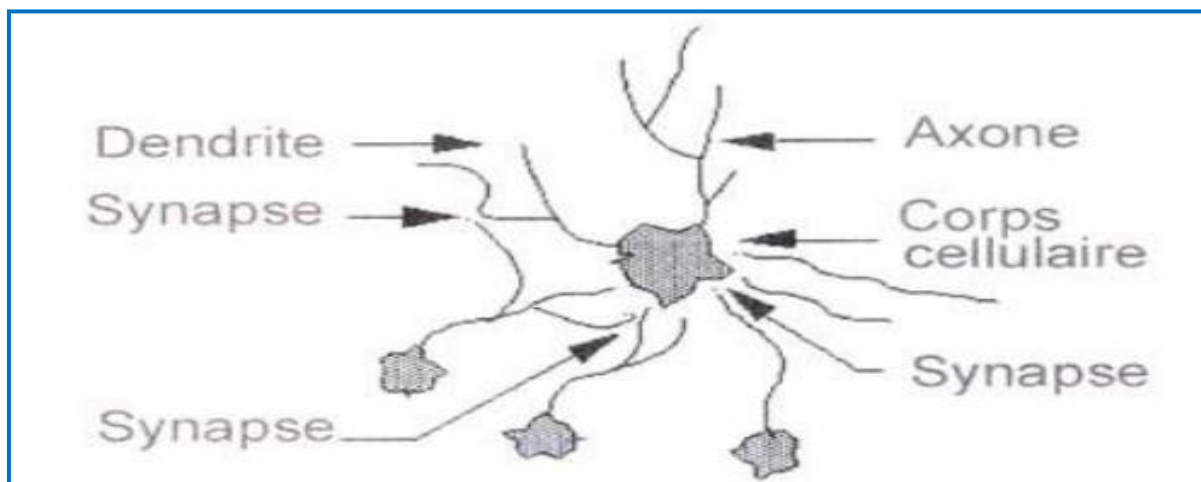


Figure. III .1. Le neurone biologique

III.3. Neurone formel

Un neurone formel est en général une fonction algébrique non linéaire et bornée, dont la valeur dépend de paramètres appelés coefficients ou poids. Les variables de cette fonction sont habituellement appelées « entrées » du neurone, et la valeur de la fonction est appelée sa « sortie ».

III.3.1. Synoptique

C'est un modèle mathématique très simple dérivé d'une analyse de la réalité des neurones biologiques. Pour le réaliser, on effectue une simple sommation des signaux arrivant au neurone, la sortie est obtenue par comparaison de cette somme au seuil de déclenchement, par exemple, elle peut être égale à 1 si la somme est supérieure au seuil et 0 dans le cas contraire.

Plus simple, il suffit de soustraire le seuil considéré à la somme des entrées, et de faire passer le résultat par la fonction de transfert la sortie du neurone est le résultat obtenu après le transfert ,ainsi, on obtient une représentation des propriétés physique du neurone biologique. Pour donner plus de puissance à ce modèle, on introduit la connexion synaptique qui est une valeur réelle, la sortie d'un neurone est alors multipliée par la valeur de la connexion synaptique durant le traitement effectué par le neurone récepteur.

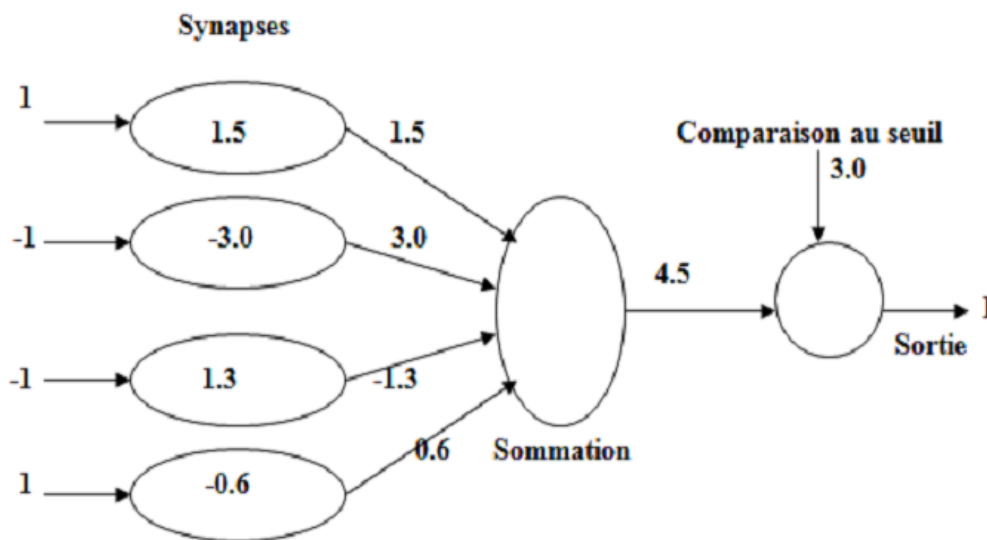


Figure III.2.Fonctionnement du neurone formel

En résumé, on peut dire que le neurone formel réalise tout simplement une somme pondérée de ces entrées, ajoute un seuil à cette somme et fait passer le résultat par une fonction de transfert pour obtenir sa sortie [31].

III.3.2. Fonctionnement

Pour fonctionner, un neurone formel utilise des entrées qui sont des grandeurs réelles. Si on relie un neurone au monde extérieur par des capteurs, il peut réaliser une simple analyse de ce qu'il « perçoit ». Si on représente les valeurs observées par le neurone sous forme d'un vecteur, le neurone réalise alors un découpage de son espace d'entrée (l'espace vectoriel auquel appartient le vecteur d'observation) en deux zones :

- ❖ La zone d'activité : dont les vecteurs donnent une sortie égale à 1 ;
- ❖ La zone d'inactivité : dont les vecteurs donnent une sortie égale à 0.

Comme le calcul effectué est en fait linéaire, la séparation l'est aussi. Les coefficients synaptiques et le seuil définissent l'équation d'un hyperplan qui est la frontière de la séparation entre les deux zones.

Mathématiquement, le neurone formel est considéré comme une fonction non linéaire, bornée et dont la valeur de sortie dépend de certains paramètres appelés coefficients ou poids. Les entrées de cette fonction représentent les entrées du neurone, tandis que son résultat représente la sortie [30].

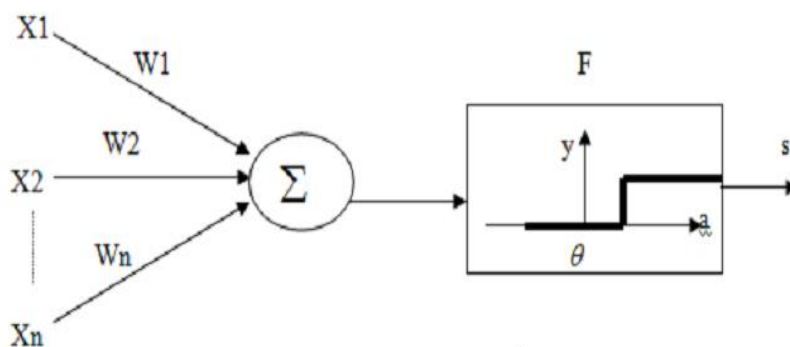


Figure III.3. Neurone Formel

Les X_1, X_2, \dots, X_n sont les n signaux d'entrée, et les W_1, W_2, \dots, W_n représentent les poids qui relient les neurones i, j ; L'activation pondérée « a » est donnée par :

$$a = X_1 W_{i_1} + X_2 W_{i_2} + \dots + X_n W_{i_n} = \sum x_i w_{i,j} = f(a)$$

La sortie finale est donnée par le seuil d'activation comme suit :

$$\text{Seuil} = \begin{cases} \mathbf{1}, & \mathbf{a} \geq \mathbf{0} \\ \mathbf{0}, & \mathbf{a} < \mathbf{0} \end{cases}$$

La fonction d'activation f , détermine l'état interne du neurone en fonction de son entrée totale. Plusieurs modèles de cette fonction existent dans la littérature, on les décrit par la suite.

Fonction Heaviside définie par :

$$\mathbf{H}(\mathbf{x}) = \begin{cases} \mathbf{1}, & \mathbf{x} \geq \mathbf{0} \\ \mathbf{0}, & \mathbf{x} < \mathbf{0} \end{cases}$$

Fonction signe définie par :

$$\text{Signe}(\mathbf{x}) = \begin{cases} +\mathbf{1}, & \mathbf{x} > \mathbf{0} \\ -\mathbf{1}, & \mathbf{x} < \mathbf{0} \end{cases}$$

La fonction linéaire est définie par : $\mathbf{F}(\mathbf{x}) = \mathbf{x}$

La fonction sigmoïde est définie par : $\mathbf{F}(\mathbf{x}) = \mathbf{1}/\mathbf{1}+\mathbf{e}^{-\mathbf{x}}$

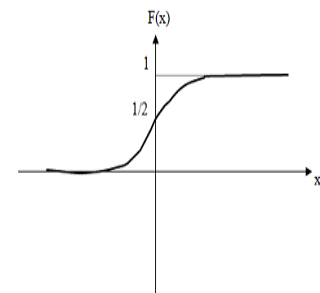
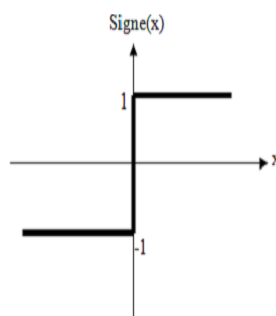
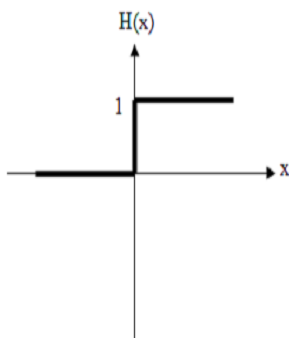


Figure.III.4.Fonction Heaviside **Figure.III.5. Fonction signe** **Figure.III.6. Fonction sigmoïde.**

Donc on peut conclure qu'un neurone formel réalise simplement une somme pondérée de ces entrées, ajoute un seuil à cette somme et fait passer le résultat par une fonction de transfert pour obtenir sa sortie.

On distingue deux grands types d'architectures de réseaux de neurones : les réseaux de neurones non bouclés et les réseaux de neurones bouclés.

III.3.3. Réseaux de neurones non bouclés

Un réseau de neurones non bouclé est représenté graphiquement par un ensemble de neurones "connectés" entre eux, l'information circulant des entrées vers les sorties sans "retour en arrière" ; si l'on représente le réseau comme un graphe dont les nœuds sont les

neurones et les arêtes les "connexions" entre ceux-ci, le graphe d'un réseau non bouclé est acyclique. Le terme de "connexions" est une métaphore : dans la très grande majorité des applications, les réseaux de neurones sont des formules [32].

algébriques dont les valeurs numériques sont calculées par des programmes d'ordinateurs, non des objets physiques (circuits électroniques spécialisés) ; néanmoins, le terme de connexion, issu des origines biologiques des réseaux de neurones, est passé dans l'usage, car il est commode quoique trompeur. Il a même donné naissance au terme de connexionnisme.

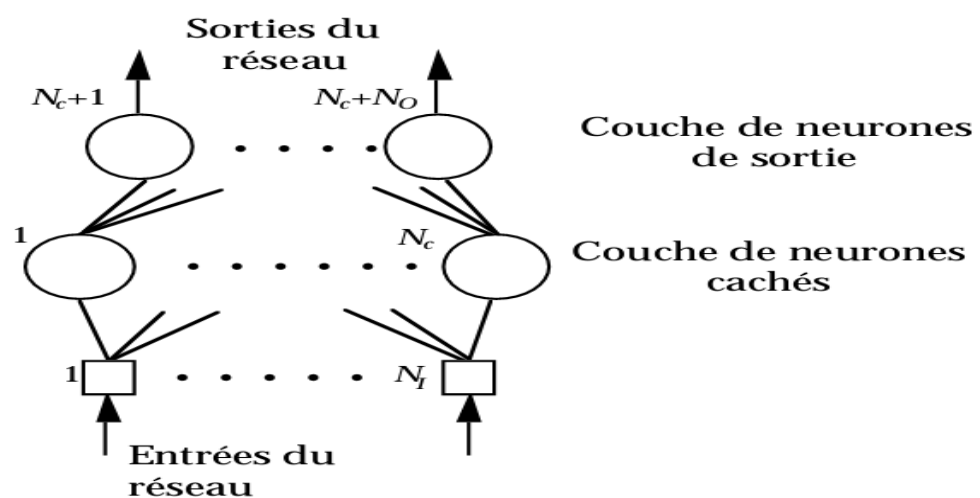


Figure III.7. Un perceptron multicouche

Les réseaux de neurones non bouclés sont des objets statiques : si les entrées sont indépendantes du temps, les sorties le sont également. Ils sont utilisés principalement pour effectuer des tâches d'approximation de fonction non linéaire, de classification ou de modélisation de processus statiques non linéaires[32].

III.3.4. Réseaux de neurones bouclés

Nous introduisons ici l'architecture la plus générale pour un réseau de neurones, les «réseaux bouclés», dont le graphe des connexions est cyclique : lorsqu'on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ. La sortie d'un neurone du réseau peut être fonction d'elle-même; cela n'est évidemment concevable que si la notion de temps est explicitement prise en considération. Ainsi, à chaque connexion d'un réseau de neurones bouclé (ou à chaque arête de son graphe) est attaché, outre un poids comme

pour les réseaux non bouclés, un retard, multiple entier de l'unité de temps choisie. Une grandeur, à un instant donné, ne pouvant pas être fonction de sa propre valeur au même instant, tout cycle du graphe du réseau doit avoir un retard non nul[33].

Les réseaux de neurones bouclés sont utilisés pour effectuer des tâches de modélisation de systèmes dynamiques, de commande de processus, ou de filtrage[32].

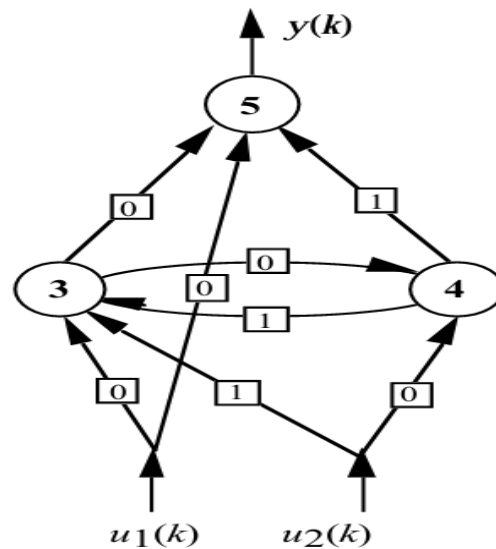


Figure III.8. Un réseau de neurones bouclé.

III.4. Propriétés d'un réseau de neurones

Les réseaux de neurones sont pourvus trois grandes propriétés :

- ❖ Approximation de fonction
- ❖ Classification
- ❖ Apprentissage

En tant qu'approximation, le réseau de neurones est une méthode qui présente les qualités d'approximation universelle comme les méthodes de polynômes, de séries de Fourier. Ces méthodes permettent par régression d'approcher une fonction à un degré de précision fixé. L'apport du réseau de neurones sur les autres méthodes est d'effectuer une **approximation parcimonieuse**, c'est à dire, que cette méthode est capable d'approcher une fonction pour une précision donnée avec moins de paramètres que les approches usuelles.

Les réseaux de neurones sont capables de **classifier**. En fait, cette propriété découle de la capacité d'approximation. Si on possède un approximateur pour deux fonctions (f et g) qui est basé sur une identification du couple entré e-sortie, alors le degré d'appartenance d'un nouveau couple entrée-sortie est directement lié à l'erreur faite par l'approximateur : plus cette

erreur est grande moins cet approximateur est adapté et donc moins ce couple entrée-sortie appartient à la fonction.

L'**apprentissage** apporte des propriétés d'adaptabilité des réseaux de neurones. Ainsi une même structure peut approcher une multitude de fonctions grâce à l'apprentissage.

L'apprentissage réside en une procédure qui actualise les poids du réseau de neurones pour atteindre un objectif. Dans le cas d'un approximateur l'objectif sera la précision du modèle [34].

III.5. Apprentissage

L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré.

L'apprentissage neuronal fait appel à des exemples de comportement [29].

III.5.1. Apprentissage supervisé

L'apprentissage est dit supervisé lorsque les exemples sont constitués de couples de valeurs du type : (valeur d'entrée, valeur de sortie désirée). Tout le problème de l'apprentissage supervisé consiste, étant donné un ensemble d'apprentissage E de N couples (entrée - sortie désirée) (x_i, y_i) $i = 1, 2, \dots, n$, à déterminer le vecteur des poids w d'un réseau F_w capable de mettre ces informations en correspondance, c'est à dire un réseau tel que : $F_w(x_i) = y_i$ avec $i=1,2,\dots, n$ [35].

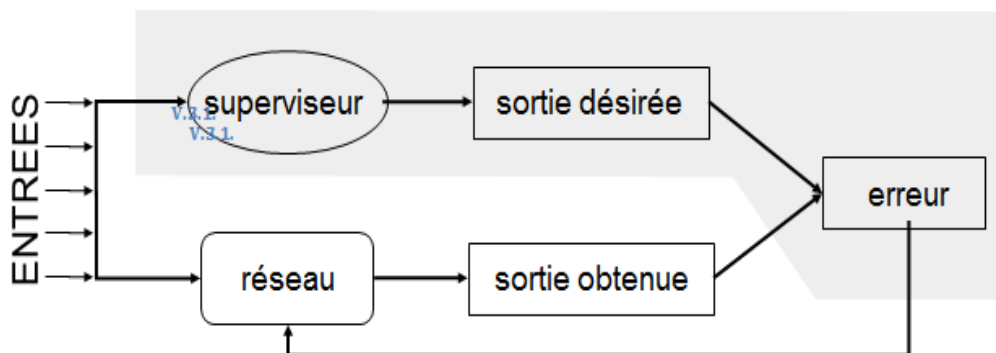


Figure III.9. Apprentissage supervisé

III.5.2. Apprentissage non supervisé

L'apprentissage est qualifié de non supervisé lorsque seules les valeurs d'entrée sont disponibles. Dans ce cas, les exemples présentés à l'entrée provoquent une auto-adaptation du

réseau afin de produire des valeurs de sortie qui soient proches en réponse à des valeurs d'entrée similaires (de même nature) [35].

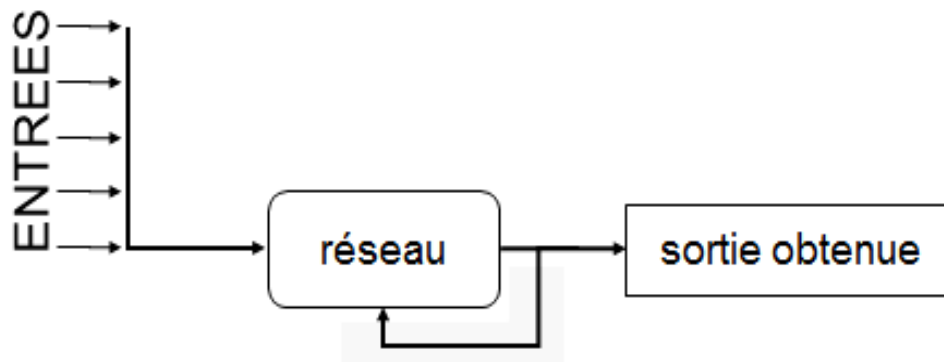


Figure III.10. Apprentissage non supervisé

III.6. Modèles des réseaux de neurones

III.6.1. Modèle de Hopfield

Le modèle de **Hopfield** fut présenté en 1982. Ce modèle très simple est basé sur le principe des mémoires associatives.

C'est d'ailleurs la raison pour laquelle ce type de réseau est dit associatif (par analogie avec le pointeur qui permet de récupérer le contenu d'une case mémoire).

Le modèle de Hopfield utilise l'architecture des réseaux entièrement connectés et récurrents (dont les connexions sont non orientées et où chaque neurone n'agit pas sur lui-même). Les sorties sont en fonction des entrées et du dernier état pris par le réseau.

III.6.2. Modèle Kohonen

Ce modèle a été présenté par Kohonen en 1982 en se basant sur des constatations biologiques. Il a pour objectif de présenter des données complexes et appartenant généralement à un espace discret de grandes dimensions dont la topologie est limitée à une ou deux dimensions.

Les cartes de Kohonen sont réalisées à partir d'un réseau à deux couches, une en entrée et une en sortie. Notons que les neurones de la couche d'entrée sont entièrement connectés à la couche de sortie. (figure.11)

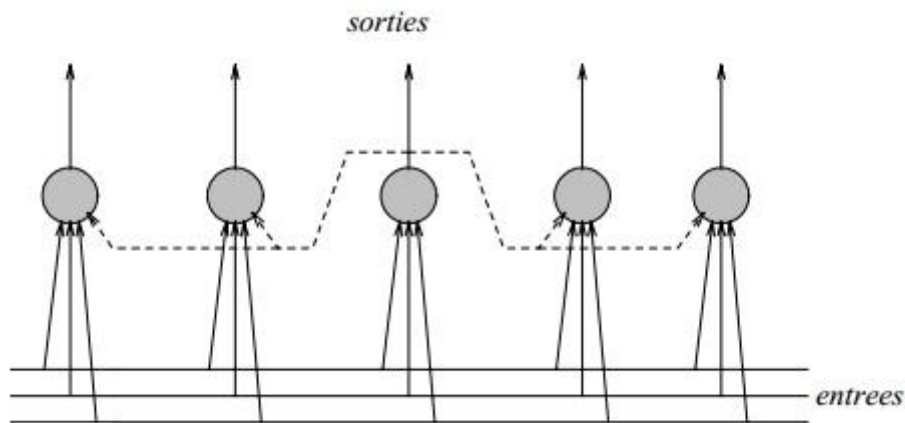


Figure III.11. Le modèle de Kohonen

III.6.3. Les Perceptrons Multicouches (PMC)

Un réseau de neurones est un ensemble de processeurs élémentaires, les neurones qui sont largement connectés les uns aux autres et qui sont capables d'échanger des informations au moyen des connexions qui les relient. Les connexions sont directionnelles et à chacune d'elle est associé un réel appelé poids de la connexion. L'information est ainsi transmise de manière unidirectionnelle du neurone j vers le neurone i , affectée du coefficient pondérateur w_{ij} . Un neurone calcule son état à partir d'informations venues de l'extérieur, ou bien il détermine son entrée à partir des neurones auxquels il est connecté et calcule son état comme une transformation souvent non linéaire de son entrée. Il transmet à son tour son état vers d'autres neurones ou vers l'environnement extérieur.

Un neurone est donc défini par trois caractéristiques: son état, ses connexions avec d'autres neurones et sa fonction de transfert. Nous utiliserons dans la suite, les notations suivantes:

- O : l'ensemble des états possibles des neurones.
- O_i : l'état du neurone i , où $o_i \in O$
- f_i : la fonction de transfert associée au neurone i .
- s_i : l'entrée du neurone i , $s_i \in \mathbb{R}$
- w_{ij} : le poids de la connexion du neurone j vers le neurone i ; $w_{ij} \in \mathbf{W}$ l'ensemble des Poids du PMC [36].

Ainsi le neurone i recevant les informations de n_i neurones effectue l'opération suivante :

$$o_i = f(s_i) \text{ avec } s_i = \sum_{j=1}^{n_i} w_{ij} o_j$$

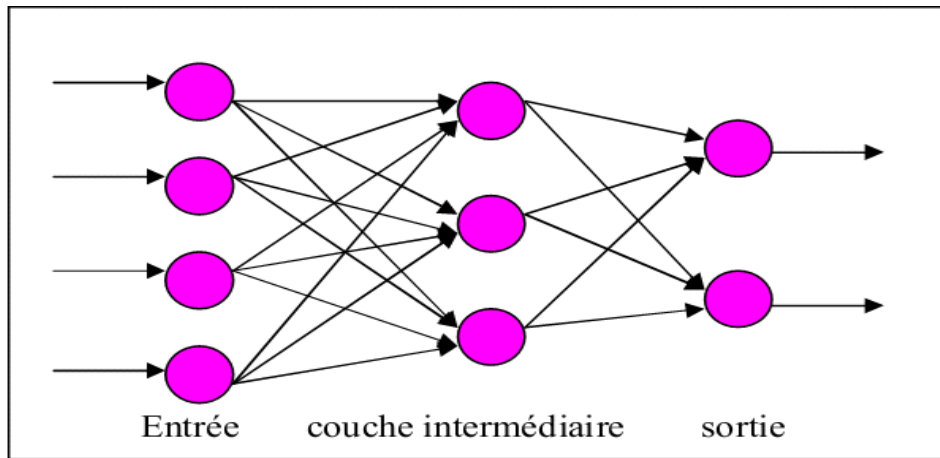


Figure III.12. Le modèle du perceptron

III.6.4. Le modèle Adaline

L'adaline (Adaptatif Linear Neurone) de Widrow et Hoff est un réseau à trois couches : une d'entrée, une couche cachée et une couche de sortie. Ce modèle est similaire au modèle de perceptron, seule la fonction de transfert change, mais reste toujours linéaire : $F(x) = x$ (voir figure III.13). Les modèles des neurones utilisés dans le perceptron et l'adaline sont des modèles linéaires pour la séparation linéaire : on dit que deux classes A et B, sont linéairement séparables si on arrive à les séparer par une droite coupant le plan en deux.

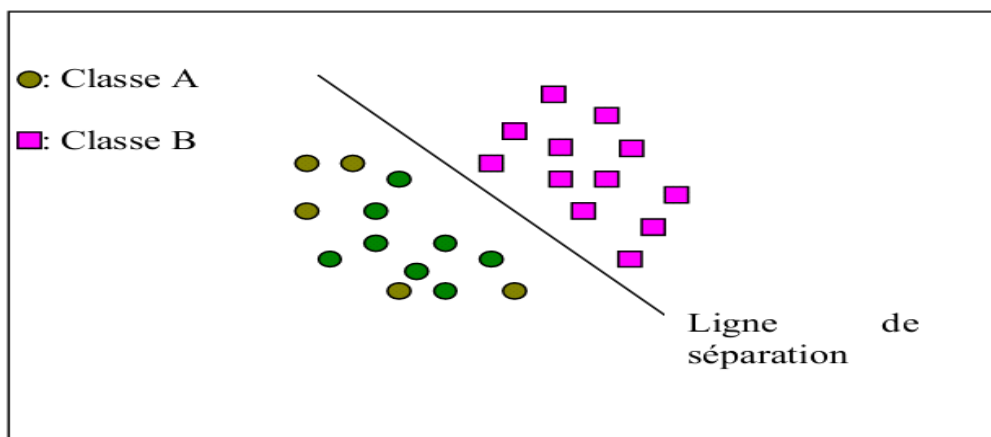


Figure III.13. La séparation linéaire entre la classe A et B

III.7. Règle d'apprentissage du Perceptron

L'algorithme d'apprentissage du Perceptron est semblable à celui utilisé pour la loi de Hebb. Les différences se situent au niveau de la modification des poids.

- 1) Initialisation des poids et du seuil S à des valeurs (petites) choisies au hasard.
- 2) Présentation d'une entrée $E_l = (e_1, \dots, e_n)$ de la base d'apprentissage.
- 3) Calcul de la sortie obtenue x pour cette entrée :

$$a = \sum (w_i \cdot e_i) - S$$

$$x = \text{signe}(a) \text{ (si } a > 0 \text{ alors } x = +1 \text{ sinon } a \leq 0 \text{ alors } x = -1 \text{)}$$

- 4) Si la sortie x du Perceptron est différente de la sortie désirée dl pour cet exemple d'entrée
Alors modification des poids (μ le pas de modification) :

$$w_i(t+1) = w_i(t) + \mu \cdot ((dl - x) \cdot e_i)$$

Rappel : dl = +1 si E est de la classe 1, dl = -1 si E est de la classe 2 et (dl - x) est une estimation de l'erreur.

- 5) Tant que tous les exemples de la base d'apprentissage ne sont pas traités correctement (i.e. Modification des poids), retour à l'étape 2 [37].

III.8. Description et principe

Un perceptron multicouche peut être considéré comme une généralisation du perceptron classique. Chaque neurone de la couche cachée calcule son activité en fonction des activités des neurones de la couche d'entrée. Puis les neurones de la couche cachée suivante calculent leur activité en fonctions de la couche précédente, et ainsi de suite jusqu'à la couche de sortie [38].

III.9. Classification par le MLP

III.9.1. Architecture

Un MLP (Multi Layer Perceptron) est composé d'une couche d'entrée, une couche de sortie et une ou plusieurs couches cachées [39].

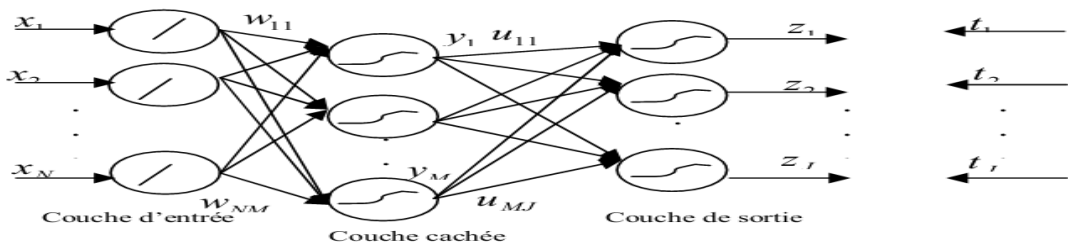


Figure III.14. Présentation schématique d'un MLP.

III.9.2. Phase d'apprentissage

L'apprentissage du perceptron multicouche est supervisé, et consiste à adapter les poids des neurones de manière à ce que le réseau soit capable de réaliser une transformation donnée, représentée par un ensemble d'exemples constitué d'une suite de N vecteurs d'entrées

$\mathbf{X}_k = [\mathbf{x}_{k1}, \mathbf{x}_{k2}, \dots, \mathbf{x}_{kd}]^T$ associée à une autre suite de vecteurs de sorties désirées

$\mathbf{T}^{(k)} = [t_1^{(k)} \quad t_2^{(k)} \quad \dots \quad t_{h_l}^{(k)}]$ Le critère des moindres carrés de l'erreur peut être utilisé pour définir la fonction de coût à minimiser, qui s'exprime alors :

$$E = \frac{1}{2} \sum_{k=1}^N \sum_{i=1}^{h_L} \left(y_{L,i}^{(k)} - t_i^{(k)} \right)^2$$

Où:

- N est le nombre d'exemples d'apprentissage;
- L est le nombre de couches du réseau;
- h_l vaut le nombre de neurones que contient la couche l ;
- $y_{l,i}^{(k)}$ désigne la sortie du neurone i de la couche l lorsque le vecteur \mathbf{X}_k est présenté à l'entrée du réseau;
- $t_i^{(k)}$ représente la valeur de sortie désirée pour le neurone i de la dernière couche lorsque le vecteur \mathbf{X}_k est présenté à l'entrée du réseau [37].

III.10. Conclusion

La technique de compression réversible est la préférable pour photos aériennes qui est basée sur les réseaux de neurones développés et mis en application en langage C. Ces méthodes ont pris essentiellement leur essor dans le cadre des approches d'apprentissage automatique en intelligence artificielle, pour obtenir un taux de compression élevé, sans perte d'information importante et pas d'influence.

Chapitre IV : L'application et les résultats

IV.1. Introduction	45
IV.2. Environnement de développement	45
IV.2.1. Les ressources physiques	45
IV.2.2. Les ressources logicielles	45
IV.3. Description de l'application	46
IV.4. Description du travail réalisé	48
IV.5. Phase des tests & résultats	51
IV.5.1. Prétraitement	51
IV.5.2. Base Image	51
IV.5.3. Résultats	52
IV.5.3.1. Test N°1	53
IV.5.3.2. Test N° 2	54
IV.5.3.3. Test N° 3	55
IV.5.4. Discussion	56
IV.6 Conclusion	57
Conclusion générale	58
Bibliographié	59

IV.1. Introduction

Dans ce travail, nous avons proposé une nouvelle approche de compression appliquée sur les photos aériennes.

Dans ce chapitre, on va expliquer l'approche de compression en détaillant les différentes étapes par laquelle une image passe avant d'être compressée, ainsi les différents algorithmes utilisés et l'environnement de programmation, illustrant des résultats de traitement en se basant sur des analyses et des tests.

Terminons par une conclusion qui englobe le fruit du travail.

IV.2. Environnement de développement

Pour l'implémentation de notre algorithme établi dans ce chapitre on a utilisé les ressources suivantes :

IV.2.1. Les ressources physiques

- ❖ Processeur Intel® Core (TM) i3-2370M CPU @ 2.40Hz
- ❖ Une mémoire vive d'une capacité de 4.00Go

IV.2.2. Les ressources logicielles

- ❖ Système d'exploitation : Windows 7
- ❖ Editeur utilisé est le C++ Builder Version 6

Notre logiciel est écrit en Borland C++ Builder version 6 sous Windows. Le choix de la programmation sous Windows a été pris à cause de l'interface graphique qu'offre l'environnement Windows ; le choix du langage Borland C++ Builder 6 était fondamental puisqu'il renferme des outils très puissants pour traiter des données de grande taille telle que les images.

C++ est un outil RAD, c'est à dire tourné vers le développement rapide d'applications (Rapid Application Development) sous Windows. En un mot, C++ Builder permet de réaliser de façon très simple l'interface des applications et de relier aisément le code utilisateur aux événements Windows, quelle que soit leur origine (souris, clavier, événement système, etc.)

C++ Builder repose sur un ensemble très complet de composants visuels prêts à l'emploi. La quasi-totalité des contrôles de Windows (boutons, boîtes de saisies, listes déroulantes, menus et autres barres d'outils) y sont représentés, regroupés par famille. Leurs caractéristiques sont

éditables directement dans une fenêtre spéciale intitulée éditeur d'objets. L'autre volet de cette même fenêtre permet d'associer du code au contrôle sélectionné.

Il s'agit de programmation « événementielle », des séquences de programme sont exécutées, suite à des actions de l'utilisateur (clique, touche enfoncée, etc...), détectées par Windows 6.

Notre application est une application de traitement d'image, et le C++ Builder donne des résultats satisfaisants concernant ce thème.

C++ Builder possède un langage de programmation à la fois puissant et simple d'utilisation, il permet d'exprimer les problèmes et les solutions d'une façon aisée.

Il s'impose dans le monde universitaire et industriel comme un outil puissant de réalisation des applications interactives.

C++ Builder intègre des outils très puissants pour générer des applications de base de données. Dans notre application nous avons opté pour le SGBD PARADOX lié à Borland C++ Builder 6.

IV.3. Description de l'application

On peut résumer les étapes d'application par un Schéma générale qui va représenter wxl'architecture de notre programme.

Pour qu'une photo aérienne soit compressée par notre application, elle doit passer par les étapes présentes dans l'organigramme (voir la figure IV.1), tel qu'elle doit en premier lieu être ajuster, puis passer par l'apprentissage supervisé, pour former l'image compressée.

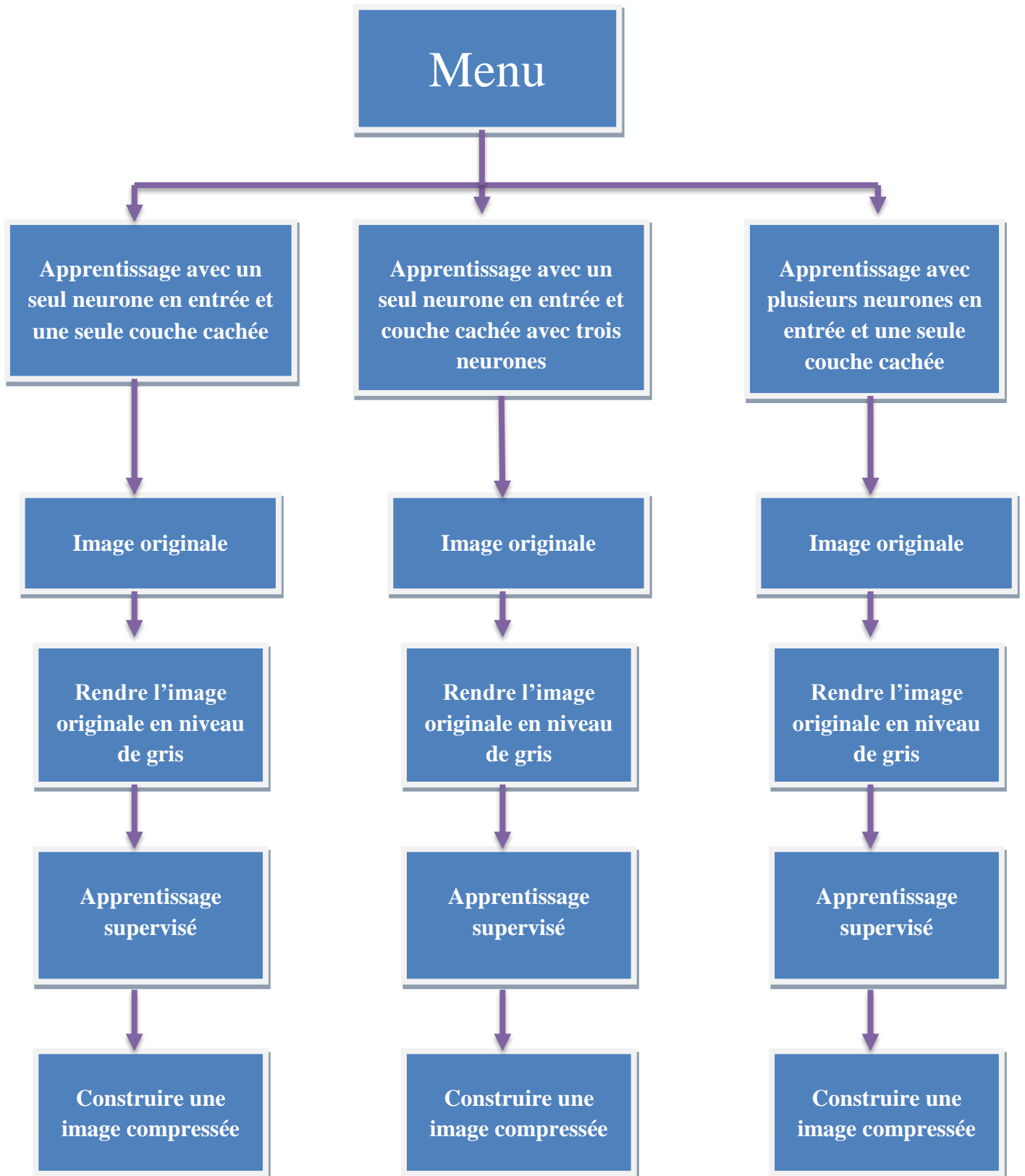


Figure IV.1 Schéma général de l'application

IV. 4. Description du travail réalisé

Cette partie est consacrée à la description de la phase de la réalisation et d'implémentation de ce projet, on va donc, montrer les différents modules de notre application afin d'illustrer plus clairement les diverses utilisations de l'application.



Figure IV.2 Fenêtre principale

En appuyant sur le bouton OK, la fenêtre ci-dessous va être affichée (voir Figure IV.3), c'est la fenêtre qui permet de choisir l'action voulue.

Tous D'abord il faut ouvrir une image en choisissant Ouvrir Image, et puis ouvrir une Photo aérienne qui va être affichée à la fenêtre la plus à gauche (la taille maximum 180x180), l'emplacement de l'image ouvrante sera affichée en tête de l'image (haut), et la taille à côté. Comme on peut prendre des informations sur les valeurs (Rouge, Vert et Bleu) d'un pixel en déplaçant la souris sur l'image.

Si la photo en couleur on peut la convertir en niveau de gris, et on peut lire la valeur du pixel (gris).

Si in décide de cliquer sur Apprentissage, une image compressée va être affichée, et chaque fois qu'on lance la compression le taux de compression s'affiche.

Notre application est constituée de trois parties distinctes, la première est consacrée à l'apprentissage supervisé avec un seul neurone d'entrée et une seule couche cachée. La deuxième partie est réservée à l'apprentissage supervisé avec un seul neurone d'entrée avec trois neurones en couches cachées et, la dernière partie est réservée à notre modèle de réseau de neurones avec plusieurs neurones en couche d'entrée et un seul neurone en couche cachée.

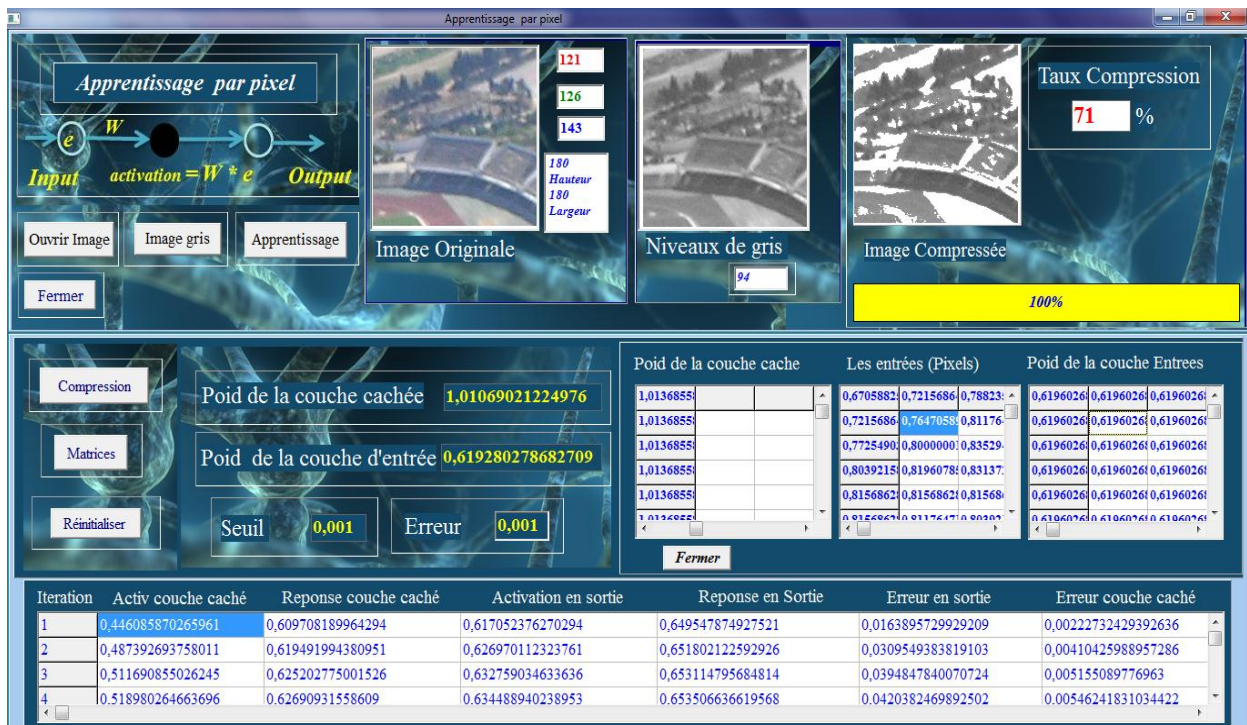


Figure IV.3 Fenêtre des actions N° 1

On peut également afficher des calculs visant la valeur d'activation de la couche cachée, la réponse de la couche cachée la valeur d'activation en sortie, la réponse en sortie, l'erreur en sortie et l'erreur de la couche cachée dans un tableau, et dans la deuxième grille s'affiche des calculs concernant les poids de la couche cachée et, les valeurs de chaque neurone en entrée. Chaque fenêtre contient des champs de saisir des poids de la couche cachée et les poids de la couche d'entrée.

Comme il y a la possibilité d'enregistrer l'image compressée par les réseaux de neurones.

Remarque : la valeur d'entrée de chaque neurone c'est la valeur de pixel en niveau de gris.

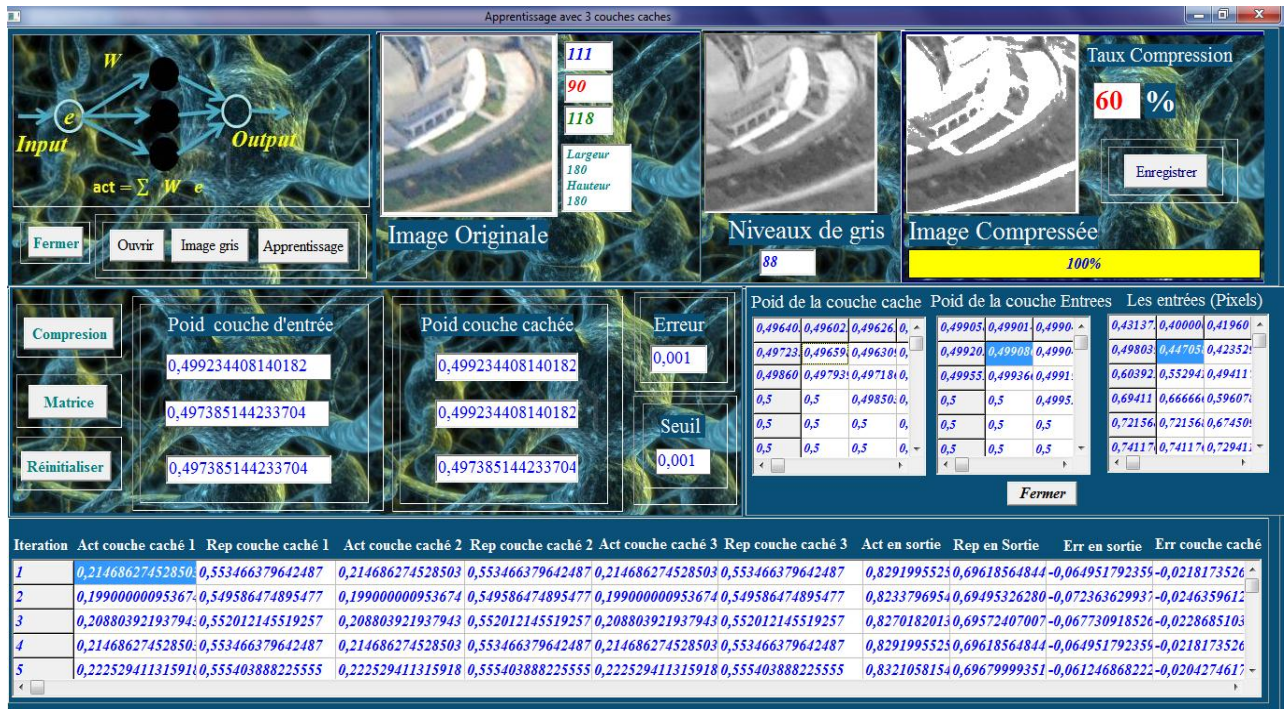


Figure IV.4 Fenêtre des actions N° 2



Figure IV.5 Fenêtre des actions N° 3

IV.5. Phase des tests & résultats

IV.5.1. Prétraitement

Un résultat avec un taux de compression satisfaisant est relié par un choix des poids (entrée /cachée), et avec une erreur qui optimise l'apprentissage supervisé.

Cette étape est appelée le prétraitement, qui est décomposé en deux phases :

- ✓ **Phase d'apprentissage** : elle nécessite la constitution d'une base d'exemples, son but est d'entraîner le réseau de neurones, afin qu'il ajuste ses paramètres (poids).
- ✓ **Phase de reconnaissance** : elle consiste seulement à présenter les valeurs d'entrée (poids de la couche cachée et entrée) au réseau de neurones qui permettent de donner la valeur de sortie en fonction des poids ajustés en cours de la phase d'apprentissage.

Après plusieurs tâtonnements nous avons fixé les valeurs des poids de la couche cachée, la couche d'entrée le seuil et l'erreur (voir table IV.1)


Image	Dimension (pixels)	Poids couche cachée	Poids couche d'entrée	Seuil	Erreur
	10 * 10	1,0136855840683	0,619602680206299	0,001	0,001

Table IV.1 Résultats optimaux

IV.5.2 Base Image

Pour manipuler des images de différentes tailles, les dimensions des images vont être ajustées pour être à la puissance de deux. Par exemple, les dimensions de l'image 1720 × 1450 vont être ajustées pour être 1440 x 1440.

Dans les trois méthodes d'apprentissage on a choisi trois types de taille variable (grande 180 x 180, moyenne 90x 90 et petite 45 x45), et formats bitmaps (.Bmp).

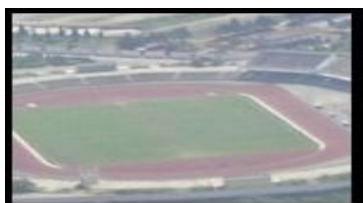


Figure.IV.6. Dimension 180x180 pixels

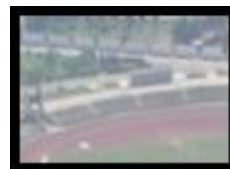


Figure.IV.7. Dimension 90x90 pixels

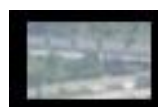


Figure.IV.8. Dimension 45x45 pixels

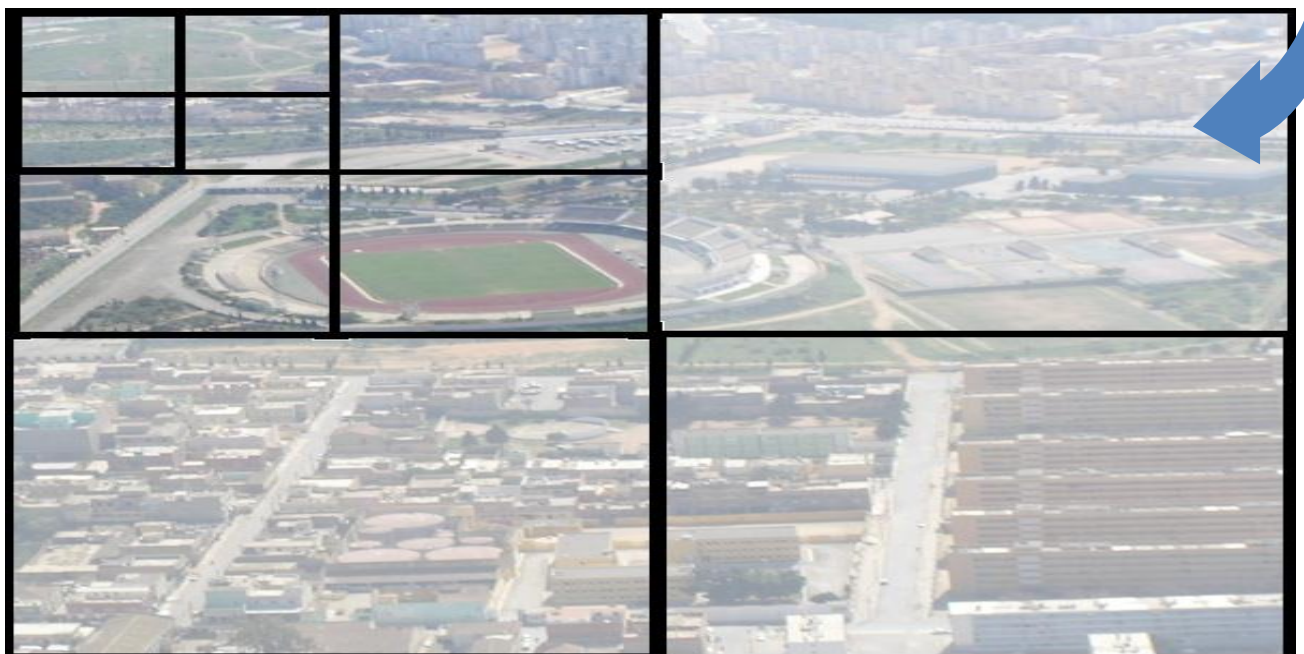
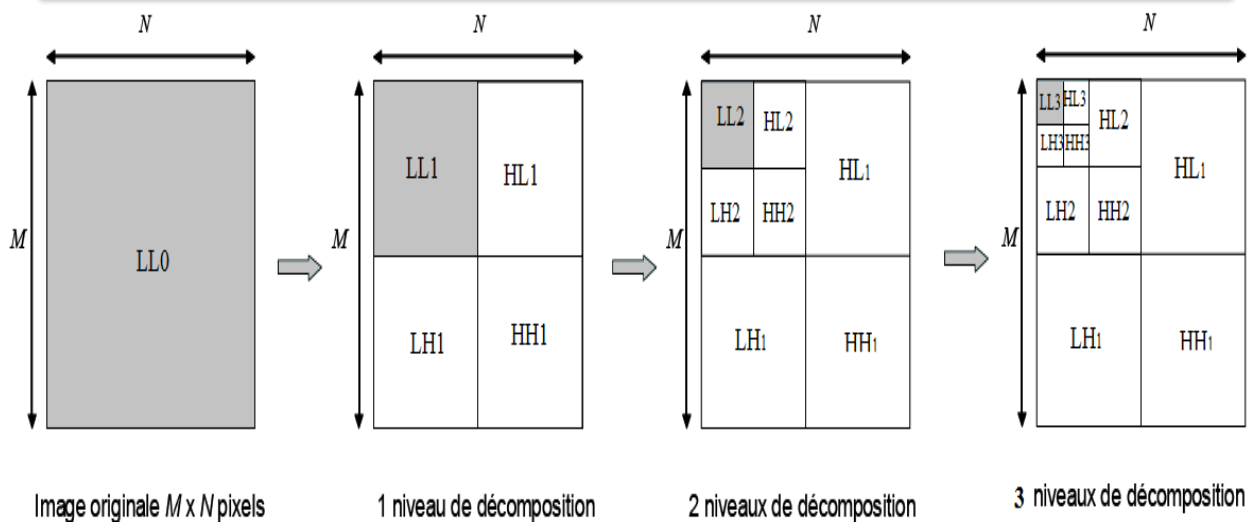


Figure IV .9. Décomposition de la photo aérienne (région Mostaganem)

IV.5.3 Résultats

1. Chargement de l'image originale (taille 180 x 180 max) en couleur (Rouge, Vert, Bleu) ;
2. Rendre l'image originale au niveau de gris ;
3. Apprentissage du réseau de neurones;
4. Calcul de l'erreur quadratique entre la sortie désirée et la sortie obtenue ;
5. Répétition des étapes 1, 2, 3 sur une nouvelle image avec une taille moins que la première (90 x 90pixels / 45 x 45pixels).

IV.5.3.1. Test N°1

Dans ce cas, on va utiliser un apprentissage avec un seul neurone et une seule couche cachée


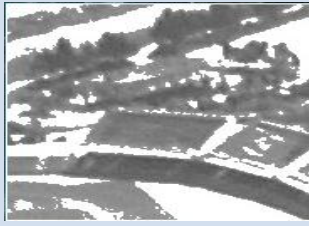
Image Originale	Image Compressée	Taux compression	Temps 'H : M : S : MS'
		71%	00 : 1 : 56 : 10

Table IV.2 Résultat obtenu pour l'image test N°1 de 180x180 pixels

- ✓ Nombre de pixels de l'image originale = 32 400 pixels
- ✓ Nombre de pixels de l'image compressée = 23026 pixels


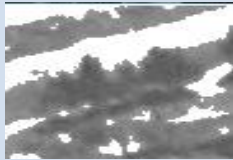
Image originale	Image compressée	Taux Compression	Temps 'H : M : S : MS'
		72%	00 : 00 : 31 : 18

Table IV.3 Résultat obtenu pour l'image test N°1 de 90x90 pixels

- ✓ Nombre de pixels de l'image originale = 8100 pixels
- ✓ Nombre de pixels de l'image compressée = 5878 pixels

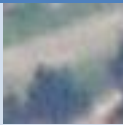

Image originale	Image compressée	Taux Compression	Temps 'H : M : S : MS'
		60%	0 : 00 : 09 : 31

Table IV.4 Résultat obtenu pour l'image test N°1 de 45x45 pixels

- ✓ Nombre de pixels de l'image originale = 2025 pixels
- ✓ Nombre de pixels de l'image compressée = 1217 pixels

IV.5.3.2. Test N° 2

Dans ce cas, on va utiliser un apprentissage avec un seul neurone et trois couches cachées :



Image originale	Image compressée	Taux Compression	Temps 'H : M : S : MS'
		97%	00 :00 :19 :75

Table IV.5 Résultat obtenu pour l'image test N°2 de 180x180 pixels

- ✓ Nombre de pixels de l'image originale = 32 400 pixels
- ✓ Nombre de pixels de l'image compressée = 31 543 pixels



Image originale	Image compressée	Taux Compression	Temps 'H : M : S : MS'
		98%	00 :00 :06 :40

Table IV.6 Résultat obtenu pour l'image test N°2 de 90x90 pixels

- ✓ Nombre de pixels de l'image originale = 8100 pixels
- ✓ Nombre de pixels de l'image compressée = 7997 pixels



Image originale	Image compressée	Taux Compression	Temps 'H : M : S : MS'
		99%	00 :00 :01 :35

Table IV.7 Résultat obtenu pour l'image test N°2 de 45x45 pixels

- ✓ Nombre de pixels de l'image originale = 2025 pixels
- ✓ Nombre de pixels de l'image compressée = 2022 pixels

IV.5.3.3. Test N° 3

Dans ce cas, on va utiliser un apprentissage avec plusieurs neurones en entrée et un seul neurone en couche cachée:



Image originale	Image compressée	Taux Compression	Temps 'H : M : S : MS'
		89%	00 :00 :07 :60

Table IV.8 Résultat obtenu pour l'image test N°3 de 180x180 pixels

- ✓ Nombre de pixels de l'image originale = 32 400 pixels
- ✓ Nombre de pixels de l'image compressée = 28 924 pixels


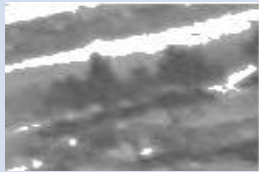
Image originale	Image compressée	Taux Compression	Temps 'H : M : S : MS'
		89%	00 :00 :03:82

Table IV.9 Résultat obtenu pour l'image test N°3 de 90x90 pixels

- ✓ Nombre de pixels de l'image originale = 8100 pixels
- ✓ Nombre de pixels de l'image compressée = 7254 pixels



Image originale	Image compressée	Taux Compression	Temps 'H : M : S : MS'
		75%	00 :00 :02 :74

Table IV.10 Résultat obtenu pour l'image test N°3 de 45x45 pixels

- ✓ Nombre de pixels de l'image originale = 2025 pixels
- ✓ Nombre de pixels de l'image compressée = 1537 pixels

IV.5.4. Discussion

Malgré l'identité de la taille pour les trois tests, il est clair que le taux et le temps de compression est différent. Ces résultats sont interprétés par les graphes ci-dessous pour visualiser la différence plus clairement.

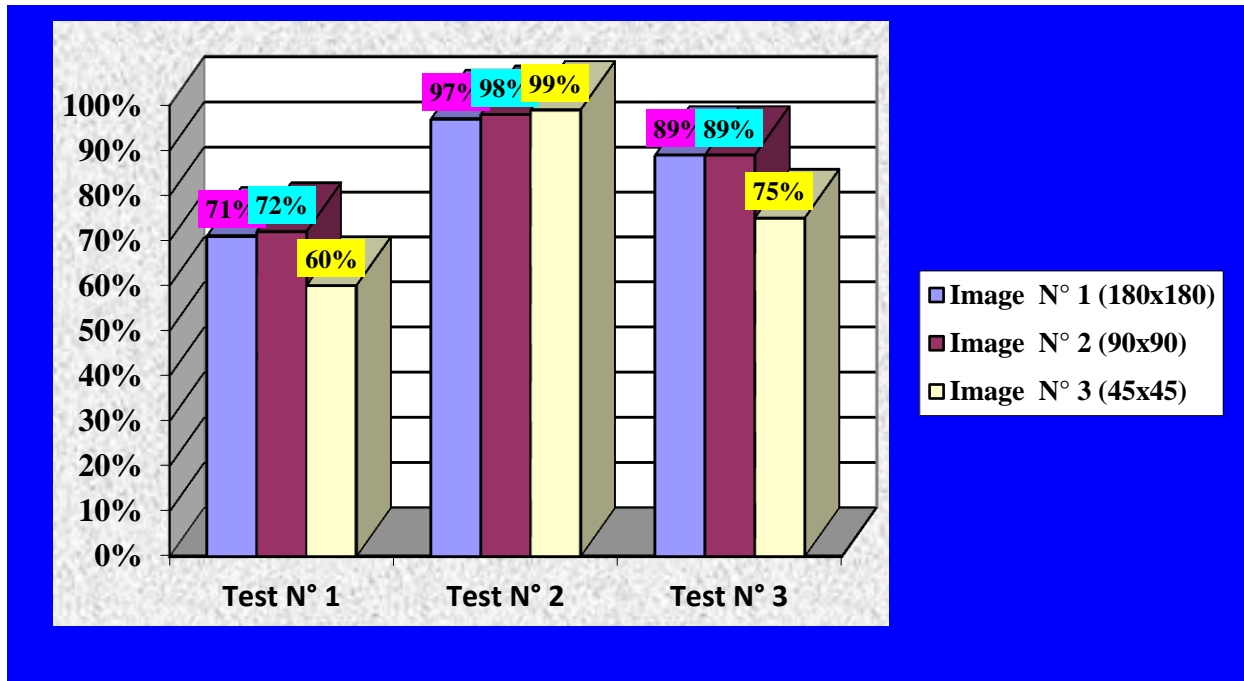


Figure IV.10 Taux de compression dans les trois tests

Nous remarquons dans le graphe (figure IV.10) que, les résultats de test N°2 (Apprentissage avec un seul neurone et trois couches cachées) est meilleur que les autres tests.

Les résultats présentés dans le test N°3 (Apprentissage avec plusieurs neurones en entrée et un seul neurone en couche cachée), soulignent la rapidité de la phase de codage de l'algorithme proposé. Cela revient essentiellement à la rapidité de construction des neurones en entrée permettent de construire un réseau plus rapide.

Dans notre application, le cas où le taux de compression est égale à 100 % (pas d'apprentissage), dans ce cas les valeurs de la couche cachée et de la couche d'entrée et même la valeur du seuil et l'erreur, sont égaux.

IV.6 Conclusion

En général, on peut dire que l'approche proposée permet d'atteindre de très bons résultats de point de vue qualité des images reconstruites d'une part et le taux de compression d'une autre part. En effet, elle permet de réduire énormément la quantité de données contenues dans les photos aériennes tout en conservant leurs propriétés originales. Ces résultats dépendent du type de réseaux de neurones, le code d'apprentissage supervisé et du nombre des couches cachées utilisées. Les résultats de l'apprentissage dépendent du choix des neurones et des tailles des images.

Le choix des réseaux de neurones pour la compression porte essentiellement sur la conservation de l'information, l'erreur ou/et le seuil utilisé. On peut utiliser une autre photo aérienne pour test, avec un format bitmap.

Ainsi, nous avons conclu que l'identité des tailles pour certaines images n'implique pas qu'elles doivent avoir le même taux de compression.

Nous sommes convaincus que la clé de succès dans ce travail est de chercher toujours à optimiser les différentes fonctions pour avoir un temps de réponse plus ou moins adaptable.

Conclusion générale

Pour bien mener ce projet qui portait sur la compression des images numériques nous avons utilisé les réseaux de neurones.

L'objectif de notre travail est la compression des photos aériennes non seulement des informations importantes mais aussi critiques, car la perte d'une petite information peut conduire à des résultats catastrophiques, et comme la compression sert à modifier la représentation initiale des données, pour qu'elles occupent moins de place, nous avons préféré une compression sans perte d'information en se basant sur le réseau de neurone.

Nous proposons aussi de compléter notre application par un travail très important en intégrant la décompression des images compressées par le réseau de neurone qui sera une étape dans le sens inverse, sachant qu'une étape de la décompression a été incorporée dans la compression afin de minimiser le temps de décompression.

Pour améliorer toujours le taux de compression, une autre approche pourra être utilisée une décompression avec perte en se basant sur l'apprentissage, et l'aide d'un expert de son domaine.

Bibliographié

- [1] **C. Foucher et G Vaucher** « Compression d'images et réseaux de neurones » *Revue Valgo*, Valgo 2001, P01-02,19/12/2012 à 11 h54.
- [2] **M.Benabdellah** « Outils de compression et de crypto compression », thèse de Doctorat, Université mohammed V-Agdal Faculté des Sciences Rabat, Soutenue publiquement le 20 Juin 2007 à 16h30, p 40, N° d'ordre 2339 ,18/01/2013 à 00h04
- [3]**D. Cavet** numérisation des images,www.cndp.fr/notestech/13/numerima.htm, consulté le 18/01/2013à 00h04.
- [4] **J.P. Gastellu-Etchegorry** « Acquisition et Traitement d'image numérique», Avril 2008 Université Paul Sabatier IUT, cours.
- [5] **Y.Gaudeau**: « Contributions en compression d'images médicales 3D et d'images naturelles 2D », thèse de Doctorat de l'Université Henri Poincaré, spécialité automatique, traitement du signal et génie informatique, Soutenue le 13 décembre 2006. p1.
- [6] **X.Delaunay**: « Compression d'images satellite par post-transformées dans le domaine ondelettes », thèse de Doctorat de l'université de Toulouse, Soutenue le 12 novembre 2008, p9
- [7] **M.Midoun**: Spécialité en système d'information géographique, Cours source de donnée, 2ème année Master, Université de Abdelhamid Ibn Badis, Mostaganem, le 23/12/2012 à 10h10.
- [8] **B.Takarli**: Spécialité en système d'information géographique, Cours photogrammétrie, 1^{er} année Master, Université de Abdelhamid Ibn Badis, Mostaganem, le 15/04/2012.
- [9]**P. Vincent, L. Franck et H. Vincent** : « Compression de données », Décembre 2004, 17/01/2013 à 14h52.
- [10]**C. C. CUTLER** : « *Differential quantization of communication signals* », U.S. Patent, vol. 2, pages 605,361, 29 july 1952.
- [11] **C. W. HARRISON** :Experiments with linear prediction in television, Bell System Tech. J., vol. 31, pages 764–783, 1952.
- [12]**D. A. HUFFMAN** : A method for the construction of minimum redundancy codes, IRE Proc., vol. 40, pages 1098–1101, 1952.
- [13] **D. GABOR** :Theory of communication, J. IEEE, vol. 93, pages 429–457, 1946.
- [14] **C. E. SHANNON** : A mathematical theory of communication, Bell System Tech. J., vol. 27, pages 379–423 and 623–656, 1948.

- [15] **J. J. Y. HUANG et P. M. SCHULTHEISS** :Block quantisation of correlated Gaussian random variables, IEEE Trans. Commun. Vol. COM-11, pages 289–296, 1963.
- [16] **W. K. PRATT, J. KANE et H. C. ANDREWS**: Hadamard transform image coding, IEEE Proc., vol. 57, pages 58–68, 1969.
- [17] **J. O. LIMB** : Source-receiver encoding of television signals, IEEE Proc., vol. 55, pages 364–379, 1967.
- [18] **D. E. PEARSON** : A realistic model for visual communication systems, IEEE Proc., vol. 55, pages 380–389, 1967.
- [19] **W. K. SCHREIBER** : Picture coding, IEEE Proc., vol. 55, pages 320–330, 1967.
- [20] **G. S. KANG et D. C. COULTER** : 600 bps voice digitizer, in ICASSP Proc., pages 91–94, 1976.
- [21] **A. GERSHO et B. RAMAMURTHI** : Image coding using vector quantization, in ICASSPProc., pages 428–431, 1982.
- [22] **M. BARNSLEY et A. D. SLAON** : A better way to compress images, BYTE Magazine, pages 215–223, january 1988.
- [23] **A. N. NETRAVALI et J. D. ROBBINS** : Motion-compensated television coding : Part 1, Bell System Tech. J., vol. 58, pages 631–670, 1979.
- [24] **C. WAGNER** : « De l’image vers la compression », Rapport de Recherche de l’INRIA, Septembre 1993.
- [25] **MM. S.Maadi, Y.Peneveyre, et C.Lambercy** : « Compression de données sans pertes », œuvre, Services de Téléinformatique, Chapitre 2, p25.
- [26] **A.Martin** :« Représentations parcimonieuses adaptées à la compression d’images », thèse de Doctorat, Université de Rennes 1, soutenue à l’IRISA le 2 Avril 2010, p7, N° d’ordre 4117.
- [27] **L. Diane** «Cours de Traitement d’Images », Rapport de recherche, Centre National de la Recherche Scientifique, 22 Janvier2004, p180, ISRN I3S/RR–2004-05–FR.
- [28] **H. Meraoubi, Z. Brahimi, K.Ait Saadi et A. Zemouri** « Un Système de Crypto-Compression des Images Médicales Basé sur la DCT 2x2-IDS et l’AES », Article, Centre de Développement des Technologies Avancées, p2.
- [29] **C.Touzet** : Les réseaux de neurones artificiels, introduction au connexionnisme, cours et travaux pratiques, juillet 1992, l’université d’Aix-Marseille, p6.
- [30] **CHIKH Mohammed Tahar** : « Amélioration des images par un modèle de réseau de neurones (Comparaison avec les filtres de base) »,Mémoire de fin d’études pour l’obtention

du diplôme de Master en Informatique ;Université Abou-Bakr Belkaid –Tlemcen- ; soutenue le 28 Septembre 2011 ; P31.

[31] F.El Hannani et L .Koudacke : « Élaboration d'un système de détection et de reconnaissance des visages & identification des émotions faciales », Université de Sidi Bel Abess, 2002.

[32]G. DREYFUS : « LES RÉSEAUX DE NEURONES », École Supérieure de Physique et de Chimie Industrielles de la Ville de Paris (ESPCI), Laboratoire d'Électronique, PARIS.

[33] P. Quang Dung : « RÉSEAUX DE NEURONES POUR LA RECONNAISSANCE DES FORMES », Hanoi, 15 juillet 2005.

[34] L. SAINT-BAU ZEL : « De la modélisation prédictive du comportement pathologique à l'application dans l'interaction Robot-Patient », THÈSE de DOCTORAT, L'UNIVERSITÉ PARIS 6, Soutenue le 14 décembre 2007.

[35] B.Kaid Slimane: «Réseau de neurones, Cour, Reconnaissances de formes, 1ème année Master, spécialité Système d'information géographique, Université de Abdelhamid Ibn Badis, Mostaganem, 2013.

[36] F. BADRAN et S. THIRIA : « Les Perceptrons Multicouches de la régression non-linéaire aux problèmes inverses », Laboratoire d'Océanographie DYnamique et de Climatologie, PARIS, France.

[37] B.Gosselin : « Application De Réseaux De Neurones Artificiels A La Reconnaissance Automatique De Caractères Manuscrits », Thèse de Doctorat, Faculté Polytechnique de Mons, 1995-1996.

[38] S.Crahay et T.viéville : « Simulation neuronale de la vision précoce corticale avec un modèle de Heeger », Institut national de recherche en informatique et automatique, Septembre 2002.

[39] D.Rumelhart et E. McClelland, Parallel Distributed Processing, Explorations in the Microstructure of cognition. I & II, Cambridge: MIT Press ;Bishop C.M., Neural Networks for Pattern Recognition, Clarendon Press OXFORD 1995.