N° d'ordre : 143 / 2017

UNIVERSITE SULTAN MOULAY SLIMANE

Faculté des Sciences et Techniques

Béni-Mellal

Centre d'Études Doctorales : Sciences et Techniques
Formation Doctorale : Mathématiques et Physique Appliquées (MPA)

THÈSE

Présentée par

HOUDA CHAKIB

Pour l'obtention du grade de
DOCTEUR

Spécialité : Traitement de signal, Electronique

Contribution à la Compression d'Images par la Transformation en Ondelettes et les Réseaux de Neurones Artificiels

Soutenue le 16/12/2017 à 9h30 devant la commission d'examen composée de :

Président	:	Pr. Mohammed Sajieddine	PES	FST - Béni Mellal
Rapporteurs	:	Pr. Nejma Fazouan	PES	FST - Mohammedia
		Pr. Najlae Idrissi	PH	FST - Béni Mellal
Directeur de thèse	:	Pr. Brahim Minaoui	PES	FST - Béni Mellal
Co-Directeur de thèse	:	Pr. Mohamed Fakir	PES	FST - Béni Mellal

Remerciements

Cette étude a été réalisée à la Faculté des Sciences et Techniques de l'Université Sultan Moulay Slimane au sein du Laboratoire de Traitement de l'Information et Aide à la Décision (TIAD), dirigé par le Professeur M. Fakir.

Ce travail a été mené sous la direction du Professeur **B. Minaoui**, et je tiens à lui exprimer mes plus vifs remerciements pour son suivi, sa disponibilité et pour les conseils scientifiques qu'il m'a prodigués tout au long de ce travail.

Je désire également remercier mon co-directeur le Professeur **M. Fakir**, pour sa collaboration, son encouragement et le soutien moral qu'il m'a apportés durant la réalisation de ce travail.

Je tiens à remercier en particulier, le Professeur **M. Sajieddine**, qui me fait l'honneur d'assurer la présidence du jury d'examen. Qu'il trouve en ces mots ma sincère gratitude.

Je tiens à exprimer mes plus sincères remerciements au Professeur **N.Idrissi**, qui a constamment suivi l'évolution de ce travail, et m'a fait profiter de sa riche expérience en traitement d'images. Et qui me fait l'honneur de faire partie de ce jury en tant que rapporteur.

Je remercie également le Professeur **N. Fazouan,** pour sa collaboration et pour l'honneur qu'elle me fait en acceptant de participer au jury en qualité de rapporteur.

Je ne saurais oublier mes collègues du département de physique qui n'ont pas cessé de m'encourager depuis le début de ce travail. Je voudrais remercier tout particulièrement **A. Salhi** pour les diverses discussions scientifiques enrichissantes.

Enfin, je tiens à exprimer mes vifs sentiments à mon mari, le Professeur **E.Agouriane**, pour ses encouragements tout au long de ce travail, son aide précieuse pour la rédaction de ce manuscrit et surtout pour sa présence et son soutien moral dans les moments difficiles.

Résumé

Dans le présent travail de cette thèse, nous avons élaboré un système de compression des images fixes intégrant des techniques de l'intelligence artificielle et de traitement du signal à savoir les réseaux de neurones et les transformées en ondelettes. Pour élaborer ce système, nous avons, dans un premier temps, étudié les performances de différents types de transformation en ondelettes en compression d'images en termes du Taux de Compression (TC), d'Erreur Quadratique Moyenne (EQM), du Nombre de Bits par Pixels (BPP) (NBP) et du Rapport Signal sur le Bruit Crête (PSNR). Cette étude a mis en évidence la dépendance de la qualité de la compression d'une image du type de la transformée en ondelettes adopté.

En se basant sur le résultat de cette étude, nous avons, dans un deuxième temps, développé un système de compression d'images basée sur une approche de compression qui consiste à utiliser un réseau de neurones multicouche entrainé à l'aide d'un apprentissage à rétro-propagation à partir d'une base de données images, pour sélectionner la transformée en ondelettes la plus appropriée pour la compression d'une image. L'association de ces deux techniques, classification par réseau de neurone et compression par transformation en ondelettes, a permis d'améliorer la compression d'images. Afin d'améliorer davantage les performances du système de compression développé, le réseau de neurones est entrainé, après modification de sa structure, à proposer en plus de la transformée en ondelettes appropriée, le taux optimal de compression convenable pour compresser une image sans perte d'information.

Une évaluation du système de compression d'images ainsi élaboré, réalisée sur un corpus d'images différent de celui utilisé au cours de l'apprentissage, a mis en évidence l'amélioration de la qualité de compression des images.

Des tests effectués sur un ensemble d'images ont mis en évidence que le système de compression ainsi élaboré permet bien d'améliorer la qualité de la compression des images.

Mots clés : compression d'images, transformation en ondelettes, réseau de neurones multicouche, apprentissage par rétro-propagation, compression sans perte d'information.

Abstract

In the work presented in this thesis, we have developed a system of compression of fixed images combining two techniques of artificial intelligence and signal processing namely neural networks and wavelet transforms. To develop this system, we first studied the performances of different types of wavelet transforms in images compression in terms of Compression Rate (CR), Mean Square Error (MSE), number of Bits Per Pixels (BBP) and Peak Signal to Noise Ratio (PSNR). This study highlighted the dependence of the compression quality of an image to the type of wavelet transform adopted.

Based on the result of this study, we have developed an image compression system based on a compression approach which consists in using a multilayer neural network, trained with a backpropagation learning algorithm from an image database, to select the most appropriate wavelet transform for compressing an image. The combination of these two techniques, classification by neuron network and compression by wavelet transformation, has made it possible to improve image compression. In order to further improve the performance of the developped compression system, the neural network is trained, after modification of its structure, to propose, in addition to the appropriate wavelet transform, the optimal compression ratio suitable for compressing an image with less information loss.

An evaluation of the image compression system thus developed, applied on a set of images different from that used during learning, has pouved the improvement of image compression quality.

Tests carried out on a set of images have shown that the image compression system thus developed managed to improve the quality of image compression.

Key words: Image compression, wavelet transformation, multilayer neural network, backpropagation learning, lossless compression.

ملخص

في الأعمال المعروضة في هذه الأطروحة، قمنا بتطوير نظام ضغط للصور الثابتة بدمج تقنيات الذكاء الإصطناعي و معالجة الإشارات نخص بالذكر الشبكات العصبية و المويجات. لتطوير هذا النظام، قمنا أولا بدراسة أداء أنواع مختلفة من المويجات في ضغط الصور من حيث معدل الضغط (CR) ، متوسط مربع خطأ (MSE) ، عدد بت لكل بكسل (BPP) و ذروة إشارة نسبة الضوضاء (PSNR). و أبرزت هذه الدراسة إرتباط جودة ضغط الصورة بنوع المويجة المعتمدة. وبناء على نتائج هذه الدراسة قمنا بتطوير نظام ضغط للصور يقوم على استخدام شبكة عصبية متعددة الطبقات تعتمد في تدريبها على نظام عكس الإنتشار، لتحديد أنسب مويجة لضغط صورة ما. الإستعمال المزدوج لهاتين التقنيات: التصنيف باستعمال شبكة الخلايا العصبية و الضغط عن طريق المويجة، جعل من الممكن ضغط الصور بشكل أفضل. ومن أجل زيادة تحسين أداء نظام الضغط المقترح، يتم تدريب الشبكة العصبية، بعد تعديلها، لإقتراح بالإضافة إلى تحديد المويجة المناسبة، معدل الضغط الأمثل لضغط الصورة دون ضياع للمعلومات

و قد تم تقييم نظام الضغط الذي تم تطويره باستعمال مجموعة من الصور مختلفة عن المستعملة أثناء تعليم الشبكة العصبية التي أبرزت تحسين جودة ضغط الصور. وقد أظهرت تلك الإختبارات، التي أجريت على مجموعة من الصور، أن نظام ضغط الصور الذي تم تطويره يجعل من الممكن تحسين جودة ضغط الصور

الكلمات الرئيسية: ضغط الصورة، تحويل المويجات، الشبكة العصبية متعددة الطبقات، التعلم بنظام عكس الإنتشار، ضغط مع الضياع .

Table des matières

Introduction

Contexte Général

"Qu'elles soient stockées numériquement dans des mémoires informatiques ou qu'elles voyagent à travers un système de transmission comme Internet, les images occupent beaucoup de place. Heureusement, il est possible de les « condenser » sans altérer leur qualité ! Une image numérisée se comprime, tout comme un jus d'orange que l'on réduit à quelques grammes de poudre concentrée. Il ne s'agit pas d'un tour de passe-passe, mais de techniques mathématiques et informatiques permettant de réduire la place occupée par une image dans un ordinateur ou dans un câble de communication. Elles sont aujourd'hui indispensables pour stocker de l'information ou la transmettre par Internet, téléphone, satellite ou tout autre moyen de transmission de données", **Stéphane Mallat.**

Au cours des dernières années, on assiste à un développement rapide des applications informatiques ce qui a entraîné une forte augmentation de l'utilisation des données numériques principalement dans le domaine du multimédia, de l'imagerie médicale, de la transmission par satellite, des jeux etc... Et malgré que de nos jours de hauts débits de données soient disponibles, la compression des images reste un outil nécessaire afin de les gérer et de les stocker efficacement, en réduisant la mémoire utilisée, par conséquence le coût de transmission. D'un autre côté, un système de compression d'image idéal doit être capable de reproduire une image avec un taux très élevé toute en gardant une haute qualité visuelle de celle-ci.

Problématique et objectif

De nos jours, des avancées considérables dans le domaine de la compression des données ont eu lieu grâce au développement de nouvelles techniques comme la théorie des ondelettes. Cette technique a été utilisée avec beaucoup de succès dans un large éventail d'applications grâce au nombre important de recherches faites dans ce domaine, recherches qui ont pu enrichir sa bibliothèque d'un nombre considérable de familles d'ondelettes. Cependant la question qui se pose est quelle fonction ondelette est plus appropriée pour compresser telle ou telle image, sachant que plusieurs études faites ont prouvé que le choix de l'ondelette a un impact significatif sur la qualité de la compression.

Les réseaux de neurones artificiels (RNA) tiennent leur intérêt principal de leur capacité d'apprentissage et de généralisation et ceci grâce à leur architecture de multicouche qui les rend très intéressants à plusieurs égards, en particulier la possibilité de les implanter sur des machines neuronales et leur résistance aux

pannes lors du traitement d'images en temps réel. Inspirée du fonctionnement du cerveau humain, cette technique n'a pas cessé de donner de bons résultats pour la résolution quasi-optimale de nombreux problèmes de grande complexité dans plusieurs domaines comme les télécommunications, l'aéronautique, la médecine etc… De plus, la combinaison entre cette technique et la transformation par les ondelettes s'est avérée être un outil précieux et fort pour le traitement de signal en général et pour la compression des images en particulier.

L'objectif principal de cette étude est donc l'élaboration d'un système de compression basé sur l'utilisation des réseaux RNA pour relier, après apprentissage, une image à la fonction d'ondelette qui convient le mieux à sa compression. Pour prendre cette décision, le système se base sur le contenu des valeurs des pixels de l'image qui seront présentées à l'entrée du réseau.

Organisation du mémoire

Ce rapport de thèse s'articule autour de 4 chapitres. Il est structuré comme suit :

Le premier chapitre est partagé en deux parties. La première partie fait un rapide survol de définitions utiles dans le domaine du traitement d'images telles que les notions d'image, pixel, histogramme, voisinage, contraste etc…La deuxième partie est consacrée à l'étude des techniques de base les plus utilisées en compression des données numériques en général et des images en particulier. En plus, l'accent est mis sur les différents paramètres de mesures de performance et de distorsion des systèmes de compression.

Le deuxième chapitre est dédié à la compression des images. Il présente la notion de transformée en ondelettes, son application dans la compression des images toute en définissant les deux fonctions ondelettes qui nous intéressent à savoir l'ondelette de Haar et l'ondelette bi-orthogonale.

Le troisième chapitre décrit l'approche neuronale en détails. Après un bref historique sur les débuts de l'utilisation des réseaux de neurones artificiels, on présente le principe de leur fonctionnement, les différents types d'architecture, les techniques d'apprentissage et les domaines d'applications. On s'intéresse en particulier au réseau Perceptron MultiCouche (PMC) qu'on a utilisé dans notre travail et à son algorithme d'apprentissage la rétro-propagation du gradient de l'erreur « *backpropagation* ».

Le quatrième chapitre est consacré à la présentation de notre contribution dans la compression d'images. Il s'agit d'une description de la démarche expérimentale

adoptée pour élaborer un système de compression d'images sans perte. Ce système est basé sur une hybridation des réseaux de neurones avec la transformée en ondelettes.

La conclusion présente une synthèse des apports de la thèse en terme d'approches proposées et de résultats obtenus, et dégage les perspectives importantes qui font suite à ce travail.

Liste des figures

Liste des tableaux

Liste des abréviations

CPU : Central Processing Unit
CR : Compression Ratio
CWE : Contrast Weighted Entropie.
CWT : Continu Wavelet Transform
DCT : Discrete Cosine Transform
DPI : Dots Per inch
DWT : Discret Wavelet Transform
ECG : EléctroCardioGramme
EEG : ElectroEncéphaloGraphie
EQM : Erreur Quadratique Moyenne
GSM : Global System for Mobile communications
IC : Integrated Circuit
ICWT : Inverse Continu Wavelet Transform
IRM : Imagerie par Résonance Magnétique
JPEG : Joint Photographic Experts Group
LZW : Lempel-Ziv-Welch
MLP : MultiLayer Perceptron
MSE : Mean Square Error
PAO : Publication Assistée par Ordinateur
PMC : Perceptron Multi-Couche
PPI : Pixels Per Inch
PPC : Points Par Centimètres
PPM : Points Par Millimètres
PSNR : Peak Signal to Noise Ratio
QMF : Quadratique Miror Filtre
RAM : Random Access Memory
RBF : Radial Basis Function
RC : Rapport de Compression
RLE : Run Length Encoding
RNA : Réseau de Neurone Artificiel
SIG : Système d'Informatique Géographique
SQ : Scalar Quantization
SVM Support Vector Machine
VLC : Variable Lenght Coding
VQ : Vector Quantization

Généralités sur les images
&
Processus de compression

Introduction

Les données numériques, par leur utilisation posent des problèmes de temps de transmission et de volume de stockage. Or, les progrès technologiques, certes incontestables, souvent ne suffisent pas, soit parce que non matures, soit parce que trop chers, d'où l'intérêt de la compression de ces données. En effet, la compression permet de diminuer la taille de stockage et de rendre possible leurs transports sur des réseaux de communication (Internet, GSM, câble, TV, satellite). Certes, les capacités des disques durs de nos ordinateurs et le débit des réseaux ne cessent d'augmenter, mais également notre utilisation des données numériques n'a pas cessé d'augmenter.

Rares sont aujourd'hui les domaines qui n'ont pas succombé au numérique. Parmi eux, on compte le son et l'image. Or, ces derniers se caractérisent, en plus de leur utilisation terriblement croissante, par le fait que :

- Les capacités des stockages des utilisateurs, même si elles ne cessent d'augmenter, ne sont pas infinies. Par exemple, dans l'imagerie médicale, des masses gigantesques de données sont acquises chaque jour. On chiffre à 10 Téraoctets [1] de données d'un service de radiologie d'un hôpital dans un pays industrialisé.
- La durée des transmissions de ces données numériques est conditionnée par le débit du réseau qui est utilisé et qui est parfois relativement faible [1].

Ce premier chapitre est composé de deux parties. La première partie est une brève introduction à l'image et à ses importantes caractéristiques. La deuxième partie sera consacrée à l'étude des techniques de compression. Nous verrons qu'il existe un grand nombre de techniques permettant de comprimer des données numériques. Leurs avantages et leurs inconvénients sont différents, et faire un choix est une entreprise délicate.

Partie A : Généralités sur les images

I. Définition de l'image

Une image est une représentation planaire d'une scène ou d'un objet situé en général dans un espace tridimensionnel, elle est issue du contact des rayons lumineux provenant des objets formants la scène avec un capteur (caméra, scanner, rayons X, …) [2]. Il ne s'agit en réalité que d'une représentation spatiale de la lumière. Elle est considérée comme un ensemble de points auquel est affectée une grandeur physique (luminance, couleur). Ces grandeurs peuvent être continues on parle d'image vectorielle ou bien discrètes il s'agit d'une image matricielle ou numérique. On peut trouver plusieurs types d'images on cite [3]:

- Imagerie vidéo (vision industrielle, vidéo surveillance) ;
- Imagerie thermique (thermographie) ;
- Imagerie d'écho (radar, sonar, échographe, doppler) ;
- Imagerie à rayons X (radiologie, scanner) ;
- Imagerie radioactive (tomographie) ;
- Imagerie IRM (résonance magnétique) ;
- Imagerie satellitaire ;
- Imagerie multi spectrale et imagerie couleur ;
- Imagerie polarimétrique.

I.1 Image matricielle

Une telle image, appelée aussi raster ou bitmap [1,4], est un tableau rectangulaire formé par des éléments appelés pixels (PICture ELement), dont les valeurs sont des nombres entiers positifs. C'est le type d'image le plus répandu sur les ordinateurs actuels. On peut la représenter par une matrice de $nc \times nl$ pixels dont laquelle chaque pixel est localisé par ses coordonnées i et j dans l'image comme indiqué sur la figure I.1, où $i \in \{0, nl-1\}$ est l'indice de colonne, et $j \in \{0, nc-1\}$ est l'indice de ligne, nc et nl sont respectivement la largeur et la hauteur de l'image I. La valeur $I(i,j)$ représente l'intensité du pixel (i, j).

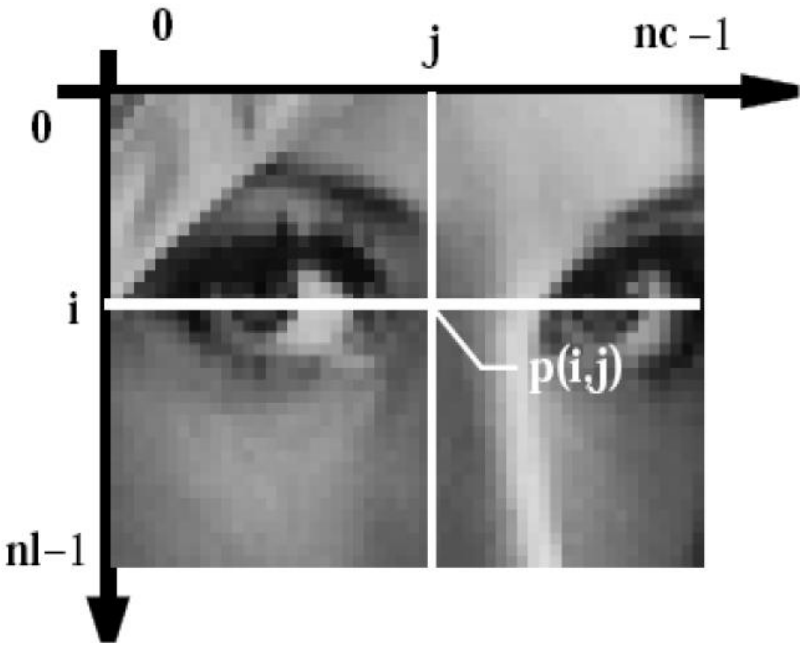


Figure I.1: Image matricielle

Ce type d'images présente des avantages et des inconvénients, on cite comme exemples [2,3]:

Avantages :

- Elles peuvent facilement être crées et stockées dans un tableau de pixels représentant l'image ;
- Lecture/écriture d'un pixel est simple du fait de la représentation de l'image comme une grille ;
- Les images raster peuvent facilement être affichées sur un écran ou être imprimées.

Inconvénients :

- Les fichiers peuvent être très gros donc nécessitent d'être compresser ;
- Problème de changement d'échelle (apparition d'effets de marches d'escalier) ;
- Les dimensions de l'image doivent être prévues pour la résolution de l'interface de sortie (écran, imprimante).

I.2 Image vectorielle

Une telle image est une liste de figures 2D : segments de droite, polygones, coniques, Bézier... remplies ou non de couleurs unies, tramées, dégradées…[1]. Ses dimensions sont des nombres réels positifs quelconques [3]. Autrement dit, une image vectorielle ne stocke pas le résultat du dessin sous la forme de pixels (bitmap), mais la façon de la dessiner par un ensemble d'objets géométriques

(lignes, cercles, texte...) définis par différents attributs (coordonnées, couleur, épaisseur de trait, remplissage…).

Ces images ont l'avantage d'occuper des places mémoires peu importantes et leurs proportions restent conservées si l'on modifie leurs tailles [5]. De plus, elles sont adaptées au stockage d'images composées de formes géométriques et peuvent aisément être redimensionnées. Leurs inconvénients est qu'elles peuvent difficilement stocker des images complexes comme des photographies et leur affichage peut prendre plus de temps que celui d'une image bitmap de complexité égale [6].

Les domaines d'utilisation de ces images sont très variés on cite : dessin industriel, PAO, Systèmes d'Information Géographiques (SIG), Internet, etc. la figure I.2.a présente une image vectorielle

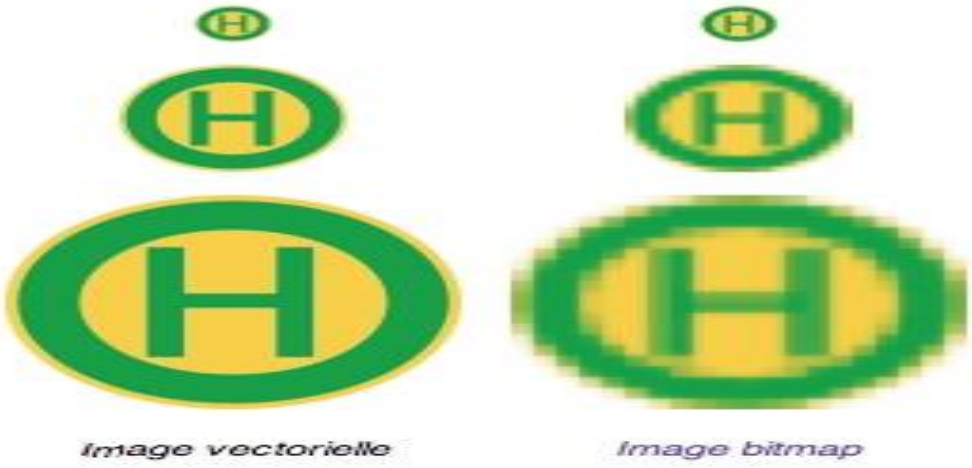


Figure I.2 : a. image vectorielle b. image matricielle (bitmap)

II. Codage d'une image matricielle

L'image numérique est une image dont la surface est divisée en éléments de taille fixe appelés cellules ou pixels. Selon la valeur de ces pixels elle se présente sous plusieurs formats, les plus utilisées sont :

- ***Image binaire***

  Dans une image binaire, les pixels sont représentés par deux états logiques 0 (noir) et 1 (blanc) [1,2,3]. Une telle image, présentée par la figure I.3, est appelée bitmap ou image binaire et il faut un seul bit pour coder ses pixels.

Figure I.3: Image codée avec deux couleurs : blanc et noir

- ***Image à niveaux de gris (intensité ou luminance)***

Chaque pixel est codé sur N bits, ce qui lui confère des valeurs entières comprises entre 0 qui représente le noir et 2^N-1 qui représente le blanc [1,7]. Chaque niveau de gris représente l'intensité ou la luminance de l'image. Une telle image est appelée parfois graymap [8] et présentée par la figure I.4

Figure I.4: Image à niveaux de gris

- ***Image couleur***

Une image couleur correspond à la synthèse additive de 3 images : rouge, vert et bleu comme indiqué par la figure I.5 [1]. La couleur d'un pixel est représentée par 3 composantes de couleurs et donne naissance à un point dans un espace tridimensionnel [2]. Chaque pixel est donc codé sur $3\times N$ bits [7,8]. Une telle image est appelée pixmap.

On trouve d'autres espaces de couleurs exemple le système YUV qui consiste à séparer la notion de couleur en deux concepts : intensité lumineuse et teinte de la lumière [2,3,4].

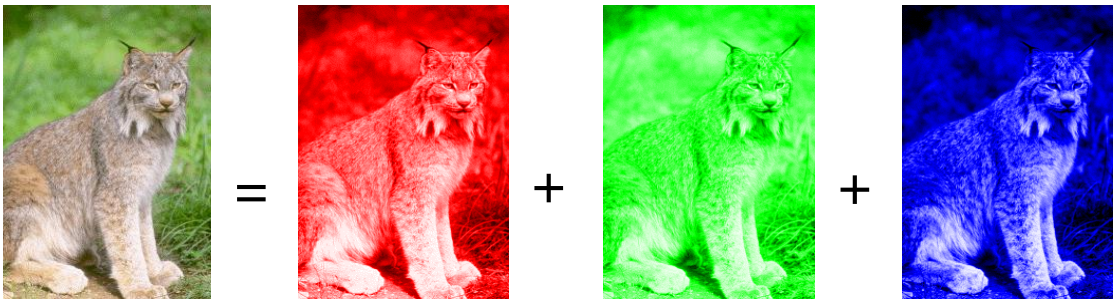


Figure I.5: Décomposition d'une image couleur

III Caractéristiques d'une image numérique

L'image numérique ou matricielle est un ensemble structuré d'informations caractérisée par les paramètres suivants :

III.1 Définition d'une image

On appelle définition le nombre total de points (pixels) constituant l'image matricielle. C'est le nombre des pixels dans les colonnes de l'image qu'on multiplie par le nombre total des pixels des lignes [6,7].

III.2 Résolution et Dimension

La résolution d'une image matricielle est le nombre de points (pixels) par unité de longueur. Elle permet d'établir le rapport entre la définition en pixels d'une image et la dimension réelle de sa représentation sur un support physique (écran, papier...) [1,6,8]]. La figure I.6 représente l'image « Lena » sous différentes résolutions.

La résolution peut être exprimée en :

DPI : points par pouce (Dots Per Inch)
PPC : Points Par Centimètre
PPM : Points Par Millimètre
PPI : pixels par pouce (Pixels per Inch)

Avec 1 pouce = 2,54 cm [1,8].

La dimension d'une image est le rapport de sa définition par sa résolution. Elle s'exprime en pouce [1,8].

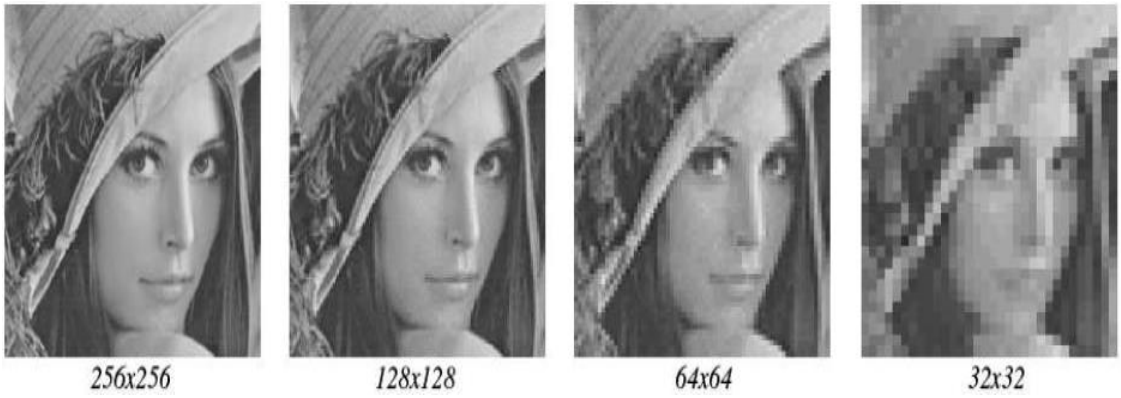


Figure I.6 : Image « Lena » sous différentes résolutions

III.3 Voisinage

Le plan de l'image est divisé en termes de formes rectangulaires ou hexagonales permettant ainsi l'exploitation de la notion de voisinage. Le voisinage d'un pixel est formé par l'ensemble des pixels qui se situent autour de lui. On définit aussi l'assiette comme étant l'ensemble de pixels définissant le voisinage pris en compte autour d'un pixel [1,2].
On distingue deux types de voisinage :

- Voisinage à 4 : dans ce cas, on ne prend en considération que les pixels qui ont un côté commun avec le pixel courant (figure I.7-a) [6,7]].
- Voisinage à 8 : ici on prend en compte tous les pixels qui ont au moins un point en liaison avec le pixel courant (figure I.7-b) [6,7].

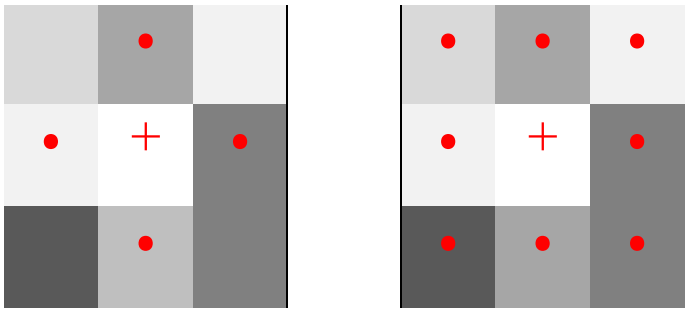


Figure I.7 : a. Voisinage 4 connexité b. Voisinage 8 connexité

III.4 Niveau de gris

C'est la valeur d'intensité lumineuse d'un pixel. Cette valeur peut aller du noir de valeur 0 jusqu'au blanc de valeur 255 en passant par les nuances qui sont contenues dans l'intervalle [0, 255] [7]. Elle correspond en fait à la quantité de la lumière réfléchie [8].

Pour 8 bits, on dispose de 256 niveaux de gris dont 40 sont reconnus à l'œil nue [8]. Plus le nombre de bits est grand plus les niveaux sont nombreux et plus la représentation est fidèle.

III.5 Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface [1,7,8]. Pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet.

III.6 Contraste

C'est la différence de luminance entre les valeurs les plus clairs proches du blanc et les valeurs les plus sombres proches du noir. Plus la différence est grande plus les détails de l'image sont nets et plus la différence est faible plus l'image est terne [1,7]. Si L_1 et L_2 sont les degrés de luminosité (valeurs des pixels) respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste est défini par le rapport [8]:

$$C = \frac{L_1 - L_2}{L_1 + L_2} \qquad \text{I. 1}$$

III.7 Bruit

Un bruit dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins. Il provient de l'éclairage des dispositifs optiques et électroniques du capteur. C'est un parasite qui représente certains défauts (poussière, petits nuages, baisse momentanée de l'intensité électrique sur les capteurs, …etc.). Il se traduit par des taches de faible dimension et dont la distribution sur l'image est aléatoire [8,9]. La figure I.8.b représente une image qui comprend du bruit.

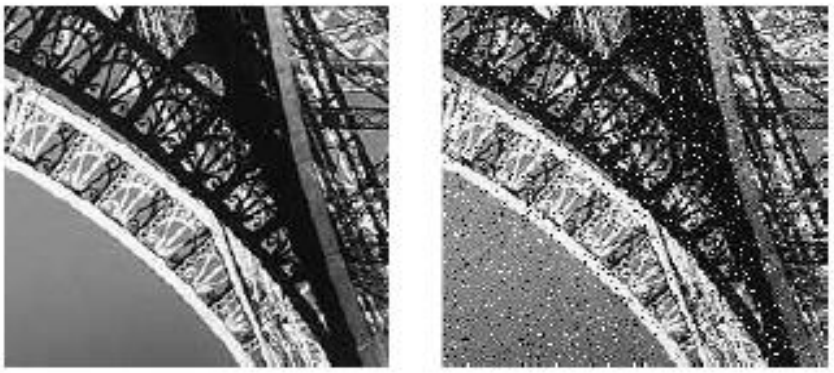


Figure I.8 : a. Image originale b. Image bruitée

III.8 Contour et Texture

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative [8,9]. Dans une image numérique, les contours peuvent se situer entre les pixels appartenant à des régions ayant des intensités moyennes différentes; il s'agit de contours de type *«saut d'amplitude»* [9], ou correspondre à une variation locale d'intensité présentant un maximum ou un minimum; il s'agit alors de contour *« en toit »* [9].

Les textures décrivent la structure des contours [8]. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes.

En plus des deux caractères définis plus haut, on peut citer la notion de zone homogène dont les pixels ont la même valeur [6,8].

Figure I.9: Contour, texture et zone homogène indiqués sur l'image « Barbara »

III.9 Histogramme d'une image

L'histogramme d'une image est un vecteur de dimension *N*, tel que *N* est le nombre total des intensités lumineuses des pixels (N=256 dans le cas d'une image à niveau de gris). Chaque élément de ce vecteur qu'on note *h(i)* représente le nombre de pixels ayant l'intensité *i* c'est sa fréquence d'apparition dans l'image. On peut aussi interpréter l'histogramme en termes de probabilité en divisant chaque terme *h(i)* par le nombre total des pixels de l'image. De ce fait, chaque case *h(i)* du vecteur représente la probabilité d'avoir l'intensité *i* dans l'image [1,6,8].

L'histogramme peut être utilisé pour améliorer la qualité d'image (rehaussement d'image) [9], ceci en introduisant quelques modifications pour pouvoir extraire les informations utiles de celle-ci. Il permet aussi d'obtenir des renseignements rapides sur l'image. En effet, on peut notamment faire la distinction entre une image trop foncée (niveaux en majorité près de 0) et une image trop claire (niveaux en majorité près de 255) [9]. Il permet aussi de faire la différence entre le fond de l'image (arrière-plan) et les objets intéressants de l'image (premier-plan).

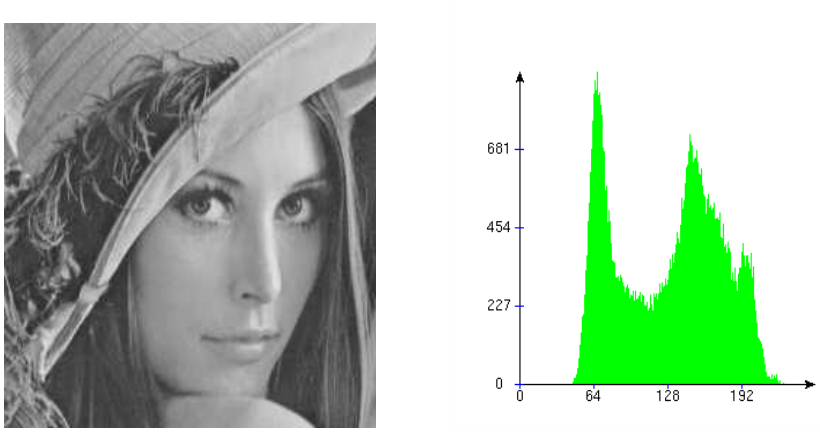


Figure I.10: Image « Lena » et son histogramme

Partie B : Processus de compression

I. Objectif et Introduction

L'utilisation de données en général et les images en particulier sous leurs formes numériques ne serait pas possible aujourd'hui sans la compression préalable de celles-ci. Et pour comprendre l'intérêt de la compression des images, le tableauI.1 présente des tailles de fichiers d'images numériques ayant une résolution permettant de faire ressortir avec une qualité convenable une feuille de taille A4 [10,11].

Comme nous pouvons le voir sur le tableau, la taille d'une image en couleur est importante. Il ne serait pas possible d'en stocker beaucoup sur la carte mémoire d'un appareil photo numérique. D'où la nécessité de comprimer les images ce qui revient à réduire le volume de données nécessaires à leurs codages. Ceci facilite

leurs stockages sur des supports de mémoire et permet d'accélérer leur transmission par réseau.

Type	Résolution	Nbpp	Taille du Fichier
Page en noir et blanc (fax)	1000×2000	1bit	2Mbits=250Ko
Page en niveau de gris (photo)	1000×2000	8bits	16Mbits=2Mo
Page en couleur (photo)	1000×2000	8bits×3=24bits	48Mbits=6Mo

Tabeau I.1 : Tailles des fichiers pour divers formats d'images numériques.

Plusieurs algorithmes de compression ont vu le jour depuis que les chercheurs ont compris l'importance et la nécessité de celle-ci. D'autre part, tous ces algorithmes se basent sur 3 points essentiels à savoir [2,10,11]:

- Détection de redondances dans le signal à comprimer (image fixe, image vidéo, son…), afin d'éliminer l'information inutile ou les choses que l'humain ne perçoit pas bien, par exemple, des nuances de couleurs invisibles pour la plupart des yeux (ou impossibles à reproduire fidèlement sur la plupart des écrans), des sons inaudibles pour la plupart des oreilles parce que masqués par d'autres sons plus forts (mp3, téléphonie) etc...

- Un algorithme de compression qui permet essentiellement de réduire la place occupée par les données du signal sans changer la signification de celles-ci. Autrement dit, cet algorithme consiste à représenter les éléments du signal par des codes plus petits que les codes d'origine.

- Un algorithme (inverse) de décompression qui permet de retrouver la représentation initiale du signal comprimé, soit sous une forme identique, soit légèrement dégradé.

Un système global de compression d'une image se décompose généralement des étapes [11,12]:

1. Prétraitements ;
2. Quantification ;
3. Compression sans pertes.

Les rôles de ces trois étapes ne sont pas toujours clairement séparés, et l'utilisation d'une technique particulière dans une étape a généralement des conséquences sur les deux autres étapes. Ainsi, les prétraitements ont pour but

d'optimiser l'efficacité de la quantification. La deuxième étape produit des données dont le niveau de redondance conditionne la troisième étape [13,14]

II. Mesure des performances de la compression

II.1 Rapport et Taux de Compression

Le rapport de compression est l'une des caractéristiques les plus importantes de toutes les méthodes de compression. Il permet d'évaluer la qualité de la compression, il est défini comme étant le rapport du nombre de bits utilisés pour coder l'image compressée par le nombre de bits nécessaires pour coder l'image originale et il est donné en pourcentage [8,9,14]:

$$RC = \frac{nombre\ de\ bits\ pour\ coder\ l'image\ compressée}{nombre\ de\ bits\ pour\ coder\ l'image\ originale} \qquad \text{I.2}$$

Ceci signifie que l'image compressée occupe *RC* % de l'espace utilisé par l'image d'origine.

Le taux de compression *Tr* défini le pourcentage de compression de l'image originale. Il est défini par la relation suivante [8,9]:

$$Tr = (1 - RC) \times 100 \qquad \text{I.3}$$

La valeur de Tr dépend en général de la nature de l'image d'origine ainsi que de l'algorithme utilisé pour sa compression. En plus, la relation I.3 liant *Tr* à *RC* implique que si *RC* est petit alors l'image est compressée avec un taux *Tr* élevé.

II.2 Mesure de distorsion

Pour mesurer la distorsion entre l'image compressée et l'image originale, il existe deux paramètres couramment employés : l'erreur quadratique moyenne et le rapport signal à bruit crête.

Si on considère *X* une image matricielle de type *M*×*N* et *X'* son image compressée. On note *I(i,j)* la valeur du pixel *(i,j)* de *X* et *I'(i,j)* sa valeur dans *X'* alors on appelle l'Erreur Quadratique Moyenne EQM (en anglais Mean Square Error MSE) la valeur [8,9,11] :

$$EQM = MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(I(i,j) - I'(i,j))^2 \qquad \text{I.4}$$

Et :

Le rapport signal à bruit crête (en anglais Peak Signal to Noise Ratio *PSNR*) permet de mesurer la qualité de perception et de distorsion au niveau de l'image compressée, il s'exprime en décibel et défini par la relation suivante [7,8,11] :

$$PSNR = 10log_{10}\left(\frac{(\text{Max}(I(i,j)))^2}{MSE}\right) \qquad \text{I.5}$$

III. Les types de compression

L'objectif de la compression est de réduire la quantité de mémoire nécessaire pour le stockage d'une image ou de manière équivalente de réduire le temps de transmission de celle-ci. Cette compression peut soit conserver l'image intacte, on parle alors de compression sans perte, soit autoriser une dégradation de l'image pour diminuer encore l'empreinte mémoire, on parle ici de compression avec perte.

Un algorithme de compression **sans perte** restitue après les opérations successives de compression et de décompression une suite de bits strictement identique à l'originale [8,14]. Les algorithmes de compression sans perte sont utiles pour les documents, les archives, les fichiers exécutables ou les fichiers texte.

Avec un algorithme de compression **avec perte**, la suite de bits obtenue après les opérations de compression et de décompression est différente de l'originale, mais l'information restituée est en revanche voisine [8,14]. Les algorithmes de compression avec perte sont utiles pour les images, le son et la vidéo.

III.1 Compression sans perte

La compression sans perte (en anglais lossless), appelé aussi codage réversible, permet de retrouver la valeur exacte du signal comprimé lorsqu'il n'y a aucune perte de données sur l'information d'origine. En fait, la même information est réécrite d'une manière plus concise. Le processus de codage sans perte crée des "mots-codes" à partir d'un dictionnaire statique ou d'un dictionnaire construit

dynamiquement. Ces processus s'appuient sur des informations statistiques de l'image [14].

III.1.1 Quantité d'information

La théorie de l'information permet d'évaluer la quantité d'information dans une image [1]. Chaque pixel d'une image est considéré comme une variable aléatoire par conséquent une image est considérée comme une suite de $M \times N$ variables aléatoires avec M nombre de lignes et N le nombre de colonnes.

Si on considère P un point d'une image en niveaux de gris donc sa valeur est un entier n_i de l'intervalle [0, 255]. On appelle l'occurrence $O(n_i)$ le nombre de fois que la valeur n_i apparait dans l'image matricielle. Donc la probabilité pour que la valeur d'un pixel soit n_i est donnée par la relation suivante [9,10] :

$$p(n_i) = \frac{O(n_i)}{256} \qquad \text{I.6}$$

En 1948, Shannon a défini la quantité d'information véhiculée par un événement $I(n_i)$, mesurée en bits, en terme de probabilité de cette événement $p(n_i)$, par l'équation suivante [1,8,9,14] *:*

$$I(n_i) = log_2\left[\frac{1}{p(n_i)}\right] \qquad \text{I.7}$$

Un événement qui est complètement prédictible (p=1) n'a aucune information ($\log_2(1)=0$).

III.1.2 L'entropie de Shannon

L'une des nombreuses questions que s'est posées Shannon à la fin des années 40 (et qu'il a bien souvent résolues) est la suivante : étant donnée une source X produisant des symboles x_i choisis dans un ensemble fini avec une probabilité $p(x_i)$, quel est le nombre minimal de bits nécessaire pour transmettre l'information. Shannon a pu montrer qu'il existe une quantité $H(X)$, qu'il a appelé entropie de la source, définie par [8,9,10,15]:

$$H(X) = -\sum_i p(xi) log_2[p(xi)] \qquad \text{I.8}$$

L'entropie est la mesure du désordre ou la mesure de l'imprévisibilité. Dans les systèmes informatiques, le degré de l'imprévisibilité d'un message peut être utilisé comme une mesure de l'information véhiculée par le message. Le concept

important dans la définition de l'entropie est celui de la prévisibilité. Un message parfaitement prévisible ne véhicule aucune information.

III.1.3 L'entropie pondérée de contraste

L'entropie pondérée du contraste calcule le changement des valeurs de contraste entre pixels voisins. Elle mesure la relation non linéaire entre les intensités d'une image, et reliée à l'entropie de l'image par l'équation suivante [14,16]:

$$CWE = -\sum_{i}(p_i - M) \times p_i \times log_2(p_i) \times R \qquad \text{I.9}$$

Où :

R: le nombre de pixels dans l'image.
N: nombre total des valeurs d'intensité des pixels.
p_i: probabilité d'apparition de la valeur du pixel i.
M: valeur moyenne des fréquences des intensités.

III.1.4 Les algorithmes de compression sans perte

Les algorithmes de compression sans perte permettent de coder d'une manière plus courte les données d'une information, c'est la raison pour laquelle ils sont appelés les algorithmes de compactage [8] car il n'y a aucune perte de données sur l'information d'origine : Il y'a autant d'information après la compression qu'avant. Ces algorithmes sont classés en trois catégories :

- *Algorithmes à base de redondances :* qui exploite la redondance des données au sein de la source. Le plus célèbre parmi eux : le codage par plage (RLE) Run length Encoding [8,10].

- *Algorithmes statistiques appelés aussi entropiques :* ils permettent de s'approcher au mieux de l'entropie de la source à coder. Ils ont pour principe de coder les événements d'une source donnée qui ont une probabilité d'apparition forte avec des mots de longueur plus faible que les symboles qui ont une probabilité d'apparition faible [8,10]. Les méthodes les plus connues de codage entropique sont le code de Shannon, le code de Huffman appelé encore Code à Longueur Variable (VLC) et le code arithmétique.

- *Algorithmes à base de dictionnaires :* Ce sont des techniques de codage qui utilisent un dictionnaire qui mémorise les adresses (indices) des chaînes qui

se répètent. La technique la plus connue est celle établie par LZW proposée par Lempel, Ziv et Welch [10,11].

III.1.4.1 Code RLE (Run-Length-Encoding)

Cet algorithme consiste à regrouper les événements identiques consécutifs et les représenter en couple <événement, occurrence> [1,8,10]. Utilisé dans la compression d'images, cet algorithme code la répétition de pixels identiques en donnant le nombre de répétitions et le pixel à répéter. En effet, on considère une image comme un signal 1D en la balayant comme un livre : de haut en bas et de gauche à droite. On pourrait tout à fait employer un autre balayage : en colonnes ou en zig-zag [13].

Cette technique est intéressante pour des images présentant de grandes plages uniformes, telles que les images vectorielles. Ce n'est pas le cas des images réelles qui sont bruitées. Il y a de faibles variations des pixels, mais pas de zone totalement identique ; dans ce cas, le codage sera plus coûteux que l'image d'origine. De plus, il est inefficace pour des images en 16 millions de couleurs à cause de la présence de plusieurs nuances différentes pour qu'il y ait de longues séquences de pixels identiques [13,14].

III.1.4.2 Codage de Shannon-Fano

Shannon et R. M. Fano ont développé à peu près en même temps une méthode de codage basée sur de simples connaissances de la probabilité d'occurrence de chaque symbole dans le message [8,15].

Le procédé de Shannon-Fano construit un arbre descendant à partir de la racine, par divisions successives. Le classement des fréquences se fait par ordre décroissant ce qui exige une première lecture du fichier et la sauvegarde de l'en-tête. Le principe suit les étapes suivantes [8,9,10]:

1. Les symboles sont triés et classés en fonction de leur fréquence d'apparition en commençant par le plus fréquent.

2. La liste des symboles est ensuite divisée en deux parties de manière à ce que le total des fréquences de chaque partie soit aussi proche que possible.

3. Le chiffre binaire 0 est affecté à la première partie de la liste, le chiffre 1 à la deuxième partie.

4. Chacune des deux parties fait à son tour l'objet des démarches 2 et 3.

Le processus continue jusqu'à ce que chaque symbole soit devenu une feuille de l'arbre correspondant à un code déterminé.

III.1.4.3 Code de Huffman

Le code de Huffman est plus complexe : il s'agit de travailler au niveau binaire, et d'utiliser moins de bits pour coder les symboles qui reviennent le plus souvent. On va construire un arbre binaire où les feuilles sont les symboles de la table, et où les branches les plus profondes portent les symboles les plus rares.

En effet, en 1952, Huffman a mis en évidence une méthode pour construire des codes compacts de longueur variables [1,8,17]. Elle consiste à attribuer un code de sortie à chaque symbole. Ces codes pouvent être aussi court qu'un bit, ou beaucoup plus longs que les symboles d'entrée en fonction de leurs probabilités. Le nombre de bits optimum pour chaque symbole est $-\log_2(p_i)$ [8,17], où p_i est la probabilité du symbole en question.

Les étapes nécessaires pour générer le code Huffman pour des symboles dont les probabilités d'occurrence sont connues peuvent être regroupées comme suit [17,18]:

Phase 1 : Construction de l'arbre.

1. Trier les différentes valeurs par ordre décroissant de fréquence d'apparition des symboles pour former la table de symboles ;

2. Fusionner les deux symboles minimaux (les deux symboles les moins probables) dans un arbre binaire et affecter leur somme à la racine pour former un symbole combiné, et reclasser encore les symboles dans l'ordre de probabilité. Cela génère une arborescence dont chaque nœud représente la probabilité cumulative de tous les nœuds en dessous ;

3. Réordonner la table de poids par poids décroissants et tracer un chemin vers chaque feuille, en notant la direction sur chaque nœud ;

4. Recommencer à partir de 2 jusqu'à obtenir un seul arbre.

Phase 2 : Construction du code

Le code se construit à partir de l'arbre obtenu dans la phase 1 comme suit :

à partir de la racine, attribuer des 0 aux sous arbres de gauche et des 1 à ceux de droite. Cette procédure continue jusqu'à la dernière colonne où la probabilité de 1 est atteinte. A partir de la dernière colonne, chaque branche de probabilité de l'arborescence, est marquée d'un 0 à gauche, et d'un 1 à droite et on construit ainsi le code.

III.1.4.4 Codage arithmétique

La particularité du code de Huffman réside dans le fait qu'il s'approche de l'entropie de la source et donc de l'optimum, ce qui peut s'avérer peu intéressant dans le cas où l'entropie de celle-ci est faible, et où un surcoût de 1 bit devient important [19]. De plus le codage de Huffman impose d'utiliser un nombre entier de bit pour un symbole source, ce qui peut s'avérer peu efficace. La solution à ce problème est de travailler sur des blocs de N symboles ce qui est apportée par le code arithmétique.

Le codage arithmétique traite l'ensemble d'une séquence de symboles comme une seule entité, ce qui lui permet de réussir à sortir de la contrainte imposée par les approches VLC (Variable Length Coding) comme le code de Huffman, qui attribuent un mot de code court aux symboles probables et des mots plus longs aux symboles moins fréquents.

En effet, l'idée fondamentale de ce code est d'utiliser une échelle sur laquelle les intervalles de nombre réels du codage sont représentés entre 0 et 1 [1,19]. Le principe repose sur le découpage de l'intervalle [0,1] et chaque symbole de la source à coder se voit attribuer une partition de l'intervalle dont la taille est égale à sa probabilité d'occurrence ce qui correspond à la fonction de la densité de probabilité cumulative de tous les symboles. Il est à signaler que chaque séquence à coder est représentée d'une manière unique par une valeur appartenant à l'intervalle [0,1]. A mesure que la séquence s'allonge, l'intervalle requis pour la représenter diminue, et le nombre de bits qui servent à préciser cet intervalle s'accroît. Finalement, les données comprimées consistent en la partie fractionnaire du plus court nombre binaire qui se trouve dans l'intervalle final.

Si on considère X une séquence de m symboles $X=x_1x_2\ldots x_m$, qui prend ses valeurs à partir d'une source $S=\{s_1, s_2, \ldots, s_n\}$. L'algorithme du codage arithmétique statique peut être résumé aux étapes suivantes [18,19]:

1. Calculer la probabilité associée à chaque symbole de la source S. Notons $p(s_k)$ la probabilité de s_k.

2. Initialiser la limite inférieure L de l'intervalle de codage à la valeur 0 et la limite supérieure H à la valeur 1, cet intervalle a une largeur :

$$largeur = H - L = 1 \qquad \text{I.10}$$

3. Associer à chaque symbole s_k un sous intervalle $[L(s_k), H(s_k)[$ proportionnel à sa probabilité, avec $H(s_k) - L(s_k) = p(s_k)$ et tel que :

$$L(s_k) = L + largeur \times \sum_{i=1}^{k-1} p(s_i) \qquad \text{I.11}$$

$$H(s_k) = L + largeur \times \sum_{i=1}^{k} p(s_i) \qquad \text{I.12}$$

4. On choisit le sous-intervalle correspondant au prochain s_k à coder dans la séquence et on met à jour les valeurs L et H de la manière suivante:

$$L = L + largeur \times L(s_k) \qquad \text{I.13}$$

$$H = L + largeur \times H(s_k) \qquad \text{I.14}$$

5. Avec le nouvel intervalle [L,H] on recommence le processus de l'étape 3.

6. Les étapes 3, 4 et 5 sont répétées jusqu'à épuisement des symboles de la séquence et obtention du dernier intervalle [L,H].

7. A la fin, la représentation binaire de tout réel x_c de l'intervalle [L,H] est un code de la séquence.

Généralement, on choisit comme représentant de la séquence la limite inférieure L de l'intervalle [18] et on appelle code(*X*) la valeur choisie comme représentant de la séquence à coder.

Il est à noter qu'à la fin du processus de codage, le code arithmétique génère un fichier composé d'un en-tête suivi de la représentation binaire du code choisi. De plus, l'en-tête peut contenir soit la table des probabilités, soit la table des fréquences de tous les symboles. Après le codage de la séquence X, on peut facilement décoder cette séquence en utilisant la table des probabilités (ou bien la table des fréquences)

Le codage arithmétique est généralement plus performant que le code de Huffman [20]. Il tend vers la limite inférieure théorique mais cependant il est gourmand en ressources et nécessite de connaître à priori l'intégralité du signal avant de pouvoir procéder au codage.

III.1.4.5 Codage Lempel-Ziv

C'est une technique de codage qui utilise un dictionnaire. On cherche dans le fichier les chaînes qui se répètent, puis on les mémorise par leur adresses (ou indices) construites dans un dictionnaire. Il existe trois versions de cet algorithme [1,8,9] :

• LZ77 c'est la version originale proposée par Lempel et Ziv en 1977 [1,2]. Son principe consiste à faire passer le signal à coder dans une fenêtre glissante. Les symboles du signal défilent de droite à gauche dans ce dispositif. L'idée est de regarder si le début de la partie droite se trouve dans la partie gauche. Si c'est le cas, on code la position de la sous-chaine et sa longueur.

• LZ78 : dans cette version, la recherche s'effectue sur tout le fichier. La taille du dictionnaire est limitée en fonction du mode de codage (16, 32, ou 64 bits).

• LZW : c'est une technique proposée par Welch et qui s'inspire d'une amélioration de LZ78. Elle consiste à remplacer la fenêtre de gauche par un dictionnaire. En effet, les chaînes déjà rencontrées dans le message à compresser sont rassemblées dans un tableau dans lequel, on attribue une case numérotée à chaque séquence. Au départ, ce tableau est initialisé avec tous les symboles de l'alphabet du signal à coder et aucune hypothèse n'est faite sur les probabilités des symboles, donc aucune optimisation à ce sujet.

III.2. Codage avec perte

L'enjeu de la compression avec pertes est de réduire la quantité de données d'un fsignal tout en préservant la qualité perceptible et en évitant l'apparition d'artefacts. Elle ne s'applique qu'aux données «perceptibles », en général sonores ou visuelles, qui peuvent subir une modification, parfois importante, sans que cela soit perceptible par l'humain. La perte d'information est irréversible, il est impossible de retrouver les données d'origine après une telle compression. La compression avec perte est appelée parfois *compression irréversible* ou *non conservative* [1,21,22].

Dans le cas des images, cette technique est fondée sur une idée simple : seul un sous-ensemble très faible de toutes les images possibles possède un caractère *exploitable et informatif* pour l'œil humain. En effet, l'œil humain est plus sensible à la luminance (intensité) qu'aux nuances de couleur et plus sensible aux basses fréquences [8,13]. Dans la pratique, l'œil a besoin pour identifier des zones de l'existence de corrélations entre pixels voisins, c'est-à-dire qu'il existe des zones contiguës de couleurs voisines. Les programmes de compression s'attachent à découvrir ces zones et à les coder de la façon aussi compacte que possible. Puisque l'œil ne perçoit pas nécessairement tous les détails d'une image, il est possible de réduire la quantité de données de telle sorte que le résultat soit très ressemblant à l'original, voire identique, pour l'œil humain.

Autrement dit, on peut réduire soit la partie de codage consacrée aux nuances de couleur il s'agit de la quantification soit supprimer les hautes fréquences d'une image on parle de la transformation. Dans les deux cas, l'information perdue rend possible une compression plus importante avec un taux de compression plus élevé au prix d'une dégradation toujours plus importante [22,23].

III.2.1. Quantification

Par définition, l'opération de quantification permet une transcription des données depuis un espace de taille "infini" constitué par exemple de l'ensemble des nombres flottants, ou de taille très importante, vers un espace contenant un nombre limité de coefficients [1,8,23]. Dans le cas d'une image c'est le nombre de valeurs différentes (niveaux de gris) qu'on peut associer à un pixel.

La quantification fait partie de plusieurs méthodes de compression d'image. L'objectif est de réduire la taille des coefficients de façon que cette réduction n'apporte pas de dégradations visuelles à l'image [21,23]. On distingue deux types familles de quantification:

- Quantification scalaire qui traite les données coefficient par coefficient.
- Quantification vectorielle qui procède sur un groupe de coefficients.

III.2.1.1 Quantification Scalaire

La quantification scalaire SQ (Scalar Quantization) est réalisée indépendamment pour chaque élément. D'une manière générale, on peut la définir comme étant l'association de chaque valeur réelle x, à une autre valeur q qui appartient à un ensemble fini de valeurs. La valeur q peut être exprimée en fonction

de la troncature utilisée : soit par l'arrondi supérieur, l'arrondi inférieur, ou l'arrondi le plus proche [1,8,9].

III.2.1.2 Quantification Vectorielle

La quantification vectorielle VQ (Vector Quantization) a été développée par Gersho et Gray [1,23,24]. Elle fait aujourd'hui l'objet de nombreuses publications dans le domaine de la compression numérique. Le principe de la quantification vectorielle est issu du travail de Shannon qui montre qu'il est toujours possible d'améliorer la compression de données en codant non pas des scalaires, mais des vecteurs.

Un quantificateur vectoriel Q associe à chaque vecteur d'entrée $X_i = (x_j, j=1...k)$, un vecteur $Y_i = (y_j, j=1\ldots k) = Q(Xi)$, tel que ce vecteur Y_i est choisi parmi un dictionnaire (code book) de taille finie. La quantification vectorielle *VQ* produit de meilleurs résultats que la quantification scalaire *SQ*, néanmoins la *VQ* nécessite un codage complexe et de grandes capacités de mémoire [23,24].

Le principe de la quantification vectorielle *VQ* appliqué au codage des images consiste à coder chaque bloc d'une image par l'index du bloc qui lui ressemble de plus au sein d'un dictionnaire (au sens d'un critère d'erreur par exemple) [24,25]. La décompression s'effectue après lecture des différents index et du dictionnaire.

III.2.2 Transformée de Fourier

Le mathématicien-physicien Joseph Fourier en 1802, dans son mémoire à l'Académie des Sciences sur la propagation de la chaleur, a notamment montré que, pour représenter de façon compacte et commode une fonction *x(t)* (du point de vue mathématique, une telle fonction est un point dans un espace ayant une infinité de dimensions), on peut utiliser des « axes » construits à l'aide d'un ensemble infini de fonctions sinusoïdales. En des termes un peu plus précis : Fourier a montré que l'on peut représenter une fonction *x(t)* par une somme d'une infinité de fonctions sinus et cosinus de la forme *sin(at)* ou *cos(at)*, chacune affectée d'un certain coefficient [1,22,26].

Soit *x(t)* une fonction de la variable t qui peut être à valeurs complexes. Cette fonction peut être décomposée sous la forme d'une intégrale de Fourier selon la relation [1,2,26]:

$$x(t) = \int_{-\infty}^{+\infty} X(f) e^{j2\pi ft} df \qquad \text{I.15}$$

$$\text{Où :} \quad X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi ft} dt \qquad \text{I.16}$$

X(f) est la transformée de Fourier de x(t). x(t) doit remplir les conditions suivantes :

- x(t) possède un nombre fini de discontinuités sur tout intervalle fini,
- x(t) possède un nombre fini de maxima et de minima sur tout intervalle fini,
- x(t) est absolument intégrable c'est-à-dire :

$$\int_{-\infty}^{+\infty} |x(t)| dt < +\infty \qquad \text{I.17}$$

III.2.3 Transformation en Cosinus Discrète (DCT)

Le noyau de projection de la transformation en cosinus discrète DCT (Discrete Cosine Transform) est une base de cosinus. En effet, les coefficients de la transformée ne sont pas complexes mais réels ce qui présente un avantage pour le codage et la quantification. La formule qui permet de transformer une image *I(x,y)* en sa transformée *T(u,v)* est la suivante [1,22,26] :

$$DCT\big(I(x,y)\big) = T(u,v)$$

$$T(u,v) = \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \cos\left[\frac{\pi}{2N} u(2x+1)\right] cos\left[\frac{\pi}{2N} v(2y+1)\right] I(x,y) \qquad \text{I.18}$$

Avec: $C(0) = \frac{1}{\sqrt{2}} \;\; et \;\; \forall \alpha \neq 0 \; C(\alpha) = 1.$

La figure I.10 représente l'image « Lena » et sa transformée de Fourier discrète.


Figure I.11 Image « Lena » et sa transformée de Fourier discrète

III.2.4 Compression JPEG

La compression JPEG (Joint Photographic Experts Group) concerne aussi bien les images fixes couleurs que les images en gris. Elle se base sur l'utilisation de la transformée en cosinus discrète DCT et s'organise autour des étapes suivantes [27,28]:

1. La décomposition de l'image à traiter en blocs (de 8×8 pixels) ce qui permet de réduire le temps de calcul et le nombre de variables à traiter.
2. Le calcul de la transformée DCT de chaque bloc.
3. Quantification des coefficients obtenus en utilisant une certaine matrice de quantification. Cette opération est responsable de la perte d'information mais aussi de la valeur élevée du facteur de compression qu'on obtient en utilisant cette méthode de compression.
4. Après la quantification on réalise un codage sans pertes.
5. A la fin, un fichier contenant les données obtenues après la compression est généré. Celui-ci a l'extension '.jpg'.
6. Pour la reconstruction de l'image traitée on applique la transformée en cosinus discrète inverse.

Conclusion

Les techniques de traitement d'image sont des techniques très diverses et le choix de l'une parmi elles, dépend essentiellement de la nature de l'application et des résultats qui peuvent être obtenus par l'application de l'une ou de l'autre. Cependant, chaque ensemble de ces techniques est destiné à une application spécifique. Certains ensembles ont des applications communes, mais les résultats

obtenus seront différents du point de vue avantages et inconvénients. La majorité de ces techniques ne cherchent pas à comprendre la sémantique de l'image, et se limite à la transformation et à la manipulation brute des pixels sans prendre en compte leur signification.

Que ce soit en mathématiques ou en physique, la transformée de Fourier a été pendant longtemps un outil essentiel, car elle servait à représenter de nombreux types de fonctions, donc de nombreuses grandeurs physiques. Cette représentation, basée sur la notion physique de fréquence, est bien adaptée pour traiter des signaux stationnaires. Or, la plupart des signaux du monde réel ne sont pas stationnaires comme les signaux vocaux et les images. Devant ces phénomènes transitoires, la transformée de Fourier s'est révélée imparfaite et sa principale limitation est qu'elle ne permet pas une description locale (sur une partie finie) d'un signal non stationnaire, et toute notion de localisation temporelle ou spatiale pour celui-ci disparaît dans l'espace de Fourier. Il faut donc trouver un compromis, une transformation qui renseigne sur le contenu fréquentiel tout en préservant la localisation. C'est ainsi que des représentations dites « temps-fréquence » ont été proposées afin d'analyser un signal à l'aide d'une transformation paramétrée par deux variables : le temps (ou la position) et la fréquence (ou l'échelle). La plus importante de ces représentations est la transformée en ondelette qui sera l'objet du prochain chapitre dans lequel on va s'intéresser particulièrement à l'utilisation de cette technique dans la compression des images fixes.

Bibliographie

[1] D. Lingrand, "*Introduction au traitement d'images,*". 2è édition Vuibert, février 2008.

[2] N. Moreau. "*Techniques de compression des signau*x,". Edition Masson, 1994.

[3] J. Froment. "*Traitement d'images et applications de la transformée en ondelettes,*". Thèse de doctorat, Université Paris IX, France (1990).

[4] J. Marconi, "*Transfert Sécurisé D'images par Combinaison de Techniques de Compression, Cryptage et Marquage,*". Thèse doctorat de l'université Montpellier (2006).

[5] C. Damerval "*Ondelettes pour la détection de caractéristiques en traitement d'images Applications à la détection de régions d'intérêt*,". Thèse doctorat de l'université Joseph Fourier, Grenoble I, France (2008).

[6] M.T. Chikh, "*Amélioration des images par un modèle de réseau de neurones (Comparaison avec les filtres de base),*". Mémoire de master de l'université Abou-Bakr Belkaid, Tlemcen, Algérie, 2011.

[7] G. Burel, "*réseaux de neurones en traitement d'images: Des modèles théoriques aux applications industrielles,*". Thèse de doctorat de l'université de Bretagne Occidentale, 1991.

[8] F. Douak, "*Reconstruction des images compressées en utilisant les réseaux de neurones artificiels et la DCT,*". Mémoire de majester de l'université de Batna, Algérie (2008).

[9] A. Ninassi, "*de la perception locale des distorsions de codage à l'appréciation globale de la qualité visuelle des images et vidéos. apport de l'attention visuelle dans le jugement de qualité,* ". Thèse de doctorat de l'université de Nantes, France (2009).

[10] A. Isar, A. Cubitchi, M. Nafornita, "*Algorithmes et techniques de compression,*". Edition orizonturi politehnice 2002.

[11] O. Le Cadet, "*Méthodes d'ondelettes pour la segmentation d'images. Applications à l'imagerie médicale et au tatouage d'images,*". Thèse de doctorat de l'institut national polytechnique de Grenoble, France (2004).

[12] C. Taouches, "*Implémentation d'un Environnement Parallèle pour la Compression d'Images à l'aide des Fractales,*". Mémoire de magister de l'université Mentouri Constantine, Algérie (2005)

[13] E. Zeybek, "*Compression multimodale du signal et de l'image en utilisant un seul codeur,*". Thèse de doctotat de l'université Paris Est, France (2011).

[14] J-P. Guillois, "*Techniques de compression des images,*". Collection informatique, Hermes, 1996.

[15] C. E. Shannon, "*A Mathematical Theory of Communication*", Bell System Technical Journal, Vol. 27, July 1948.

[16] K. Dimililer, "*Backpropagation Neural Network Implementation for Medical Image Compression*," Journal of Applied Mathimatics, November 2013.

[17] J. S. Vitter. "*Design and analysis of dynamic Huffman codes*, ". ACM Transactions on Mathematical Software, Volume 15, Issue 2(1989).

[18] N. Abramson, "*Information Theory and Coding*", McGraw-Hill Book Company, Inc., New York, NY, 1963.

[19] G. G. Langdon Jr., "*An Introduction to Arithmetic Coding*", IBM Journal of Research and Development, Volume 28, No.2, March 1984.

[20] I. H. Witten, R. M. Neal, J. G. Cleary, "*Arithmetic coding for data compression*", Communications of the ACM, 30(6), June 1987.

[21] M. Rabbani, P.W. Jones, "*Digital Image Compression Techniques*," vol.7, SPIE Press, Bellingham, Washington, USA, 1991.

[22] K. Ratakondu, N. Ahuja, "*Loless Image Compression with Multiscale Segmentation,*" IEEE, Transactions on Image Processing, vol. 11, N°. 11, 2002,

[23] É. Incerti, "*Compression d'image – Algorithmes et standards*, ". Vuibert 2003.

[24] A.V. Singh, K.S. Murthy, "*Neuro-Wavelet based Efficient Image Compression Using Vector Quantization*," International Journal of Computer Applications, vol. 49-N°.3, July 2012.

[25] R. Lancini, S. Tubaro. "*Adaptive Vector Quantization for Picture Coding using Neural Networks,* ". IEEE Transactions on Communications, 1995.

[26] G.Strang, G.Fix, "*A Fourier analysis of the finite element variational method*", Edition C.I.M.E., 1978.

[27] C. Christopoulos, A.N. Skodras, T. Ebrahimi, "*The JPEG2000 still image coding system: an overview*", *IEEE* Transactions on Consumer Electronics, vol. 46, No. 4, November 2000.

[28] A. Skodras, C. Christopoulos, T. Ebrahimi, "*The JPEG2000 Still Image Compression Standard*", *IEEE* Signal Processing Magazine, vol. 18, September 2001.

La Compression par les Ondelettes

Introduction

Les spécialistes du traitement des signaux n'étaient pas les seuls à prendre conscience des limitations des bases de Fourier [1,2]. En effet, dans les années 1970, un ingénieur-géophysicien français, Jean Morlet, s'est rendu compte qu'elles n'étaient pas le meilleur outil mathématique pour explorer le sous-sol. Et même bien avant, dès les années 1930, des physiciens s'étaient rendu compte que les bases de Fourier n'étaient pas bien adaptées pour analyser les états d'un atome ce qui a poussé des mathématiciens à tenter de les améliorer, avec les moyens du bord. Ces recherches ont conduit à la découverte de la transformée en ondelettes [1,2]. Cette théorie mathématique, fondée sur un ensemble de fonctions de base différentes des fonctions sinusoïdales utilisées dans la méthode de Fourier, va remplacer avantageusement celle-ci dans certaines situations.

Mais la découverte la plus intéressante a été faite par le mathématicien français Yves Meyer vers l'année 1980, qui a découvert les premières bases d'ondelettes orthogonales [3]. Cet outil puissant a ensuite été très étudié et développé, tant du point de vue pratique que du point de vue théorique, par des chercheurs telle que : Y. Meyer, I. Daubechies, S. Mallat, P.G. Lemarié, JP. Antoine ou R. Murenzi et d'autres [3], Ce qui a permet d'aboutir à la théorie de l'analyse par ondelettes telle qu'elle existe aujourd'hui.

Dans ce chapitre on va faire une étude bibliographique concernant les ondelettes dans le traitement de signal en général et leur implication dans la compression des images en particulier.

I. L'analyse multi-résolutions

L'analyse multi-résolution est une technique qui a été développée par Stéphane Mallat et Yves Meyer [4]. Comme son nom l'indique, le but de cette théorie est de décomposer un signal suivant différentes résolutions. On procède ainsi à une dé-corrélation de l'information qu'il contient. L'analyse multi-résolution a le même effet qu'un microscope aux pouvoirs de grossissement variables. Les basses résolutions représentent la forme grossière du signal tandis que les hautes résolutions encodent les détails du signal [5]. Dans une approche traitement du

signal, les hautes résolutions représentent les hautes fréquences et les basses résolutions représentent les basses fréquences [4].

Pour bien comprendre cette technique, imaginons qu'on veut construire un bonhomme en utilisant des briques de ligots, On place d'abord les grosses briques pour avoir une version grossière du personnage: quelques grosses briques pour le corps, les bras, les jambes et la tête, puis on ajoute des briques moyennes pour le cou, les pieds, les mains, les articulations, puis des briques un peu plus petites pour les yeux, le nez, les doigts, puis éventuellement des toutes petites pour fignoler les détails [2]. Cette approche qui permet d'utiliser un nombre d'éléments très réduit est appelé multi-résolution, car on construit un objet d'abord de manière grossière puis de plus en plus précise: en augmentant graduellement la résolution. C'est dans cet esprit qu'ont été inventées les ondelettes.

Dans le cas des images, les ondelettes sont des ensembles de pixels de différentes tailles. Pour être précis, les ondelettes ne sont pas de gros blocs rectangulaires mais des paquets plus ou moins gros qui ont la forme d'une vague, ou d'une onde d'où leur nom. Les grosses ondelettes vont permettre de construire une version grossière de l'image et les petites ondelettes vont apporter des détails plus fins dont on peut se passer. Ainsi, la transformation par les ondelettes consiste à décomposer l'image perçue comme un signal en un ensemble d'images de plus petite résolution [6].

L'avantage de la représentation multi-résolution réside dans la dualité contenu-fréquence. Contrairement à la transformée de Fourier qui projette le signal dans l'espace des fréquences, l'analyse multi-résolution représente le signal conjointement dans son espace réel et dans son domaine fréquentiel en le décomposant en sous-bandes orthogonales ou non [7]. Pour des signaux 2D comme les images, des propriétés topologiques (orientations, agencement du contenu) sont ainsi conservées après la transformation [2,6].

La décomposition par analyse multi-résolution se fait de la manière suivante [6,7]:

1. On définit d'abord une famille de fonctions d'échelle $\varphi_{j,k}(t)$ pour $j,k=1,\ldots,m$ à partir d'une fonction mère $\varphi(t)$ par la relation :

$$\varphi_{j,k}(t) = \varphi(j2^j - k) \qquad \text{II.1}$$

2. Ce choix définit les espaces vectoriels emboités $V_0 \subseteq V_1 \subseteq,\ldots, \subseteq V_m$ et les sous-espaces complémentaires W_j de V_j dans V_{j+1}:

$$W_j \oplus V_j = V_{j+1} \qquad \text{II. 2}$$

3. Définir un produit scalaire sur V_m ce qui induit le même produit scalaire sur V_0, V_1 ,..., V_m.

4. Choisir un ensemble d'ondelettes $\psi_{j,k(t)}$ servant de base aux W_j et si possible orthogonales aux $\varphi_{j,k}(t)$.

5. La base ainsi obtenue peut ensuite être normée en divisant chacune des fonctions de base par sa norme.

II. La transformée en ondelettes continue

La transformée en ondelette continue permet d'analyser le comportement local d'un signal en réduisant progressivement le paramètre d'échelle et ceci en cherchant une représentation plus compacte localisée simultanément en temps et en fréquence [6]. Cette procédure de zoom est un outil puissant pour détecter et caractériser les irrégularités d'un signal. En effet, l'idée de l'analyse par ondelettes continues est de décomposer un signal sur une base de fonction d'un espace fonctionnel ayant des propriétés biens déterminés en d'autre terme : une ondelette permet de représenter une fonction (ou un signal) comme une somme pondérée de petites ondes translatées ou dilatées issues à partir d'une fonction appelée « ondelette mère » [8].

En effet, si *f(t)* est une fonction réelle de variables réelles représentant un signal alors la transformée en ondelettes continue CWT (Continu Wavelet Transform) de *f(t)* est définie par l'équation [9] :

$$CWT(a,b) = \frac{1}{\sqrt{a}} \int_{t=-\infty}^{t=+\infty} f(t) \psi^*{}_{a,b}(t) dt \qquad \text{II. 3}$$

a≠0; et la fonction $\psi^*{}_{a,b}(t)$ étant le conjugué de $\psi_{a,b}(t)$ obtenu par translation et dilatation d'une fonction particulière appelée " **ondelette mère** " définie par [9] :

$$\psi_{a,b}(t) = \Psi\left(\frac{t-b}{a}\right) \qquad \text{II. 4}$$

* b : Facteur de translation « position/temps »

* a : Facteur de dilatation « échelle/fréquence ».
* $\Psi\left(\frac{t-b}{a}\right)$: Ondelette mère.
* $\psi_{a,b}(t)$: Ondelettes enfants.
* $\frac{1}{\sqrt{a}}$ Facteur de normalisation de l'énergie afin que le signal transformé ait la même énergie à toutes les échelles.

L'ondelette mère Ψ est une fonction de l'espace L^2 (ensemble des fonctions de carré intégrable) qui doit vérifier les propriétés suivantes [6,9] :

► $\Psi \in L^2$ donc de carré intégrable:

$$\int_{-\infty}^{+\infty} |\Psi(t)|^2 \, dt < +\infty \qquad \text{II. 5}$$

$|\Psi(t)|$ est le module de $\Psi(t)$

► Ψ est une fonction de moyenne nulle :

$$\int_{-\infty}^{+\infty} \Psi(t) dt = 0 \qquad \text{II. 6}$$

Cette relation montre que le moment d'ordre zéro de $\Psi(t)$ est nul ce qui implique que $\Psi(t)$ est oscillante. En général une ondelette a m moments nuls si et seulement si :

$$\int_{-\infty}^{+\infty} t^k \Psi(t) dt = 0 \quad avec\ k = 1 \dots\dots\dots m \qquad \text{II. 7}$$

► $\Psi(t)$ doit vérifier la condition d'admissibilité suivante :

$$\int_{0}^{+\infty} \frac{|TF(\Psi(\omega))|^2}{\omega} < +\infty \qquad \text{II. 8}$$

$TF(\Psi(\omega))$ est la transformée de Fourier de la fonction $\Psi(t)$ établie par l'équation I.16 dans le chapitre I.

► $\Psi(t)$ doit être d'énergie finie :

$$|\Psi(0)|^2 = 0 \qquad \text{II. 9}$$

La fonction *f(t)* peut être reconstruite à partir de la transformée CWT(a,b) par la formule suivante [10,11]:

$$f(t) = \frac{1}{C_\psi} \int_{a=-\infty}^{+\infty} \int_{b=-\infty}^{b=+\infty} \frac{CWT(a,b)}{a^2} \psi_{a,b}(t) dadb \qquad \text{II. 10}$$

C_ψ est une constante qui dépend de l'ondelette mère $\Psi(t)$ choisie. Il s'agit de la transformée en ondelette continue inverse ICWT (Inverse Continu Wavelet Transform).

Il existe une infinité de fonctions d'ondelettes parce que toute fonction oscillante localisée est une ondelette mère possible. Toutefois, elles ne possèdent pas toutes des propriétés intéressantes [12]. Parmi les familles d'ondelettes les plus célèbres et les plus simples on trouve les ondelettes de Haar qui sont les seules ondelettes symétriques à support temporel fini [13] et les ondelettes de Daubechies qui sont à support compact et qui permettent d'utiliser des filtres de taille finie [14]. Une autre famille d'ondelettes est la famille des ondelettes splines dont la réponse fréquentielle est bien localisée [15]. Les différentes familles d'ondelettes sont utilisées selon leurs propriétés en fonction du problème à résoudre.

Cette transformation en ondelette continu reste en théorie infiniment redondante puisque l'ondelette est translatée de manière continue, cependant il existe des méthodes pour diminuer cette redondance et l'une de ces méthodes consiste en l'emploi de la transformée en ondelettes discrète DWT (Discret Wavelet Transform) [16].

III. La transformée en ondelettes discrète

La transformée en ondelettes discrète DWT a été découverte pour surmonter le problème de redondance de la transformée en ondelettes continu CWT. En effet, la transformée en ondelette DWT fournit suffisamment d'information, tant pour l'analyse que pour la reconstruction du signal original en un temps de calcul notablement réduit [16,17].

La transformée en ondelette discrète peut être calculée à partir d'un signal discret, comme les images, en translatant et dilatant l'ondelette mère selon des valeurs discrètes [15]. Ses coefficients a et b seront discrétisés de la manière suivante :

$$a = a_0^j \;\; et \;\; b = k b_0 a_0^j \;\; avec \; a_0 > 1 \; et \; b_0 > 0 \qquad \text{II. 11}$$

a_0 et b_0 des entiers fixes appartenant à Z.
Les ondelettes sont alors définies de la manière suivante [16] :

$$\psi(a_0, b_0) = \frac{1}{\sqrt{a_0^j}} \Psi\left(\frac{t - k a_0^j b_0}{a_0^j}\right) \qquad \text{II. 12}$$

La transformée DWT de la fonction f(t) est donnée par la formule suivante [16]:

$$DWT(a_0, b_0) = \frac{1}{\sqrt{a_0^j}} \int_{-\infty}^{+\infty} f(t)\ \Psi\left(\frac{t - k a_0^j b_0}{a_0^j}\right) dt \qquad \text{II. 13}$$

Tel que :

- a_0^j : Facteur d'échelle.
- b_0 : Facteur de translation.
- k et j : Des entiers.

La figure II.1 représente quelques ondelettes célèbres [16]:

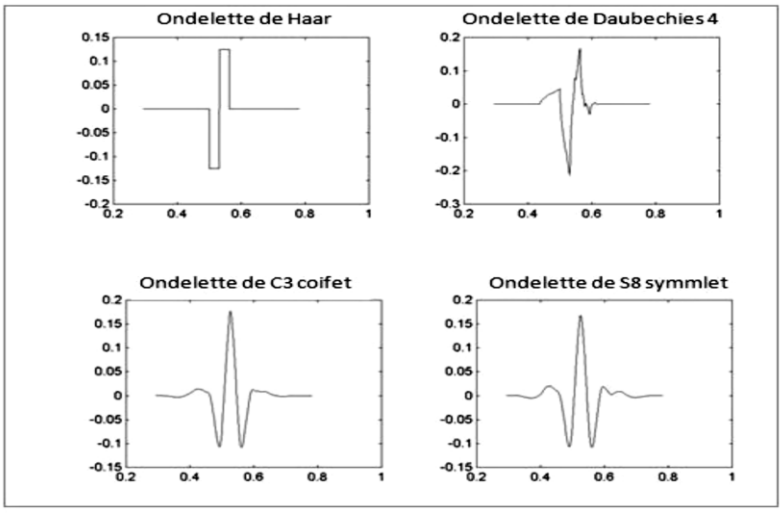


Figure II.1 : Quelques types d'ondelettes

III.1 Transformée d'ondelette de Haar

Le mérite revient à Alfred Haar d'avoir construit en 1909 [12] des bases considérées aujourd'hui comme le fondement de la théorie des ondelettes.

Pour un signal représenté par la fonction *x(t)* définie sur l'intervalle [0, 1] et qui est échantillonnée en $\mathrm{N} = 2^{\mathrm{m}}$ points, m étant un entier. On note :

$$x_k = x(t_k) \quad pour\ t_k = 0,\ \frac{1}{N}, \ldots\ldots, \frac{N-1}{N} \qquad \text{II. 14}$$

Haar a défini la fonction [16,18]:

$$\phi(t) = \begin{cases} 1 & pour\ 0 \le t \le 1 \\ 0 & autrement \end{cases} \quad \text{II. 15}$$

Et les fonctions [13,16] :

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k) \quad \text{II. 16}$$

Avec :

$$\phi_{j,k}(t) = \begin{cases} 1 & pour\ \frac{k}{2^j} \le t \le \frac{k+1}{2^j} \\ 0 & autrement \end{cases} \quad \text{II. 17}$$

On peut reconstruire le signal x(t) de la façon suivante [13,16] :

$$x(t) = \sum_{k=0}^{k=N-1} x_k \phi_{m,k}(t) \quad \text{II. 18}$$

Les fonctions $\phi_{j,k}(t)$ vérifient les propriétés suivantes [19] :

- Elles forment une base orthogonale de $L^2(\mathbb{R})$:

$$\langle \phi_{j,k}, \phi_{j,l} \rangle = 0 \ \text{ pour } k \neq l \quad \text{II. 19}$$

- Elles forment une base orthonormée :

$$\langle \phi_{j,k}, \phi_{j,k} \rangle = \left\| \phi_{j,k} \right\|^2 = 1 \quad \text{II. 20}$$

- Pour chaque j, elles engendrent un sous espace vectoriel noté V_j de dimension 2^j de fonctions réelles intégrables sur [0, 1[.

La fonction $\phi_{j,k}(t)$ est appelée **la fonction d'échelle de Haar** [19].

De plus, Haar a défini la famille des ondelettes [13]:

$$\psi(t) = \begin{cases} 1 \ pour\ 0 \le t \le \frac{1}{2} \\ -1 \quad pour\ \frac{1}{2} \le t \le 1 \\ 0 \quad ailleurs \end{cases} \quad \text{II. 21}$$

Et: $$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \quad \text{II. 22}$$

Soit : $$\psi(t)_{j,k} = \begin{cases} 1 & pour\ \ \frac{k}{2^j} \le t < \frac{k}{2^j} + \frac{1}{2^{j+1}} \\ -1 & pour\ \ \frac{k}{2^j} + \frac{1}{2^{j+1}} \le t \le \frac{k+1}{2^j} \\ 0 & ailleurs \end{cases} \quad \text{II. 23}$$

Les fonctions $\psi_{j,k}(t)$ sont orthogonales pour tout $j \neq k$ et forment aussi une base orthonormée de $L^2(\mathbb{R})$ et pour une même valeur de j elles définissent un espace vectoriel W_j de dimension 2^j [19].

Les deux espaces vectoriels W_j et V_j sont complémentaires donc ils vérifient la relation de complémentarité suivante :

$$V_j \oplus W_{j=} V_{j+1} \qquad \text{II. 24}$$

Avec : V_{j+1} est l'espace vectoriel de dimension 2^{j+1} engendré par les fonctions $\phi_{j+1,k}(t)$. Les fonctions $\psi_{j,k}(t)$ sont appelées **les ondelettes de Haar** [13,19].

L'ondelette de Haar n'a qu'un seul moment nul. Si on effectue une projection d'une fonction sur cette base, la projetée aura une allure de fonction en escalier [13]. Si on désire plus de régularité, il faut utiliser des fonctions avec plus de moments nuls.

III.2 Transformée d'ondelette bi-orthogonale

La notion de bi-orthogonalité a été introduite en 1987 par Tchamitchian [2,6] et celle d'ondelettes bi-orthogonales en 1992 par Cohen et all. [20]. Les ondelettes bi-orthogonales permettent d'introduire une certaine souplesse par rapport aux ondelettes orthogonales. En effet, la base permettant l'analyse (phase de décomposition en coefficients d'ondelettes) ne sera pas la même que celle permettant la synthèse (phase de reconstruction de la fonction) d'où des conditions moins strictes pour la construction des ondelettes.

Pour définir les ondelettes bi-orthogonales, il est nécessaire d'introduire les fonctions duales $\tilde{\psi}(t)$ et $\tilde{\phi}(t)$ respectivement des fonctions $\psi(t)$ et $\phi(t)$ qui vérifient l'équation suivante [21]:

$$\begin{cases} \langle \phi_{j,k} , \tilde{\phi}_{j,l} \rangle = \delta_{k,l} \\ \langle \psi_{j,k} , \tilde{\psi}_{j,l} \rangle = \delta_{k,l} \end{cases} \qquad \text{II. 25}$$

Avec : $$\delta_{k,l} = \begin{cases} 1 \ \ pour \ \ l = k \\ 0 \ \ ailleurs \end{cases} \qquad \text{II. 26}$$

$\delta_{k,l}$ étant le symbole de Kronecker

Donc une ondelette bi-orthogonale est telle que la fonction d'échelle originale est orthogonale à l'ondelette duale et de même l'ondelette originale est orthogonale à la fonction d'échelle duale. Autrement dit, les ondelettes bi-orthogonales vérifient l'équation suivante [21] :

$$\begin{cases} \langle \phi_{j,k} , \tilde{\psi}_{j,l} \rangle = 0 \\ \langle \psi_{j,k} , \tilde{\phi}_{j,l} \rangle = 0 \end{cases} \quad pour \ \ tout \ j,k,l \qquad \text{II.27}$$

Les ondelettes bi-orthogonales les plus utilisées sont appelées « splines » elles ont la fonction d'échelle $\phi(x)$ qui est linéaire quadratique et cubique. L'unique problème qui s'oppose en utilisant ces ondelettes c'est que la transformée nécessite deux ondelettes au lieu d'une. En plus, elles peuvent toutes deux avoir des régularités très différentes [6].

III.3 Application de la transformée DWT aux images

La décomposition d'une image en sous-bandes par une ondelette s'effectue en utilisant deux filtres à réponse impulsionnelle finie, l'un passe-haut Ha(n) qui permet d'avoir les hautes fréquences, l'autre passe-bas La(n) qui ne filtre que les basses fréquences. Cette étape est appelée étape *d'analyse*. La reconstruction de l'image, aussi appelée étape de *synthèse*, nécessite également l'utilisation de deux filtres Hs(n) et Ls(n) qui peuvent être construits à partir des filtres Ha(n) et La(n) [22]. Ces derniers sont connus sous le nom de filtres à miroirs quadratiques QMF (Quadratique Miror Filtre) [23].

Le fonctionnement de la DWT repose sur une série de cycles successifs à deux temps [22]:

1er temps :

On calcule la moyenne des pixels de l'image d'origine deux à deux en suivant l'axe horizontal, on obtient en sortie une image sous-échantillonnée appelée **« image d'approximation »** de même longueur que l'origine et de largeur la moitié (largeur initiale de l'image divisée par 2). En même temps, on crée une 2éme image en calculant les écarts (erreurs) entre l'image sous-échantillonnée et l'image d'origine toujours dans le sens vertical pour obtenir en sortie « l'image de détails ».

2ème temps :

On moyenne les pixels des images d'approximation et de détails dans le sens vertical pour obtenir, pour chacune, deux matrices de résolution la moitié de celle

de l'image d'origine. Et en calculant les écarts entre l'image d'approximation et ces deux sous-images d'une part et l'image de détails et ces sous-images d'autre part, on obtient quatre images qui sont : **l'image d'approximation** qui contient les importantes informations sur l'image d'origine qui sont les basses fréquences et les trois autres **images de détails** : horizontaux, verticaux et diagonaux. Les sous-bandes de résolutions supérieures (les plus grandes) possèdent un contenu qui est relativement faible alors que les sous-bandes basse-fréquences (les plus petites) sont beaucoup plus riches en contenus.

Ces deux temps constituent le premier niveau de décomposition en ondelettes d'une image ainsi le processus peut être répétée autant de fois que nécessaires. Les étapes de décomposition et de reconstitution d'une image sont illustrées sur les figures II.2 et II.3 [24].

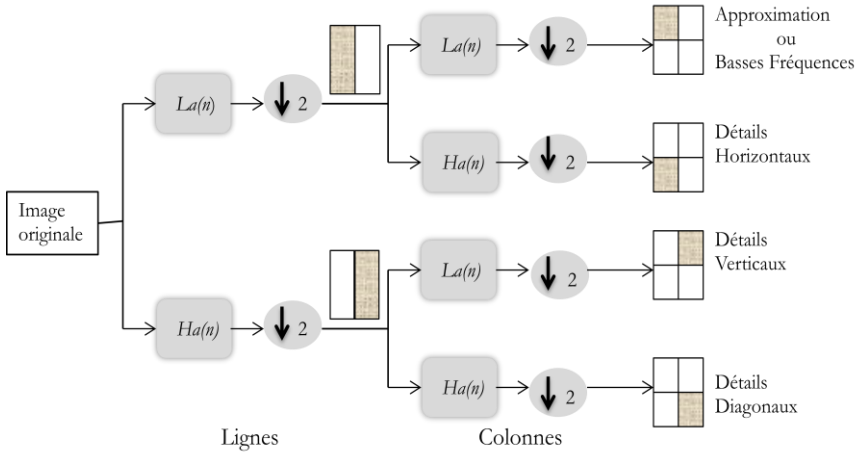


Figure II.2 Décomposition multi-résolution d'une image

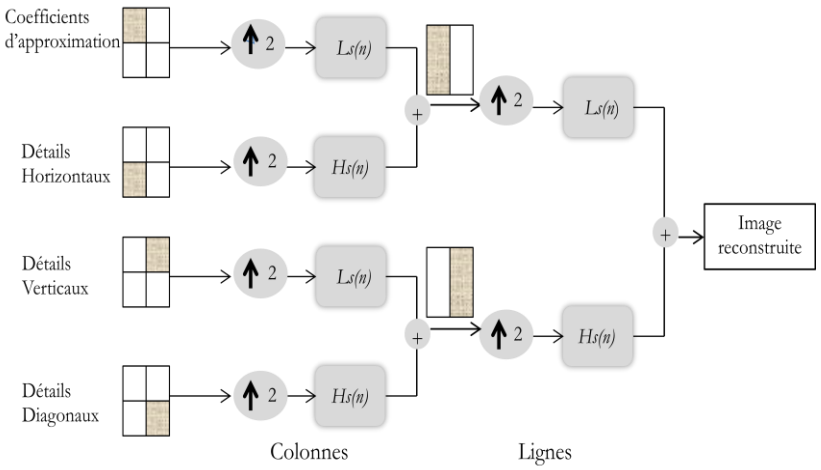


Figure II.3 Reconstruction d'une image décomposée par les ondelettes.

Illustration sur une image réelle :

L'utilisation de la transformée en ondelettes va permettre de représenter une image initiale « Cameraman » de résolution 512×512 donnée dans la figure II.4.a, sous la forme d'une image **d'approximation** et des images **de détails,** figure II.4.b [25].

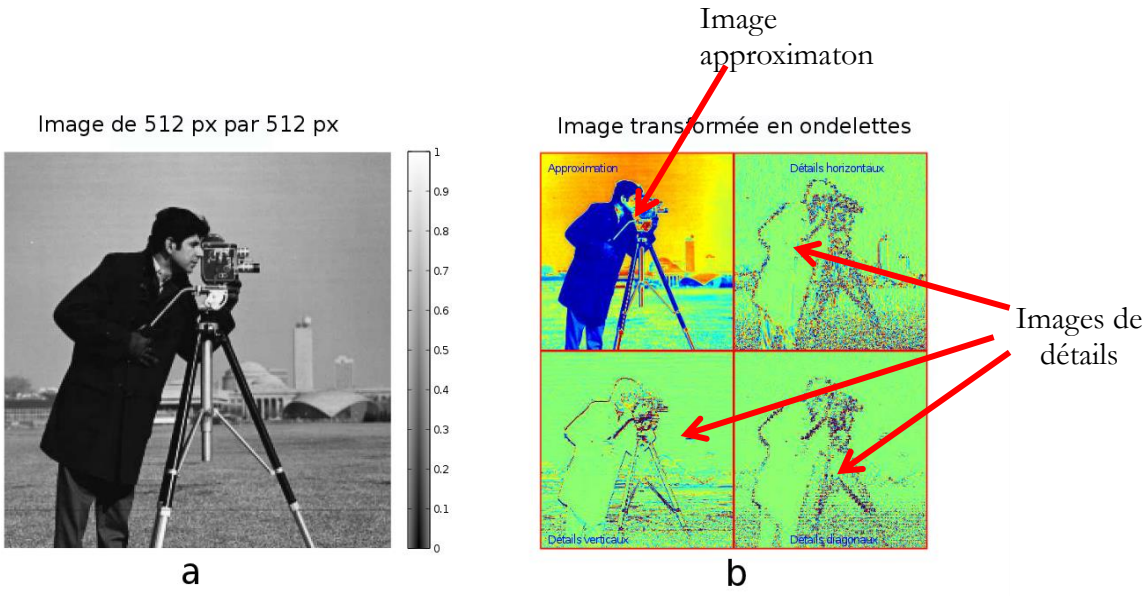


Figure II.4 : Décomposition multi-résolution de l'image « cameraman » au niveau 1

III.4 Autres application des ondelettes en traitement d'images

III.4.1 La détection de contours

La détection de contours est une tâche ardue lorsque les images traitées présentent des variations brusques dans des zones intéressantes. C'est pour cela qu'il est préférable d'ignorer certains contours et ne conserver que les plus représentatifs [25]. En fonction de l'application qu'on veut, on peut conserver uniquement les contours principaux ou bien l'aspect de texture. Ce type d'analyse est possible grâce au principe de la multi-résolution qui caractérise l'utilisation des ondelettes. En analysant l'image à une résolution grossière les informations non intéressantes peuvent être éliminées.

III.4.2 La réduction de bruit

Le signal électrique à partir duquel on construit l'image numérique est toujours accompagné d'un bruit. Les causes de ce phénomène sont variées mais son élimination reste une étape indispensable pour laquelle les recherches se font

actuellement par différents moyens. La transformée en ondelettes en fait partie, à cause de son analyse multi-résolution, qui est un outil très efficace pour la réduction du bruit dans l'image numérique. En effet, comme nous l'avons vu, les coefficients d'ondelettes marquent les discontinuités qui interviennent dans l'image. Ils correspondent donc aux détails. Si, maintenant, on fixe un seuil pour ces coefficients, cela permet d'éliminer les détails les plus fins de l'image en les mettant à zéro et on reconstruit l'image qu'avec les coefficients les plus grands [26]. Le bruit correspond en général à des détails faibles donc il est éliminé par ce seuillage des coefficients d'ondelettes. Il en découle le dé-bruitage (ou "*denoising*"), nous obtenons alors une image plus "lisse" donc dé-bruitée [27,28]. Un exemple est montré sur la figure II.5.

Image « Mandrill » initiale | Image bruitée | Après seuillage

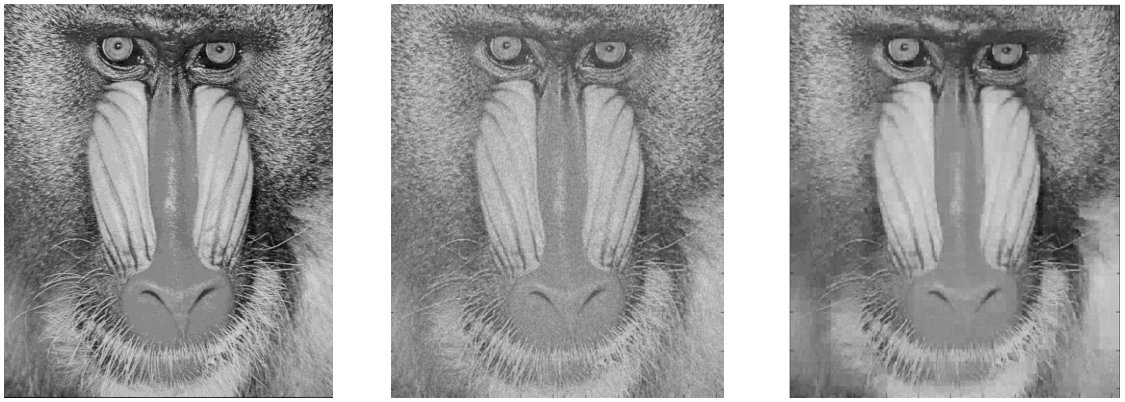


Figure II.5 : Dé-bruitage de l'image « Mandrill » par les ondelettes.

III.4.3 Détection et reconnaissance de texture

L'analyse multi-résolution apporte un avantage considérable dans le domaine de la reconnaissance de texture [29], puisque dans ce cas l'échelle est prise en compte en plus des paramètres habituels de détection comme les motifs.

Bien évidemment, la transformée en ondelettes ne se limite pas aux applications mentionnées ci-dessus. On peut également citer la reconnaissance de visage, la détection de mouvements, et d'une manière générale la plupart des applications reposant sur l'analyse d'images. D'autre part, un avantage non négligeable de la transformée en ondelettes est qu'elle n'est pas liée à une fonction prédéfinie, comme l'est la transformée de Fourier qui utilise les fonctions sinus et cosinus exclusivement. Ainsi, le choix de l'ondelette utilisée pour l'analyse pourra dépendre de l'application envisagée, voire même du type de données traitées.

IV Compression des images par les ondelettes

En compression d'image, l'utilisation des ondelettes a permis d'adapter le taux de compression à la qualité de vision de l'image désirée [17]. En effet, la décomposition en sous-bandes d'une image conduit à l'élimination des détails contenus dans certains niveaux de résolution, jugés peu importants, et de ne conserver que l'information nécessaire. Ainsi, on peut diminuer l'espace mémoire occupée par l'image d'origine et par la suite augmenter le taux de compression et l'adapter à la qualité désirée [15,17]. Si l'on désire conserver une bonne qualité d'image, il suffira alors de conserver tous les détails envisagés.

Bien que la librairie des ondelettes contienne un nombre important d'ondelettes, le choix de l'ondelette adaptée n'est pas facile [30]. Il convient de bien cerner le problème à étudier et d'identifier le type de transformée à utiliser (continue ou discrète). En général, un système de compression d'images basé sur les ondelettes est composé de trois principales étapes comme indiqué par la figure II.6 [9] :

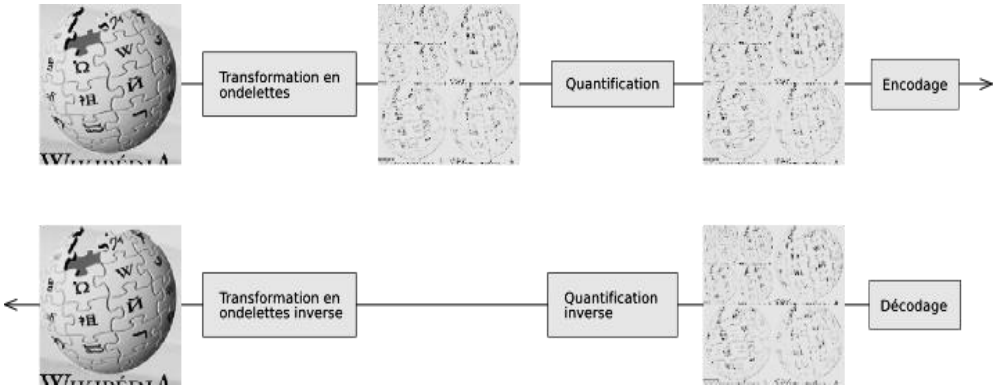


Figure II.6 : Schéma d'un système de compression et de décompression d'une image basée sur les ondelettes.

► **1ére étape :**

C'est l'étape de prétraitement de l'image, elle consiste à décomposer l'image en sous-bandes ou blocs rectangulaires pour obtenir les matrices d'approximation et celles de détails en utilisant la décomposition par l'ondelette convenable. Le but de cette étape est de compacter au mieux l'information contenue dans l'image, c'est à dire d'avoir un nombre de coefficients représentatifs aussi faible que possible.

► **2eme étape :**

Un processus de quantification, scalaire ou vectorielle, est ensuite entamé pour approximer les valeurs des pixels transformés de l'image constituant la matrice d'approximation avec un pas de quantification convenable. C'est au niveau de la quantification qu'on assiste à une perte d'information contenue dans l'image, perte mesurée par le facteur PSNR (Peak Signal to Noise Ratio). La quantification est donc la phase qui provoque une dégradation dans l'image reconstruite mais c'est aussi cette opération qui permet d'obtenir des taux de compressions beaucoup plus importants que dans le cas d'une compression sans perte [26,31].

► **3^{éme} étape :**

Après la quantification vient le codage qui consiste à coder les pixels quantifiés pour préparer l'image à la transmission ou au stockage. C'est cette étape qui va exploiter au mieux la redondance d'information qui s'est accumulée après les étapes de transformation et de quantification. La reconstitution de l'image compressée se fait en suivant le chemin inverse du processus de compression comme indiqué dans la figure II.6.

Illustration sur une image réelle

Le principe de la transformée en ondelettes et compression d'une image est présenté dans le figure II.8. On effectue trois fois de suite l'itération sur l'image initiale du « Cameraman » de la figure II.7, ainsi on passe d'une approximation de 256 x 256 pixels et 3 types de détails (horizontaux, verticaux et diagonaux) à une approximation de 64 x 64 pixels et toujours 3 niveaux de détails donnés par les coefficients d'ondelettes. La figure II.8.a représente la transformée en ondelettes compressée lorsque l'on garde 20 % des coefficients durant la quantification, avant de reconstruire l'image. La différence entre l'image reconstruite figure II.8.c et l'image initiale est quasi-invisible. Dans la figure II.8.d, on a gardé seulement 4 % des coefficients avant de reconstruire l'image et l'on peut voir que des artéfacts commencent à apparaître.

Figure II.7 : Image originale « Cameraman » de résolution 256 x256

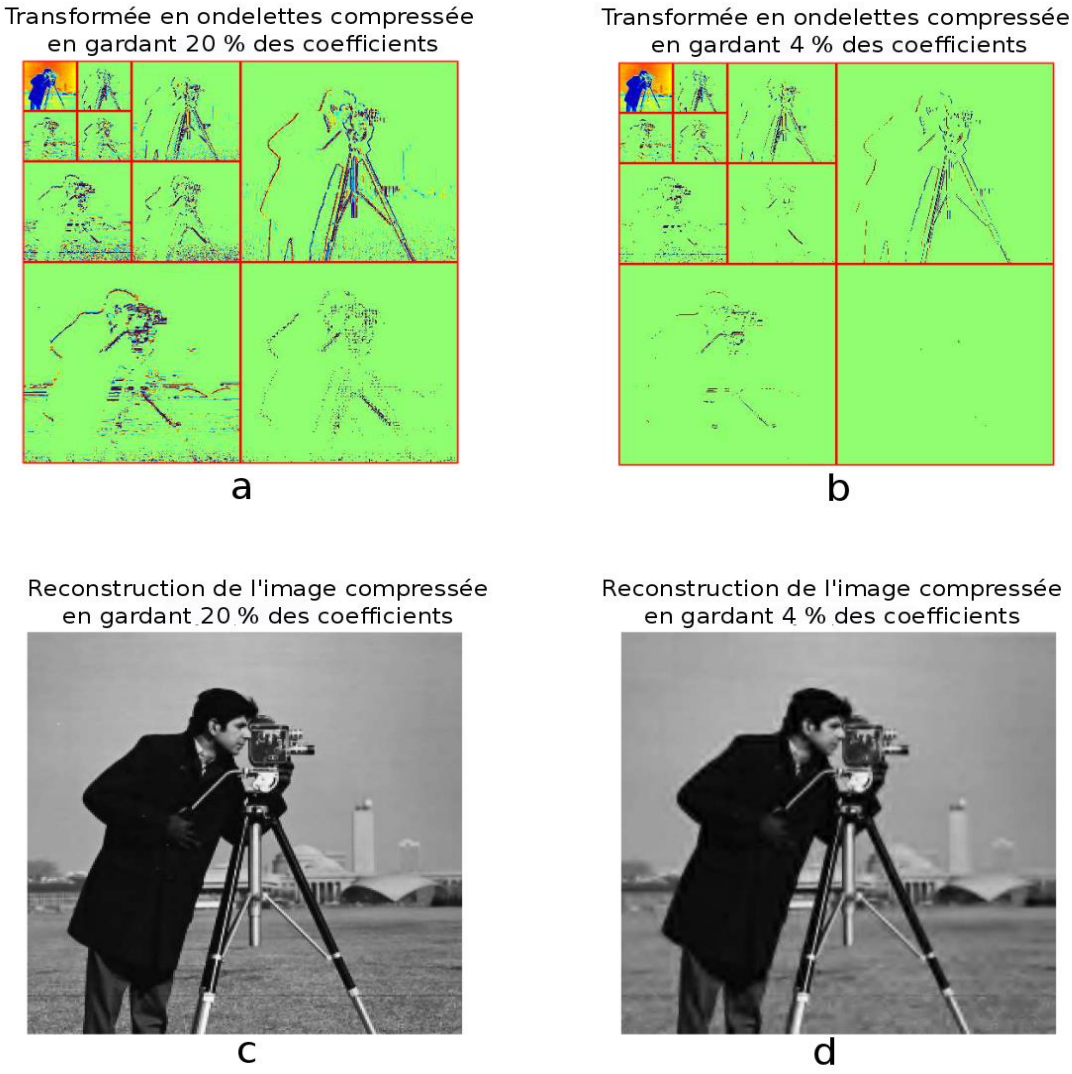


Figure II.8 : Principe de compression de l'image « Cameraman » par les ondelettes

Conclusion

La transformée en ondelettes présente de nombreux avantages dans le domaine du traitement du signal et de l'image. L'analyse par ondelettes ne se limite plus à l'image telle qu'elle nous apparaît, mais permet l'étude des objets présents dans l'image à différentes échelles, elle permet de réduire la redondance pour améliorer la compression d'une image, elle peut également extraire les informations importantes (texture, contours, etc.) contenues dans une image et aussi de réduire le bruit contenu dans l'image. C'est un outil puissant de transformation du signal qui permet de «préparer» le signal afin de faciliter son traitement. En plus cette technique possède de nombreux avantages tel que:

- Elle permet de prévoir le taux de compression contrairement à d'autres techniques comme le JPEG.
- Elle n'entraîne pas d'effet de mosaïque.
- L'algorithme est plus simple et plus souple donc plus rapide.
- Il est possible d'avoir des images très compactes (de l'ordre du ko).

Bibliographie

[1] N. Moreau. "*Techniques de compression des signau*x,". Edition Masson, 1994.

[2] A. Isar, A. Cubitchi, M. Nafornita, "*Algorithmes et techniques de compression,*". Edition orizonturi politehnice 2002.

[3] Y. Meyer. "*Ondelettes, filtres miroirs en quadrature et traitement numérique de l'image,*". P. G. Lemarié (editeur), Springer-Verlag, 1990.

[4] S.G. Mallat, "*Multifrequency channel decomposition of images and wavelet models*,". IEEE trans. Acoust, speech, Signal processing, volume. 37, 1989.

[5] S. G. Mallat "*A Theory for Multiresolution Signal Decomposition: The Wavelet Representation,*". IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989.

[6] A. Cohen. "*Ondelettes et traitement numérique du signal,*". *Masson*, 1992.

[7] O. Rioul, "*A discrete time multiresolution theory,*". *IEEE Trans.on SP*, volume. 41, no. 8, août 1993.

[8] S.G. Mallat, "*A wavelet tour of signal processing,* ". Academic Press, Second Edition 2001.

[9] M. Misiti, Y. Misiti, G. Oppenheim, JM Poggi, "*Les ondelettes et leurs applications,* ". Hermes Sciences, 2003.

[10] A. Calderon, "*Intermediate space and interpolation, the complex method,* ". Stud. Math, 1964.

[11] A. Grossman, J. Morlet, "*Decomposition of Hardy functions into square intergrable wavelets of constant shape,* "SIAM Journal of Math. juillet 1984.

[12] É. Incerti, "*Compression d'image – Algorithmes et standards*, ". Vuibert 2003.

[13] K.H. Talukder, K.Harada, "*Haar Wavelet based approach for image compression and quality assessment of compressed image,*". IAENG, International Journal of Applied Mathematics, 2007.

[14] I. Daubechies. "*Orthonormal Bases of Compactly Supported Wavelets*". *Comm. Pure. Appl. Math.*, No. 41, 1988.

[15] J-P. Guillois, "*Techniques de Compression des Images,*". Collection informatique, Hermes, 1996.

[16] F. Truchetet, "*Ondelettes pour le signal numérique,* ". Editions Hermes, Paris, 1998.

[17] M. Rabbani, P.W. Jones, "*Digital image compression techniques*," vol.7, SPIE Press, Bellingham, Washington, USA, 1991.

[18] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, "*Image Coding using Wavelet Transform,*". IEEE Trans. Image Process. 1(2), 1992.

[19] S. Mallat, "*A wavelet tour of signal processing,*". Academic Press, Second Edition 2000.

[20] I. Daubechies. "*Ten Lectures on Wavelets,*". *SIAM*, Philadelphia 1992.

[21] A. Cohen, I. Daubechies, J.C. Feauveau, "*Bi-orthogonal bases of compactly supported Wavelets, "*. Comm. Dans Pure and Applied Math. volume XLV.

[22] J.W. Woods, S.D. O'Neil, "*Sub-bands Coding of Images,*". IEEE Trans. Acoust, Speech, Signal Processing, vol. 34, 1986.

[23] D. Esteban et C. Galland, "*Application of quadrature mirror filters to split-band voice coding schemes,*". Proceedings IEEE International Conference on Acoustics and Signal Speech Processing 1977, Hardford.

[24] J. Froment. "*Traitement d'images et applications de la transformée en ondelettes,*". Thèse de doctorat, Université Paris IX, France (1990).

[25] C. Damerval "*Ondelettes pour la détection de caractéristiques en traitement d'images Applications à la détection de régions d'intérêt*,". Thèse doctorat de l'université Joseph Fourier, Grenoble I, France (2008).

[26] K. Ratakondu, N. Ahuja, "*Loless Image Compression with Multiscale Segmentation,*" IEEE, Transactions on Image Processing, vol.11, N°11, 2002.

[27] S. Mallat, "*Une exploration des signaux en ondelettes, "* Editions de l'Ecole Polytechnique, 2000.

[28] D. L. Donoho, "*De-Noising By Soft-Thresholding,*" IEEE Transactions on Information Theory, 1994.

[29] O. Le Cadet, "*Méthodes d'ondelettes pour la segmentation d'images. Applications à l'imagerie médicale et au tatouage d'images,*". Thèse de doctorat de l'institut national polytechnique de Grenoble, France (2004).

[30] T. Gandhi, B.K. Panigrahi, S. Anand, "*A Comparative Study of Wavelet Families for EEG Signal Classification,*" Neurocomputing 74, 2011.

[31] P. Mathieu, M. Barlaud, M. Antonini, "*Compression d'Images par Transformée en Ondelette et Quantification Vectorielle*", Journal Traitement de Signal vol7, N°2, 1990.

Les réseaux de neurones artificiels

Introduction

Le principe du réseau de neurones est de reproduire schématiquement le cerveau humain et son mode d'association afin de l'automatiser dans une machine : c'est ce qu'on appelle l'intelligence artificielle [1]. Comme notre cerveau, le réseau de neurones débute par une phase d'apprentissage, avec l'association de données d'entrées (image, son, texte…) à des catégories spécifiques comme références (ex : reconnaissance de mots à partir de la photo d'un texte). Une fois cet apprentissage effectué, lorsque de nouvelles données d'entrées lui seront présentées, le réseau de neurones les associera à telle ou telle catégorie en fonction de ce qu'il aura appris : c'est la classification [1,2]. D'autre part, le réseau de neurones fonctionne par analyse statistique, par opposition à d'autres systèmes capable d'apprendre et qui prennent des décisions par comparaison à un ensemble de règles déductives comme la logique algorithmique etc… [3].

Aujourd'hui en intelligence artificielle, on en est encore qu'aux débuts. On arrive tout juste à créer des comportements intelligents primitifs à l'aide des réseaux de neurones, appelés aussi les réseaux neuro-mimétiques [1] qui interviennent dans différents domaines de la vie courante : classifier des espèces, reconnaissance de motifs (chèques, cartes postales…), estimation boursière…. En effet, les progrès théoriques, l'expérience croissante et l'étude de nouvelles structures neuro-mimétiques semblent très prometteurs pour la résolution quasi-optimale de nombreux problèmes de grande complexité dans plusieurs domaines comme [1,2,3]:

- Télécommunications : compression des images et de données, transmission de la parole en temps réel, reconnaissance de caractères et de signatures, reconnaissance de forme, cryptage…
- Médecine: analyse des cellules cancéreuses, analyse des signaux EEG et ECG, optimisation du temps de transplantation…
- Aéronautique : pilote automatique des avions, détection des anomalies dans les avions, simulation de boite noire, prévision météorologique…
- Militaire: orientation et direction des armes, suivi des cibles, reconnaissance faciale, identification du signal / image…

- Finance : évaluation immobilière, conseiller en prêts, analyse financière d'entreprise, prédiction de la valeur de change, évaluateurs de demandes de crédit, dépistage hypothécaire…
- Industrie: contrôle de processus de fabrication, conception et analyse de produits, système d'inspection de la qualité, analyse de la conception des produits chimiques, analyse de la maintenance des machines…
- Traitement du signal : filtrage, identification de la source, traitement et classification de la parole, conversion d'un texte en parole…
- Transport: diagnostic du système de freinage des camions, planification des véhicules, systèmes de routage, commande de processus, diagnostic, contrôle qualité, asservissement…
- Electronique : prédictions des séquences de codage, mise en couches des puces électroniques IC, analyse des anomalies des puces, synthèse vocale…
- Automobile: systèmes de guidage des automobiles…

Dans ce qui suit, nous allons décrire l'approche neurale en détail en définissant entre autre le fonctionnement des réseaux de neurones artificiels, les différentes architectures possibles, les différentes technique d'apprentissage etc…. et ceci après un bref historique sur les débuts d'utilisation des réseaux de neurones.

I. Historique

Le champ des réseaux neuronaux a démarré par la présentation en *1943* par W. Mac Culloch et W. Pitts d'un neurone formel mimant le neurone biologique et capable de mémoriser des fonctions booléennes simples [3,4]. Les réseaux de neurones artificiels réalisés à partir de ce type de neurones sont ainsi inspirés du système nerveux. Ils sont conçus pour reproduire certaines caractéristiques des mémoires biologiques par le fait qu'ils sont [4]:

- Massivement parallèles ;
- Capables d'apprentissage ;
- Capables de mémoriser l'information dans les connexions entre les neurones ;
- Capables de traiter des informations incomplètes.

En *1949*, D. Hebb a mis en évidence l'importance du couplage synaptique dans l'apprentissage par renforcement ou dégénérescence des liaisons inter-neuronales lors de l'interaction du cerveau avec le milieu extérieur [3,4].

En *1958*, F. Rosenblatt développa le premier modèle opérationnel appelé le Perceptron qui est inspiré du neurone visuel et considéré comme étant le premier système artificiel capable d'apprentissage [3,4]. Dans la même période, le modèle de L'Adaline (ADAptive LINar Element) a été présenté par B. Widrow et Hoff. Ce modèle sera par la suite le modèle de base des réseaux multicouches [3,4].

En *1969*, les mathématiciens M. Minsky et S. Papert publièrent une critique des propriétés du Perceptron monocouche montrant ses limites du point de vue performance. Cela va avoir une grande incidence sur la recherche dans ce domaine qui va fortement diminuer jusqu'en *1972*, où T. Kohonen présenta ses travaux sur les mémoires associatives et proposa des applications à la reconnaissance de formes [3,4].

En *1982*, Hopfield a remis à la mode le connexionnisme en donnant une architecture solide inspirée de la physique en montrant que les réseaux de neurones étaient capables de résoudre des problèmes d'optimisation et ceux de Kohonen ont montré qu'ils étaient capables de résoudre des tâches de classification et de reconnaissance [3,4].

II. Présentation générale

Le cerveau humain est la machine de traitement de l'information la plus sophistiquée, en effet même si l'être humain est incapable de réaliser des calculs aussi rapides et précises que l'ordinateur, il est le mieux adapté que ce dernier aux problèmes de traitement de l'information dans son ensemble : perception, traitement du signal, extraction de caractéristiques complexes et prise de décision. De telles tâches sont accomplies par l'être humain avec une grande rapidité et le cerveau adapte son comportement aux situations selon deux processus [5]:

- L'adaptation rapide et automatique comme le reflexe
- L'apprentissage qui autorise une adaptation lente d'un individu à l'exécution d'une nouvelle tâche.

Toutes ces remarques et constatations ont constitué le déclencheur motivant qui a conduit à la schématisation du fonctionnement du premier model du neurone artificiel de Mac Culloch & Pitts qui est une simplification extrême du neurone biologique dont le comportement temporel est beaucoup plus complexe. Et pourtant c'est ce neurone statique simplifié qui est à la base de la plupart des systèmes mettant en œuvre des réseaux neuronaux sur des applications. Avant

d'expliquer comment fonctionne un réseau de neurones artificiels il est préférable de comprendre comment fonctionne le neurone humain.

II.1. Le neurone biologique

II.1.1 Description

Le fonctionnement général d'un neurone consiste à recevoir et émettre des signaux électriques en utilisant ses principaux éléments qui sont, d'après la figure III.1, le corps cellulaire ou le noyau, l'axone, les dendrites et les synapses ou la connexion synaptique [6].

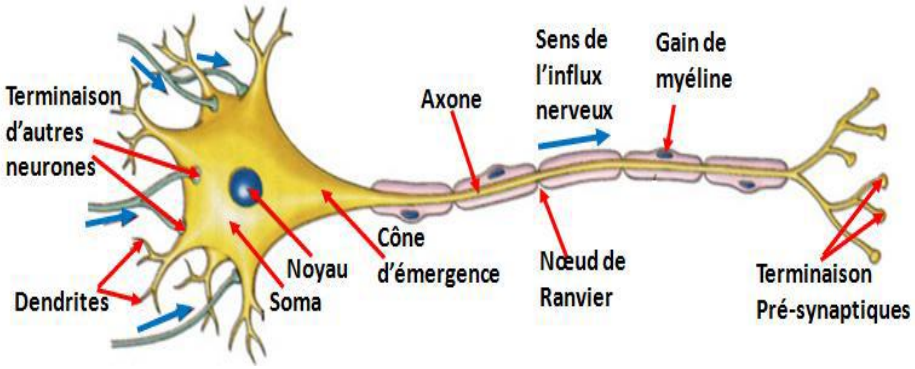


Figure III.1 : Schéma d'un neurone biologique

- Les dendrites sont de fins tubes de quelques dixièmes de micron de diamètre et d'une longueur de quelques dizaines de microns. Elles se présentent sous la forme de petites branches qui se ramifient à leur extrémité. Elles forment une sorte d'arborescence autour du corps cellulaire pour faciliter l'acheminement des informations vers le neurone [5,6,7].
- Le corps cellulaire contient le noyau du neurone et effectue les transformations biochimiques nécessaires à la vie du neurone [5,6,7].
- L'axone est la fibre nerveuse qui permet de transporter les signaux émis par le neurone. Sa membrane externe possède certaines propriétés, il est plus long que les dendrites et se ramifie à son extrémité là où il communique avec les autres neurones [5,6,7].
- Les synapses constituent la jonction entre deux neurones. En effet, deux neurones sont connectés entre eux via les synapses et ils ne sont pas directement reliés l'un à l'autre, mais sont séparés par un petit espace, appelé espace synaptique. En plus, certains traitements sont effectués à leurs niveaux [5,6,7].

II.1.2 Fonctionnement

Au niveau de chaque neurone, l'information qui vient d'un autre neurone ou provient de capteurs sensoriels tels que le touché, la vue, l'odorat…. ou d'autres capteurs internes à l'organisme, est traitée par les dendrites en effectuant une intégration des signaux reçus au cours du temps. C'est en quelque sorte une sommation des signaux électriques reçus qui se produit au niveau de chaque neurone puis elle est acheminée le long des axones jusqu'aux synapses [5,7]. Lorsqu'une impulsion électrique atteint la terminaison d'un axone, des neuromédiateurs sont libérés et se lient à des récepteurs post-synaptiques présents sur les dendrites ce qui donne un effet excitateur ou inhibiteur [5,7]. Cette opération ne se fait pas d'une manière linéaire mais il y'a un seuil à respecter. Quand ce dernier est dépassé, le neurone émet un signal électrique [7]. Ainsi les neurones communiquent en émettant des trains de potentiels rapides et très courts de l'ordre de quelques millisecondes. En générale, la membrane externe d'un neurone exécute les cinq fonctions suivantes [8]:

- Propager les impulsions électriques le long de l'axone et des dendrites.
- Libérer les médiateurs à l'extrémité de l'axone.
- Réagir à ces médiateurs chimiques au niveau des dendrites.
- Réagir au niveau du corps cellulaire aux impulsions électriques issues des dendrites pour générer ou non une nouvelle impulsion.
- Permettre au neurone de reconnaître ses voisins pour se situer au cours de la formation du cerveau et trouver à quelles autres cellules il doit ce connecté.

II.2 Le neurone formel

II.2.1 Présentation

Il existe un grand nombre de modèles de neurones. Les modèles les plus utilisés sont basés sur le modèle développé par Mac Culloch & Pitts, dans lequel le neurone peut être représenté par une cellule connectée à des sources d'information en entrée et renvoie une information en sortie et peut être modélisé par deux opérateurs d'après la figure III.2 [6]:

- Un opérateur de sommation qui élabore un « potentiel post-synaptique » **P** égal à la somme pondérée des entrées de la cellule [5,8].

- Un opérateur de décision appelé aussi « fonction d'activation » qui calcule l'état de la sortie **S** du neurone en fonction de son potentiel **P** [5,8].

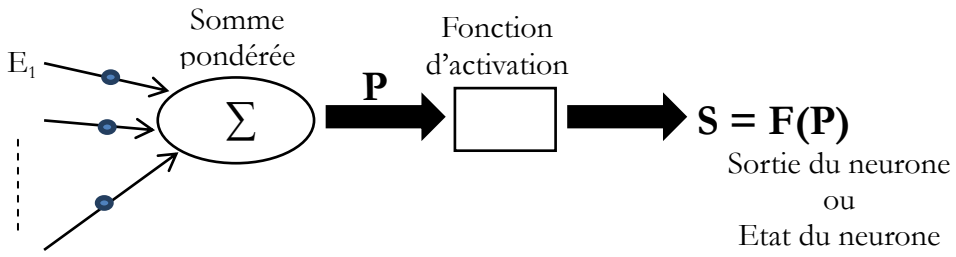


Figure III.2 : Modélisation du neurone formel

S : état du neurone (valeur de sortie).
F : fonction de transition ou d'activation.
W_i : coefficient de pondération associé à la connexion i (poids)
P : potentiel du neurone (résultat de la somme pondérée).

II.2.2 Fonctionnement du neurone formel

Le neurone formel reçoit à l'entée l'information *Ei* provenant d'autres neurones via des connexions qui n'ont pas la même importance. En effet, certaines sont plus importantes que d'autres et cette importance relative est mesurée par ***un poids*** $\boldsymbol{W_i}$ affecté à chaque connexion [3,6,7]. En fait, tout se passe comme si le neurone ne recevait qu'une entrée $\boldsymbol{P}$ telle que [6,7]:

$$P = \sum_1^n E_i W_i \qquad \text{III. 1}$$

Une fois l'entrée connue, le neurone applique une fonction $\boldsymbol{F}$ appelée **fonction d'activation** à la valeur *P* et on obtient la sortie *S* qui représente l'état du neurone telle que [6,7]:

$$S = F(P) = F\left[\sum_1^n E_i W_i\right] \qquad \text{III. 2}$$

Le choix de cette fonction *F* se révèle être un élément constitutif et important des réseaux de neurones. Ainsi, l'identité est rarement suffisante et des fonctions non linéaires et plus évoluées sont nécessaires [6,7].

En résumé, on peut dire que le neurone est une machine élémentaire qui reçoit à son entrée les variables issues des autres neurones en amont et associe à

chaque entrée un poids qui représente la force de connexion entre les deux neurones. A la sortie, il y a une seule donnée basée sur les informations d'entrée.

II.2.3 Le choix de la fonction d'activation

Le choix de la fonction de transition F est a priori laissé au concepteur de l'application neuronale. La seule contrainte impérative est que F soit dérivable. Toutefois, il est souhaitable de respecter les contraintes suivantes:

► ***La fonction est non-linéaire***

En effet, un réseau entièrement linéaire peut toujours se ramener à un réseau équivalent sans couche intermédiaire. Il suffit pour cela de faire le produit des matrices de transformation entre les différentes couches. On perd donc tout l'avantage d'une structure multicouche.

► ***La fonction est strictement croissante***

Cette condition s'exprime par l'équation [8]:

$$F'(X_j) > 0; \quad \forall X_j \in \mathcal{R} \qquad \text{III. 3}$$

Il est en effet souhaitable que la dérivée ne s'annule pas afin d'éviter un blocage de l'apprentissage (gradient nul) [7]. De plus des changements de signe de la dérivée provoquent des inversions brutales du gradient qui déstabilisent l'apprentissage [7]. Donc F doit être croissante. Même si F est décroissante, il est toujours possible de la remplacer par la fonction croissante F_c et d'inverser les poids W_{ij} telle que [7]:

$$F_c(\mathrm{X_j}) = \mathrm{F}(-\mathrm{X_j}) \qquad \text{III. 4}$$

► ***La fonction est saturante***

Cette condition s'exprime par l'équation suivante :

$$|F(\pm\infty)| < \infty \qquad \text{III. 5}$$

Cette condition procure au réseau une forte robustesse par rapport au bruit et lui permet de fonctionner comme un calculateur binaire [6].

A titre illustratif voici quelques fonctions couramment utilisées comme fonctions d'activation [9]:

Sigmoïde ou Logistique	Tangente hyperbolique	Fonction Gaussienne	Fonction à seuil

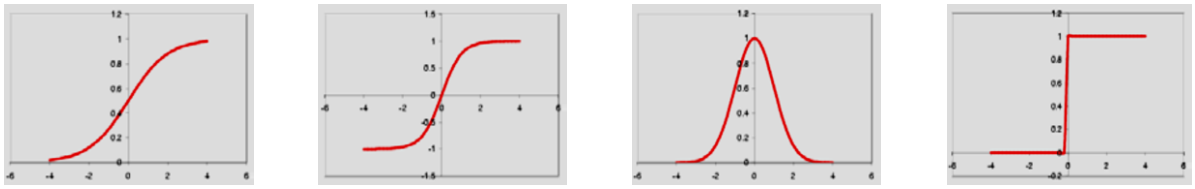


Figure III.3 : Exemples de fonctions usuelles utilisées comme fonction d'activation

III Topologie d'un réseau de neurones

III.1 Définition

Un réseau de neurones artificiels RNA est défini comme étant un ensemble de petites unités de calculs reliés entre elles par des liens de communication, et qui permettent d'acheminer l'information numérisée et codée de différentes manières [6]. Chaque unité possède sa propre mémoire et agit sur ses données locales en se servant des entrées reçues, via les canaux de communication, des autres unités [6]. En général, un réseau de neurone présente la même structure que chacun de ses neurones si on le regarde dans sa globalité donc on peut le modéliser comme indiqué par la figure III.4.

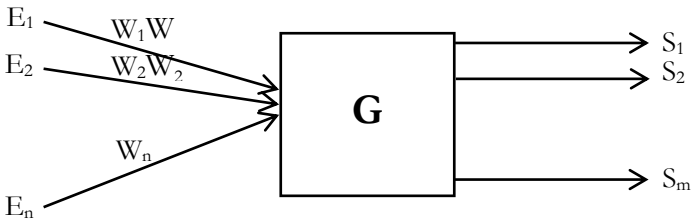


Figure III.4 : Modélisation d'un réseau de neurones

D'après cette figure, le réseau de neurones doit pouvoir calculer les valeurs de sorties *(S_1, S_2,……S_m)* en fonction des variables d'entrées *(E_1, E_2,….. E_n)*. Le principe consiste à ce qu'une première série de neurones applique aux entrées E_1, E_2,…. E_n, leur propre fonction d'activation F_1, ce qui fournit un certain nombre de résultats. Une seconde série de neurones prend ces résultats en entrée et calculent de nouveau, avec leur propre fonction d'activation F_2, des résultats qu'ils transmettent à la série de neurones suivante, etc..., ainsi jusqu'à atteindre la dernière série de neurones. Les sorties de ces derniers neurones sont alors les sorties du réseau S_1, S_2,……S_m.

Contrairement à chacune des fonctions d'activation, la fonction G qui transforme les valeurs d'entrée en valeurs de sortie à l'échelle du réseau ne peut pas être explicitée facilement [3]. Elle est en effet beaucoup plus compliquée puisqu'elle est en quelque sorte constituée de la superposition de toutes les fonctions d'activation F_i de chaque neurone *i*. De ce fait, on peut dire que la topologie d'un réseau de neurone est définie par son architecture (ou structure) et la nature des connexions entre ses neurones [9].

III.2 Architecture d'un réseau de neurones

La plupart des réseaux de neurones ont une topologie définie sous forme de couches [8,9,10]. Il existe quelques exceptions lorsque le réseau n'est pas explicitement défini sur plusieurs couches (comme par exemple certaines mémoires associatives), mais elles peuvent alors être considérées comme n'ayant qu'une seule couche. En général, les réseaux de neurones existent sous deux formes : **monocouches** ou **multicouches** [9,10]. De ce fait, l'architecture du réseau peut alors être décrite par le nombre de couches et le nombre de neurones dans chaque couche. Ainsi, un réseau de neurones est constitué de trois types de couches :

- Couche d'entrée composée des neurones d'entrées appelés aussi cellules perceptives, du fait de leur rôle de collecter les données qui proviennent de l'extérieur du réseau [11,12];
- Couche cachée dont les neurones n'interagissent qu'avec les autres neurones du réseau sans aucun lien avec le monde extérieur à celui-ci [11,12];
- Couche de sortie dont les neurones constituent la sortie ou la réponse du réseau [11,12].

Un exemple d'architecture de réseau est schématisé sur la figure III.5, il comprend une couche d'entrée, une couche cachée et une couche de sortie [2].

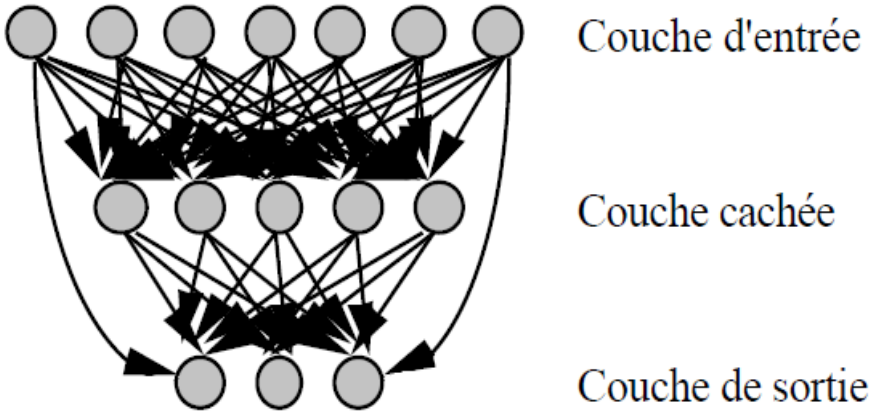


Figure III.5 : Exemple d'architectures de réseaux de neurones

III.3 Connectivité

La connectivité des réseaux est la manière dont les neurones sont connectés entre eux et en se basant sur la structure à couches, on distingue différents types de connexions [2,3,6]:

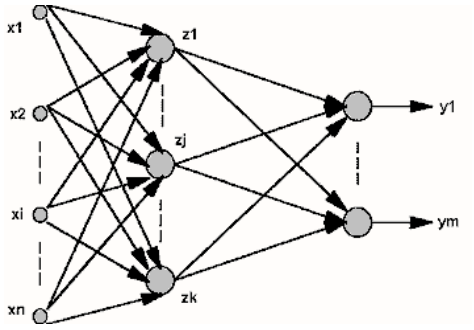
- les connexions inter-couches, correspondent à l'interconnexion entre neurones de couches voisines ;
- les connexions supra-couches, il s'agit de l'interconnexion entre neurones de couches qui ne sont pas adjacentes ;
- les connexions intra-couches correspondent à l'interconnexion entre neurones voisines d'une même couche ;
- l'auto-connexion c'est lorsqu'un neurone se connecte à lui-même.

D'autre part, le sens de transfert de l'information dans un réseau est défini par la nature des connexions qui peuvent être soit directes ou récurrentes.

Les connexions directes (*feedforward*) [3,7], appelées aussi non bouclées, se sont celles qui sont dirigées d'une couche d'indice inférieur vers une couche d'indice supérieur comme il est indiqué par la figure III.6.a.

Les connexions sont dites récurrentes *(feedback)* [3,7] ou bien bouclées, lorsque des sorties de neurones d'une couche sont connectées aux entrées d'une couche d'indice inférieur Figure III.6.b [2].

Par ailleurs, entre deux couches les connexions peuvent être partielles ou totales [12].



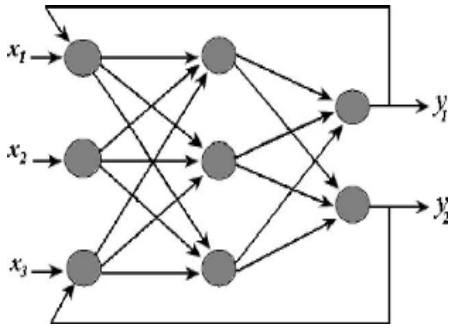


Figure III.6 : a. connexions directes b. Connexions récurrentes

IV Quelques célèbres réseaux de neurones.

IV.1 Le Perceptron

C'est un des premiers réseaux de neurones, conçu en 1958 par Rosenblatt, en s'inspirant du neurone visuel humain. Sa particularité c'est qu'il est linéaire et monocouche [5,6]. La première couche (d'entrée) représente la rétine [5]. Les neurones de la couche suivante (unique, d'où le qualificatif de monocouche) sont les cellules d'association et la couche finale correspond aux cellules de décision. Les sorties des neurones ne peuvent prendre que deux états (-1 et 1 ou 0 et 1) [5,6,7]. Seuls les poids des liaisons entre la couche d'association et la couche finale peuvent être modifiés. La règle de modification des poids utilisée est la règle de Widrow-Hoff qui stipule que si la sortie du réseau (donc celle d'une cellule de décision) est égale à la sortie désirée, le poids de la connexion entre ce neurone et le neurone d'association qui lui est connecté n'est pas modifié. Dans le cas contraire le poids est modifié "proportionnellement" à la différence entre la sortie obtenue et la sortie désirée selon l'équation suivante [2,7]:

$$w_{i+1} \leq w_i + k(d - s) \qquad \text{III. 6}$$

Où : s : la sortie obtenue,
d : la sortie désirée,
k : une constante positive.

En 1969, Marvin Lee Minsky et Seymour Papert publièrent un ouvrage mettant en exergue quelques limitations théoriques du perceptron, notamment l'impossibilité de traiter des problèmes non linéaires ou de connexité [7]. Ils étendirent implicitement ces limitations à tous les modèles de réseaux de neurones artificiels.

IV.2 Le Perceptron Multi-Couche (PMC)

C'est une nouvelle génération de réseaux de neurones, capables de traiter avec succès des phénomènes non linéaires et résoudre les défauts mis en évidence par Marvin Minsky. Proposé pour la première fois par Paul Werbos, le perceptron multicouche est apparu en 1986 et introduit par David Rumelhart [2,6]. Il est organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement à travers une ou plusieurs couches intermédiaires dites couches cachées, capables de traiter avec succès les phénomènes non linéaires. Il s'agit donc d'un réseau à propagation directe (*feedforward*) [14]. Chaque couche est constituée d'un nombre variable de neurones et chacun d'eux est relié directement à tous les neurones des couches précédentes et suivantes comme indiqué par la figure III.7 [6]. Les neurones de la dernière couche (dite « de sortie ») étant les sorties du système global autrement dit la réponse du réseau au problème à traiter. Ce type de réseau possède les caractéristiques suivantes [8]:

1. C'est un réseau qui est facilement et entièrement reconfigurable, afin de pouvoir tester rapidement de nouvelles architectures sans avoir à tout reprogrammer. Par exemple, il est immédiat de modifier le nombre de neurones d'une couche donnée, de l'interconnecter à toute autre couche….
2. Sa structure hiérarchique basée sur la connexité directe entre les couches successives.
3. les différentes couches sont indépendantes les unes des autres, dans la mesure où l'on peut modifier une couche (ou supprimer, ou rajouter) avec un minimum voire pas de changements sur le reste du réseau.
4. Sa fiabilité comme outil de classification grâce à son apprentissage qui est du type supervisé.

Pour leur apprentissage, les PMC utilisent l'algorithme de rétro-propagation du gradient qui sera traité en détails dans un prochain paragraphe. Grâce à leurs architectures, les PMC agissent comme un séparateur non linéaire et peuvent être utilisés pour la classification, le traitement de l'image ou l'aide à la décision [14].

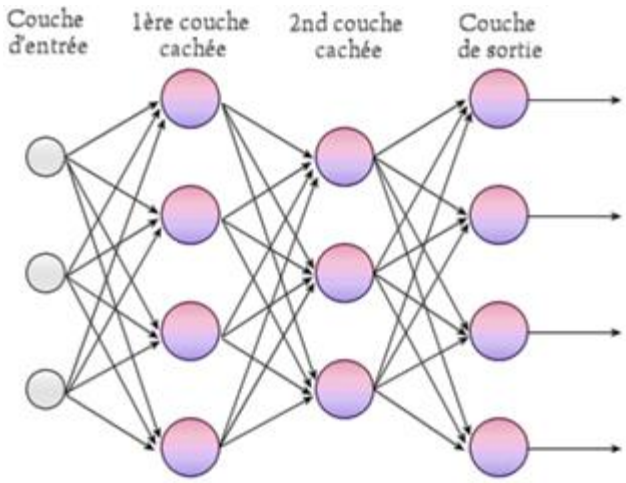


Figure III.7 : Architecture d'un réseau de type PMC

IV.3 Réseau RBF

Un autre type d'architecture de réseaux de neurones est connu sous le nom de *Fonctions à Base Radiale* (RBF). Les réseaux RBF constituent probablement le type de réseaux de neurones le plus utilisé après les PMC. Ils ont une seule couche cachée dont les fonctions d'activation sont des fonctions à base radiale, le plus souvent des gaussiennes [4,9]. La fonction d'activation du neurone de la couche de sortie est l'identité [13]. Les entrées sont directement connectées aux neurones de la couche cachée [14].

À de nombreux égards, les réseaux RBF s'apparentent aux réseaux PMC. Tout d'abord, ils possèdent également des connexions unidirectionnelles vers l'avant (*feedforward*) et chaque neurone est entièrement connecté aux unités de la couche suivante. Il n'en reste pas moins que les modèles de réseaux de neurones RBF sont fondamentalement différents dans la manière dont ils modélisent la relation entre les entrées et les sorties [13]. Tandis que les réseaux PMC modélisent la relation entrée-sortie en une seule étape, un réseau RBF va partitionner ce processus d'apprentissage en deux étapes indépendantes et bien distinctes. Au cours de la première étape, et grâce aux neurones de la couche caché connus sous le nom de *fonctions à base radiale* [15], les réseaux RBF vont modéliser la distribution de probabilités des données d'entrée. Au cours d'une seconde étape, le réseau RBF va apprendre comment mettre en relation ces données d'entrée avec une variable cible [10,13].

IV.4 Réseau de Hopfield

Il s'agit d'un réseau constitué de neurones à deux états (-1 et 1, ou 0 et 1), dont la loi d'apprentissage est la règle de Hebb (1949) [6,7]. Cette règle suggère que lorsque deux neurones sont excités conjointement, ils se créent ou renforcent un lien les unissant. Cette théorie est souvent résumée par « des neurones qui stimulent en même temps, sont des neurones qui se lient ensemble. » ce qui veut dire qu'une synapse améliore son activité si et seulement si l'activité de ses deux neurones est corrélée (c'est-à-dire que le poids d'une connexion entre deux neurones augmente quand les deux neurones sont activés en même temps).

IV.5 Réseau de Kohonen (Carte de Kohonen)

Contrairement aux réseaux de Hopfield où les neurones sont modélisés de la façon la plus simple possible, on recherche ici un modèle de neurone plus proche de la réalité. Ces réseaux sont inspirés des observations biologiques du fonctionnement des systèmes nerveux de perception des mammifères [5]. En effet, ce type de réseau de neurones consiste en une situation où plusieurs neurones entrent en compétition pour être activés. Dans cette situation, le seul neurone activé est le gagnant. Selon une certaine fonction, ainsi que, dans certains cas, un certain voisinage de ce neurone gagnant, verra son poids modifié. D'autre part, seul ce neurone et son voisinage profiteront de l'apprentissage. On calcule le neurone gagnant en choisissant celui dont le vecteur de poids est le plus près du vecteur des entrées selon une distance euclidienne [6,16].

L'apprentissage d'un tel réseau se base sur une loi de Hebb modifiée qui tient en compte d'un facteur d'oubli et qui consiste à rapprocher les poids des neurones gagnants des valeurs des entrées. De cette façon, une application successive de plusieurs signaux entrants différents « organisera » les poids du réseau. Les neurones d'une couche de Kohonen possèdent également des liaisons entre eux, dans le but de « compétition » constituant ainsi la base du réseau. La taille et la manière de calculer le voisinage d'un neurone de la couche de Kohonen peut varier d'un réseau à l'autre. Ce voisinage peut aussi bien être nul, c'est-à-dire que seul le neurone activé est entraîné à chaque présentation d'un élément. Ceci se résume par la formule [16]:

$$\frac{dw}{dt} = kes - wB(s) \qquad \text{III. 7}$$

Où w : le poids associé à une certaine connexion,
e : la valeur que le neurone reçoit en entrée par cette connexion,
s : la valeur qu'il renvoie en sortie (toujours positive),
$B(s)$: la fonction d'oubli,
k : une constante positive.

V. Apprentissage d'un réseau RNA

L'apprentissage est une phase de développement durant laquelle le comportement du réseau est modifié en changeant l'état des neurones de la première couche cachée jusqu'à la couche de sortie du réseau. En effet, le réseau va prendre des décisions par rapport aux données d'entrées qu'il reçoit en fonction de ce qu'il a appris par le passé, grâce à une « base » d'apprentissage. Le choix de classification se portera sur la probabilité de ressemblance à une classe qu'il connaît déjà, on appelle ceci l'apprentissage par expérience. Ainsi, Ce phénomène se répète jusqu'à l'obtention du comportement désiré et elle recouvre deux réalités à savoir [2]:

- **La mémorisation**: le fait d'assimiler sous une forme dense des exemples éventuellement nombreux.
- **La généralisation**: le fait d'être capable, grâce aux exemples appris, de traiter des exemples distincts, encore non rencontrés mais similaires.

V.1 Techniques d'Apprentissage

L'apprentissage est réalisé par des algorithmes de calcul dont le but est d'adapter les poids, liés aux connexions entre les neurones, en fonction des informations présentées à l'entrée du réseau [6,7]. Une fois l'apprentissage fini, les poids ne sont plus modifiés. Les procédures d'apprentissage peuvent être classées en trois catégories: apprentissage supervisé, apprentissage non supervisé, et apprentissage par renforcement. Et ces procédures se font en général de deux manières différentes:

En paquet on parle d'apprentissage « batch » qui utilise tous les exemples d'entrainement qui sont groupées dans une base d'exemples et sont traités simultanément de façon répétée [3,4]. Ce type d'apprentissage requiert souvent de

charger en mémoire l'ensemble des poids et des données d'apprentissage. En général c'est un processus qui met beaucoup de temps avant que le réseau de neurones converge vers le résultat désiré [12].

En ligne « on-line » il s'agit d'un apprentissage plus dynamique qui met à jour l'estimation courante par l'observation des nouvelles données une par une [11,12]. C'est une procédure itérative, lente qui demande moins de mémoire et elle est recommandée dans des environnements changeants.

V.1.1 Apprentissage supervisé

L'apprentissage est qualifié de supervisé lorsque le réseau reçoit en même temps la donnée à traiter et la sortie attendue on dit qu'il se fait sous le contrôle d'un expert. En effet, Chaque neurone collecte les informations venant des neurones précédents et en déduit une information à transmettre aux neurones suivants tout en donnant plus ou moins d'importance à certaines que d'autres. L'apprentissage supervisé consiste à trouver les ajustements pour qu'il accorde la bonne importance à la bonne information. Pour cela, à chaque prévision faite par le réseau au moment de son apprentissage, on lui présente la réponse qu'il aurait dû sortir. Il analyse les écarts pour s'adapter [1,2]. Après l'apprentissage, le réseau est testé en lui présentant des données qui n'ont pas servi à la phase d'apprentissage et l'on observe ses réponses il s'agit de la phase de test. Les performances du réseau sont alors évaluées par le pourcentage d'erreurs, la magnitude du gradient de l'erreur, la matrice de confusion et autres [4,6]. Si le réseau de neurones donne de meilleures performances, alors en utilisation il ne fait que des opérations mathématiques, ce qui permet de faire un travail à une rapidité permettant des applications en temps réel.

V.1.2 Apprentissage non supervisé

Ce type d'apprentissage est plutôt adapté à des réseaux appelés généralement « auto-organisateurs» ou à apprentissage compétitif [6,13]. L'apprentissage s'effectue alors par présentation, à un réseau autonome, des données possédant une certaine redondance. L'objectif du réseau est alors de dégager des régularités. On le laisse évoluer librement jusqu'à ce qu'il se stabilise (s'auto-organise) utilisant les lois régissant l'évolution des synapses. Un tel réseau est autodidacte et peut être utilisé pour l'obtention de prototypes [9].

V.1.3. Apprentissage par renforcement

C'est un apprentissage par assignation de crédits qui ne nécessite pas de comportement de référence explicite mais seulement d'informations grossières, comme un encouragement ou une pénalisation [6]. Autrement dit, le réseau de neurone interagit avec l'environnement qui lui donne récompense pour une réponse satisfaisante et lui assigne une pénalité dans le cas contraire [8]. Le réseau doit ainsi, découvrir les réponses qui lui donnent un maximum de récompenses.

V.2 Algorithme d'Apprentissage : la rétro-propagation

V.2.1 Introduction à la rétro-propagation

La technique de rétro-propagation du gradient est l'un des algorithmes d'apprentissage les plus utilisés pour entrainer un réseau de neurones multicouches (PMC) en se basant sur la règle de la descente du gradient de l'erreur (Delta Rule) ou l'algorithme de Windrow-Hoff [2,11], qui consiste à rétro-propager l'erreur commise par un neurone à ses synapses (poids) et aux neurones qui y sont reliés et à minimiser cette erreur par le calcul de son gradient [11,12]. Autrement dit, l'erreur est corrigée selon l'importance des éléments qui ont participé à sa cause c'est-à-dire les poids des différentes connexions entre les neurones. Ainsi, les poids qui ont contribué à engendrer une erreur importante se verront modifiés de manière plus significative que les poids qui ont engendré une erreur marginale. D'autre part pour cet algorithme, les poids sont modifiés au fur et à mesure que les exemples se présentent au réseau et non pas après que tous les exemples aient défilé ce qui permet de minimiser l'erreur d'une manière précise [10,14]. Et en appliquant ce calcul plusieurs fois, l'erreur tend à diminuer et le réseau offre une meilleure prédiction.

V.2.2 Technique de la rétro-propagation

L'algorithme de rétro-propagation, comme il est mentionné avant, est un algorithme d'apprentissage de type « online » qui se résume par les étapes suivantes :

- On considère un réseau PMC défini par une architecture à N entrées et à N' sorties, soit $W=(w_i)_i$ le vecteur des poids synaptiques associés à tous les liens du réseau. On désigne par le couple (X^n, Y^n) le n^{ème} échantillon d'apprentissage avec : $X^n = (x_i^n)_{1 \le i \le p}$ l'entrée, et $Y^n = (y_i^n)_{1 \le i \le q}$ sa sortie désirée.

- La première étape de l'apprentissage c'est une phase de calcul dans le sens direct où chaque neurone effectue la somme pondérée de ses entrées et applique ensuite la fonction d'activation *F* pour donner une sortie qui représente son état. Ceci est illustré par l'équation III.8 [1,2] avec p_i^n le potentiel post-synaptique du *i*ème neurone dans une couche quelconque, x_j^n est l'état d'un neurone de la couche précédente à celle-ci et w_{ij}^n le poids de la connexion entre les deux neurones *j* et *i* :

$$x_i^n = F(p_i^n) \quad avec \;\; p_i^n = \sum_j w_{ij}^n x_j^n \qquad \text{III. 8}$$

- Une fois la propagation vers l'avant est terminée, on obtient à la sortie du réseau pour le n^{ème} échantillon, le résultat $S^n = (s_j^n)_{1\leq j\leq q}$. Ce résultat est obtenu avec des poids qui sont au préalable initialisés avec des valeurs aléatoires. Cette phase, dite de propagation, permet de calculer la sortie du réseau en fonction de son entrée.

- Pour chaque neurone *j* du *n*ème échantillon d'apprentissage, une erreur quadratique entre la sortie donnée par le réseau s_j^n et le résultat désirée y_j^n est calculée :

$$e_j^n = (y_j^n - s_j^n) \qquad \text{III. 9}$$

  Cette formule permet d'aboutir à la somme des erreurs quadratiques observées sur tous les neurones de la sortie pour le *n*ème échantillon appelée aussi fonction de « coût » et définie par [4,17]:

$$E^n = \frac{1}{2}\sum_{j=1}^{j=q}(e_j^n)^2 = \frac{1}{2}\sum_{j=1}^{j=q}(y_j^n - s_j^n)^2 \qquad \text{III. 10}$$

  On voit donc que l'erreur est nulle si le réseau de neurones ne se trompe sur aucun des exemples, c'est-à-dire s'il parvient à calculer la bonne sortie pour chaque échantillon correctement ce qui n'est pas très rare. C'est pour cela que l'objectif de l'algorithme de rétro-propagation est d'adapter les poids des connexions du réseau de manière à minimiser cette fonction de coût en la propageant dans le sens inverse du gradient autrement dit en calculant, pour chaque échantillon (n), la dérivée partielle de l'erreur par rapport aux poids. Si *j* est le neurone pour lequel on veut adapter les poids, deux cas se présentent [2,3,4,17]:

a. Le neurone *j* du $n^{ème}$ échantillon appartient à la couche de sortie :

D'après l'équation III.2, la sortie s_j^n du neurone *j* est définie par :

$$s_j^n = F\left[p_j^n\right] \; avec \; p_j^n = \sum_{i=1}^{i=r} w_{ji}^n s_i^n \qquad \text{III. 11}$$

- F [ ] la fonction d'activation du neurone *j ;*
- p_j^n le potentiel post-synaptique du neurone *j ;*
- w_{ji}^n est le poids de la connexion entre le neurone *i* de la couche précédente et le neurone *j ;*
- s_i^n est la sortie du neurone *i* de la couche précédente qu'on suppose qu'elle contient *r* neurones ce qui nous donne *r* poids à adapter.

Pour corriger l'erreur, on modifie le poids w_{ji}^n dans le sens opposé au gradient $\frac{\partial E^n}{\partial w_{ji}^n}$ de l'erreur. Par la règle de chaînage des dérivées partielles, on obtient :

$$\frac{\partial E^n}{\partial w_{ji}^n} = \frac{\partial E^n}{\partial e_j^n} \times \frac{\partial e_j^n}{\partial s_j^n} \times \frac{\partial s_j^n}{\partial p_j^n} \times \frac{\partial p_j^n}{\partial w_{ji}^n} \qquad \text{III. 12}$$

On va évaluer chaque terme de l'équation III.12:

$$\frac{\partial E^n}{\partial e_j^n} = \frac{\partial \left[\frac{1}{2}\sum_{k=1}^{k=q}(e_k^n)^2\right]}{\partial e_j^n} = \frac{1}{2}\frac{\partial (e_j^n)^2}{\partial e_j^n} = e_j^n \qquad \text{III. 13}$$

$$\frac{\partial e_j^n}{\partial s_j^n} = \frac{\partial (y_j^n - s_j^n)}{\partial s_j^n} = -1 \qquad \text{III. 14}$$

$$\frac{\partial s_j^n}{\partial p_j^n} = \frac{\partial F\left(p_j^n\right)}{\partial p_j^n} = F'\left(p_j^n\right) \qquad \text{III. 15}$$

$$\frac{\partial p_j^n}{\partial w_{ji}^n} = \frac{\partial \left[\sum_{l=1}^{l=r}(w_{jl}^n s_l^n)\right]}{\partial w_{ji}^n} = \frac{\partial (w_{ji}^n s_i^n)}{\partial w_{ji}^n} = s_i^n \qquad \text{III. 16}$$

Nous obtenons à la fin :

$$\frac{\partial E^n}{\partial w_{ji}^n} = -e_j^n F'(p_j^n) s_i^n \quad \text{III. 17}$$

D'autre part les poids sont mis à jour en utilisant la règle dite « Delta » exprimée par l'équation suivante :

$$\Delta w_{ji}^n = -\mu \frac{\partial E^n}{\partial w_{ji}^n} = \mu e_j^n F'(p_j^n) s_i^n = \mu \delta_j^n s_i^n \quad \text{III. 18}$$

Avec : $\delta_j^n = e_j^n F'(p_j^n)$ III. 19

Où : $0 \leq \mu \leq 1$ représente le taux d'apprentissage ou gain de l'algorithme [2,11].

b. Le neurone j du $n^{ème}$ échantillon appartient à une couche cachée:

On reprend l'expression de la dérivée partielle de l'erreur E^n par rapport à w_{ji}^n sans dériver par rapport à l'erreur e_j^n car celle-ci est dans ce cas est inconnue. On obtient alors :

$$\frac{\partial E^n}{\partial w_{ji}^n} = \frac{\partial E^n}{\partial s_j^n} \times \frac{\partial s_j^n}{\partial p_j^n} \times \frac{\partial p_j^n}{\partial w_{ji}^n} \quad \text{III. 20}$$

Les deux derniers termes de l'équation III.20 ne changent pas :

$$\begin{cases} \dfrac{\partial s_j^n}{\partial p_j^n} = F'(p_j^n) & \text{III. 21} \\ \dfrac{\partial p_j^n}{\partial w_{ji}^n} = s_i^n & \text{III. 22} \end{cases}$$

Il reste à déterminer la valeur de $\frac{\partial E^n}{\partial s_j^n}$ sachant que dans ce cas chaque erreur e_k^n au niveau d'une couche k dépend de s_j^n du fait de la propagation de l'erreur de la sortie à l'entrée à travers les couches cachées. On obtient alors l'équation suivante :

$$\frac{\partial E^n}{\partial s_j^n} = \frac{\partial \left[\frac{1}{2}\sum_{k=1}^{k=q}(e_k^n)^2\right]}{\partial s_j^n} \quad \text{III. 23}$$

$$= \sum_{k=1}^{k=q} \left[e_k^n \times \frac{\partial e_k^n}{\partial s_j^n} \right] \quad \text{III. 24}$$

$$= \sum_{k=1}^{k=q} \left[e_k^n \times \frac{\partial e_k^n}{\partial p_k^n} \times \frac{\partial p_k^n}{\partial s_j^n} \right] \quad \text{III. 25}$$

$$= \sum_{k=1}^{k=q} \left[e_k^n \times \frac{\partial (y_k^n - s_k^n)}{\partial p_k^n} \times \frac{\partial (\sum_l w_{kl}^n s_l^n)}{\partial s_j^n} \right] \quad \text{III. 26}$$

$$= \sum_{k=1}^{k=q} \left[e_k^n \times \frac{\partial (y_k^n - F(p_k^n))}{\partial p_k^n} \times \frac{\partial (\sum_l w_{kl}^n s_l^n)}{\partial s_j^n} \right] \quad \text{III. 27}$$

Comme la sortie désirée y_k^n est indépendante de p_k^n on alors :

$$\frac{\partial (y_k^n - F(p_k^n))}{\partial p_k^n} = -F'(p_k^n) \quad \text{III. 28}$$

Aussi :

$$\frac{\partial (\sum_l w_{kl}^n s_l^n)}{\partial s_j^n} = w_{kj}^n \quad \text{III. 29}$$

On obtient alors:

$$\frac{\partial E^n}{\partial s_j^n} = \sum_{k=1}^{k=q} \left[e_k^n \times (-F'(p_k^n) w_{kj}^n \right] \quad \text{III. 30}$$

En substituant l'équation III.19 on obtient :

$$\frac{\partial E^n}{\partial s_j^n} = -\sum_{k=1}^{k=q} \delta_k^n w_{kj}^n \quad \text{III. 31}$$

Finalement en tenant compte des équations III.31, III.21 et III.22, l'équation III.20 devient:

$$\frac{\partial E^n}{\partial w_{ji}^n} = -F'(p_j^n) s_i^n \sum_{k=1}^{k=q} \delta_k^n w_{kj}^n \quad \text{III. 32}$$

Les poids sont mis à jour en utilisant toujours la règle « delta » :

$$\Delta w_{ji}^n = -\mu \frac{\partial E^n}{\partial w_{ji}^n} = -\mu s_i^n \left[-F'(p_j^n) \sum_{k=1}^{k=q} \delta_k^n w_{kj}^n \right] \quad \text{III. 33}$$

Ou encore :

$$\Delta w_{ji}^n = \mu s_i^n \delta_j^n \quad \Leftrightarrow \quad w_{ji}^n = w_{ji}^{n-1} + \mu s_i^n \delta_j^n \quad \text{III. 34}$$

Avec :

$$\delta_j^n = F'(p_j^n) \sum_{k=1}^{k=q} \delta_k^n w_{kj}^n \qquad et \qquad 0 \leq \mu \leq 1 \qquad \text{III. 35}$$

Ce qui est détaillé avant représente l'algorithme d'apprentissage par rétro-propagation (*back-propagation*) appliqué à un seul neurone, et il faudra donc l'appliquer sur chacun des neurones du réseau. La valeur du taux d'apprentissage μ à un effet significatif sur les performances du réseau, si ce taux est petit l'algorithme converge lentement, par contre s'il est grand l'algorithme risque de générer des oscillations [2,6].

Appliqué plusieurs fois, cet algorithme permettra d'affiner la correction d'erreurs et d'obtenir un réseau de neurones de plus en plus performant. Avec cette méthode d'apprentissage le réseau de neurones s'améliore mieux et tend plus rapidement à classer parfaitement (presque) chacun des exemples qui se présentent à lui, ce qui n'est pas le cas lorsqu'on corrige sur la globalité des exemples, car dans ce cas le réseau ne s'adapte aux exemples qu'après un certain moment ce qui prend plus de temps.

L'algorithme de rétro-propagation est résumé par les étapes suivantes [2]:

1. Initialisation des poids $(W_i)_i$, à des valeurs aléatoires de faible grandeur.
2. Sélection d'un exemple d'apprentissage dans la base d'apprentissage.
3. Calcul de la sortie S, par propagation depuis l'entrée, par la formule :

$$S = F(P) = F\left[\sum_1^n E_i W_i\right]$$

4. Comparer pour chaque neurone *j*, la sortie désirée avec la sortie calculée

$$e_j = y_j - s_j.$$

5. Si erreur $e_j = y_j - s_j \neq 0$ en sortie, alors pour tous les neurones *j* depuis la sortie jusqu'à l'entrée) :

   - Si *j* est un neurone de sortie alors $\delta_j = e_j F'(p_j)$.
   - Si *j* est un neurone caché alors $\delta_j = F'(p_j) \sum_{k=1}^{k=q} \delta_k \, w_{kj}$

   (k : neurones compris entre la couche actuelle et la couche de sortie)
6. Mettre à jour les nouvelles valeurs des poids

$$w_{ji}(t+1) = w_{ji}(t) + \mu \delta_j s_i \qquad 0 \leq \mu \leq 1 \, .$$

7. Tant que l'erreur est trop importante, retour à l'étape 2 (on prend alors l'exemple suivant).

V.2.3 Amélioration de l'algorithme de rétro-propagation

Un grand nombre de propositions d'amélioration de l'algorithme de rétro-propagation du gradient total ont déjà été publiées et continuent encore à l'être. Dans la plupart des cas, le principe consiste à apporter quelques modifications afin de résoudre certains problèmes tels que ceux de lenteur de convergence, d'arrêt prématuré de l'apprentissage, ou de sur-apprentissage. Les principales modifications qui ont été introduit dans l'algorithme de rétro-propagation consistent en l'accélération d'apprentissage et le pouvoir de généralisation [10,11].

V.2.3.1 Accélération de l'algorithme

L'accélération de l'algorithme est réalisable au travers de différentes caractéristiques liées à l'apprentissage à savoir le taux d'apprentissage appelé aussi le gain d'adaptation et la manière d'initialiser les poids lors du lancement de l'apprentissage.

V.2.3.1.1 Ajout de l'inertie

Le choix des paramètres d'un algorithme d'apprentissage influe beaucoup sur la rapidité de calculs. Dans le cas de l'algorithme de rétro-propagation, le calcul du gradient consiste à définir, dans un espace contenant autant de dimensions qu'il y a de poids, la direction dans laquelle doit s'effectuer la modification des poids. Le principe de descente de gradient consiste alors à effectuer de manière itérative (pas par pas) une modification des poids suivant cette direction jusqu'à arriver à un minimum sur la fonction de coût représentant l'écart entre les sorties obtenues et les sorties de référence. Si ce pas, aussi appelé « gain d'adaptation » [2,10], est défini trop petit, le nombre de pas nécessaires peut s'avérer relativement important et contribue donc à ralentir de manière non négligeable l'apprentissage. Notons qu'il est même possible que l'algorithme, rencontrant un minimum local, ne puisse plus en sortir. A l'inverse, si le pas est trop important, l'algorithme peut devenir instable.

Pour pallier à ce problème, un terme d'inertie appelé *« momentum »* est souvent rajouté [10,12]. Son usage, proportionnel à la modification de poids effectuée au cycle précédent, revient à faire varier le pas en fonction de la progression dans la direction du gradient. Cette solution permet d'accélérer la convergence du réseau d'une part et de faire sortir des minimums locaux, dans la mesure du possible, afin d'accomplir la descente de la fonction d'erreur.

A chaque itération, le changement de poids conserve les informations des changements précédents. Cet effet de mémoire permet d'éviter les oscillations et

accélère l'optimisation du réseau. Par rapport à la formule de modification des poids présentés par l'équation III.18, le changement des poids avec inertie se traduit par l'équation suivante [12]:

$$w_{ji}^{n} = w_{ji}^{n-1} + \mu s_i^n \delta_j^n + \alpha \Delta w_{ji}^{n-1} \qquad \text{III. 36}$$

Où: α est un paramètre nommé *« momentum »* compris entre 0 et 1 qui représente une espèce d'inertie dans le changement de poids. Le choix de la valeur de ce coefficient constitue un grand défi pour le réseau, en effet une grande valeur proche de 1 provoquera un comportement totalement aléatoire et des fluctuations incessantes des poids on dit que le système est instable. Un coefficient nul donne un comportement déterministe où l'on descend la pente du gradient de l'erreur sans s'autoriser la moindre remontée, dans ce cas on risque d'être piégé dans un minimum local.

V.2.3.1.2 Initialisation des poids

Au lancement de l'apprentissage, les valeurs initiales des poids doivent être différentes de zéro pour que l'algorithme de rétro-propagation puisse fonctionner. D'autre part, l'utilisation de valeurs élevées peut provoquer un phénomène de saturation prématurée qui contribue à diminuer la vitesse de convergence de l'apprentissage. Une méthode consiste à effectuer une initialisation des poids selon une distribution uniforme dans l'intervalle [-M, M], avec M défini par l'équation suivante [10]:

$$M = \frac{0{,}87}{k\sqrt{n}\sqrt{E(x_j^2)}} \qquad \text{III. 37}$$

Où : *k* étant la pente de la fonction d'activation dans la zone linéaire, *n* le nombre d'entrées d'un neurone *j* de la première couche cachée et $E(x_j^2)$ la variance des données de la base d'apprentissage.

V.2.3.2 Augmentation du pouvoir de généralisation

Nous avons évoqué plusieurs fois le problème du sur-apprentissage, qui est provoqué par la capacité d'un réseau de neurones, possédant un nombre d'unités de mémorisation plus que nécessaire (on parle aussi de 'surparamétrisation'), à apprendre parfaitement les exemples de la base d'apprentissage. Afin d'arrêter l'apprentissage juste avant que ne se produise ce phénomène de sur-apprentissage, plusieurs méthodes ont été proposées. La plus simple consiste à disposer de trois bases de données distinctes [2]: une base d'apprentissage, une base de test et une base dite de « validation ». Cette dernière base est utilisée pendant l'apprentissage

afin d'examiner le comportement du réseau pour des données qui lui sont inconnues. Ainsi, l'apprentissage est arrêté lorsque l'erreur sur la courbe de la base de validation croisée atteint un minimum. Ce qui nécessite d'avoir suffisamment de données pour constituer trois bases à la fois représentatives et distinctes.

VI. Procédure de développement d'un réseau RNA

Le réseau de neurones est utilisé pour réaliser une fonction particulière, dans notre cas il s'agit d'une classification, qui va être élaborée lors d'une phase d'apprentissage. Le résultat de cette fonction est obtenu lors d'une phase d'utilisation (ou propagation) à travers le réseau qui s'effectue par modification de l'état des neurones, de la première couche cachée jusqu'à la sortie du réseau. En général, un cycle de conception et de mise en œuvre d'un réseau de neurones suit les étapes suivantes :

1- La collecte et l'analyse des données,
2- Le prétraitement et la séparation des bases de données,
3- L'élaboration de la structure d'un réseau,
4- Le dimensionnement du réseau,
5- L'apprentissage du réseau,
6- Le tast et la validation.

VI.1 Collecte et Analyse des données:

Le processus d'élaboration d'un réseau de neurones commence toujours par le choix et la préparation des échantillons de données qui lui seront présentées en entrée. L'objectif de cette étape est de recueillir des données, à la fois pour développer le réseau de neurones et pour le tester. Si les données d'apprentissage sont variées et si la qualité des exemples présentés est meilleure alors le réseau de neurones généralisera mieux et il aura plus de chance de répondre correctement. Autrement dit, si le réseau sait répondre correctement pour un nombre fini de situations les plus diverses, il sera alors plus proche de ce que l'on veut dans une situation nouvelle. Ainsi, la présentation de données très différentes de celles qui ont été utilisées lors de l'apprentissage peut entraîner une sortie totalement imprévisible.

Il est souvent préférable d'effectuer une analyse des données de manière à déterminer les caractéristiques discriminantes pour détecter ou différencier ces données qui constituent l'entrée du réseau de neurones. Cette détermination des caractéristiques a des conséquences à la fois sur la taille du réseau (et donc le temps

de simulation), sur les performances du système (pouvoir de classification, taux de détection), et sur le temps de développement (temps d'apprentissage). Dans le cas d'un problème de classification, ce qui est le cas dans cette thèse, une étude statistique sur les données peut permettre d'écarter celles qui sont aberrantes et redondantes. Et il appartient à l'expérimentateur de déterminer le nombre de classes auxquelles ses données appartiennent et de déterminer pour chaque donnée la classe à laquelle elle appartient.

VI.2 Prétraitement et Séparation des bases de données

De manière générale, les bases de données doivent subir un prétraitement afin d'être adaptées aux entrées et sorties du réseau de neurones. Un prétraitement courant consiste à effectuer une normalisation appropriée, qui tienne compte de l'amplitude des valeurs acceptées par le réseau. Une fois ces données sont prêtes à être présentés, elles sont partagées en deux types :

1. Une partie est utilisée pour l'ajustement des paramètres (les poids des connexions) du réseau au cours de son apprentissage.
2. L'autre partie est utilisée pour tester et évaluer les performances du réseau.

L'ensemble d'apprentissage contient des observations ainsi que la connaissance de leur vraie classe. On ajuste alors les paramètres du réseau en fonction de ces exemples. Puis on peut mesurer le pourcentage de réponses correctes sur un ensemble d'évaluation. Ce dernier ne doit pas contenir d'exemples ayant servi à l'apprentissage sans quoi les mesures seraient faussées (il y a en effet peu de chances pour qu'en situation réelle le classificateur retrouve exactement des observations identiques à celles de l'apprentissage).

Une troisième base de données appelée « base de validation croisée » peut être ajoutée afin de contrôler la phase d'apprentissage. Il n'y a pas de règle pour déterminer ce partage de manière quantitatif. Il résulte souvent d'un compromis tenant compte du nombre de données dont on dispose et du temps imparti pour effectuer l'apprentissage. Chaque base doit cependant satisfaire aux contraintes de représentativité de chaque classe de données et doit généralement refléter la distribution réelle, c'est à dire la probabilité d'occurrence des diverses classes.

VI.3 Elaboration de la structure d'un réseau

Il existe un grand nombre de types de réseaux de neurones, avec pour chacun des avantages et des inconvénients. Il faut d'abord choisir le type de réseau :

un perceptron standard, un réseau multicouche PMC, un réseau de Hopfield, un réseau de Kohonen etc…une telle action peut dépendre d'une part de la tâche à effectuer (classification, association, contrôle de processus, séparation aveugle de sources...), de la nature des données (des données présentant des variations au cours du temps). D'autres considérations sont à prendre en considération lors du choix du type du réseau comme d'éventuelles contraintes d'utilisation temps-réel et des différents types de réseaux de neurones disponibles dans le logiciel de simulation que l'on compte utiliser.

VI.4 Dimensionnement d'un réseau

Dimensionner un réseau de neurones revient à déterminer le nombre de ses couches et le nombre des neurones dans chacune d'elles. La couche d'entrée est constituée des données collectées et analysées en vue de l'apprentissage du réseau. Les neurones de la couche de sortie représentent la réponse attendue de la part du réseau autrement dit, si on a besoin de n nombres de sortie alors on doit choisir n neurones sur la couche de sortie.

Mis à part les couches d'entrée et de sortie, l'analyste doit décider du nombre de couches intermédiaires ou cachées. Sans couche cachée, le réseau n'offre que de faibles possibilités d'adaptation; avec une couche cachée, il est capable, avec un nombre suffisant de neurones, d'approximer toute fonction continue. D'autre part, dans le cas des couches cachées, les choses ne sont pas si simples en ce qui concerne la détermination du nombre de neurones, du fait qu'il n'existe aucune règle à suivre pour déterminer le nombre des neurones à placer dans une couche cachée pour avoir un réseau optimal. La manière la plus suivie est d'essayer au « hasard » plusieurs nombres de neurones jusqu'à obtenir des résultats satisfaisants après l'apprentissage.

VI.5 Apprentissage du réseau

Tous les modèles de réseaux de neurones requièrent un apprentissage. Plusieurs types d'apprentissages peuvent être adaptés à un même type de réseau de neurones. Les critères de choix sont souvent la rapidité de convergence ou les performances de généralisation. Le critère d'arrêt de l'apprentissage est souvent calculé à partir d'une fonction de coût, caractérisant l'écart entre les valeurs de sortie obtenues et les valeurs de références (réponses souhaitées pour chaque exemple présenté). Durant la phase d'apprentissage le réseau de neurones ajuste les poids des différentes connexions entre les neurones en calculant l'erreur entre la sortie désirée et la réponse du réseau pour un neurone de sortie. La technique de

validation croisée, qui sera précisée par la suite, permet un arrêt adéquat de l'apprentissage pour obtenir de bonnes performances de généralisation.

Certains algorithmes d'apprentissage se chargent de la détermination des paramètres architecturaux du réseau de neurones. Si on n'utilise pas ces techniques, l'obtention des paramètres architecturaux optimaux se fera par comparaison des performances obtenues pour différentes architectures de réseaux de neurones. Des contraintes dues à l'éventuelle réalisation matérielle du réseau peuvent être introduites lors de l'apprentissage.

VI.6 Test et Validation

Une fois le réseau de neurones entraîné (après apprentissage), il est nécessaire de le tester sur une base de données différente de celles utilisées pour l'apprentissage ou la validation croisée. Ce test permet à la fois d'apprécier les performances du système neuronal et de détecter le type de données qui pose problème. Si les performances ne sont pas satisfaisantes, il faudra soit modifier l'architecture du réseau, soit modifier la base d'apprentissage. Alors que les tests concernent la vérification des performances du réseau de neurones hors échantillon et sa capacité de généralisation, la validation est parfois utilisée lors de l'apprentissage pour mesurer la généralisation du réseau et interrompre l'apprentissage quand celle-ci s'arrête de s'améliorer. Dans le cas général, une partie des échantillons d'apprentissage est écartée et conservée pour être utilisée pour tester la performance du réseau. Dans les cas de petits échantillons, on ne peut pas toujours utiliser une telle distinction, simplement parce qu'il n'est pas toujours possible d'avoir suffisamment de données dans chacun des groupes. On a alors parfois recours à d'autres procédures pour établir la structure optimale du réseau.

VII. Avantages et Inconvénients des réseaux RNA

Comme toute technique, les réseaux de neurones présentent des avantages et des inconvénients. Parmi les avantages on peut citer [8,9]:

- Apprentissage automatique des poids (les règles d'apprentissage) ;
- Capacité de généralisation (la phase des tests) ;
- Possibilité de faire le parallélisme (les éléments de chaque couche peuvent fonctionner en parallèle) ;
- Résistance aux pannes (si un neurone ne fonctionne plus, le réseau ne se perturbe pas) ;
- Modélisation des fonctions inconnues ;

- Bonne approximation.

Par contre, les réseaux de neurones présentent aussi quelques inconvénients, à savoir [8,9] :

- Représentation complexe (généralement l'architecture de réseau de neurones est complexe, et même si on la schématise, elle reste peu lisible) ;
- Paramètre difficiles à interpréter (comme une boite noire) ;
- Inexistence des règles définissant l'architecture du réseau en fonction du problème à résoudre ;
- Les algorithmes d'apprentissage consomment un temps de calcul considérable.

Conclusion

L'intérêt porté aujourd'hui aux réseaux de neurones tient sa justification dans les propriétés fascinantes qu'ils possèdent. Ces outils ont permis aussi le dépassement des limites de l'informatique traditionnelle, en offrant des services variés tels que la classification, la modélisation, l'approximation…, tous ces facteurs ont permis à l'informatique de conquérir plusieurs domaines comme le contrôle et la commande automatique, le traitement d'image et du signal numérique en général, la reconnaissance des formes etc…

En se basant sur les avantages des réseaux de neurones en général et celles des PMC en particulier, nous avons opté pour cette technique pour implémenter un classificateur d'ondelettes pour la compression des images fixes.

Bibliographie

[1] J.P. Renard, "R*éseaux neuronaux et modèle objet,* " Vuibert, 2006.

[2] C. Touzet, "*Les réseaux de neurones artificiels - Introduction au connexionnisme,*". 2éme édition. Edition. la Machotte, 2016

[3] G. Dreyfus, " *Réseaux de neurons,* ". Edition Eyrolles, 2002.

[4] O. Sarzeaud, "*Les Réseaux de neurones: Contribution à une théorie,*". Ouest Editions 1995.

[5] P. Buser, M. Imbert, "*Vision,* ". Editions Herman 1987.

[6] P. Borne, M. Benrejeb, J. Haggège, "*Les réseaux de neurons, presentation et application,* ". Edition Technip 2007.

[7] J. M. Zurada, "*Introduction to artificial neural systems,* ". West Publishing Company 1992.

[8] M.T. Chikh, "*Amélioration des images par un modèle de réseau de neurones (Comparaison avec les filtres de base),* ". Mémoire de master de l'université Abou-Bakr Belkaid, Tlemcen, Algérie, 2011.

[9] F. Douak, "*Reconstruction des images compressées en utilisant les réseaux de neurones artificiels et la DCT,*". Mémoire de Majester de l'université de Batna, Algérie (2008).

[10] G. Burel, "*Réseaux de neurones en traitement d'images: des modèles théoriques aux applications industrielles,*". Thèse de doctorat de l'université de Bretagne Occidentale, 1991.

[11] R. Lepage, "*Reconnaissance d'algues toxiques par vision artificielle et réseau de neurones,* ". Thèse de l'université de Québec, Rimouski (Canada) 2004.

[12] B. Gosselin, *"Application de réseaux de neurones artificiels à la reconnaissance automatique de caractères manuscrits,* " Thèse de la faculté polytechnique de Mons, France 1995.

[13] L. BAGHLI, "*Contribution à la Commande de la Machine Asynchrone, Utilisation de la Logique Floue, des Réseaux de Neurones et des Algorithmes Génétiques*", thèse doctorat de l'université Henri Poincaré, Nancy I, France 1999.

[14] S. Osowski, R. Waszczuk, P. Bojarczak, "*Image Compression Using Feed Forward Neural Networks- Hierarchical Approach,*" Lecture Notes in Computer Science, Book chapter, Springer – Verlag, 2006,

[15] A.V Singh, K.S. Murthy, "*Neuro-Wavelet based Efficient Image Compression Using Vector Quantization*," International Journal of Computer Applications vol. 49-N°.3, 2012.

[16] T. Kohonen, "*Self-organizing Maps, Springer-Verlag*," volume 30, Berlin 1995.

[17] A.K. Alexandridis, A.D. Zaprani, "*Wavelet Neural Networks: A Pratical Guide,*" neural networks 42, 2013,

Contribution à la compression des images

Introduction & Etat d'Art

La compression d'image numérique est un champ de recherche fertile dans le domaine du traitement de l'image en raison de son grand nombre d'applications telles que la surveillance aérienne, la reconnaissance, la médecine et les communications multimédias. Par conséquent, il a reçu une attention particulière de la part des chercheurs dont l'objectif principal est de développer différents schémas de systèmes de compression qui fournissent une bonne qualité visuelle des données avec moins de bits pour le stockage ou la transmission [1,2].

Comme la transformation par les ondelettes est l'une des techniques de compression des images les plus utilisées [2,3], les chercheurs se sont intéressés aux trois étapes constituant le processus de compression par celle-ci à savoir: les transformations de pixels [4,5], la quantification scalaire ou vectorielle [6,7] et le codage entropique [8,9]. Les résultats de ces recherches ont abouti à l'émergence d'un ensemble de transformées en ondelettes qui ont permis la résolution de plusieurs problèmes dans le traitement d'images [2,10]. Malgré cette diversité, des études récentes ont montré que le choix de l'ondelette convenable a un grand impact sur la qualité de compression d'une image [10,11].

Les approches de réseaux neuronaux utilisées pour le traitement des données ont prouvé leur efficacité grâce à leurs structures qui offrent un traitement parallèle des données et un processus de formation qui rend le réseau adapté à différents types de données [12,13,14]. La mise en œuvre du réseau neuronal à chaque étape d'un système de compression d'images a suscité beaucoup d'attention de la part des scientifiques et de nombreux travaux ont démontré que la combinaison entre réseau de neurones artificiels et transformée en ondelettes est un outil précieux pour le traitement d'images [15,16,17,18].

I. Problématique

Il existe un certain nombre de familles d'ondelettes orthogonales couramment utilisées. Les plus connues et les plus simples sont sans doute les ondelettes de Haar. Ce type d'ondelettes est extrêmement utile en pratique car les ondelettes orthogonales sont non redondantes [10] ce qui est bénéfique pour la rapidité de calcul de la transformation et leur inversibilité est assurée. En revanche, comparées aux ondelettes bi-orthogonales, elles n'offrent pas la même souplesse dans leur conception et plusieurs études ont monté que ces dernières assurent les meilleurs taux de compression [10,11].

Cependant, il existe des images dont le contenu ne supporte pas une très grande perte au niveau de l'information. Autrement dit, ces images ne doivent pas être compressées avec un taux élevé et elles se contentent d'un petit taux afin de conserver l'essentiel de leur information d'où l'intérêt d'avoir un système qui peut décider du choix de l'ondelette convenable pour compresser une image précise. Un tel système sera plus bénéfique pour la transmission ou le stockage des images d'importance cruciale comme les images médicales, les images satellitaires, etc… avec un minimum de perte.

II. Les objectifs de la recherche

L'objectif de cette recherche est d'entrainer un réseau de neurones artificielle multicouche PMC, à classer une image fixe avec la transformée en ondelette qui peut servir le mieux dans sa compression et ceci en se basant sur le contenu de l'image et la relation non linéaire entre les valeurs de ses pixels.

Dans ce travail de recherche, l'expérimentation s'est déroulée en suivant deux axes complémentaires:

1. Dans un premier temps, le réseau PMC a été entrainé à relier chaque image à sa transformée en ondelette.

2. Dans la seconde étape, le réseau PMC a été entrainé, en plus de classer l'image avec sa transformée en ondelette, à décider du taux de compression optimal pour sa compression.

III. Outils et méthodologies de travail

Pour la réalisation de ce travail, la simulation a été effectuée en utilisant un ordinateur possédant les propriétés suivantes :

- Processeur de type : Intel(R)Core(TM)i5-6200CPU ;
- Mémoire RAM de 8,00 Go ;
- Système d'exploitation 64 bits.

La programmation a été faite par logiciel MATLAB (R2013a) utilisant les librairies :

- « **Image Processing »** pour le traitement d'image ;
- « **Wavelet** » pour la compression des images ;
- « **Neural Network** » pour la conception des réseaux de neurones à différentes configurations.

La démarche adoptée pour la réalisation de ces deux axes est illustrée par la figure IV.1 qui représente les étapes suivies pour accomplir notre projet :

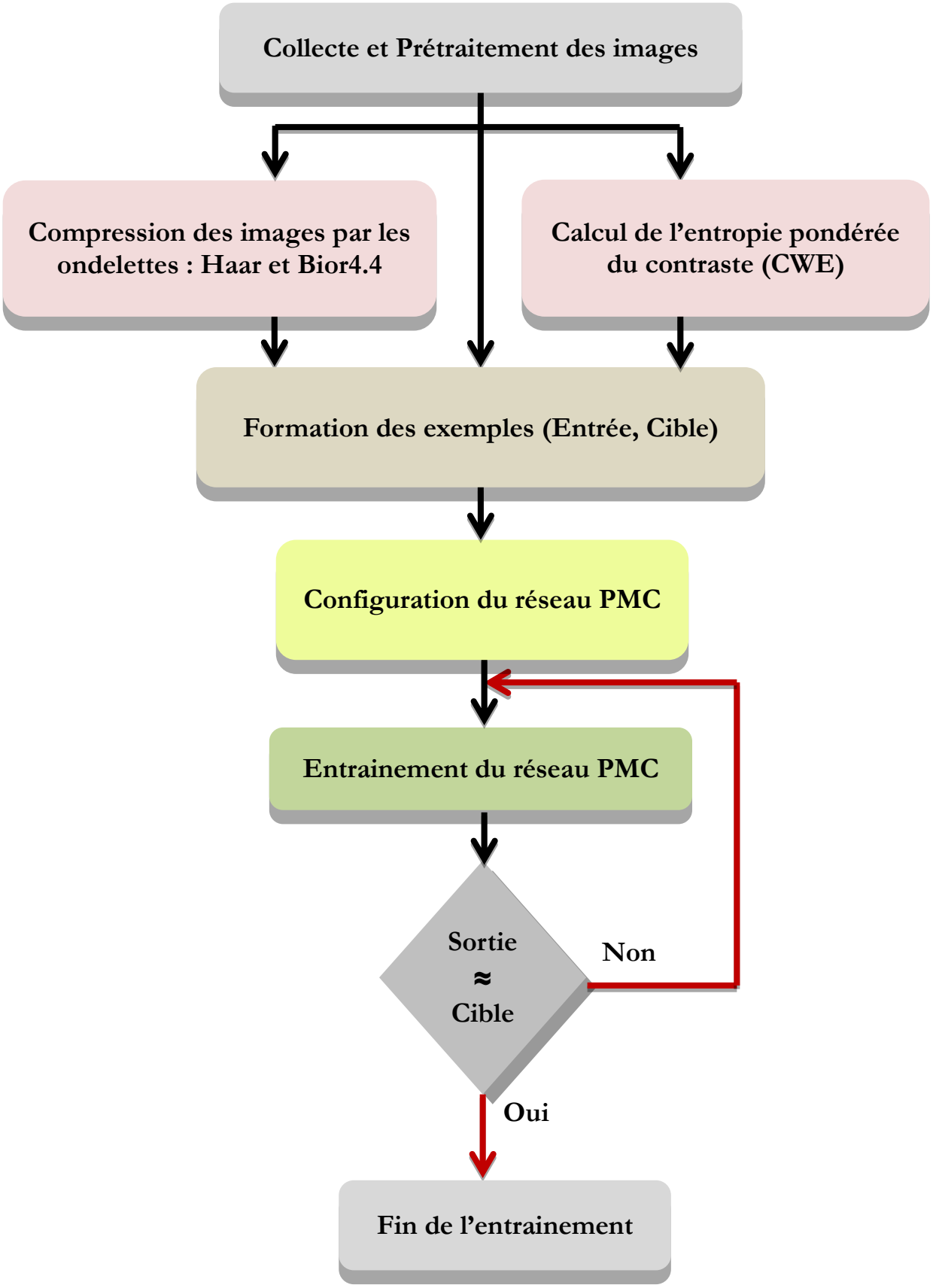


Figure IV.1 La démarche adoptée pour la réalisation du projet

Partie I

Notre but dans cette partie est d'entrainer un réseau de neurones à associer des images en gris à leurs transformations en ondelettes convenables en suivant les pas illustrés dans l'organigramme de la figure IV.1.

I. La collecte et le prétraitement des images fixes

Pour la réalisation de ce projet, 90 images de type (.jpeg) ont été collectées à partir des deux bases de données Wang (http://wang.ist.psu.edu/docs/related/) réservées uniquement pour les travaux de recherche. Une contient 10000 images et l'autre 1000 images. On a pris soin de choisir des images d'objets, de contrastes et d'intensités variés afin de permettre au classificateur une meilleure généralisation. Quelques images de cette base de données sont indiquées par la figure IV.2. On remarque bien la diversité de ces images au niveau du contraste, de la texture et de la luminance.

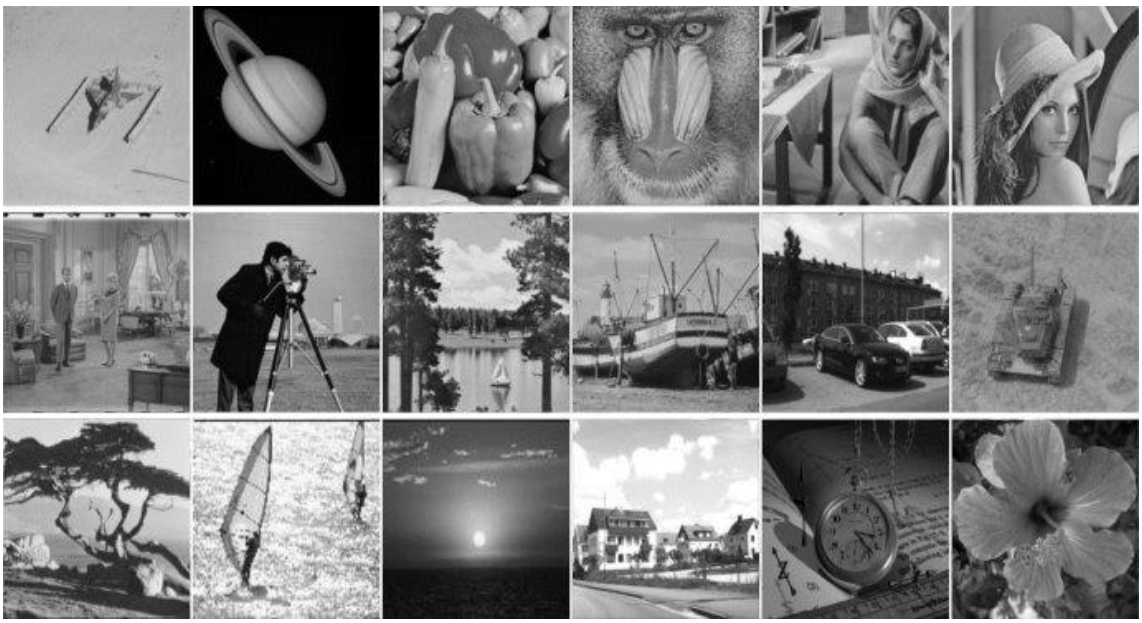


Figure IV.2 Exemples d'images utilisées au cours de cette étude

Avant d'entamer les étapes suivantes du projet, ces images devaient être préparées pour l'apprentissage du réseau de neurones pour cela elles ont subi les traitements suivants :

1. Convertir les images fixes couleurs en images gris. Cette conversion se fait par une commande Matlab qui permet d'éliminer les informations concernant le teint et la saturation et ne garder que la luminance de l'image qui représente ses niveaux de gris.

2. Redimensionner les images en gris obtenues après conversion. En effet, les images initiales ont des résolutions très grandes et il a fallu les réduire, dans un premier temps, à une résolution de 256×256, pour construire une base d'images uniformes, ensuite à une résolution de 32×32 afin de de diminuer le nombre de pixels dans la couche d'entrée de notre réseau. La réduction de la résolution d'une image consiste à diminuer le nombre de pixels qui la composent, c'est un processus avec pertes certes, mais il a fallu passer par cette étape dans le but de réduire le temps d'apprentissage du réseau.

Le tableau IV.1 montre des exemples d'images utilisées subissant les transformations citées en haut.

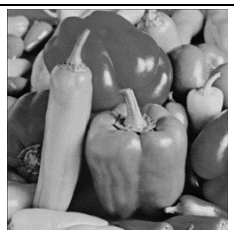
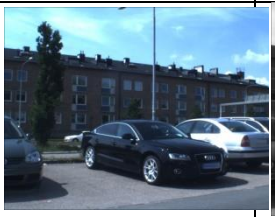
Image couleur originale	Image en gris 256×256	Image en gris 32×32
 1024×768×3	 256×256	 32×32
 490×733×3	 256×256	 32×32
 512×512×3	 256×256	 32×32
 1280×960×3	 256×256	 32×32

Tableau IV.1 Exemples d'images originales converties et redimensionnées.

II. Compression des images et position du problème

II.1 Etape de compression

L'étape suivante consistait à compresser les 90 images en utilisant les transformées en ondelette Haar et Bior4.4. L'algorithme de cette compression est le suivant :

1. Lire l'image couleur.
2. Convertir l'image couleur en image gris.
3. Réduire la taille de l'image à 256×256.
4. Choisir les caractéristiques de la compression :
    - L'ondelette (Haar ou Bior4.4).
    - Nombre de résolution (niveau de décomposition).
    - Type de la quantification.
    - Type du codage.
5. Compresser l'image.
6. Calculer les paramètres de compression : MSE ; PSNR; CR ; BBP.
7. Enregistrer l'image compressée.

Dans le tableau IV.2, on a regroupé quelques images compressées avec leurs paramètres métriques de compression. En se basant sur ces valeurs, on peut dire que la compression par une ondelette bi-orthogonale est meilleure que celle par l'ondelette Haar.

Or du point de vue contenu des images c'est-à-dire, l'information portée par les valeurs des pixels, certaines parmi elles ne tolèrent pas une compression avec un grand taux de compression. En effet, en plus de la quantité d'informations perdue au cours du processus de compression par les ondelettes on a de plus une perte causée par le taux élevé due au mauvais choix de l'ondelette utilisée.

Pour pouvoir distinguer les images dont le contenu est sensible à une compression avec un taux élevé, on a utilisé le coefficient d'entropie pondérée du contraste CWE qui mesure la relation non linéaire entre les intensités des pixels voisins d'une image et son entropie. Ainsi, une image avec un coefficient CWE élevé est une image dont les pixels voisins sont très liés entre eux, raison pour

laquelle il faut la compresser avec un faible taux de compression Tr donc un CR élevé, et ceci afin d'éviter la perte de l'information existante entre les pixels.

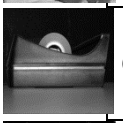
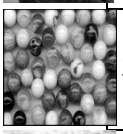
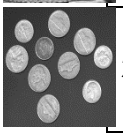
	Ondelette Haar				**CWE**	Ondelette Bior4.4			
	MSE	PSNR (db)	BPP	CR%		MSE	PSNR (db)	BPP	CR%
	26.850	33.841	0.838	10.48	**- 4144.56**	25.670	34.036	0.708	8.85
	29.888	33.375	0.952	11.90	**- 8738.13**	24.803	34.185	0.683	8.55
	6.8979	39.743	0.787	9.84	**- 124833.08**	6.6505	39.902	0.654	8.18
	10.407	37.957	1.482	18.53	**- 1189140.61**	9.2549	38.467	1.020	12.76
	28.646	33.560	0.690	8.63	**- 66413.42**	19.366	35.260	0.459	5.74
	38.235	32.306	1.197	14.96	**- 2528233.5**	29.091	33.493	0.753	9.42
	51.216	31.036	1.726	21.58	**- 78647.74**	46.379	31.467	1.327	16.59
	25.481	34.068	0.751	9.39	**- 4519.72**	21.429	34.820	0.581	7.27

Tableau IV.2 Les coefficients métriques de compression et les CWE de quelques images utilisées.

II.2 Position du problème et Solution

Pour expliquer le but derrière cette étude, prenons l'exemple de l'image « coins » dernière image dans le tableau IV.2. Cette image a été compressée en utilisant les deux transformées en ondelette Bior4.4 et Haar, les résultats de ces deux compressions sont représentés respectivement sur les figues IV.3 et IV. En tenant compte des paramètres métriques de compression et la bonne visualité de l'image compressée qui doit être le plus possible identique à l'originale on peut dire que la meilleure transformation pour la compression de cette image est bien sûr l'ondelette bi-orthogonale Bior4.4. D'un autre côté la grande valeur (-4519.7238)

calculée pour son coefficient d'entropie de contraste indique que l'information au niveau des pixels de cette image est importante et on n'a pas intérêt d'en perdre beaucoup par conséquent une compression avec un taux Tr faible (CR grand) est souhaitable. On conclue donc qu'il est préférable dans ce cas et dans le but de préserver le maximum d'information transmis par cette image, de la compresser via l'ondelette Haar, d'où la nécessité de réaliser un système qui peut décider automatiquement du choix de l'ondelette convenable en s'appuyant sur le contenu des pixels de l'image à compresser.

Pour réaliser ce système décisionnel, on propose d'utiliser un classificateur qui va permettre d'associer le contenu d'une image fixe en gris à l'ondelette convenable à sa compression.

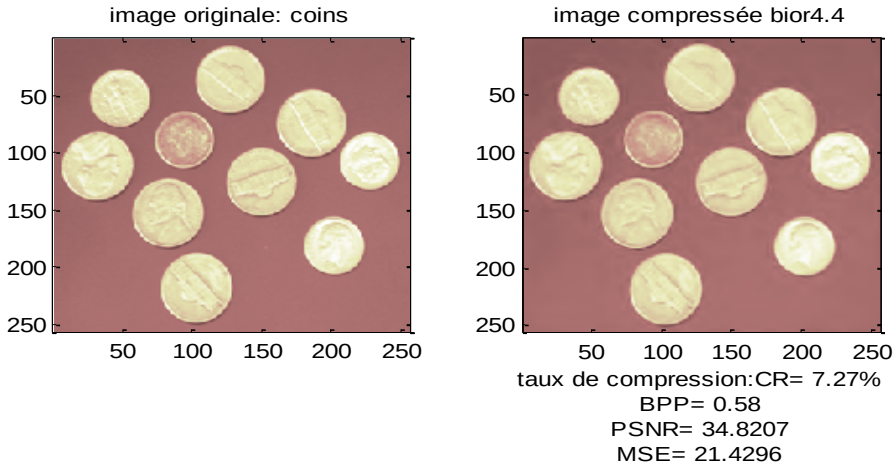


Figure IV.3 Compression de l'image « Coins » par la transformée de Bior4.4

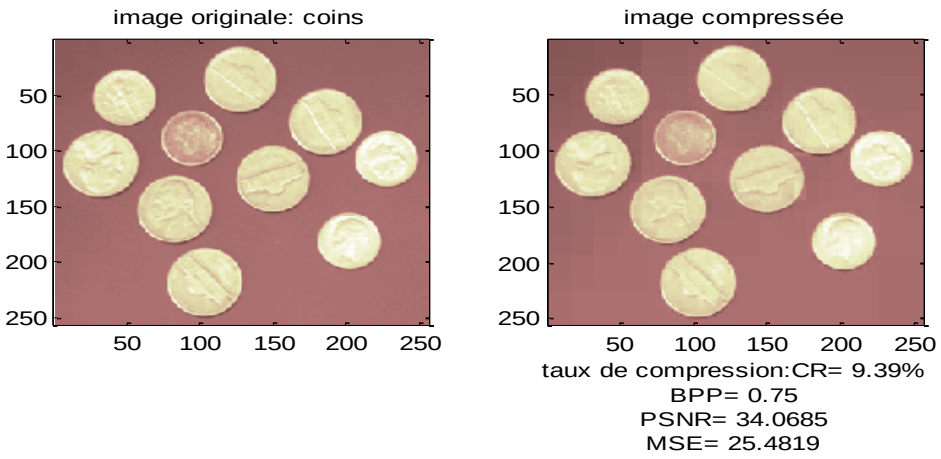


Figure IV.4 Compression de l'image « Coins » par la transformée de Haar

III. Choix et Configuration du classificateur

III.1 Choix et architecture du classificateur

Pour le choix du classificateur convenable pour mener à terme ce travail et pour les raisons citées dans le chapitre III, on a opté pour les réseaux de neurones artificiels et plus précisément les perceptrons multicouches PMC. Un tel choix a été justifié par le fait que plusieurs recherches ont prouvé qu'ils sont les plus performants en cas de problème de classification.

En vue de classer les 90 images collectées et traitées par le réseau de neurones, ce dernier nécessitait un dimensionnement. Mis à part les couches d'entrée et de sortie, l'analyste doit décider du nombre de couches intermédiaires ou cachées. Sans couche cachée, le réseau n'offre que de faibles possibilités d'adaptation; avec une couche cachée, il est capable, avec un nombre suffisant de neurones, d'approximer toute fonction continue. Une seconde couche cachée prend en compte les discontinuités éventuelles. Ainsi notre réseau est constitué des couches suivantes :

- ***Couche d'entrée***

  En utilisant l'approche un pixel par neurone, cette couche contient donc 1024 neurones représentant les pixels de l'image.

- ***Couche cachée***

  Chaque neurone supplémentaire de la couche cachée permet de prendre en compte des profils spécifiques des neurones d'entrée. Un nombre très élevé de neurones diminue la capacité de généralisation du réseau. Pour cela, on commence aléatoirement par un petit nombre de neurones et après plusieurs essais et tentatives on décide du nombre à choisie. Dans notre cas, avec un nombre de 58 neurones dans la couche cachée, on a obtenu la meilleure performance du réseau.

- ***Couche de sortie***

  Cette couche contient 2 neurones représentant les deux décisions que doit prendre le réseau à savoir l'ondelette Haar ou l'ondelette Bior4.4.

Un schéma fonctionnel du réseau utilisé est représenté sur la figure IV.5

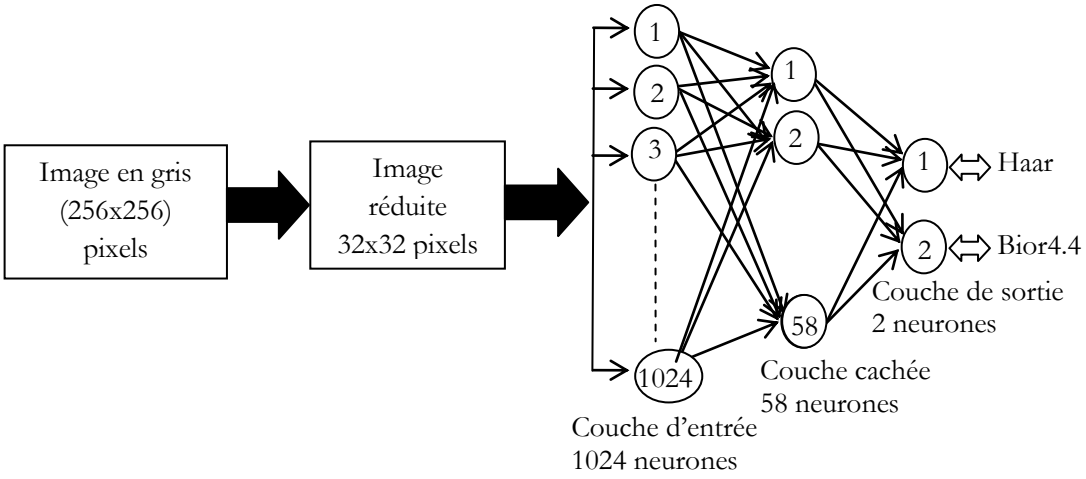


Figure IV.5 Schéma fonctionnel du réseau PMC utilisé

III.2 Configuration du réseau

La configuration du réseau consiste à partager la base des images en trois ensembles. Il n'existe aucune règle à suivre pour partager ces données, cependant tant que le nombre d'échantillons réservés à l'apprentissage du réseau est important tant la convergence de celui-ci est assurée. La base des images est donc subdivisée en :

1. **Une base d'apprentissage** qui contient des images avec leur transformée en ondelette convenable en se basant sur la valeur de leur coefficient d'entropie de contraste. Cette base est présentée à plusieurs reprises au réseau durant la phase d'apprentissage et à chaque fois, la sortie du réseau a été comparée à la cible afin de calculer l'erreur quadratique. Cette erreur est utilisée pour ajuster les poids du réseau de neurones de manière à ce que le modèle se rapproche enfin de la reproduction de la cible.
   Dans notre cas, cette base contient 54 images parmi les 90 images réservées pour cette étude soit 60% de cette base.

2. **Une base de validation de l'apprentissage** qui contient 18 images, soit 20% de la base globale, utilisées pour mesurer la généralisation du réseau et arrêter l'entraînement lorsque la généralisation cesse de s'améliorer.

3. **Une base de test** qui contient aussi 18 images dont la transformée en ondelette est inconnue pour le réseau. Le but de ce jeu de données et de tester la performance du réseau.

IV. Apprentissage du réseau

L'apprentissage du réseau de neurones PMC est basé sur l'algorithme de rétro-propagation de l'erreur qui nécessite la détermination des paramètres d'ajustement des poids synaptiques à chaque itération.

Du point de vue calcul, l'entrainement du réseau se repose sur la série d'étapes ci-dessous :

1. Entrer la matrice d'entrée (couche d'entrées : [1024 90])
2. Entrer la matrice des cibles (couche de sortie : [2 90])
3. Configurer le réseau de neurone :
   - Indiquer le nombre de couches cachées
   - Indiquer le nombre de neurones dans la couche cachée
4. Subdiviser les données en :
   - Base d'apprentissage : 60/100.
   - Base de validation : 20/100.
   - Base de test : 20/100.
5. Choisir l'algorithme d'apprentissage (rétro-propagation)
6. Choisir la fonction de performance MSE (Mean squared error))
7. Entrainer le réseau de neurones.
8. Tester le réseau de neurones en calculant MSE entre les sorties et les cibles
9. Afficher les courbes de variations :
   - De l'erreur MSE en fonctions du nombre d'itérations.
   - Du gradient de l'erreur en fonction du nombre d'itérations
10. Afficher les matrices de confusion.

V. Résultats et Discussions

Comme on a mentionné précédemment, le réseau de neurones utilisé dans ce travail est un réseau multicouche PMC dont l'apprentissage se base sur l'algorithme de rétro-propagation du gradient.

Pendant la phase d'apprentissage, les paramètres du réseau ont été initialisés aléatoirement et ajustés pour optimiser ses performances qui sont évaluées par l'une des fonctions suivantes:

- L'erreur quadratique moyenne MSE (Mean Square Error) qui représente la différence entre les sorties du réseau et les cibles.
- La magnitude du gradient de l'erreur MSE.
- Les matrices de confusion

Après 61 itérations et au bout de 25s, l'erreur MSE a atteint la valeur minimale $5{,}758.10^{-8}$. La figure IV.6 représente la variation de la fonction de performance MSE en fonction du nombre d'itération durant l'apprentissage du réseau. Un tel résultat montre que l'apprentissage du réseau s'est fait à 100% vue que l'erreur entre les sorties et les cibles au cours de la phase d'apprentissage est très faible.

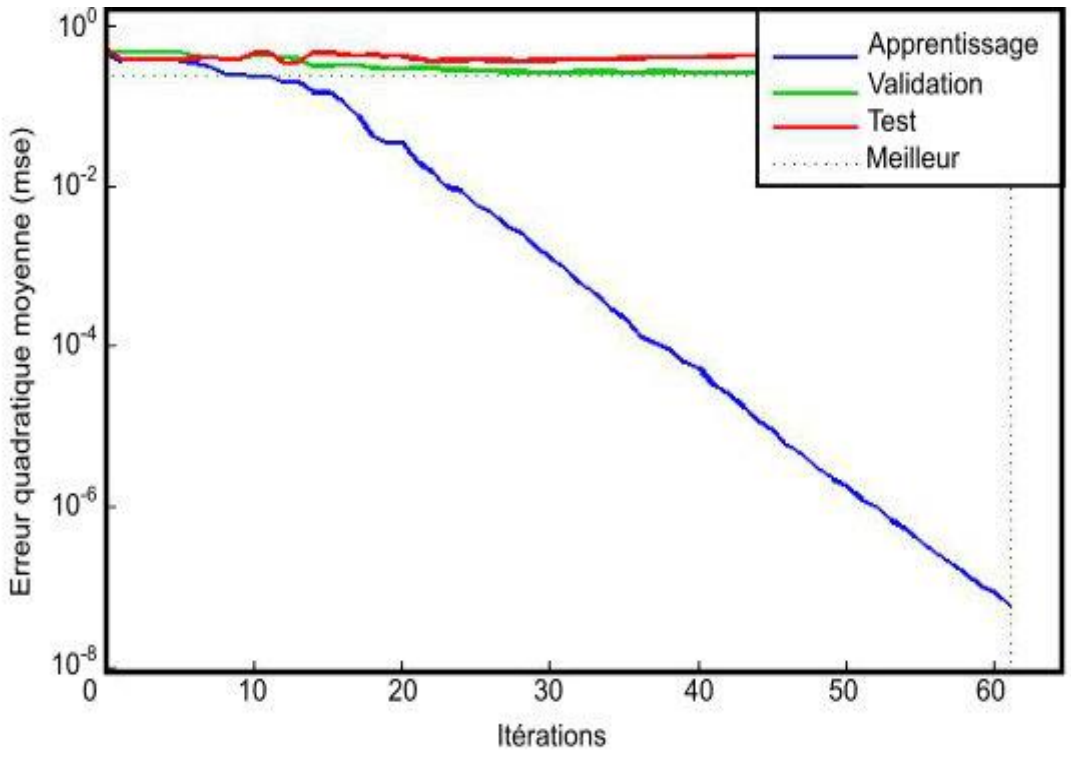


Figure IV.6 Variation de l'erreur MSE en fonction du nombre d'itérations pour un taux d'apprentissage de 100%.

Un autre paramètre qui permet de mesurer la performation du réseau est la magnitude du gradient présenté sur la figure IV.7 qui montre la variation de ce paramètre en fonction du nombre d'itérations. Après 61 itérations, le gradient

atteint la valeur de 9,92 10^{-7} qui est très faible ce qui implique que la fonction de performance MSE a été minimisée autrement dit que les prévisions du réseau (sorties) étaient très proches des cibles.

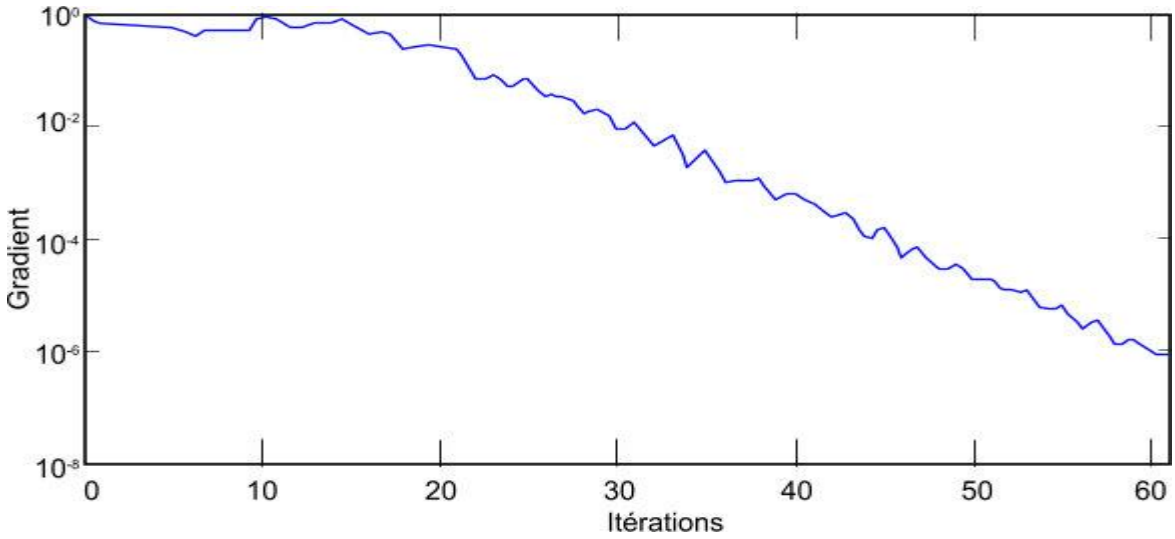


Figure IV.7 Variation du gradient durant la phase d'apprentissage pour un taux d'apprentissage de 100%.

Pour plus d'analyse de la performance du réseau, on utilise les matrices de confusion, présentées sur la figure IV.8, relatives aux trois phases de l'entrainement du réseau. Elles montrent la capacité de généralisation du réseau pour les trois types de bases. D'après les résultats, on peut dire que le réseau a été capable de classer toutes les 54 images de la base d'apprentissage avec leurs ondelettes avec un pourcentage de reconnaissance de 100%. Cependant durant la phase de test, et avec cette configuration de poids, le réseau n'a pas pu classer que 8 images parmi les 18 de la base de test, offrant ainsi un pourcentage de reconnaissance de 44,4% ce qui est insuffisant.

Une nouvelle configuration de poids a permis d'avoir un pourcentage d'apprentissage de 94,4% après 45 itérations et au bout de 17s. Autrement dit, que durant la phase d'apprentissage, le réseau a pu classé 51 images parmi 54. Une fois entrainé, ce réseau a été testé sur les images de la base de test et il a pu classé 14 parmi les 18 résérvées pour sa généralisation réalisant ainsi un pourcentage de reconnaissace de 77,8% comme le montre les matrices de confusion représentées sur la figure IV.9.

FigureIV.8 Matrices de confusion avec un taux d'apprentissage de 100%

FigureIV.9 Matrices de confusion avec un taux d'apprentissage de 94,4%

La figure IV.10 présente des exemples de quelques images de la base de test qui sont compressées en utilisant l'ondelette choisie par le réseau de neurones après son entrainement.

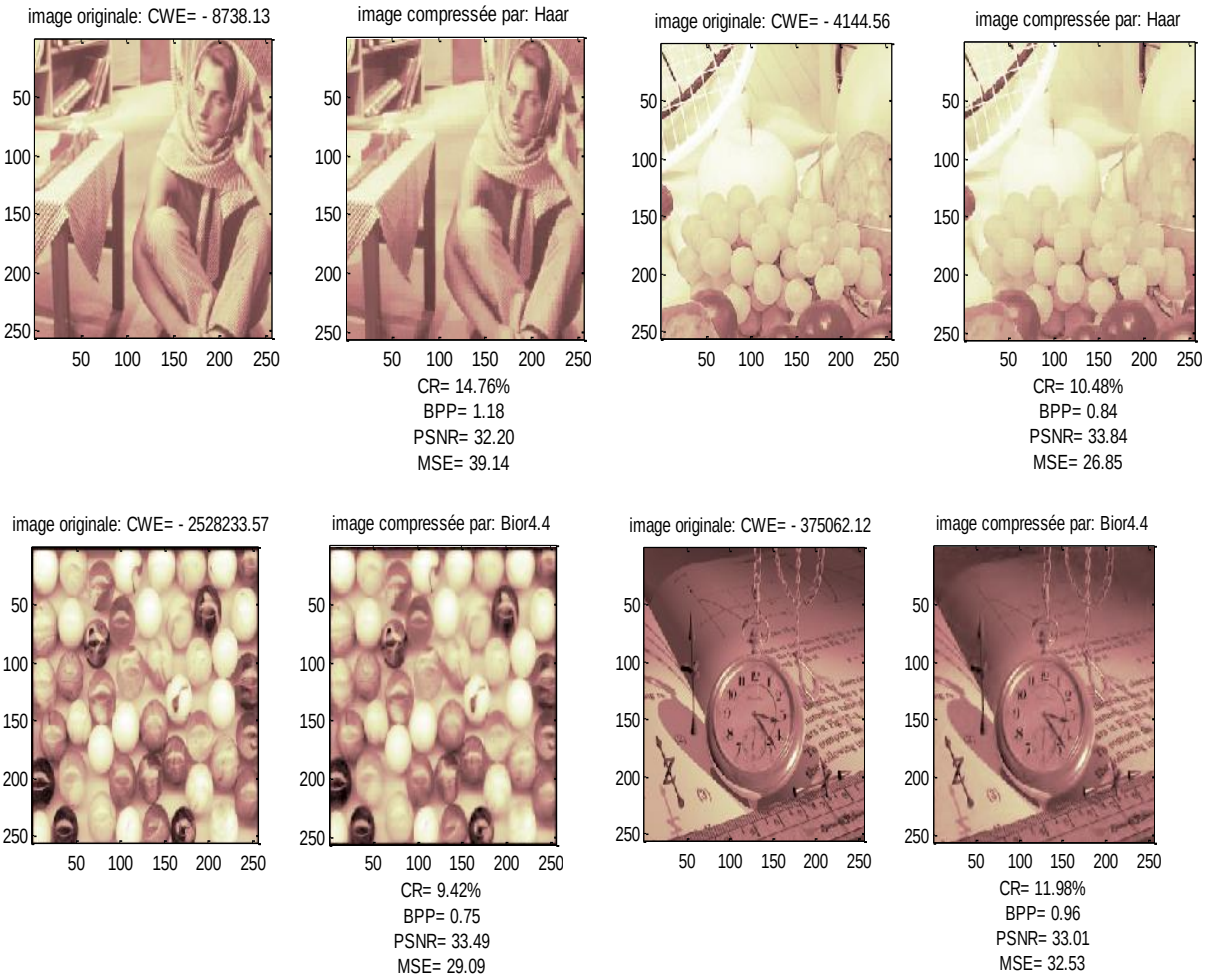


Figure IV.10 Quelques images de test compressées selon l'ondelette choisie par le réseau

Les paramètres du réseau de neurone entrainé pour cette partie du projet sont regroupés dans le tableau IV.3.

	Taux d'apprentissage 100%	Taux d'apprentissage 94.4%
Nombre de neurones de la couche d'entrée	1024	1024
Nombre de neurones de la couche cachée	58	58
Nombre de neurones de la couche de sortie	2	2
Nombre d'itérations	61	45
Temps d'apprentissage (s)	25	17
Performance	$5{,}758.10^{-8}$	$0{,}4.10^{-3}$
Gradient	$9{,}92\ 10^{-7}$	$0{,}156.10^{-2}$
Taux de généralisation (%)	44,4	77,8

Tableau IV.3 Paramètres d'entrainement du réseau PMC

Partie II

I. Collecte et le Prétraitement des images fixes

Dans cette partie on a voulu pousser un peu notre défit en entrainant notre réseau en plus de connecter une image à sa transformée en ondelette convenable mais aussi à son taux de compression optimal.

Pour augmenter la performance du réseau on a augmenté le nombre des échantillons à lui présenter donc on a pris une base de données de 120 images d'objets, de contrastes et d'intensités variés. Pour la mise en forme de ces images on a procédé de la même manière que la première partie c'est-à-dire que les images ont été converties en gris et réduites à une résolution de 32×32. Le tableau IV.3 présente des exemples d'images de la nouvelle base de données collectées pour cette étude en plus des valeurs de leurs coefficients d'entropie de contraste sur lesquels on s'est basé pour former les réponses (cibles) auxquelles le réseau doit comparer ses sorties.

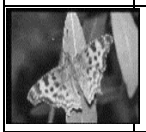
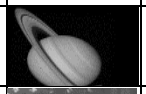
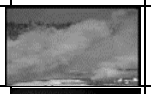
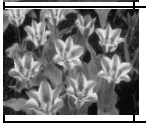
Image originale	CWE	Image originale	CWE	Image originale	CWE	Image originale	CWE
	-287191.26		-6968686.29		-5957081		-5825.42
	-11045.39		-8024.81		-8591882.81		-21947559.9
	-375062.12		-4228.12		-6944.21		-2150289.31
	-4443.11		-1675197.72		-671150.41		-4519.72
	-1720238.8		-1619143.25		-395664.54		-78030.90

Tableau IV.4 Quelques images utilisées pour la 2éme partie avec leurs CWE

II. Configuration et Dimensionnement du réseau

Notre réseau de neurones est toujours un PMC basé sur un apprentissage par rétro-propagation du gradient de l'erreur quadratique. Sa nouvelle topologie est illustrée par la figure IV.11.

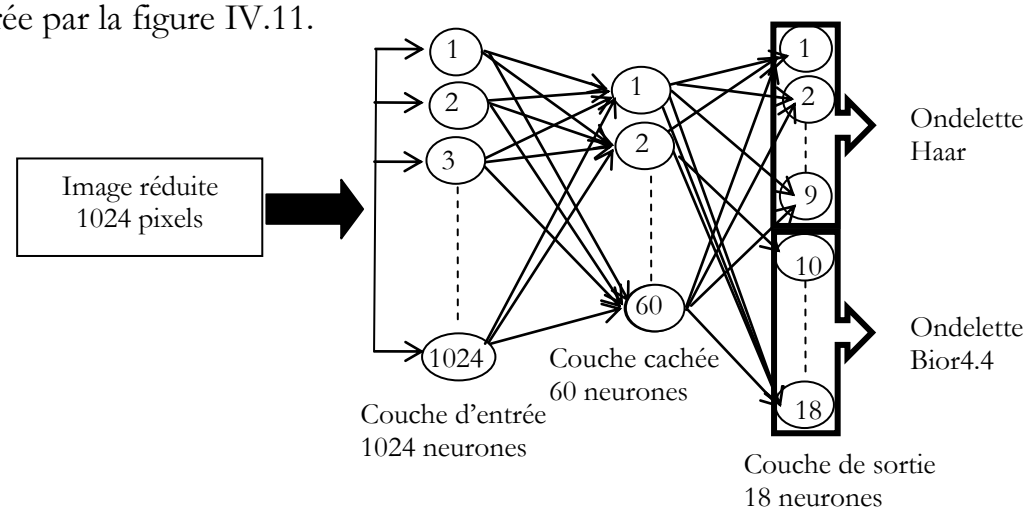


Figure IV.11 Schéma fonctionnel du réseau PMC pour la 2éme partie

Le réseau possède toujours trois couches :

- Une couche d'entrée de 1024 neurones représentant les pixels de l'image à présenter au réseau.

- Une couche cachée avec 60 neurones. Ce choix a été fait après plusieurs expériences qui ont abouti à la valeur minimale significative de l'erreur MSE.

- Une couche de sortie de 18 neurones dans lesquels les 9 premiers neurones représentent les 9 taux de compression par l'ondelette Haar {10%, 20%, ……, 90%} et les 9 derniers neurones représentent les mêmes taux de compression par l'ondelette Bior4.4.

De même les données présentées au réseau de neurones au cours de son entrainement sont subdivisées aléatoirement en trois bases :

- **Base d'apprentissage** contenant 72 images présentées au réseau avec leur transformée en ondelette convenable et le taux de compression optimal.

- **Base de validation** contenant 24 images.

- **Base de test** contenant 24 images jamais présentées au réseau auparavant et qui vont servir pour tester la performance du réseau et sa capacité de généralisation.

III. Résultats et Discussions

Pendant la phase d'apprentissage, les paramètres du réseau ont été initialisé aléatoirement et ajustés pour optimiser les performances du réseau évaluées toujours par la fonction erreur quadratique moyenne MSE entre les sorties du réseau et les cibles désirées et la magnitude du gradient.

Après plusieurs essais, le nombre de neurones de la couche cachée a été fixé à 60, ce qui a assuré une meilleure performance du réseau par la valeur $6{,}06 \times 10^{-6}$ atteinte par la fonction MSE. La figure IV.12 présente la variation de la fonction de performance MSE en fonction du nombre d'itérations au cours des trois phases de l'entrainement du réseau à savoir : phase d'apprentissage, de validation et de test. D'après cette figure, on peut dire que l'entraînement est raisonnable puisque d'une part, l'erreur quadratique MSE est très faible et se stabilise après 55 itérations et d'autre part la courbe de validation et la courbe de test convergent en même temps et de la même manière.

De plus avec cette configuration de poids, le réseau neuronal apprend et converge après 55 itérations et au bout de 33s reconnaissant correctement les 72 images de la base d'apprentissage et offrant ainsi un pourcentage de reconnaissance de 100%. Cependant, seules 10 images sur les 24 de l'ensemble de test ont été associées avec leurs méthodes de compression idéale et leurs taux optimal offrant ainsi un pourcentage de classification de 40%, ce qui n'est pas suffisant.

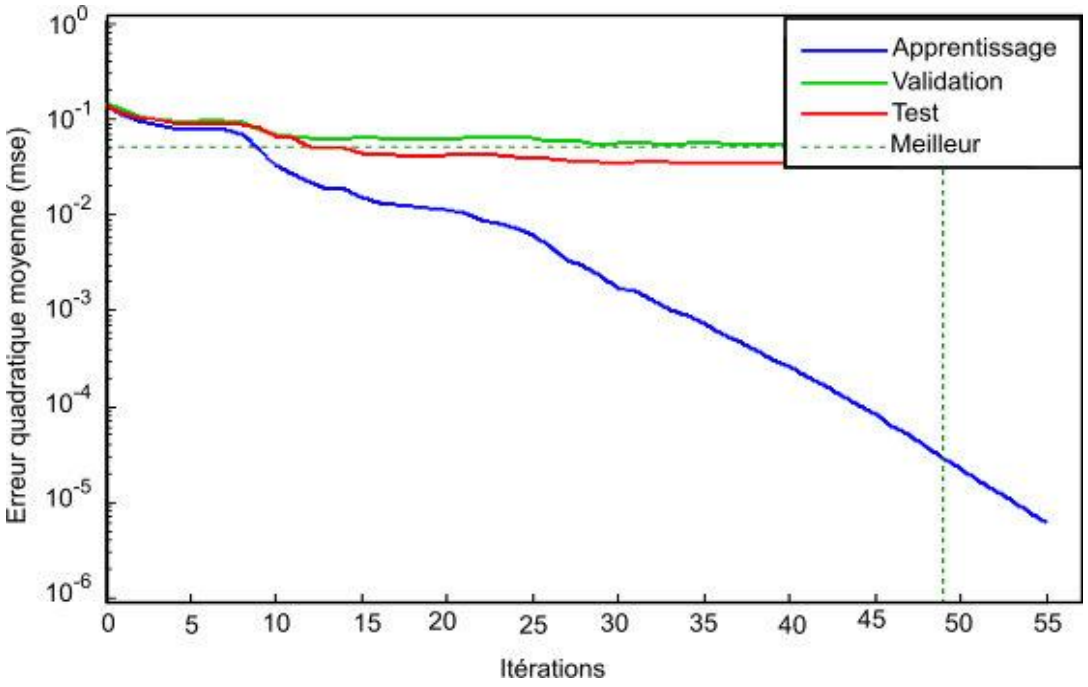


Figure IV.12 Variation de l'erreur MSE en fonction du nombre d'itérations pour un taux d'apprentissage de 100%.

Pour la même configuration de poids, le gradient de la fonction de performance MSE a atteint une valeur minimale de $6{,}98 \times 10^{-5}$ d'après la courbe de la figure IV.13, ce qui implique que la fonction MSE a été bien minimisée et que l'apprentissage du réseau a été fait avec succès.

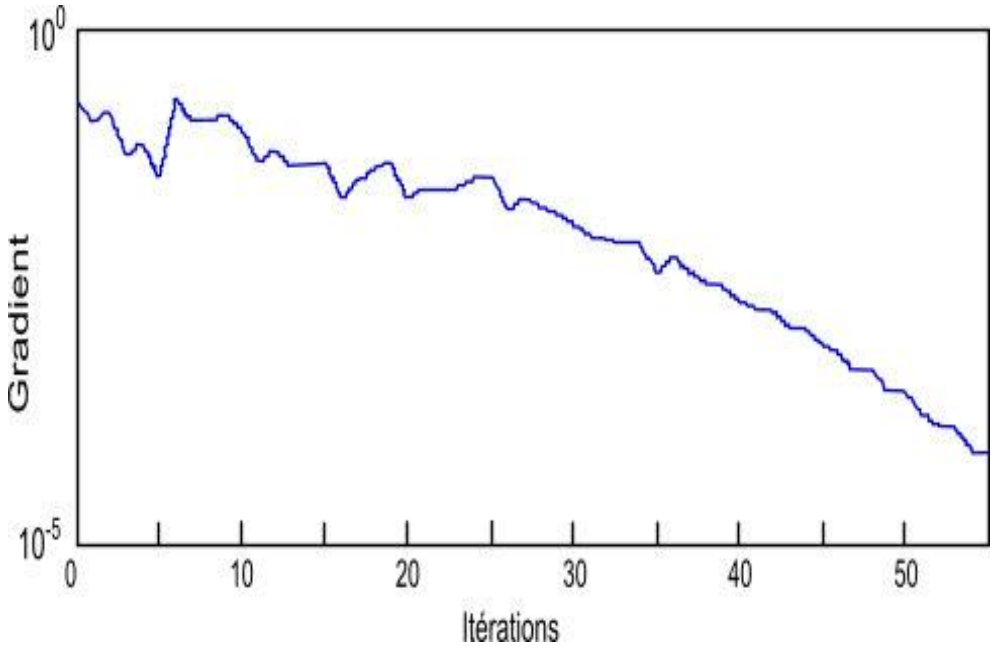


Figure IV.13 variation du gradient de MSE en fonction du nombre d'itérations

Cependant pour notre étude un pourcentage de 40% durant la phase de test est insuffisant c'est pour cela qu'on a adopté une autre configuration des poids qui a permis un apprentissage du réseau de l'ordre de 98%. Avec ce pourcentage le réseau PMC a pu classer 70 images parmi les 72 formant la base d'apprentissage après 71 itérations et au bout de 29s. La fonction de performance MSE a atteint la valeur basse de 0,00169 qu'on considérée comme acceptable. La figure IV.14 présente la variation de la fonction de performance MSE en fonction du nombre d'itération durant les trois phases de l'entrainement du réseau.

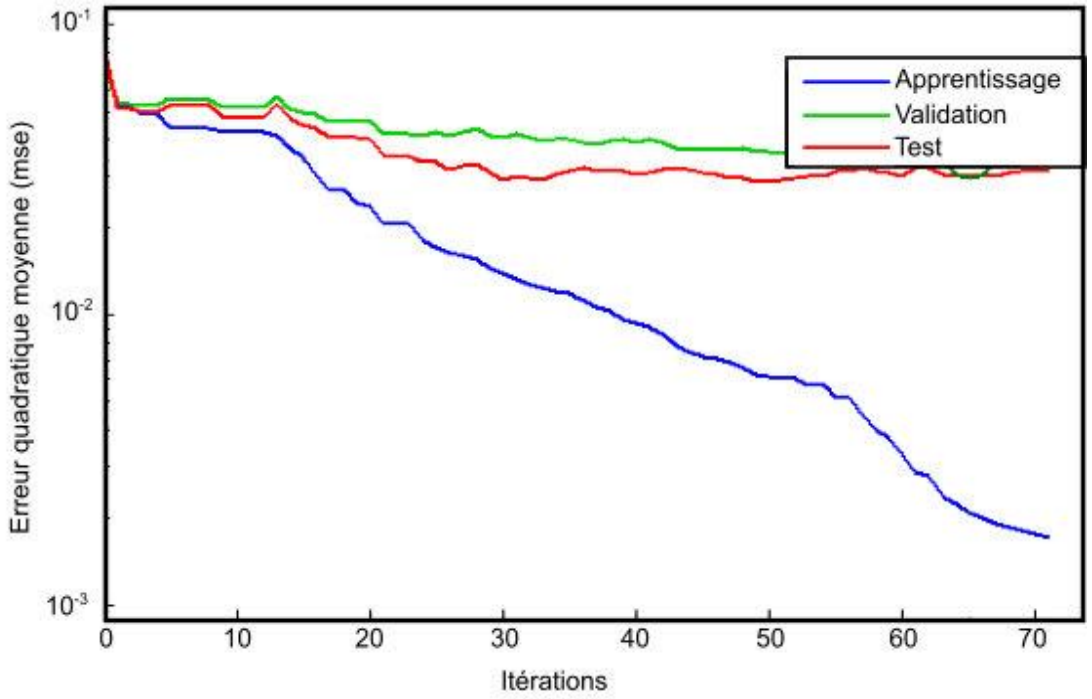


Figure IV.14 Variation de MSE en fonction du nombre d'itérations pour un taux d'apprentissage de 98%

De plus, le gradient de la fonction de performance, représentée par la figure IV.15 montre bien que celui-ci a atteint la valeur 0.00113 après 71 itérations ce qui indique que les sorties données par le réseau sont proches des cibles désirées.

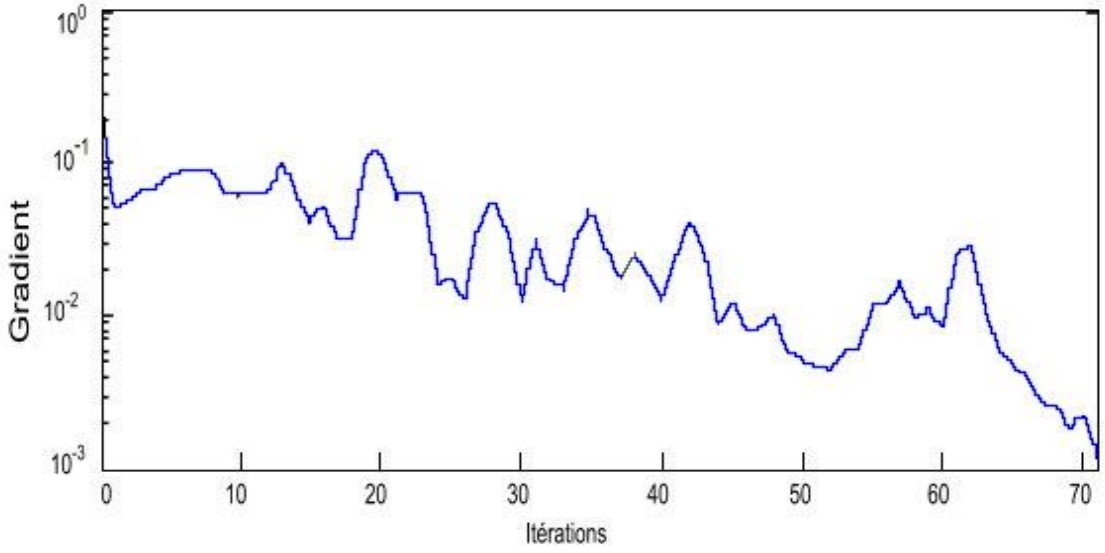


Figure IV.15 Variation du gradient de MSE en fonction du nombre d'itérations pour un taux d'apprentissage de 98%

En adoptant cette configuration de poids pour les connexions du réseau, celui-ci a pu correctement classer 18 images, parmi les 24 constituants la base de test, avec leur transformée idéale et leur taux de compression optimal ce qui fait un pourcentage de 75% de reconnaissance.

On présente sur la figure IV.16 quelques images qui n'ont jamais été présenté au réseau et que celui-ci a pu classer avec l'ondelette convenable et le taux optimal.

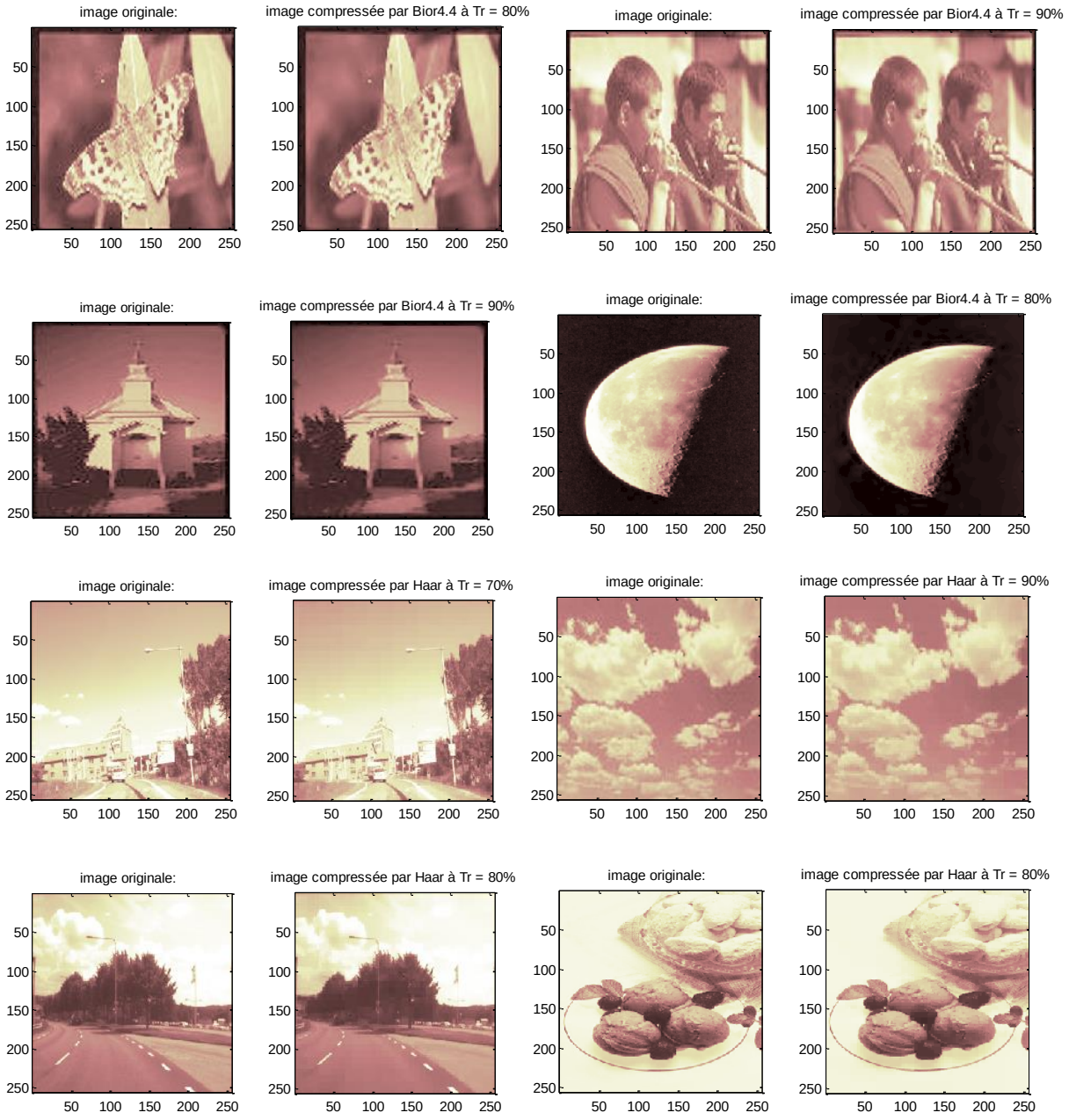


Figure IV.16 Des images de la base de test classées par le réseau de neurones PMC

Les paramètres de ce réseau sont regroupés dans le tableau suivant :

	Taux d'apprentissage 100%	Taux d'apprentissage 98%
Nombre de neurones de la couche d'entrée	1024	1024
Nombre de neurones de la couche cachée	60	60
Nombre de neurones de la couche de sortie	18	18
Nombre d'itérations	55	71
Temps d'apprentissage (s)	33	29
Performance	$6{,}06\text{x}10^{-6}$	0,00169
Gradient	$6{,}98\text{x}10^{-5}$	0.00113
Taux de généralisation (%)	40	75

Tableau IV.5 Paramètres d'entrainement du réseau PMC

Conclusion

Dans ce chapitre on a proposé une nouvelle approche pour la compression des images fixes se basant sur la technique des réseaux neurones artificiels. Notre étude a été faite en deux parties :

Première partie: le réseau de neurone a été entrainé à associer 90 images d'objets, de contrastes et d'intensités variés à leur fonction de compression idéale. Pour un taux d'apprentissage de 94,4%, le réseau de neurones a produit un pourcentage de classification des images de test de 77,8%. Autrement dit, il a pu classer 14 images parmi 18 celles qu'il n'a jamais vu auparavant et qui étaient réservées pour tester sa performance. Et que uniquement 4 parmi elles étaient mal classées

Deuxième partie : on a jugé de pousser plus loin le travail du réseau qui va cette fois-ci associer le contenu d'une image fixe en gris à sa fonction ondelette convenable tout en précisant le taux de compression optimal. Dans le but d'améliorer la performance du réseau, on a augmenté le nombre d'échantillons en utilisant une base de données constituée de 120 images. Une fois entrainé, la capacité de généralisation de ce réseau à des situations nouvelles (non vues durant son apprentissage) a atteint taux de reconnaissance de 75%. Autrement dit, ce

réseau neuronal a pu classer 18 images, parmi les 24 de l'ensemble de test, à l'ondelette idéale et le taux de compression optimal.

Etant donné que la subdivision de la base des images en trois regroupements se fait aléatoirement, on ne peut pas éviter le risque de tomber sur des incohérences au niveau de la base d'apprentissage ou celle de test : comme avoir deux images presque identiques avec des décisions du réseau qui sont différentes. Donc, on peut dire que les résultats obtenus sont concluants et satisfaisants et que les objectifs de ce projet de recherche sont atteints.

Bibliographie

[1] A. Said, W.A. Pearlman, "*A new Fast and Efficient Image Codec based on Set Partitioning in Hierarchical Trees*," IEEE, Trans. Circuits syst. Video Technol. 6 (3), 1996.

[2] K. Ratakondu, N. Ahuja, "*Loless Image Compression with Multiscale Segmentation,*". IEEE, Transactions on Image Processing, vol. 11, N°. 11, 2002.

[3] K. Ahmadi, A.Y. Javaid, E. Salari, "*An efficient Compression Scheme based on Adaptive Thresholding in Wavelet Domain using Particle Swarm Optimization,*" signal processing: image communication 32, 2015.

[4] I. Daubechies. "*Orthonormal Bases of Compactly Supported Wavelets*". Comm. Pure. Appl. Math., No. 41, 1988.

[5] R.R.Coifman, M.V.Wickerhauser "*Wavelets and adapted waveform analysis*" in Proceedings of symposia in applied mathematics,SIAM vol. 47, 1993, éditeur Ingrid Daubechies. Ed. Politehnica, Timişoara, 1998.

[6] P.C. Cosman, R.M. Gray, M. Vetterli, "*Vector Quantization of Image Sub-bands: A Survey,*" IEEE Trans. Image Processing, vol. 5, 1996.

[7] P. Mathieu, M. Barlaud, M. Antonini, "*Compression d'images par transformée en ondelette et quantification vectorielle*", Journal Traitement de Signal vol7, N°2, 1990

[8] A. Moffat, R. M. Neal, I. H. Witten, "*Arithmetic coding revisited*", ACM Transactions on Information Systems, 16(3), July 1998.

[9] J. S. Vitter. "*Design and analysis of dynamic Huffman codes,*" ACM Transactions on Mathematical Software, Volume 15 , Issue 2 (1989),

[10] T. Gandhi, B. K. Panigrahi, and S. Anand, *"A Comparative Study of Wavelet Families for EEG Signal Classification,"* Neurocomputing 74, 2011.

[11] K.H. Talukder, K.Harada, "*Haar Wavelet based approach for image compression and quality assessment of compressed image,*". IAENG International Journal of Adpplied Mathematics 2007.

[12] T. Denk, K. Perhi, V. cherkassky, " *Combining neural network and wavelet transform for image compression, "*. Proceeding of intl conf 1993.

[13] S. Osowski, R. Waszczuk, P. Bojarczak, "*Image Compression Using Feed Forward Neural Networks- Hierarchical Approach,"* Lecture Notes in Computer Science, Book chapter, Springer – Verlag, vol. 3497, 2006.

[14] C. Ben Amar, O. Jemai, " *Wavelet Networks Approch for Image Compression,*" ICGST, GVIP special issue on image compression, 2007.

[15] A.V. Singh, K.S. Murthy, "*Neuro-Wavelet based Efficient Image Compression Using Vector Quantization*," International Journal of Computer Applications (0975-08887), vol. 49-N°.3, July 2012.

[16] V. Krishnanaik, G.M. Someswar, K. Purushotham, A. Rajaiah,"*Implementation of Wavelet Transform, DPCM and Neural Network for Image Compression,"* International Journal of Engineering and Computer science, vol. 2, issue. 8, August 2013.

[17] K. Dimililer, A. Khashman, " *Image Compression using Neural Networks and Haar wavelet*," Transaction on Signal Processing, ISSN: 1790-5052, vol. 4, issue: 5, May 2008.

[18] A.J. Hussain, D. Al-Jumeily, N. Radi, P. Lisboa," *Hybrid Neural Network Predictive-Wavelet Image Compression System,*" neurocomputing 151, 2015.

Conclusion Générale & Perspectives

Dans le présent travail de cette thèse, on a proposé une nouvelle approche pour la compression des images fixes en gris, combinant deux techniques qui ont prouvées leur efficacité dans le traitement de signal en général, et dans le traitement d'images en particulier à savoir la transformée par les ondelettes et les réseaux de neurones artificiels.

Les transformées en ondelettes ont gagné un intérêt considérable pour le traitement des images, qui s'explique par le fait que l'analyse multi-résolution qu'elles utilisent ne fait apparaitre des coefficients significatifs qu'à un petit nombre de position dans le plan temps-échelle, permettant ainsi d'éliminer tout ce qui est redondant dans l'image y compris les détails les plus insignifiants. Cette caractéristique est très utile en compression d'image car un petit nombre de coefficients est suffisant pour en reconstruire l'essentiel.

Les réseaux de neurones dont le fonctionnement s'appuie sur l'apprentissage, n'ont cessé de prouver leur efficacité dans la réalisation de différentes applications et la résolution de plusieurs problèmes de classification ou d'optimisation.

Notre objectif dans ce projet est d'associer une image à la fonction d'ondelette qui convient le mieux à sa compression avec un taux qui lui assure une bonne qualité visuelle. Le classificateur proposé va se baser sur le contenu de l'image et la relation non linéaire entre les valeurs de ses pixels, pour la relier à l'une des deux ondelettes choisies pour cette étude à savoir: l'ondelette orthogonale Haar et celle bi-orthogonale Bior4.4.

Ce travail s'est déroulé en deux axes :

Premier axe : le réseau de neurone a été entrainé à associer 90 images d'objets, de contrastes et d'intensités variés à leur fonction de compression idéale. Il était capable de classer tous les échantillons réservés à son apprentissage avec une performance quasi optimale avec un taux de reconnaissance de 94,4%. Et d'un autre côté sa capacité de généralisation à des situations nouvelles (images de test) a atteint 77,8%.

Deuxième axe : le réseau a été entrainé cette fois ci à relier le contenu d'une image à la transformée en ondelette qui convient le mieux à sa compression et au taux de compression optimal. Pour atteindre notre but, nous avons utilisé une base de données constituée de 120 images. Le réseau neuronal implémenté a été capable de

classer toutes les images de la base d'apprentissage avec un taux de 98%. Testé sur de nouvelles images non vues durant son apprentissage, il a pu classer 75% d'entre elles avec l'ondelette convenable et le taux de compression optimal.

En tenant compte du choix de la base des images utilisées pour la réalisation de ce travail et le fait qu'un risque d'incohérences entre les images, que ce soit de la base d'apprentissage ou celles de test, est toujours possible, les résultats obtenus durant ce projet sont concluants et satisfaisants.

Les travaux relatés dans le présent rapport ne doivent pas pour autant être considérés comme achevés mais ils ouvrent la voie vers diverses perspectives de recherche comme :

- La généralisation de cette approche pour toutes les transformées d'ondelettes existantes.
- L'implémentation de ce système de compression pour les images médicales.
- L'utilisation d'autres techniques de classification comme les SVM (Support Vector Machine) et les comparer avec les réseaux de neurones.