

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERY DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : Sciences et Technologies

Filière : Génie électrique

Spécialité : **Commande des systèmes**

Présenté par
Houacine Noura
Khelifa Nadia

Thème

Classification des textures par les réseaux de neurones convolutifs

Mémoire soutenu publiquement le ...27.../ 09...../2018..... devant le jury composé de :

M Prénom NOM

Grade, Lieu d'exercice, Président

M Prénom NOM

Grade, Lieu d'exercice, Encadreur

Mr Hamouche.Kamel

Grade, Lieu d'exercice, Co-Encadreur

M Prénom NOM

Grade, Lieu d'exercice, Examineur

M Prénom NOM

Grade, Lieu d'exercice, Examineur

Remerciements

Remerciements

Nous tenons à remercier le bon dieu qui nous a donné force, Volonté et courage afin d'élaborer ce mémoire. A remercier notre encadreur Mr Hamouche Kamal Pour le soutien moral et intellectuel, la gentillesse et la haute compréhension qu'elle nous a apportée tout au long de notre travail on lui exprime ici nos sincères et profondes Reconnaissances. On remercie toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce mémoire. Merci à tous Les enseignants de département d'AS qui ont contribué à notre formation, Nos remerciements s'adressent aussi à notre promotion de Master2, 2018 à Mr Touat 'chef de département'

Enfin, nous tenons à remercier très vivement les membres de jury qui ont accepté de jurer ce travail.

Merci à tous

Dédicaces

Dédicaces

Je dédie ce modeste travail à mes parents, mes sœurs Kahina, ouerdya, soraya, mes frères, et mes beaux-frères Kamel et ikhelef, à mes chers Lilia ouazzi et lila hamoum, Sara, Bouzid et Mebarek, Ammar et à ma très chère nièce Ania. Sans oublier Nadia Khelifa et à toute sa famille

M^{elle} N. Houacine

Je dédie ce modeste travail à mes parents, à ma chère sœur Kenza, à mon cher frère Mourad, et à ma grand-mère, ma tante, à ma chère copine Nora Ibriz, à Loukal Amira sans oublier Noura Houacine et à toute sa famille.

M^{elle} Khelifa Nadia

Listes des figures

- Figure 1 Exemple de textures grossières et non grossières**Erreur ! Signet non défini.
- Figure 2 Exemple de textures contrastées**Erreur ! Signet non défini.
- Figure 3 illustration de la complexité des textures**Erreur ! Signet non défini.
- Figure 4 illustration de la notion de force des textures**Erreur ! Signet non défini.
- Figure 5 Exemple de textures structurales**Erreur ! Signet non défini.
- Figure 6 Exemple de textures aléatoires**Erreur ! Signet non défini.
- Figure 7 Exemples de textures directionnelle**Erreur ! Signet non défini.
- Figure 8 Direction des matrices de cooccurrence**Erreur ! Signet non défini.
- Figure 9 Exemple de calcul d'une matrice de cooccurrence**Erreur ! Signet non défini.
- Figure 10 Calcul du nombre LBP et du contraste C**Erreur ! Signet non défini.
- Figure 11 Caractérisation de la texture couleur par les attributs de texture extrait de l'image de luminance et des attributs caractérisant la distribution des couleurs dans l'espace (C1, C2, C3)**Erreur ! Signet non défini.
- Figure 12 Caractérisation de la texture couleur par les attributs de texture couleur qui ne tiennent compte que les relations intra composantes dans l'espace (C1,C2,C3)**Erreur ! Signet non défini.
- Figure 13 Caractérisation de la texture couleur par les attributs de texture couleur qui tiennent compte à la fois les relations intra et inter composantes dans l'espace (C1,C2,C3)**Erreur ! Signet non défini.
- Figure 14 Exemple d'images de la base Brodatz**Erreur ! Signet non défini.
- Figure 15 Exemple d'images de la base Vis Tex**Erreur ! Signet non défini.
- Figure 16 Exemple d'images de la base BarkTex**Erreur ! Signet non défini.
- Figure 17 Exemple d'images de la base OuTex**Erreur ! Signet non défini.
- Figure 18 Exemples d'images de la base Kth-tips**Erreur ! Signet non défini.
- Figure 19 Schéma synoptique d'un réseau de neurones bilogique**Erreur ! Signet non défini.
- Figure 20 Représentation d'un neurone formel** Erreur ! Signet non défini.
- Figure 21 Différentes connexion inter-couches dans les réseaux de neurones**Erreur ! Signet non défini.
- Figure 22 connexion intra couches**Erreur ! Signet non défini.

Figure 23 Réseau de neurones perceptron multicouches	Erreur ! Signet non défini.
Figure 24 Réseau à connexions récurrentes.....	Erreur ! Signet non défini.
Figure 25 Réseau à connexions complètes.....	Erreur ! Signet non défini.
Figure 26 Apprentissage non supervisé.....	Erreur ! Signet non défini.
Figure 27 Apprentissage supervisé	Erreur ! Signet non défini.
Figure 28 Exemple sur un réseau de neurones convolutifs	Erreur ! Signet non défini.
Figure 29 Schéma de la fenêtre de filtre sur l'image.....	Erreur ! Signet non défini.
Figure 30 Produit de convolution	Erreur ! Signet non défini.
Figure 31 Illustration de la convolution d'une image couleur	Erreur ! Signet non défini.
Figure 32 Image multi-composante de taille 28*28*6.....	Erreur ! Signet non défini.
Figure 33 Illustration de la méthode maxpolling	Erreur ! Signet non défini.
Figure 34 Allure de la fonction ReLu	Erreur ! Signet non défini.
Figure 35 Architecture de Google Net.....	Erreur ! Signet non défini.
Figure 36 Structure de VGG	Erreur ! Signet non défini.
Figure 37 Architecture de LeNet	Erreur ! Signet non défini.
Figure 38 Architecture d'Alex Net.....	Erreur ! Signet non défini.
Figure 39 Architecture de ResNet.....	Erreur ! Signet non défini.
Figure 40 Architecture d'Overfeat	Erreur ! Signet non défini.
Figure 41 Architecture du CNN utilisé pour la classification des textures	Erreur ! Signet non défini.
Figure 42 Variation du taux de classification et de l'erreur.....	Erreur ! Signet non défini.
Figure 43 Matrice de confusion.....	Erreur ! Signet non défini.

SOMMAIRE

Introduction générale.....	7
1. CHAPITRE 01: ANALYSE DES TEXTURES.....	9
1.1 Introduction.....	10
1.2 Définition de texture.....	10
1.3 Modèles de la textures.....	13
1.3.1 Textures structurelles.....	13
1.3.2 Textures aléatoires (ou micro-textures).....	13
1.3.3 Textures directionnelles.....	14
1.4 Méthodes d`analyse de la texture.....	14
1.4.1 Méthodes structurelles.....	15
1.4.2 Méthodes basées sur un modèle.....	15
1.4.3 Méthodes spatio fréquentielles.....	15
1.4.4 Méthodes statistiques.....	16
1.5 Analyse des textures couleurs.....	21
1.6. Application de l`analyse de la texture.....	23
1.7 Classification des textures.....	24
1.7.1 Définition.....	24
1.7.2 Types de méthodes de classification.....	24
1.7.3 Méthodes de classification.....	25
1.8 Bases d`images de textures.....	29
1.8.1 Base Brodatz.....	29
1.8.2 Base Vis Tex.....	30
1.8.3 Base BarkTex.....	31
1.8.4 Baseoutex.....	32
1.8.5 Base KTH-TIPS.....	32
1.9 Conclusion.....	33
2. CHAPITRE 02: Les réseaux de neurones	34
2.1. Introduction.....	35
2.2. Historique.....	35
2.3 Neurone biologique et neurone formel.....	36

2.3.1 Neurone biologique.....	36
2.3.2 Neurone formel.....	37
2.4 Réseaux de neurones artificiels	40
2.4.1 Réseaux de neurones non bouclés.....	41
2.4.2 Réseaux de neurones bouclés	43
2.5. Apprentissage des réseaux de neurones.....	44
2.5.1. Apprentissage non supervisé (unsupervised Learning).....	44
2.5.2. Apprentissage supervisé (supervised Learning).....	45
2.5.3 .Apprentissage du perceptron multicouche par la règle de retro propagation.....	46
2.6. Propriétés des réseaux de neurones.....	47
2.7. Domaines d'application des réseaux de neurones.....	48
2.8. Conclusion.....	48
3. CHAPITRE 03: RESEAUX DE NEURONES CONVOLUTIFS.....	49
3.1 Introduction.....	50
3.2 Historique.....	51
3.3 Réseaux de neurones convolutifs.....	52
3.3.1 Couches convolutives.....	52
3.2.2 Couche de mise en commun (pooling).....	55
3.2.3 Couche de correction Relu.....	56
3.2.4 Couches entièrement connectées.....	57
3.2.5 Couche de perte (LOSS).....	58
3.3 Paramètres des CNN.....	58
3.4 Exemples de CNN et transfert d'apprentissage.....	59
3.4.1 Google Net.....	59
3.4.2 VGG.....	60
3.4.3 LeNet.....	61
3.4.4 Alex Net	61
3.4.5 ResNet.....	62
3.4.6 Overfeat.....	62
3.5 Classification des textures par un CNN.....	63
3.5.1 Architecture du CNN proposé.....	63
3.5.2 Résultats obtenus.....	64

3.6 Conclusion.....	66
Conclusion générale.....	68
Bibliographie	70

Introduction générale

Introduction générale

Nous vivons dans un monde numérique, où les informations sont stockées, traitées, indexées et recherchées par des systèmes informatiques, ce qui rend leur récupération une tâche rapide et pas cher. Au cours des dernières années, des progrès considérables ont été réalisés dans le domaine de classification des textures. Ce progrès est dû aux nombreux travaux dans ce domaine et à la disponibilité des bases d'images internationales qui ont permis aux chercheurs de signaler de manière crédible l'exécution de leurs approches dans ce domaine, avec la possibilité de les comparer à d'autres approches qu'ils utilisent les mêmes bases.

Dans la fin des années 80 Yan le Cun a développé un type de réseau particulier qui s'appelle le réseau de neurone convolutionnel, ces réseaux sont une forme particulière de réseau neuronal multicouche dont l'architecture des connexions est inspirée de celle du cortex visuel des mammifères. Par exemple, chaque élément n'est connecté qu'à un petit nombre d'éléments voisins dans la couche précédente. En 1995, Yan le cun et deux autres ingénieurs ont développé un système automatique de lecture de chèques qui a été déployé largement dans le monde. À la fin des années 90, ce système lisait entre 10 et 20 % de tous les chèques émis aux États-Unis. Mais ces méthodes étaient plutôt difficiles à mettre en œuvre avec les ordinateurs de l'époque, et malgré ce succès, les réseaux convolutionnels et les réseaux neuronaux plus généralement ont été délaissés par la communauté de la recherche entre 1997 et 2012. En 2011 et 2012 trois événements ont soudainement changé la situation. Tout d'abord, les GPU (Graphical Processing Unit) capables de plus de mille milliards d'opérations par seconde sont devenus disponibles pour un prix moins cher. Ces puissants processeurs spécialisés, initialement conçus pour le rendu graphique des jeux vidéo, se sont avérés être très performants pour les calculs des réseaux neuronaux. Deuxièmement, des expériences menées simultanément à Microsoft, Google avec l'aide du laboratoire de Geoff Hinton ont montré que les réseaux profonds pouvaient diminuer de moitié les taux d'erreurs des systèmes de reconnaissance vocale. Troisièmement plusieurs records en reconnaissance d'image ont été battus par des réseaux de neurones convolutionnels. Dans notre projet on va utiliser les réseaux de neurones convolutionnels pour classer des textures, on va créer différents modèles avec différentes architectures et par la suite on va appliquer ces modèles sur les bases d'images de textures.

Pour ce faire, nous avons structuré notre mémoire en trois chapitres :

- Dans le premier chapitre on va présenter les notions de base de la classification des textures, les différents types des textures et les caractéristiques, ainsi que l'utilisation des réseaux de neurones dans la classification des images.
- Le deuxième chapitre est consacré à la description des réseaux de neurones ainsi que leurs l'intérêt dans le domaine de la classification des textures.
- Dans la troisième chapitre, nous allons décrire les réseaux de neurones convolutifs et leurs applications et on va montrer la partie expérimentale de notre travail et on discute les différents résultats obtenus et à la fin on termine par une conclusion générale

Chapitre 01: Analyse des textures

1.1 Introduction

L'analyse de la texture joue un rôle très important en traitement d'images car souvent les scènes que décrivent les images renferment des zones ou des objets dont la surface laisse apparaître un aspect structurel ou textural. Selon les applications envisagées, l'analyse de la texture est utilisée soit pour la classification des textures (reconnaissance des formes) soit pour segmenter une image.

L'analyse de la texture a pour but d'écrire ou caractériser une image ou une partie de l'image. Par un ensemble d'attributs. Selon les cas (classification ou segmentation), l'analyse peut être globale ou locale. Toute une gamme de méthodes adaptées à la description de la texture en terme qualitatif ou quantitatif a été proposée. Cette description est formalisée par un certain nombre de propriétés caractéristiques ou attributs qu'on peut exprimer sous forme paramétrique.

Nous présentons dans ce chapitre, quelques notions de base de la texture.

1.2 Définition de texture

Il n'existe pas de définition universellement admise, qui permette de caractériser la notion de texture au-delà du fait qu'elle restitue l'état de surface. En effet, toutes les définitions proposées sont, soit trop générales et imprécises, soit trop restrictives pour s'adapter à la diversité des cas rencontrés. Cependant nous retrouvons toujours un critère important, en commun. Il représente la notion d'arrangement spatial des pixels dans une image.

La texture est ainsi définie et étudiée de manières différentes, en fonction de la façon dont on la caractérise. On trouve dans la littérature des définitions telles que:

-Une texture est une région d'une image pour laquelle il existe une fenêtre de dimension réduite, telle qu'une observation au travers de celle-ci se traduise par une impression visuelle identique pour toutes les positions envisageables par translation à l'intérieur de la région considérée (Unser. 1984).

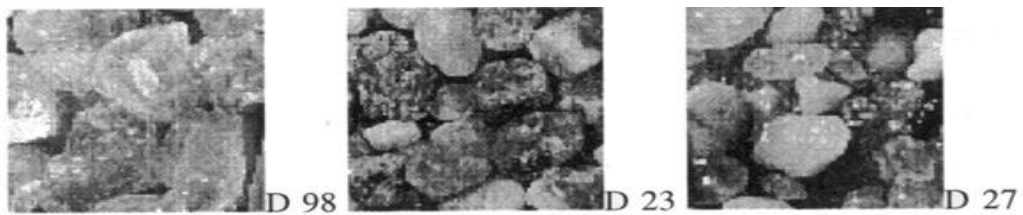
-La texture peut être décrite comme une structure hiérarchique à deux niveaux. Le premier concerne la description d'éléments de base ou primitives, à partir desquels est construite la texture. Le second niveau est relatif à la description de l'organisation spatiale de ces primitives, à l'agencement des motifs texturaux entre eux, qui peut satisfaire aussi bien à des lois stochastiques que déterministe (Haralick, R., 1979).

Elle manifeste aussi par une information visuelle qui permet de la décrire qualitativement à l'aide des adjectifs suivants: grossière, fine, lisse, tachetée, granuleuse,

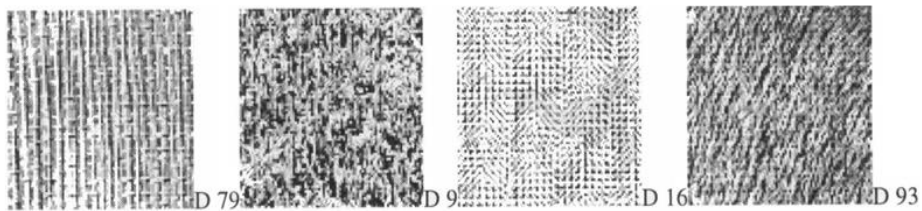
marbrée, régulière ou irrégulière. La notion de texture est utilisée pour traduire un aspect homogène de la surface d'un objet sur une image.

Les caractéristiques de base les plus utilisées sont le contraste, la complexité, la grossièreté, la forme, la direction et la force. Les figures ci-dessous illustrent quelques caractéristiques de la texture.

- **La grossièreté**: Une texture grossière possède des primitives larges: il existe alors peu de variations entre l'intensité d'un pixel et celle de ses pixels voisins (Fig.1.1)



Textures grossières



Textures non grossières

Fig.1.1 : Exemple de textures grossières et non grossières

- **Le contraste** : Une texture possède un contraste élevé si les différences d'intensité entre primitives sont importantes (Fig.1.2).

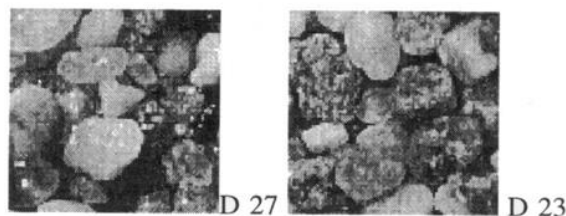


Fig.1.2 : Exemple de textures contrastées

- **La complexité :** Une texture complexe possède plusieurs types de primitives (fig.1.3). Dans ce cas le contenu d'informations présent dans la texture est important.

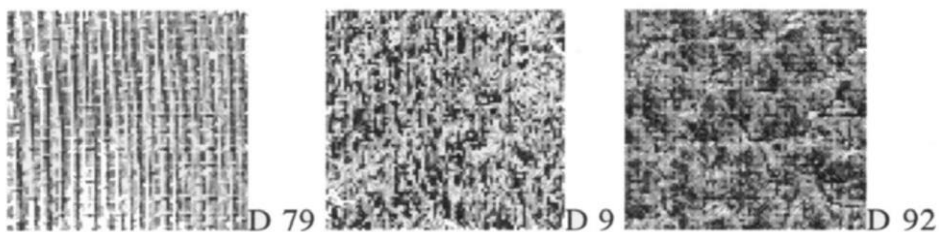


Fig.1.3 : Illustration de la complexité des textures

- **La force:** Plus la force est élevée plus les primitives sont facilement définissables et visibles.



Force élevée



Force faible

Fig.1.4 : Illustration de la notion de force des textures

1.3 Modèles de la texture

La complexité de la notion texture s'impose aussi dans la classification de l'ensemble des textures. En effet, ils existent une multitude de textures qu'on peut regrouper en trois principales familles :

- Textures structurelles.
- Textures aléatoires.
- Textures directionnelles.

1.3.1 Textures structurelles

Ce type de textures, encore appelées macro-textures, se distinguent par la répétition spatiale d'un motif de base appelé texton. La répétition spatiale de ces motifs obéit à des règles de directions et de placement. Ces textures peuvent même présenter une certaine périodicité ou cyclo stationnarité (processus aléatoire plaqué sur un processus périodique). Elles sont bien décrites par des approches fréquentielles ou structurelles. Certains exemples sont représentatifs de ce type de textures, comme la texture d'un mur de brique, de certains tissus ou d'un grillage (Fig.1.5).

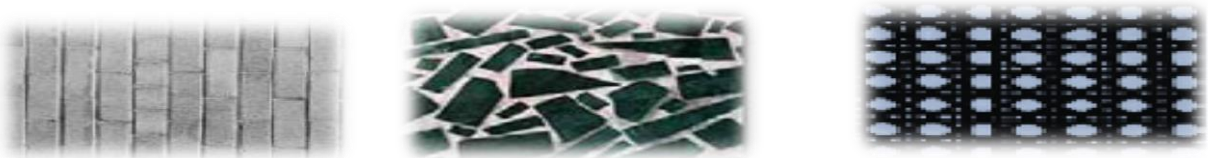


Fig. 1.5: Exemples de textures structurelles (macro-textures)

1.3.2 Textures aléatoires (ou micro-textures)

Dans ce type de textures, aucun motif particulier n'est localisable ou détectable. La distribution des intensités n'est alors l'objet d'aucune régularité apparente d'où le rôle spécifique que joue l'aléatoire dans ce type de texture. Elle sera décrite par des lois statistiques, une description spectrale en termes de densité de puissance, des propriétés de corrélation ou d'isotropie. Cependant, leur impression visuelle reste globalement homogène. Les différentes régions d'une image aérienne, les bois, les champs, représentent des exemples de textures microscopiques (Fig. 1.6).



Fig. 1.6: Exemples de textures aléatoires (micro-textures)

1.3.3 Textures directionnelles

Les textures directionnelles ne sont pas totalement aléatoires et ne présentent pas d'éléments structurants de base. Néanmoins, elles se caractérisent essentiellement par une certaine orientation. La figure (1.7) montre un exemple de texture directionnelle.

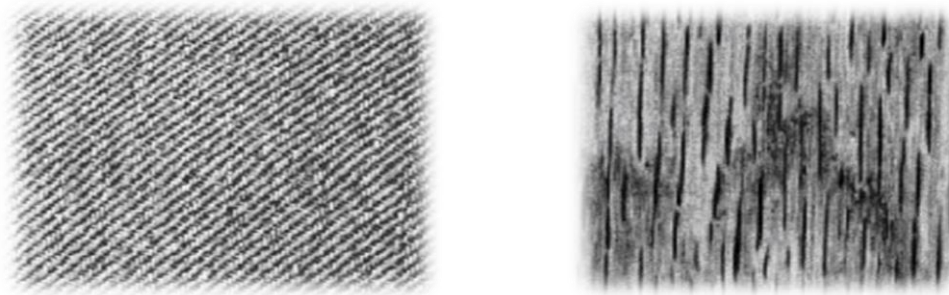


Fig. 1.7: Exemples de Textures directionnelle.

1.4 Méthodes d'analyse de la texture

L'analyse de la texture regroupe un ensemble de techniques permettant de quantifier ou de caractériser les textures caractérisant une image, une portion d'images ou un pixel de l'image. Ces méthodes ont pour but d'extraire un ensemble d'attributs ou paramètres pouvant décrire les caractéristiques de la texture. Ces attributs doivent être représentatifs, pertinents et discriminant de façon qu'on puisse discerner une texture parmi d'autres. Leurs extractions peuvent être classées essentiellement en quatre approches : l'approche structurale, l'approche basée sur un modèle, l'approche fréquentielle et l'approche statistique.

1.4.1 Méthodes structurelles

Ces méthodes tiennent compte de l'information structurelle et contextuelle d'une forme et sont particulièrement bien adaptées aux textures macroscopiques. Elles cherchent à détecter les primitives géométriques constituant les textures et à définir les règles de positionnement des différents motifs de la texture.

Ces méthodes sont sensibles à la régularité des motifs et ne permettent pas de caractériser des textures irrégulières comme celles présentes dans la majorité des images naturelles.

1.4.2 Méthodes basées sur un modèle

Le but de ces méthodes est d'obtenir un modèle générateur de la texture. Les paramètres de ce modèle permettent de caractériser ou de synthétiser une texture. On distingue en général trois types de modèles. Les modèles autorégressifs, les modèles fractals et les modèles Markoviens

1.4.3 Méthodes spatio fréquentielles

Ce type de méthodes cherche à décrire une texture à partir de ses caractéristiques fréquentielles. Certaines de ces méthodes sont basées sur le filtrage spatial (domaine spatial), d'autres extraient les attributs de texture directement à partir de la représentation de l'image dans le domaine de Fourier et d'autres exploitent à la fois le domaine spatial et le domaine fréquentiel lors du calcul des attributs.

La transformation de Fourier d'une image permet de mettre en évidence les régularités de la texture en examinant le domaine fréquentiel seulement. Le problème posé par cette opération, qui agit globalement sur l'image, est qu'elle ne tient pas compte de la localisation spatiale.

Afin d'y remédier à ce problème, des méthodes préservent à la fois les informations globales et de localisation et ont été proposées. Ces méthodes sont bien adaptées aux signaux quasi périodiques, comme les textures qui ont une énergie fréquentielle localisée. Ces méthodes s'articulent autour des transformations spatio fréquentielles afin de caractériser la texture à différentes échelles.

Parmi ces transformations, on peut citer la transformée en ondelettes, et la transformée de Gabor.

1.4.3.1 Transformation de Gabor

Une solution au problème de localisation spatial est d'utiliser une transformation alternative appelée transformation de Fourier à fenêtre glissante, dont le principe est d'appliquer la transformée de Fourier dans une fenêtre d'observation qui se déplace dans l'image. Le choix de la taille de la fenêtre et du pas de déplacement dépend des caractéristiques spatiales des textures à analyser.

Il existe différentes fenêtres d'observation : la fenêtre triangulaire, la fenêtre Gaussienne modulée. Lorsque cette dernière est appliquée, on parle de transformée de Gabor.

Un filtre de Gabor h , à deux dimensions, peut être représenté comme une gaussienne modulée par une onde plane sinusoïdale

$$h(x, y) = \exp \left[-\frac{1}{2} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) \right] \cos(2\pi(\mu_0 x + \nu_0 y) + \varphi) \quad (1.1)$$

Où μ_0, ν_0 et φ sont respectivement les fréquences et la phase de l'onde plane sinusoïdale.

σ_x et σ_y caractérisent l'étendue spatiale du filtre.

Un filtre de Gabor d'orientation arbitraire θ peut être obtenu en faisant subir une rotation au système d'axes (x, y) .

$$\begin{cases} x' = x \cos \theta + y \sin \theta \\ y' = -x \sin \theta + y \cos \theta \end{cases} \quad (1.2)$$

En faisant varier les différents paramètres $(\theta, \sigma_x, \sigma_y, \mu_0, \nu_0)$, plusieurs filtres de réponses impulsionnelles h_j sont obtenus.

Les attributs caractérisant la texture de l'image sont finalement obtenus en calculant l'énergie, l'entropie ou la variance de chaque image filtrée

1.4.4 Méthodes statistiques

Dans ces méthodes, la texture est considérée comme étant la réalisation d'un processus stochastique stationnaire. Ces méthodes exploitent directement les propriétés statistiques de l'image afin d'extraire des attributs de textures. Parmi ces méthodes on peut citer la méthode basée sur les matrices de cooccurrences et la méthode LBP (Local Binary Pattern)

1.4.4.1 Méthode basée sur les matrices de cooccurrences

La méthode basée sur les matrices de cooccurrence, proposée par Haralick et al, constitue une des méthodes référence pour la caractérisation la texture. Cette approche consiste à explorer les dépendances spatiales des pixels en construisant d'abord une matrice de cooccurrence basée sur l'orientation et la distance entre les pixels de l'image. De chacune de ces matrices, on peut extraire des caractéristiques de la texture, comme le contraste, l'entropie ou la différence inverse des moments.

La matrice de cooccurrence de niveaux de gris GLCM (Grey Level Cooccurrence Matrix) a été initialement introduite par Haralick. C'est une méthode de second ordre puisqu'elle permet d'analyser la relation entre les niveaux de gris des pixels pris deux à deux. Elle est considérée comme la méthode de référence dans le domaine de l'analyse de la texture. Une matrice de cooccurrence de niveaux de gris définie comme suit :

Soit $I(x, y)$ la fonction image définie sur un domaine D telle que $I(x, y) = 0, 1, 2, \dots, G - 1$, où G est le nombre de niveaux de gris dans l'image.

Soit R une relation spatiale liant deux pixels. La matrice de cooccurrence du domaine D relative à la relation R est définie par le terme général suivant :

$$t(i, j) = \text{card} \left\{ \begin{array}{l} ((x, y), (x', y')) \in D, \text{ vérifiant } R \\ I(x, y) = i, I(x', y') = j \end{array} \right. \quad (1.3)$$

Avec $i = 0, \dots, G - 1$ et $j = 0, \dots, G - 1$.

Chaque élément $t(i, j)$ représente le nombre de couples de points du domaine D liés par la relation R , l'un ayant le niveau de gris i et l'autre le niveau de gris j .

On peut également définir la matrice de cooccurrence normalisée en divisant chaque élément $t(i, j)$ par le nombre total de couples du domaine D vérifiant R .

$$T(i, j, d, \theta) = \frac{t(i, j, d, \theta)}{N} \quad (1.4)$$

Tel que

$$N = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} T(i, j, d, \theta) \tag{1.5}$$

$T(i, j, d, \theta)$ N'est plus un nombre mais une fréquence relative. C'est la probabilité de trouver un couple de points du domaine D , l'un ayant le niveau de gris i et l'autre de niveau j , vérifiant R . Cette relation spatiale R liant les deux points est définie par Haralick par la distance d entre ces deux points et leur orientation θ par rapport à l'horizontale.

Les matrices de cooccurrence usuelles sont définies par la distance d entre deux points et les quatre directions horizontale ($\theta = 0$) verticale ($\theta = \frac{\pi}{2}$), diagonale gauche ($\theta = \frac{\pi}{4}$) et diagonale droite

($\theta = \frac{3\pi}{4}$) (Fig. 1.8).

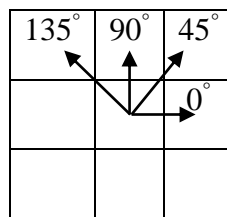


Fig. 1.8 : Directions des matrices de cooccurrence

La figure 1.9 illustre le principe de calcul d'une matrice de cooccurrence sur une région de 25 pixels quantifiée sur 4 niveaux de gris.

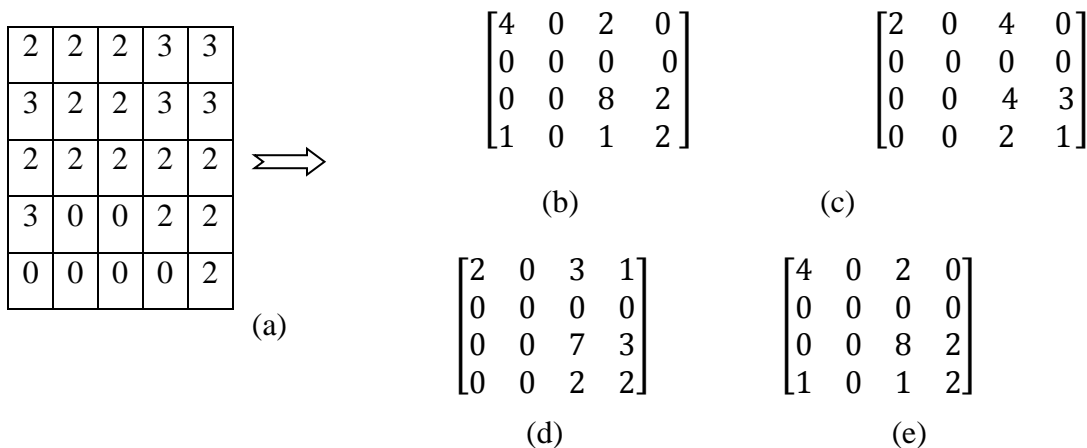


Fig. 1.9 : Exemple de calcul d'une matrice de cooccurrence.

- (a) Image originale, (b) Matricet $(0^\circ, 1)$, (c) Matricet $(45^\circ, 1)$, (d) Matricet $(90^\circ, 1)$, (e) Matrice t $(135^\circ, 1)$.

Les matrices obtenues ne sont pas symétriques, le calcul des attributs de texture est beaucoup plus performant dans une matrice symétrique.

$$t_s(0^\circ, 1) = \frac{1}{2}(t(0^\circ, 1) + t^T(0^\circ, 1)) \quad (1.6)$$

$$t_s(0^\circ, 1) = \begin{bmatrix} 4 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 8 & 1.5 \\ 0.5 & 0 & 1.5 & 2 \end{bmatrix} \quad (1.7)$$

La matrice normalisée est obtenue en divisons $t_s(0^\circ, 1)$ par $N=20$:

$$T_s(0^\circ, 1) = \begin{bmatrix} 0.2 & 0 & 0.05 & 0.025 \\ 0 & 0 & 0 & 0 \\ 0.05 & 0 & 0.4 & 0.075 \\ 0.025 & 0 & 0.075 & 0.1 \end{bmatrix} \quad (1.8)$$

La matrice de cooccurrence est riche en information mais reste difficilement

Exploitable dans son intégralité. Une solution consiste à réduire le nombre de niveaux de gris afin d'utiliser les éléments de la matrice de cooccurrence comme attributs de texture. L'autre démarche qui est la plus courante consiste à extraire plusieurs attributs afin de résumer toute l'information apportée par la matrice de cooccurrence. Les six attributs du tableau 1 sont ceux qui sont les plus utilisé

Contraste	$CON = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i-j)^2 T^2(i, j, d, \theta)$
Variance	$VAR = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i-j)^2 T(i, j, d, \theta)$
Second moment angulaire	$SMA = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} T^2(i, j, d, \theta)$
Entropie	$ENT = - \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} T(i, j, d, \theta) \log T(i, j, d, \theta)$
Corrélation	$COR = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{(i - \mu_x)(j - \mu_y) T(i, j, d, \theta)}{\sigma_x \sigma_y}$
Homogénéité locale ou moment angulaire inverse	$IDM = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{1}{1 + (i-j)^2} T(i, j, d, \theta)$

Tableau 1 : Attributs de texture extraits par les matrices de cooccurrence

Avec

$$\sigma_x^2 = \sum_{i=0}^{G-1} (i - \mu_x)^2 T_x(i) \quad \text{et} \quad \sigma_y^2 = \sum_{j=0}^{G-1} (j - \mu_y)^2 T_y(j) \quad (1.9)$$

tels que

$$\begin{aligned} T_x &= \sum_{i=0}^{G-1} T(i, j), \quad T_y = \sum_{j=0}^{G-1} T(i, j), \quad \mu_x \\ &= \sum_{i=0}^{G-1} i T_x(i) \quad \text{et} \quad \mu_y = \sum_{j=0}^{G-1} j T_y(j) \end{aligned} \quad (1.10)$$

Dans le premier indice, chaque terme de la matrice T est pondéré par sa distance à la diagonale. On obtient ainsi un indice correspondant à la notion usuelle du contraste, il est élevé quand les termes éloignés de la diagonale de la matrice sont élevés, c'est-à-dire quand on passe souvent d'un pixel très clair à un pixel très foncé ou inversement. Le SMA est d'autant plus élevé que l'on retrouve souvent le même couple de pixels, ce qui est le cas quand le niveau de gris est uniforme (texture lisse) ou quand il y a périodicité dans le cas de translation. L'entropie est faible si on a souvent le même nombre de pixels, forte si chaque couple est peu représenté. Elle fournit un indicateur de désordre que peut présenter une texture. La corrélation quantifie la dépendance directionnelle des niveaux de gris, elle atteint ses plus grandes valeurs lorsque θ est proche de l'orientation des lignes de la texture. IDM reflète l'existence de plages uniformes de texture et affecte un poids de plus en plus faible au fur et à mesure qu'on s'éloigne de la diagonale principale. Plus la valeur de l'IDM est élevée plus la texture est grossière. La variance mesure la somme des fréquences pour différents écarts de niveaux de gris.

1.4.4.2 Méthode LBP

Cette méthode initialement proposée par Ojala et al. Considère un nombre de pixels voisins p et une distance d reliant un pixel central avec ses pixels voisins. Pour chaque pixel, on définit une matrice de taille $(p \times p)$. En codant ces matrices, on obtient un nombre appelé Local Binary Patterns ou (LBP) qui peut prendre 2^p valeurs. Il est défini par l'équation suivante :

$$LBP = \sum_{i=0}^{p-1} u(g_i - g_c) 2^i \quad (1.11)$$

Où g_c est le niveau de gris du pixel central, g_i ($i = 0, 1, \dots, p - 1$) est le niveau de gris du pixel voisin et $u(x)$ est la fonction définie comme suit :

$$u(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases} \quad (1.12)$$

Ce nombre est généralement combiné avec une mesure de contraste C qui évalue la différence des niveaux de gris des pixels voisins ayant la valeur binaire 1 et ceux ayant la valeur 0. La distribution bidimensionnelle reliant les valeurs de LBP et celles de C permet de caractériser la texture. La figure (1.10) illustre le principe de calcul du nombre LBP et du

contraste C d'un pixel ayant $p=8$ pixels voisins.

6	5	2
7	6	1
9	3	7

(a)

1	0	0
1		0
1	0	1

(b)

1	2	4
8		16
32	64	128

(c)

1	0	0
8		0
32	0	128

(d)

Fig. 1.10 : Calcul du nombre LBP et du contraste C.

Dans cet exemple, nous avons : $LBP = 1 + 8 + 32 + 128 = 169$ et

$$C = (6 + 7 + 9 + 7)4 - (5 + 2 + 1 + 3)4 = 4.5.$$

1.5 Analyse des textures couleurs

Les définitions ainsi que les méthodes d'analyse de la texture évoquées précédemment ont été initialement proposées pour des images en niveaux de gris. Cependant, ces définitions restent valables pour des images couleurs.

Une image numérique couleur I est une matrice de pixels P dont la couleur C(P) est définie par trois composantes rouge, verte et bleue notées respectivement $C_R(P)$, $C_G(P)$ et $C_B(P)$. La couleur peut non seulement être représentée en utilisant ces trois composantes, mais également en utilisant d'autres systèmes de représentation de la couleur comme les espaces (x, y, z) , (L, u, v) , (L^*, u^*, v^*) , (L^*, a^*, b^*) et (L, H, S) , d'une manière générale, un pixel peut être représenté par trois composantes notées c_1, c_2, c_3 .

A partir de l'image couleur I codée dans l'espace (c_1, c_2, c_3) , on peut extraire trois images composantes notées I^{ck} ou chaque pixel est caractérisé par composante couleur $C_k(k=1, 2, 3)$

Notons qu'une image de luminance, notée I^L peut être définie à partir de l'image couleur.

A titre d'exemple : $I^L = 0,30I^{ck} + 0,59 I^{cv} + 0,11 I^{cb}$

L'analyse des textures couleur peut se faire selon 3 approches :

1- L'analyse de la texture est effectuée sur l'image de luminance I^L . Des caractéristiques couleurs sont ensuite associées aux attributs de texture (Fig.1.11)

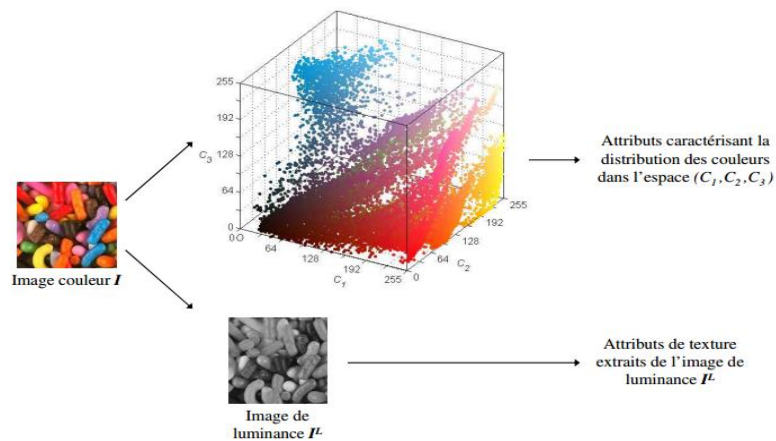


Fig.1.11 Caractérisation de la texture couleur par les attributs de texture extraits de l'image de luminance et des attributs caractérisant la distribution des couleurs dans l'espace (C_1, C_2, C_3)

2- Des attributs de texture sont extraits sur chaque image composante I^{Ck} (Fig.1.12).

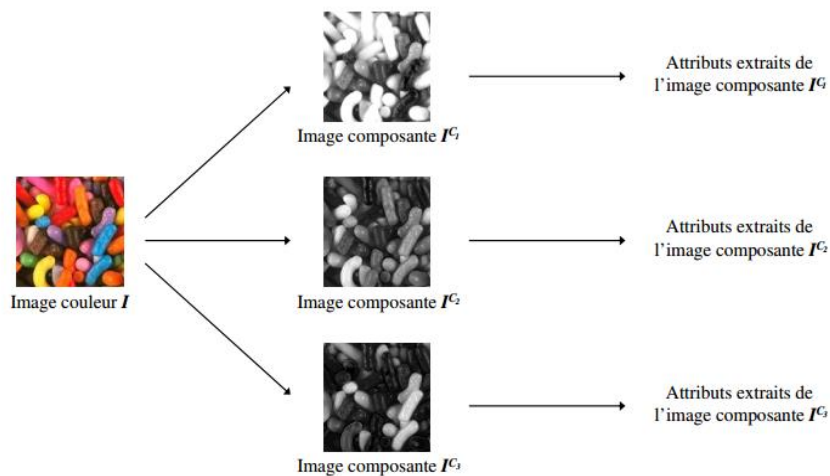


Fig.1.12 Caractérisation de la texture couleur par les attributs de texture couleur qui ne tiennent compte que les relations intra composantes dans l'espace (C_1, C_2, C_3)

3- L'analyse de la texture tient compte à la fois des relations Intra et inter composantes (Fig.1.13)

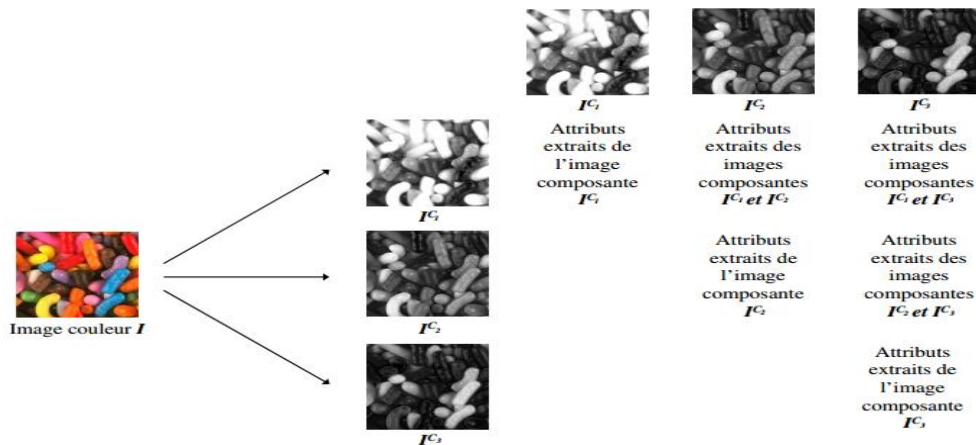


Fig.1.13 Caractérisation de la texture couleur par les attributs de texture couleur qui
Tiennent compte à la fois les relations intra et inter composantes
Dans l'espace (C_1, C_2, C_3)

1.6 Application de l'analyse de la texture

L'analyse de texture d'une image donnée, peut être essentielle pour plus d'une raison. Nous pouvons avoir besoin de trouver différentes régions dans une image, qui contient textures distinctives. Dans ce cas on parle de segmentation de textures qui est une procédure très utilisée pour analyser des images aériennes et satellitaire. La segmentation peut aussi être utilisée dans d'autres applications comme la séparation du texte à partir d'une partie restante d'une image. On peut aussi utiliser l'analyse de texture pour déterminer si une texture particulière, connue à priori, est présente ou non dans une image donnée. Ceci est connu sous le nom de la classification de texture. Une telle classification a des applications où il est souvent besoin d'égaliser ou de comparer deux textures différentes et d'identifier le degré de similarité, comme dans la biométrie où l'identification de texture à partir des images de l'iris, des visages, des empreintes, est impliquée un autre usage de la classification de texture est dans l'analyse d'images médicales, où le critère de classification aura pour objectif de différencier entre la texture du tissu normal et la structure du tissu anormal.

Une autre application peut être l'extraction ou la construction de surface 3D à partir des variations dans les propriétés texturales d'une image. Une telle application permet de recréer la surface 3D, dont la projection sur une image 2D est la texture donnée. Comme la texture ne peut être complètement décrite en une définition particulière, nous ne pouvons pas non plus trouver juste une seule manière qui nous permettra de caractériser complètement une texture donnée et d'extraire ses traits caractéristiques avec succès pour toutes les différentes

applications. En conséquence, il y a différentes approches pour analyser la texture, selon le genre d'application et le type de texture qui doit être analysée.

1.7 Classification des textures

1.7.1 Définition

La classification est une action de ranger par classes par catégories, des éléments présentant des critères en commun. Classification proprement dite est une opération de l'esprit, elle est basée sur la comparaison des caractères.

1.7.2 Types de méthodes de classification

De nombreuses méthodes classiques ont été consacrés, elles peuvent être séparées en deux grandes catégories : les méthodes de classification supervisée et les méthodes de classification non supervisée.

1.7.2.1 Méthodes supervisées

L'objectif de la classification supervisée est principalement de définir des règles permettant de classer des objets dans des classes à partir de variables qualitatives ou quantitatives caractérisant ces objets. On dispose au départ d'un échantillon dit d'apprentissage dont le classement est connu. Cet échantillon est utilisé pour l'apprentissage des règles de classement. Il est nécessaire d'étudier la fiabilité de ces règles pour les comparer et les appliquer, évaluer les cas de sous apprentissage ou de sur apprentissage (complexité du modèle). On utilise souvent un deuxième échantillon indépendant, dit de validation ou de test.

1.7.2.2 Méthodes non supervisées

Procède de la façon contraire. C'est à dire ne nécessitent aucun apprentissage et aucune tâche préalable d'étiquetage manuel. Elle consiste à représenter un nuage des points d'un espace quelconque en un ensemble de groupes appelé Cluster. Il lié généralement au domaine de l'analyse des données comme l'ACP. Un «Cluster» est une collection d'objets qui sont «similaires» entre eux et qui sont «dissemblables » par rapport aux objets appartenant à d'autres groupes.

1.7.3 Méthodes de classification

. Il existe une multitude de méthodes de classification supervisée. Les plus classiques sont la classification optimale de Bayes et la règle des K plus proches voisins. Et la méthode SVM, et les réseaux de neurones.

1.7.3.1 Classification par la méthode bayésienne

Cette méthode de nature statistique, suppose que l'échantillon à analyser est composé d N observations tirées, indépendamment, de K classes C_k où $k = 1, 2, \dots, K$ selon les lois de probabilité conditionnelles $P(X/C_k)$ connues. Par application du théorème de Bayes, il est possible de déterminer la probabilité pour qu'une observation X_q appartienne à la classe C_k . Cette probabilité, appelée aussi probabilité a posteriori de la classe C_k , est donnée par la relation :

$$P(C_k/X_q) = \frac{P\left(\frac{X_q}{C_k}\right) P(C_k)}{P(X_q)} \quad (1.13)$$

Où $P(X_q)$ désigne la valeur de la fonction de densité sous-jacente à la distribution des observations provenant de l'échantillon au point X_q , et $P(C_k)$ désigne la probabilité a priori de la classe C_k de telle sorte que :

$$\sum_{k=1}^K P(C_k) = 1 \quad (1.14)$$

$P(X_q/C_k)$ Désigne la probabilité conditionnelle de la classe C_k , c'est-à-dire, la fonction de densité de probabilité sous-jacente à la distribution des observations provenant de la classe C_k . Dans ces conditions, le risque d'erreur lié à l'affectation de l'observation X_q à la classe C_k est donné par :

$$R(C_k/X_q) = 1 - P(C_k/X_q) \quad (1.15)$$

Pour obtenir un classement optimal, il faut affecter toute observation X_q à la classe C_k qui minimise ce risque, donc qui maximise la probabilité a posteriori. Le problème de classification d'une observation X_q peut être résolu en adoptant la règle de décision suivante :

Décider C_s si : $P(C_s/X_q) > P(C_k/X_q), \forall k = 1, 2, \dots, K, (k \neq s)$

L'application de la règle de Bayes revient à trouver des fonctions de décision $d_k(X_q)$, $k = 1, 2, \dots, K$, telles que l'observation X_q soit affectée à la classe C_s si et seulement si:

$$d_s(X_q) > d_k(X_q) \quad \forall k = 1, 2, \dots, K, \quad s \neq k$$

$$\text{Où } d_k(X_q) = P(X_q/C_k)P(C_k) \quad (1.16)$$

Dans la plus part des cas, la probabilité conditionnelle de la classe C_k est de type gaussienne telle que :

$$P(X_q/C_k) = \frac{\exp(-\frac{1}{2}(X_q - \mu_k)^T \Sigma_k^{-1}(X_q - \mu_k))}{(2\pi)^{\frac{N}{2}} |\Sigma_k|^{\frac{1}{2}}} \quad (1.17)$$

Dans cette expression, μ_k est le vecteur moyenne de la classe C_k et Σ_k la matrice de covariance de la classe C_k .

Ainsi, les fonctions de décision d_k peuvent être écrites sous la forme :

$$d_k(X_q) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (X_q - \mu_k)^T \Sigma_k^{-1} (X_q - \mu_k) + \log [P(C_k)] \quad (1.18)$$

Les probabilités $P(C_k)$ peuvent être estimées par :

$$P(C_k) = \frac{N_{C_k}}{N} \quad (1.19)$$

Où N_{C_k} est le nombre d'observations appartenant à la classe C_k et N le nombre d'observations totales. μ_k Et Σ_k sont estimés par la moyenne et la variance empirique à partir des prototypes de la classe C_k .

1.7.3.2. Méthode des k plus proches voisins

Contrairement à la méthode précédente où la probabilité conditionnelle de la classe C_k est choisie à priori (généralement gaussienne), dans la méthode des k plus proches voisins (kppv), cette probabilité est supposée inconnue, elle est alors estimée de la manière suivante :

Un volume $V(X_q)$ centré sur chaque point X_q des échantillons est déterminé de façon à ce qu'il englobe k observations de l'ensemble d'apprentissage. On dénombre alors le nombre k_i de voisins de X_q appartenant à chaque classe C_k . La densité de probabilité de la classe C_k contenant N_{C_k} points prototypes est alors estimée comme suit :

$$\hat{P}(X_q/C_k) = \frac{k_i}{N_{C_k} V(X_q)} \quad (1.20)$$

En remplaçant $P(C_k)$ par $\frac{N_{C_k}}{N}$ et $P(X_q/C_k)$ par $\hat{P}(X_q/C_k)$ dans la règle de Bayes, on aboutit à la règle suivante :

$$X_q \in C_k \quad \text{si } P(C_k/X_q) = \max_{i=1, \dots, K} (k_i) \quad k = \sum_{i=1}^K k_i$$

Cette règle peut être énoncée comme suit : si les k plus proches voisins d'un point X_q appartiennent en majorité à la classe C_k , alors X_q appartiendra à la classe C_k .

1.7.3.3 Méthode SVM

La méthode SVM ou Machines à Vecteurs Supports (Support Vector Machine) a été proposée en 1995 par Vapnik (Cortes, Vapnik, 1995). Depuis, elle est très activement utilisée dans beaucoup d'applications. Son but est de rechercher une surface capable de séparer au mieux les prototypes de deux classes issus de la base d'apprentissage.

Considérons le cas où l'ensemble d'apprentissage $(X_a, Y_a) = (x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$

Sont linéairement séparables en deux classes C_1 et C_2 telles que $x_i \in R^d$ représente la i ème observation caractérisée par un vecteur de d attributs et $y_i \in \{+1, -1\}$, $y_i = 1$ si $x_i \in C_1$ et $y_i = -1$ si $x_i \in C_2$.

La méthode SVM consiste à chercher un hyperplan qui sépare les données des deux classes. Les points situés sur cet hyperplan satisfait l'équation $w^T x + b = 0$ où $w = [w_1, w_2, \dots, w_d]$ est le vecteur perpendiculaire à l'hyperplan et b un scalaire appelé bias. Dans le cas où les observations sont séparables, la méthode SVM consiste à maximiser la marge représentant la distance entre un hyperplan H_1 défini par l'équation $W^T x + b = -1$ et un hyperplan H_2 défini par l'équation $w^T x + b = 1$. Ceci constitue un problème d'optimisation avec contrainte:

$$\begin{cases} \text{minimiser } f(x) = \frac{1}{2} w^T w \\ \text{sujet à } g_i(x) = y_i(w^T x_i + b) - 1 \geq 0 \quad i = 1, \dots, N \end{cases} \quad (1.21)$$

En introduisant les multiplicateurs $\lambda_i \geq 0$ de Lagrange, ce problème devient :

$$\begin{cases} \max_{\lambda} -\frac{1}{2} \sum_{i=1}^p \sum_{j=1}^p \lambda_i \lambda_j y_i y_j x_i x_j + \sum_{i=1}^p \lambda_i & \text{avec les constantes } \lambda_i \geq 0 \quad i = 1, \dots, N \\ \sum_{i=1}^N \lambda_i y_i = 0 \end{cases} \quad (1.22)$$

Dont la résolution fournit les valeurs $\bar{\lambda}_i$ et par conséquent l'hyperplan optimal défini par la paire (\bar{w}, \bar{b}) telles que :

$$\bar{w} = \sum_{i=1}^p \bar{\lambda}_i x_i y_i \quad \text{et} \quad \bar{b} = y_j - \sum_{i=1}^p \bar{\lambda}_i x_i y_i x_j$$

Une nouvelle observation x_k est alors affectée une classe selon la règle suivante :

$$\text{Class}(x_k) = \text{sign}\left(\sum_{i=1}^M \bar{\lambda}_i x_i y_i x_k + \bar{b}\right)$$

Où M est le nombre de vecteurs support correspondant au nombre d'observations dont les valeurs $\bar{\lambda}_i \neq 0$.

Lorsque les classes sont non linéairement séparables, les observations x_i subissent la transformation non linéaire $\phi(x_i)$ et le problème SVM est posé comme suit:

$$\left\{ \begin{array}{l} \max_{\lambda} -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j K(x_i, x_j) + \sum_{i=1}^p \lambda_i \\ \sum_{i=1}^N \lambda_i y_i = 0 \end{array} \right. \quad \text{avec les constantes } 0 \leq \lambda_i \leq C \quad i = 1, \dots, N \quad (1.23)$$

Où C est un paramètre de pénalisation, $K(x_i, x_j) = \phi(x_i, x_j)$. $\phi(x_i, x_j)$ Est une fonction noyau définie par exemple par:

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (1.24)$$

L'affectation d'une nouvelle observation est donnée par :

$$\text{Class}(x_k) = \text{sign}\left(\sum_{i=1}^M \bar{\lambda}_i y_i K(x_i, x_k) + \bar{b}\right) \quad (1.25)$$

1.7.3.4 Réseaux de neurones

Cette méthode repose sur la notion de neurone formel, ce dernier est un modèle caractérisé par des signaux d'entrées, une fonction d'activation 'f.f ($\alpha_0 + \sum_i \alpha_i x_i$) Cette fonction peut être linéaire, à seuil, et plus souvent sigmoïde, le calcul des paramètres se fait par apprentissage.

1.8 Bases d'images de textures

Il existe plusieurs bases de texture. Ces bases sont très utilisées dans le cadre de l'analyse ou de la classification des textures. Elles permettent d'évaluer et de comparer la pertinence des attributs de texture à distinguer les différentes textures. Parmi ces bases nous pouvons citer :

1.8.1 Base Brodatz

Cette base est constituée de 112 images qui ont des intensités d'arrière-plan différentes, la figure ci-dessous donne un exemple de six textures Brodatz différentes organisées en deux ensembles: D109, D61 et D10 (rangée du haut); et D64, D95 et D75 (rangée du bas). Par

exemple, D109 a un arrière-plan noir tandis que D61 et D10 ont respectivement un arrière-plan gris et blanc. (fig. 1.14)

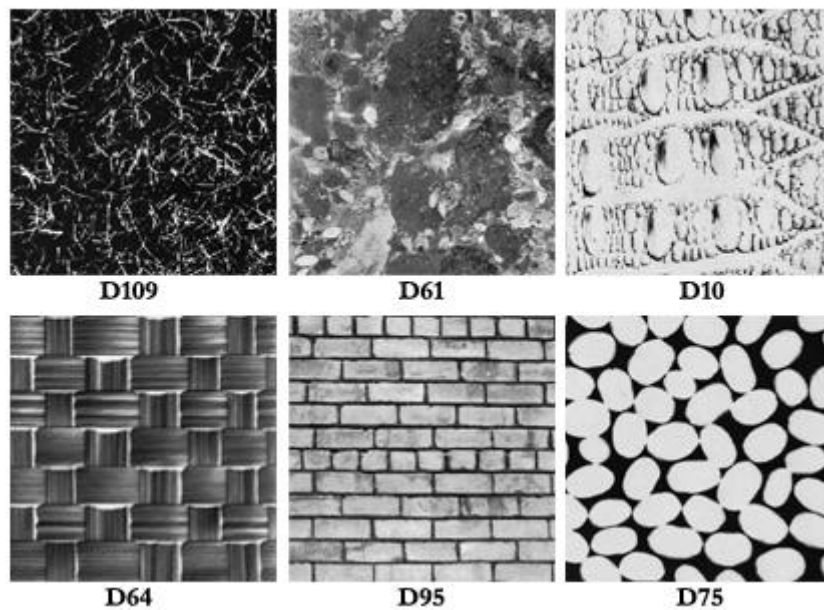


Fig. 1.14 : Exemple d'images de la base brodatz

1.8.2 Base Vis Tex

La base Vis Tex est une collection d'images de textures couleur extraites de scènes naturelles. Cette base de référence est constituée de 168 images acquises dans les conditions non contrôlés et réparties en 19 classes de textures couleur. Ces images sont disponibles en différentes tailles.

La figure (1.15) représente cinq textures couleur de taille 128* 128 pixels issus de la base VisTex, la classe Bark, représentant des écorces d'arbres, la classe Fabric, ou se trouvent divers textiles, la classe food, ou sont présents différents types d'aliments, la classe Métal, représentant la texture de différents métaux, et enfin la classe Water, présentant diverses surfaces d'eau de mer

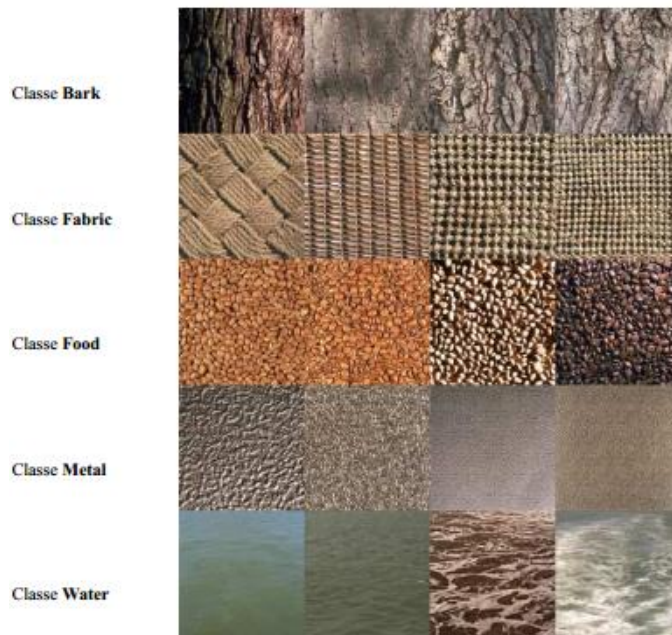


Fig.1.15 : Exemples d'images de la base Vis Tex

1.8.3 Base BarkTex

La base BarkTex comporte 6 classes d'images de texture couleur orientées verticalement correspondant à des écorces d'arbres (Fig.1.16), avec 68 images pour chacune d'entre elles, ce qui représente 408 images au total, disponibles en deux tailles : 128*192 et 256*384 pixels correspondant à deux résolutions spatiales différentes.

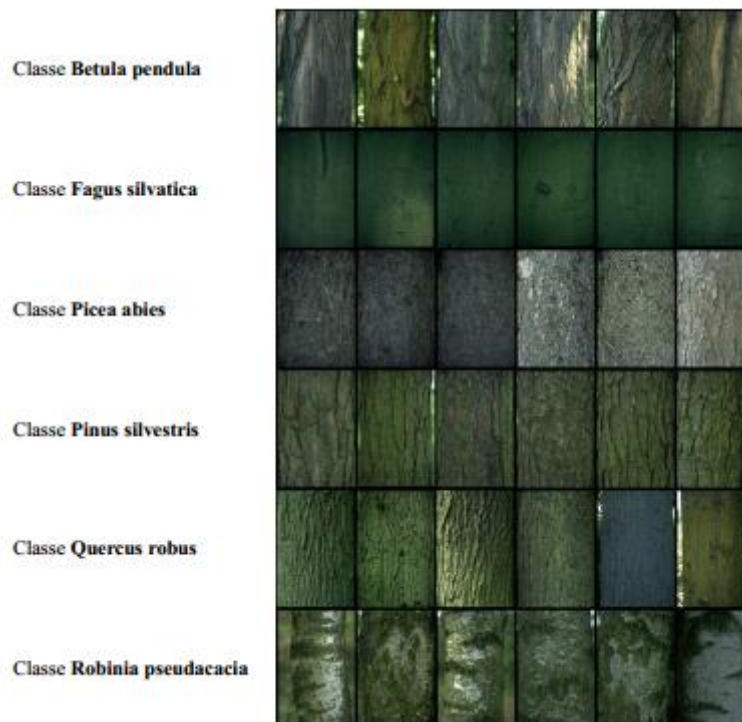


Fig.1.16 Exemple d'images de la base BarkTex

1.8.4 Base outex

La base d'images OuTex a été créée dans le but de fournir un large panel de textures pour les applications de traitement d'images. Tout comme la base VisTex, la base OuTex contient un grand nombre d'objets différents, à savoir 29 classes d'images de texture couleur. Le nombre d'images par classe varie entre 1 et 47 selon la classe considérée (Fig. 1.17).

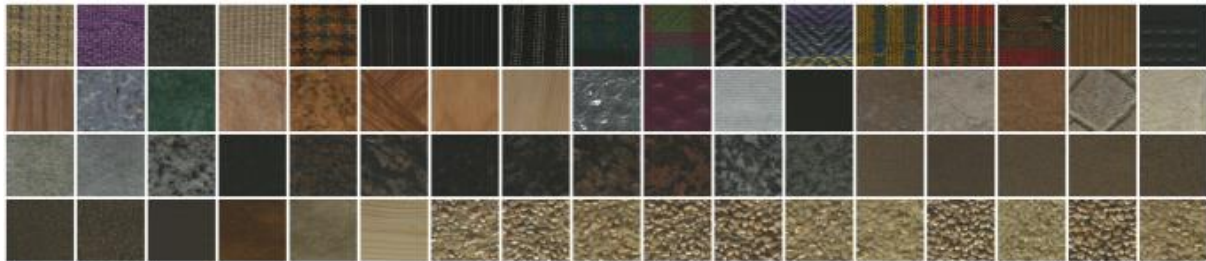


Fig.1.17 Exemple d'images de la base ouTex

1.8.5 Base KTH-TIPS

Les KTH-TIPS (Textures moins de variation d'illumination, Pose et Scale) base de données d'image a été créé par Mario Fritz pour étendre la base de données curette dans deux directions, en fournissant des variations de l'échelle. Ainsi que la pose et l'illumination, et en imageant d'autres échantillons d'un sous-ensemble de ses matériaux dans différents paramètres. (Fig. 1.18)



Fig.1.18: Exemples d'images de la base Kth-tips

1.9 Conclusion

Nous avons présenté dans ce chapitre quelques généralités sur l'analyse de la texture. Il que la texture peut être définie et étudiée de plusieurs manières selon en fonction de la manière utilisée pour la caractériser. La caractérisation de la texture est une procédure très importante en analyse de la texture. Elle consiste à extraire un ensemble d'attributs à partir soit directement de l'image de la texture soit après transformation.

Ces attributs sont en général utilisés pour classer des textures dans plusieurs domaines d'application.

Une manière de classer les textures est d'utiliser les réseaux de neurones dont une description générale sera donnée dans le chapitre suivant.

Chapitre 02: Les Réseaux de neurones

2.1. Introduction

Les neurones sont considérés comme le support physique de l'intelligence, ils troublent puisque comprendre et savoir utiliser l'intelligence permet de réaliser les buts inimaginables.

Depuis quelques années, on cherche à copier les réseaux de neurones pour faire des lois afin de traiter des images.

Les réseaux de neurones ont une capacité de mémoriser des fonctions simples, ils possèdent une architecture du cerveau humain, Une capacité d'apprentissage, et d'adaptation, Une résistance aux pannes, Une faible consommation énergétique.

2.2 Historique

L'étude sur les réseaux de neurones a débuté déjà à la fin du 19^{ème} siècle avec les travaux de Willam James en 1890, après l'élaboration d'une théorie sur les activités du cerveau, plus précisément au niveau du neurone et de la propagation de son excitation sur ces voisins

Les premiers travaux sur les réseaux de neurones artificiels sont débuté au début des années 1940 et ont été menés par Mc. Culloch et pitts, ils décrivent les propriétés du système nerveux à partir de neurones idéalisés : ce sont des neurones logiques (0 ou 1). Dix années plus tard, en constitué le premier modèle réel d'un réseau de neurones. En 1960, le premier perceptron est créé par Rosnblatt. Plus, durant les années 1970 il y eut une remise en cause de l'intérêt des réseaux car les ordinateurs apprenaient lentement, coûtaient très cher et leurs performances n'étaient pas si impressionnantes. La disponibilité croissante des mini et microordinateurs, vers la fin des années 1970, a permis aux réseaux de neurones de prendre un nouveau départ.

L'évolution des modèles du réseau et les algorithmes d'apprentissages ont donné aux réseaux de neurones, une grande richesse de comportement et une capacité à résoudre des problèmes complexes

2.3 Neurone biologique et neurone formel

Le cerveau humain est composé de milliards de cellules nerveuses appelées « neurones » connectées entre elles. Cette association de plusieurs neurones en un réseau de neurone permet aux cellules d'échanges et de transmettre des informations à travers une activité électrique ou plus précisément électrochimique

Un réseau de neurones artificiel s'inspire du fonctionnement biologique du cerveau humain. Dans un réseau de neurones l'information se propage, de la couche d'entrée à la couche de sortie en passant par aucune ou plusieurs couches intermédiaires dites couches cachées.

2.3.1 Neurone biologique

Un neurone biologique est une cellule composée d'un corps cellulaire et d'un noyau (fig.2.1). Le corps cellulaire se ramifie pour former les dendrites, également appelées chevelure dendritique ou arborisation dendritique. C'est par les dendrites que l'information est acheminée de l'extérieur de la cellule vers le stroma ou corps de neurone. L'information traitée par le neurone est ensuite acheminée le long de l'axone (unique) pour être transmise aux autres neurones. La transmission entre deux neurones n'est pas directe. En fait, il existe un espace intercellulaire de quelques angströms (10^{-9} m) entre l'axone du neurone efférent et les dendrites du neurone afférent. La jonction entre deux neurones est appelée synapse.

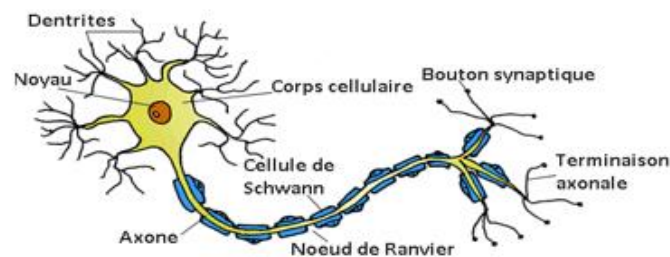


Fig. 2.1 Schéma synoptique d'un réseau de neurone biologique

Le traitement de l'information par un réseau de neurones est de nature électrochimique. Chaque neurone est asservi au maintien d'un gradient électrique (potentiel membranaire) d'environ -70mV entre l'intérieur et l'extérieur du neurone. Ainsi le neurone est dit polarisé (la tension négative est à l'intérieur).

Si les influences des autres neurones sur le potentiel membranaire suffisent pour dépolariser le neurone jusqu'à un certain seuil (environ -50 mV), alors le corps cellulaire génère une impulsion électrique de type tout ou rien appelée potentiel d'action. Le potentiel d'action ainsi généré se propage le long de l'axone et de ses ramifications.

Quand le potentiel d'action atteint une synapse reliée à un autre neurone, il déclenche une émission chimique qui modifie le potentiel membranaire du neurone récepteur soit de façon excitatrice (dépolarisation), soit de façon inhibitrice (hyperpolarisation). Ainsi chaque neurone fait en permanence l'addition des signaux excitateurs et inhibiteurs reçus par sa membrane et déclenche une impulsion nerveuse lorsque les conditions sont réalisées, c'est-à-dire le potentiel d'action est atteint.

A l'arrivée du potentiel d'action, un neurotransmetteur chimique, stocké dans les vésicules pré synaptiques, est relâché à travers la fente synaptique et il se diffuse par les fentes post synaptiques modifiant ainsi le potentiel de la membrane post synaptique.

Ce neurotransmetteur provoque habituellement une variation mineure locale et confinée dans l'espace et le temps de l'état de la membrane post synaptique.

Pour engendrer un potentiel d'action, il est donc nécessaire que de nombreux potentiels d'actions venant d'autres neurones arrivent presque simultanément de façon à pouvoir être additionnés et produire une importante dépolarisation.

2.3.2 Neurone formel

Un neurone formel est une représentation mathématique d'un neurone biologique. C'est une unité de traitement qui reçoit plusieurs données en entrées sous forme d'un vecteur $(x_1, x_2, \dots, x_n)^T$ et produit une sortie réelle. Cette sortie est une fonction des entrées et des poids des connexions appelés poids synaptiques (Fig.2.2).

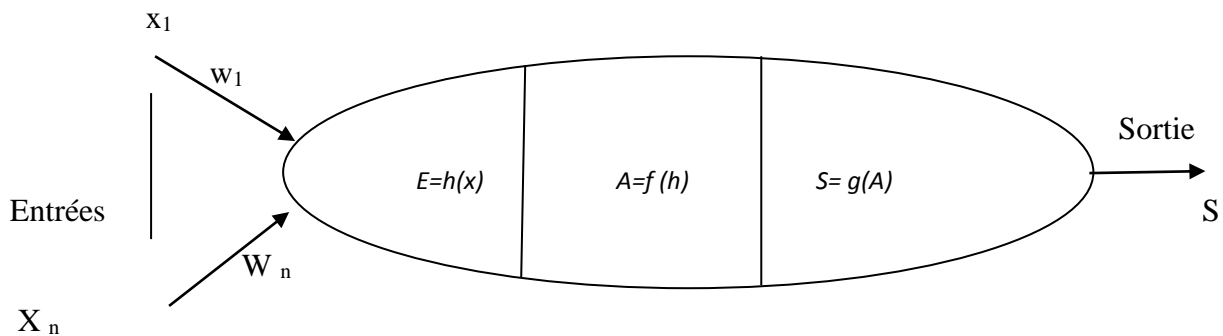


Fig.2.2 Représentation d'un neurone formel

La première modélisation d'un neurone a été présentée par Mac culloch et pitts en s'inspirant des neurones biologiques.

Un neurone formel calcule une somme pondérée des potentiels d'actions qui lui parviennent et il s'active suivant la valeur de cette sommation. Chacun de ces potentiels est une valeur numérique qui représente l'état du neurone qui l'a émis. Si cette somme dépasse un certain seuil, le neurone est activé et transmet une réponse (sous forme de potentiel d'action) dont la valeur est celle de son activation. Si le neurone n'est pas activé, il ne transmet rien. Les entrées et les sorties des neurones formels peuvent être de nature binaires $(-1, +1)$, $(0,1)$, ou réelles.

a. Fonction d'entrée totale

Notée h , définit le prétraitement effectué sur les entrée, elle peut être

- Booléenne ;

-Linéaire $h(x_1, \dots, x_n) = \sum_{i=1}^n w_i x_i$;

-Affine, $h(x_1, \dots, x_n) = \sum_{i=1}^n w_i x_i - b$ ou b est une constante appelée biais. Elle peut être considérée comme une pondération d'une entrée 0 avec $w_0=1$.

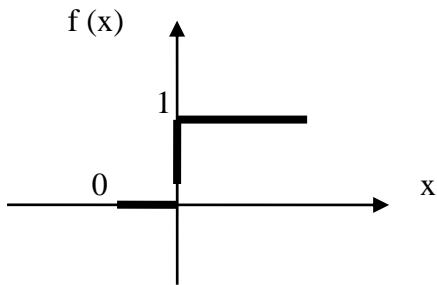
- Polynomiale de degré supérieur à deux.

On désigne l'entrée totale par $E=h(x_1, x_2, \dots, x_n)$. La combinaison linéaire est appelée « potentiel » et les coefficients W_i sont appelés « poids synaptiques » en référence à l'origine biologique des réseaux de neurones.

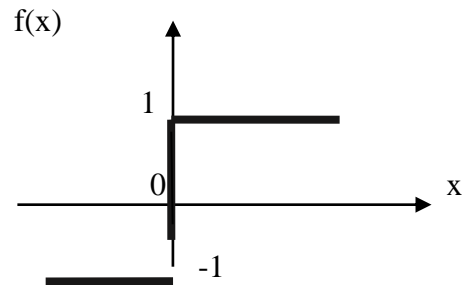
b. Fonction d'activation

Elle définit l'état interne du neurone en fonction de son entrée totale E. Notée f, elle peut prendre plusieurs formes :

-Fonction binaire à seuil



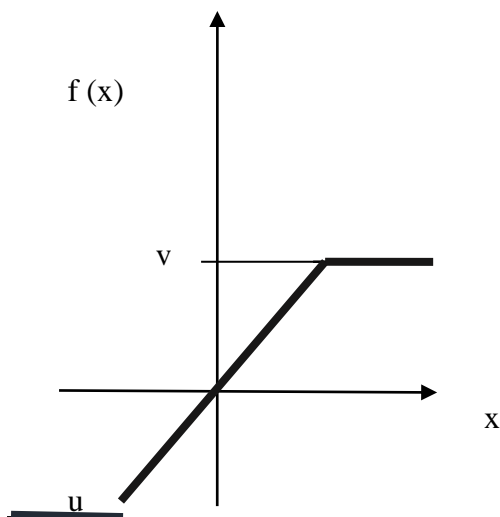
Fonction de heaviside



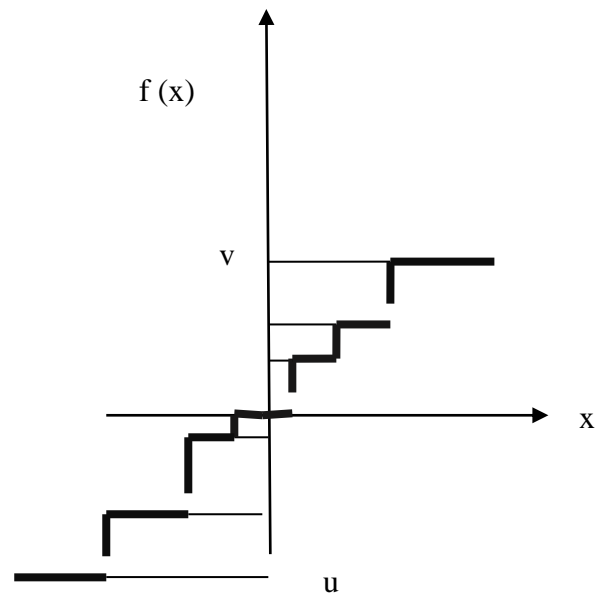
Fonction Signe

-Fonction linéaire à seuil ou multi-seuils.

$$f(x) = \begin{cases} x & \text{si } x \in [u, v] \\ u & \text{si } x \leq u \\ v & \text{si } x \geq v \end{cases}$$



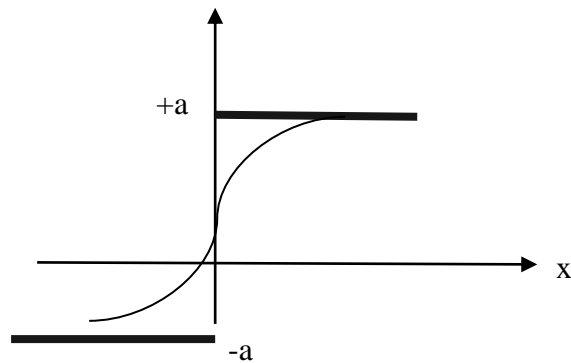
Fonction linéaire à seuil



Fonction linéaire à multi-seuil

-Fonction sigmoïdale : $f(x) = a \frac{(e^{kx} - 1)}{(e^{kx} + 1)}$

f(x)



Fonction sigmoïde

-Fonction tangente hyperbolique : $f(x) = \frac{1 - e^{-x}}{e^x + e^{-x}}$

-Fonction stochastique :

$$f(x) = \begin{cases} 1 & \text{avec probabilité } 1/(1 + \exp(-\frac{x}{t})) \\ 0 & \text{sinon} \end{cases}$$

t est un paramètre appelé température, quand t tend vers 0, f tend vers la fonction seuil.

Notons que toute autre fonction, généralement choisie croissante et impaire, peut être prise comme fonction d'activation.

c. La fonction de sortie

Elle calcule la sortie du neurone en fonction de son état d'activation. On note g cette fonction. Elle est souvent considérée comme une fonction identité. Par la suite, nous confondrons toujours activation et sortie du neurone $S = f(E) = A$.

2.4 Réseaux de neurones artificiels :

Un réseau de neurone artificiel est un ensemble de neurones formels connectés entre eux selon une architecture donnée. On distingue essentiellement deux types de réseaux de neurones :

- ✓ Réseaux de neurones non bouclés (statiques).
- ✓ Réseaux de neurones bouclés ou récurrent (dynamiques).

2.4.1 Réseaux de neurones non bouclés

Un réseau de neurones non bouclé est constitué d'une ou plusieurs couches cachées contenant chacune un certain nombre de neurones. Il réalise une (ou plusieurs) fonction algébrique de ses entrées par composition des fonctions réalisées par chacun de ses neurones. Dans un tel réseau, le flux de l'information circule des entrées vers les sorties sans "retour en arrière". Tout neurone dont la sortie est une sortie du réseau est appelé "neurone de sortie". Les autres neurones qui effectuent des calculs intermédiaires sont appelés "neurones cachés".

Les réseaux de neurones non bouclés sont souvent organisés en couches (une couche d'entrée, une couche de sortie et une ou plusieurs couches cachées ou carrément aucune couche cachée) avec différentes types de connexions entre les neurones d'une même couche et ces neurones de deux couches successives.

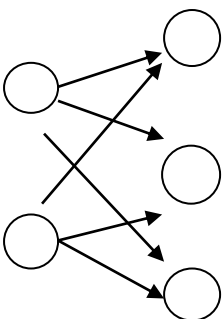
a. Connectivité inter-couches

Les connexions entre les neurones de deux couches peuvent être (fig. 2.3)

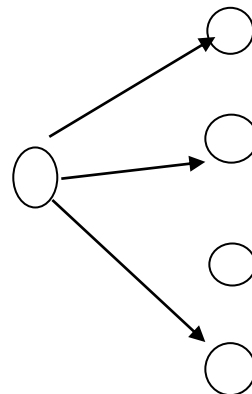
-complètes : Tous les neurones d'une couche sont reliés à tous les neurones de la couche suivante.

-Injective : Chaque neurone d'une couche partage un lien avec un seul neurone d'une autre couche.

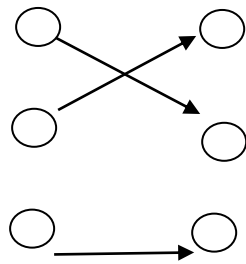
-Probabiliste : Dans ce cas la connexion est partielle selon une probabilité. Si cette probabilité est égale à un, la connexion est complète.



Connexion inter-couches complète.



Connexion inter-couches probabiliste

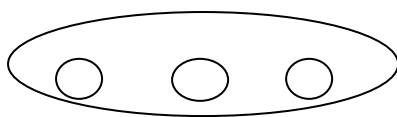


Connexion inter-couches injective.

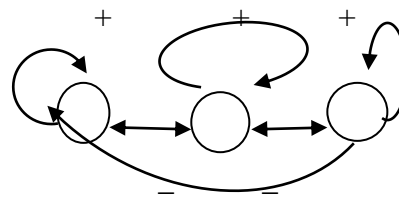
Fig2.3 : Différentes connexions inter-couches dans les réseaux de neurones

b. Connectivité intra-couches

Les connexions entre les neurones d'une même couche peuvent être interdites ou non. Dans le deuxième cas, chaque neurone s'excite lui-même et inhibe les autres qui sont sur la même couche. (Voir Fig.2.4)



Connexion intra-couches interdites



Connexions dites < on center, off surround >

Fig.2.4 : connexions intra-couches

Parmi les réseaux de neurones non bouclés, le perceptron multicouche est le plus populaire

c .Le réseau de neurone perceptron multicouches(MLP) :

Un perceptron multicouche est un type réseau neuronal non bouclé organisé en plusieurs couches, avec des connexions inter-couches et sans connexions intra-couches, au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement. Il s'agit donc d'un réseau à propagation directe chaque couche est constitué d'un certain

nombre de neurones. Les neurones de la couche de sortie représentent la sortie du système global et les neurones de la couche d'entrées représentent les entrées du système. (Fig.2.5)

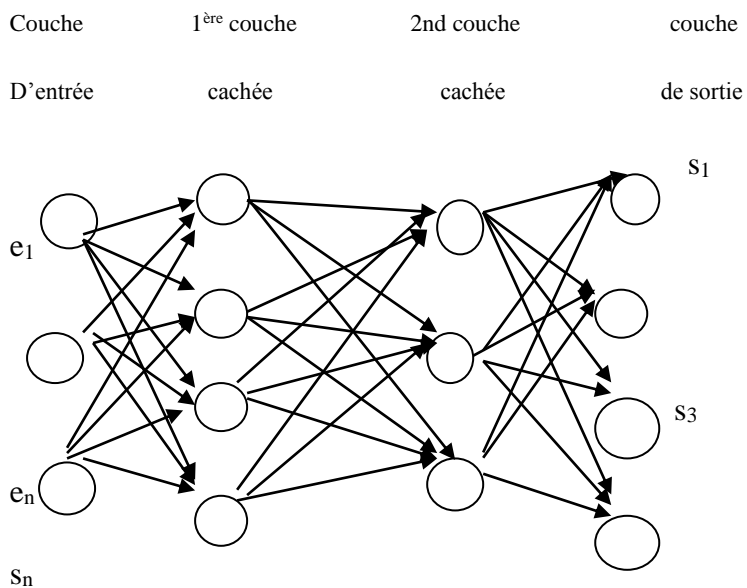


Fig. 2.5 Réseau de neurones perceptron multicouches

2.4.2 Réseaux de neurones bouclés

Ce type de réseau possède des connexions sous forme de boucles, contrairement au réseau non bouclé où les connexions sont orientées de l'entrée vers la sortie du réseau. Le réseau récurrent admet tout type de connexion, c'est-à-dire d'un neurone à n'importe quel autre y compris lui-même. Il est possible dans ce réseau de trouver au moins un chemin qui revient à son point de départ en suivant le sens des connexions, on appelle un tel chemin « cycle ».

a. Réseau à connexions récurrentes :

Les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouches (fig. 2.6).

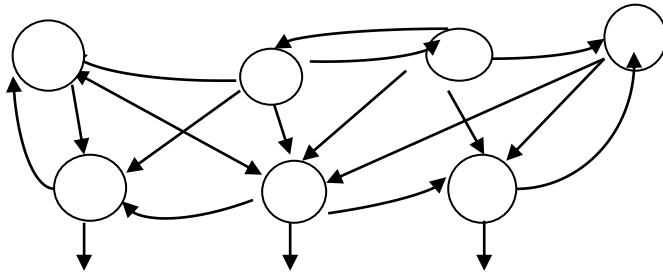


Fig. 2.6 : Réseau à connexions récurrentes

b. Réseau à connexions complètes :

C'est la structure d'interconnexion la plus générale. Chaque neurone est connecté à tous les neurones du réseau (fig. 2.7)

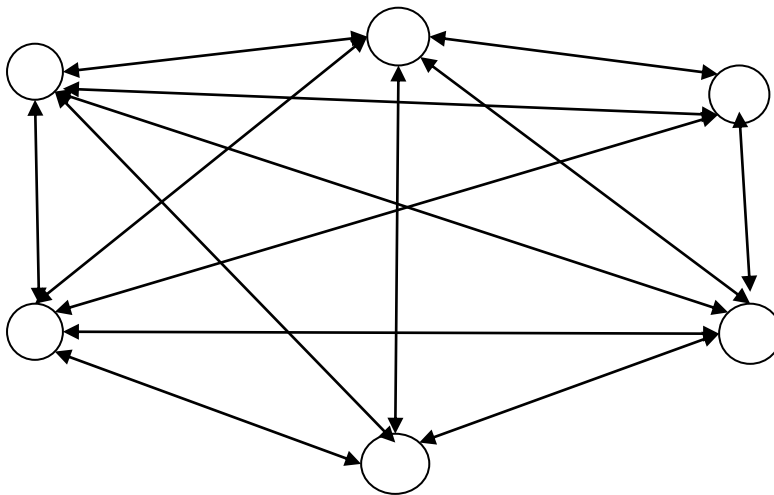


Fig.2.7: Réseau à connexions complètes

2.5. Apprentissage des réseaux de neurones

Le comportement d'un réseau de neurone est en grande partie réglé par sa connectivité, et plus particulièrement, par les poids synaptiques des liens entre les neurones.

L'apprentissage est un aspect intrinsèque de l'intelligence. L'apprentissage d'un réseau de neurone est le processus de détermination de ces poids synaptiques. En général c'est un

processus graduel, itératif ou les poids du réseau sont modifiés plusieurs fois avant d'atteindre leurs valeurs finales. L'apprentissage se fait de manière supervisée ou non supervisée.

2.5.1. Apprentissage non supervisé (unsupervised Learning)

L'apprentissage non supervisé (aussi appelé auto organisation) ou sans superviseur, modifie les poids du réseau en fonction de critères internes comme la coactivation des neurones. Les comportements résultants de cet apprentissage sont en général comparables à des techniques de classification non supervisée.

Ce type d'apprentissage consiste à organiser les entrées présentées au système en classe ou groupes présentant des caractéristiques communes. La distinction entre ces classes est basée sur des mesures de similarité entre les entrées. (Fig.2.8)

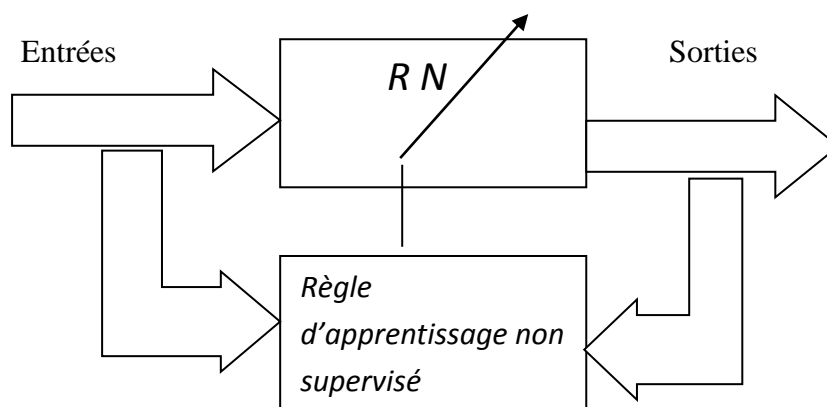


Fig.2.8 : Apprentissage non supervisé

2.5.2. Apprentissage supervisé (supervised Learning)

Dans l'apprentissage supervisé la sortie désirée ou correcte du réseau à une entrée donnée est connue à priori. L'apprentissage du réseau consiste alors à mesurer la différence entre son comportement actuel et le comportement de référence (la sortie désirée) et corriger ses poids de façon à réduire cette erreur (Fig.2.9).

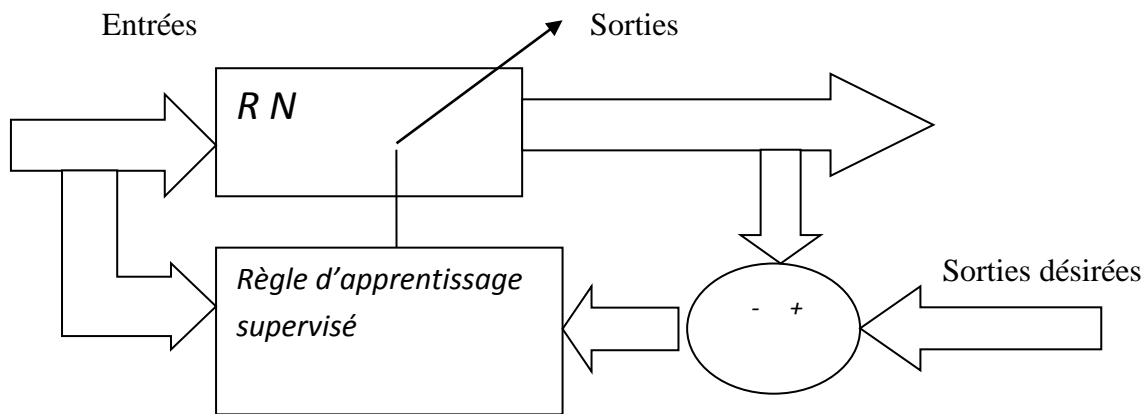


Fig.2.9. Apprentissage supervisé

2.5.3 .Apprentissage du perceptron multicouche par la règle de retro propagation.

L'apprentissage du perceptron multicouche est souvent réalisé par un algorithme supervisé basé sur la retro propagation du gradient

La retro propagation du gradient est une méthode pour calculer le gradient de l'erreur pour chaque neurone du réseau de neurones, de la dernière couche vers la première. Cette technique utilise l'algorithme classique de correction des erreurs basée sur le calcul du gradient grâce à la retro propagation

D'une manière générale, on cherche à obtenir du réseau une réponse préétablie comme étant correcte. On dispose pour cela d'une base de connaissance de type entrées /sorties désirées On compare en suite la sortie de chaque neurones de sortie à la sortie attendue. On introduit une fonction d'erreur pour chaque sortie $e_i = s_i - d_i$ que l'on va chercher à minimiser en modifiant les poids du réseau.

Une fois les poids définis par cet algorithme à partir d'exemples connus de l'utilisateur, on va chercher à extrapoler le réseau, en lui fournissant des entrées inconnues.

L'algorithme de retropropagation se déroule selon les étapes suivantes:

Etape 1 : Initialiser les poids avec les valeurs au hasard.

Etape 2 : Appliquer une donnée d'entraînement au réseau de neurones et calculer la sortie.

Etape 3 : Calculer l'erreur de chacun des nœuds de sortie.

Erreur au nœud K $\delta_K = (t_K - O_K) (1 - O_K) O_K$ sortie obtenue

Etape 4: Calculer la correction des poids pour les nœuds de sorties

$$w_{jk} = w_{jk} + \eta \delta_K Z_j \text{ valeur du nœud } j \text{ de la couche cachée}$$

Facteur d'entraînement

➤ Version stochastique : $\Delta_{jk} = \eta \delta_K Z_j$

Etape 5 : Propager l'erreur sur les poids des nœuds de la couche cachée.

L'erreur d'entrée du $j^{\text{ième}}$ nœud de la couche cachée est $\delta_{in j} = \sum_{K=1}^M \delta_K w_{jK}$

L'erreur du $j^{\text{ième}}$ nœud de la couche cachée est $\delta_j = O_j(1 - O_j) \sum_{K=1}^M \delta_K w_{jK}$

Etape 6: Calculer la correction des poids pour les nœuds de la couche cachée

$$w_{ij} = w_{ij} + \eta \delta_j x_i$$

➤ Version stochastique $\Delta_{ij} = \eta \delta_j x_i$

Etape 7 (version stochastique) :

Calculer la correction des poids pour les nœuds de la couche cachée $w_{ij} = w_{ij} + \Delta_{ij}$

Etape 8 : Vérifier la convergence

Arrêter l'algorithme si une condition de convergence a été atteinte. Sinon, on reprend à l'étape 2

2.6. Propriétés des réseaux de neurones

L'intérêt porté aux réseaux de neurones tient sa justification dans les quelques propriétés fascinantes qu'ils possèdent, parmi lesquelles on peut citer.

Le parallélisme : Son intérêt dans le traitement des données a été mis en évidence par l'échec des méthodes séquentielles pour traiter des problèmes qui nécessitent des quantités énormes de données. Le parallélisme permet une rapidité de calcul supérieure, mais exige de penser et de poser différemment les problèmes à résoudre.

La capacité d'adaptation: C'est la capacité d'apprentissage qui permet au réseau de tenir compte de nouvelles contraintes ou de nouvelles données. De plus elle se caractérise dans

certaines réseaux par leur capacité d'auto organisation qui assure leur stabilité autant que systèmes dynamiques.

La mémoire distribuée: Dans les réseaux de neurones, la mémoire d'un fait correspond à une carte d'activation des neurones. Cette carte est en quelque sorte un codage du fait mémorisé. L'intérêt de cette distribution de la mémoire sur plusieurs entités est ce qu'on appelle la résistance au bruit.

La capacité de généralisation: La capacité de généralisation des réseaux de neurones a été mise en lumière par l'exemple de l'apprentissage des formes du passé pour les verbes de l'Anglais. Cette capacité est essentielle pour reconnaître une forme non apprise.

Il est alors intéressant de disposer d'un système qui puisse à partir d'un ensemble d'exemples, apprendre à trouver des règles sous-jacentes ou à mimer les comportements qui permettent de résoudre le problème.

Facilité de construction: La simulation informatique d'un réseau de neurone pour une petite application est simple et ne nécessite qu'un temps de développement assez court.

2.7. Domaines d'application des réseaux de neurones

Les réseaux de neurones sont de puissants outils d'analyse et d'interprétation de données. Capables de s'adapter automatiquement, ils s'appliquent sur une grande variété de domaines, de la médecine à la gestion de la consommation d'eau et d'électricité ou encore en physique de particules. Les réseaux de neurones ont été principalement appliqués en :

- La reconnaissance des caractères manuscrits ou dactylographiés.
- La reconnaissance de la parole.
- Les prévisions météorologiques.
- L'analyse de tendance financière.
- Imagerie biomédicale.
- Robotique
- Diagnostic des machines.
- Reconnaissance des défauts.
- Biométrie (reconnaissance des visages, empreintes, iris)

2.8. Conclusion

Dans ce chapitre, nous avons introduit quelques notions de base sur les réseaux de neurones formels, et nous avons donné quelques informations permettant d'entrevoir les relations existantes entre ces réseaux et la classification, en basant sur quelques modèles utilisant l'apprentissage supervisé comme le réseau perceptron multicouches.

Malgré les résultats très convainquant de ce réseau dans la classification, il reste toute de même limité.

Nous présenterons dans le chapitre suivant un nouveaux type de réseaux de neurones en l'occurrence les réseaux de neurones convolutifs.

Chapitre 03 : Les Réseaux de neurones convolutifs

3.1 Introduction

La méthode de classification supervisée par les réseaux de neurones multicouches (perceptron) effectue un apprentissage à partir d'un ensemble de données prototypes (exemples des classes) où chaque donnée est décrite par un ensemble d'attributs. La difficulté de choisir les attributs et le nombre de couches, généralement faible (1 à 2 couches), fait que les performances atteintes par le perceptron multicouches sont limitées.

D'autres méthodes d'apprentissage profond ou Deep learning (par opposition au shallow learning, apprentissage peu profond) ont été proposées. Ces méthodes permettent d'éviter les inconvénients cités précédemment et sont actuellement très en vogue.

La plupart des méthodes de Deep Learning utilisent des architectures de réseaux de neurones, ce qui explique pourquoi il est souvent question de réseaux de neurones profonds pour désigner des modèles de Deep Learning. Le terme « profond » revient au nombre de couches cachées du réseau de neurones. Ces derniers comportent que 1 à 2 couches cachées, tandis que les réseaux profonds peuvent en compter jusqu'à 150. L'entraînement des modèles s'effectue à l'aide de vastes ensembles de données labellisées (prototypes) et d'architectures de réseaux de neurones qui apprennent des caractéristiques directement depuis les données, sans avoir à effectuer une extraction manuelle.

Un des types de réseaux de neurones profonds le plus populaire est le réseau de neurones à convolution (Convolutional Neural Network ou CNN). Un CNN utilise des couches à convolution afin d'extraire des caractéristiques directement à partir des données. Son architecture est parfaitement adaptée au traitement des données matricielles telles que les images. Les réseaux de neurones à convolution éliminent la nécessité d'effectuer une extraction de caractéristiques manuelle, qui est généralement une tâche fastidieuse. Les caractéristiques utilisées ne sont pas pré-entraînées, c'est à dire que le réseau les apprend lui-même en s'entraînant sur des images. Cette extraction automatisée des caractéristiques permet aux CNN d'atteindre des performances élevées en classification d'objets. La complexité des caractéristiques apprises augmente avec le nombre de couches cachées du réseau. La première couche cachée peut par exemple apprendre à détecter les contours de l'image, tandis que la dernière apprend à détecter des formes plus complexes spécialement adaptées à la forme de l'objet que l'on vise à reconnaître.

Nous allons décrire dans ce chapitre les réseaux de neurones à convolution et leurs applications à la classification des textures.

3.2 Historique

Le premier réseau de neurones convolutif a été en réalité développé en 1998 par le chercheur français Yann LeCun. Ce réseau appelé LeNet a permis d'atteindre de très bonne performance en reconnaissance des caractères,

Bien que cette approche donne des résultats, ses progrès et son évolution ont été limités par les progrès technologiques en matière de micro-processeurs, de puissance de calculs, et du manque d'accessibilités à des données afin de pouvoir entraîner les neurones.

Cependant certains chercheurs ont continué à travailler sur ce modèle pendant environ deux décennies. Et, avec l'aide des évolutions en matière de technologies mais surtout avec la disponibilité toujours plus grande de données, ont pu améliorer cette technique.

Ce n'est qu'en 2012 que le Deep Learning est remis au goût du jour en remportant avec succès le concours de reconnaissance d'image fondé par l'université de Stamford (Large Scale Visual Recognition Challenge: ILSVRC) grâce à l'évolution en matière de technologies et à la disponibilité toujours plus grande de données. Un nouvel algorithme de Deep Learning explose les records Il s'agit d'un réseau de neurones convolutif appelé AlexNet, largement inspiré du réseau LeNet. ImageNet regroupant 15 000 000 d'images naturelles comportant différents objets et diverses scènes (véhicules, animaux,...).

Aujourd'hui, les réseaux de neurones convolutifs, sont toujours les modèles les plus performants pour la classification d'images. Google, Microsoft, Facebook, Baidu (le moteur de recherche chinois), Alibaba (site marchand chinois), Nvidia (géant du processeur graphique)...utilisent les CNN dans leurs applications.

3.3 Réseaux de neurones convolutifs

Un réseau de neurone convolutif se compose de plusieurs couches comme le montre la figure (3.1)

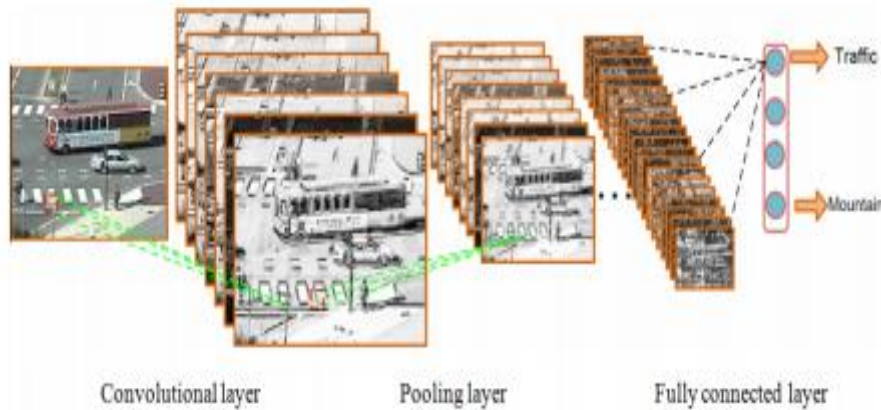


Fig.3.1 : Exemple sur un réseau de neurones convolutifs

3.3.1 Couches convolutives

Les couches convolutives constituent le cœur des réseaux convolutifs. Ces couches se composent d'une grille rectangulaire de neurones ayant un petit champ réceptif étendu à travers toute la profondeur du volume d'entrée.

Le terme "convolution" vient du fait que les réseaux CNN utilisent des opérations de convolution comme dans le traitement du signal ou d'images. Dans la couche de convolution, on applique un produit de convolution entre l'image originale ou celle obtenue dans la couche précédente avec un filtre dont des coefficients sont assimilés à des poids synaptiques.

Le produit de convolution d'une image I avec un filtre F de taille $(s*s)$ est:

$$I_F(x, y) = \sum_{i=-s/2}^{s/2} \sum_{j=-s/2}^{s/2} I(x+i, y+j) F(i+\frac{s}{2}, j+\frac{s}{2})$$

La convolution agit comme un filtrage. Pour effectuer cette opération sur toute l'image, on définit une fenêtre de voisinage de taille $(s*s)$ qui va se déplacer à travers toute l'image. Au tout début de la convolution, la fenêtre sera positionnée tout en haut à gauche de l'image puis elle va se décaler d'un certain nombre de cases (appelé le pas) vers la droite et lorsqu'elle

arrivera au bout de l'image, elle se décalera d'un pas vers le bas ainsi de suite jusqu'à ce que le filtre est parcourue la totalité de l'image (fig. 3.2)

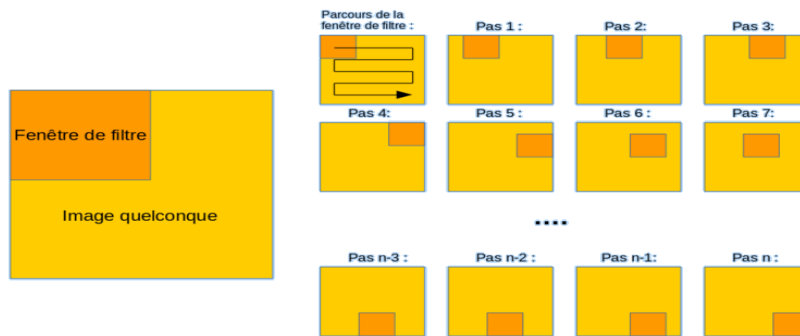


Fig.3.2 : Schéma du parcours de la fenêtre de filtre sur l'image.

Le but est de se servir des valeurs présentes dans le filtre à chaque pas. Par exemple si l'on définit une fenêtre 3 par 3, cela représentera 9 cases du tableau (c'est à dire 9 pixels). La convolution va effectuer une opération avec ces 9 pixels. La figure (3.3) illustre le principe du calcul du produit de convolution

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Image I

0.1	0.2	0.3
0.4	0.5	0.6
0.7	0.8	0.9

Filtre de taille (3*3)

34.8	39.3
52.8	57.3

Image résultat I_F

Fig.3.3 Produit de convolution

L'opération de convolution produit, en sortie une image appelée "carte de caractéristiques" ou "featuremaps". De plus, elle peut être appliquée sur des images multi composantes (plusieurs canaux).

Supposons qu'on a une image de dimension $N \times N \times 3$ (le 3 correspond aux 3 canaux R, G, B). Prenons $N=32$ dans notre exemple. (Fig. 3.4)

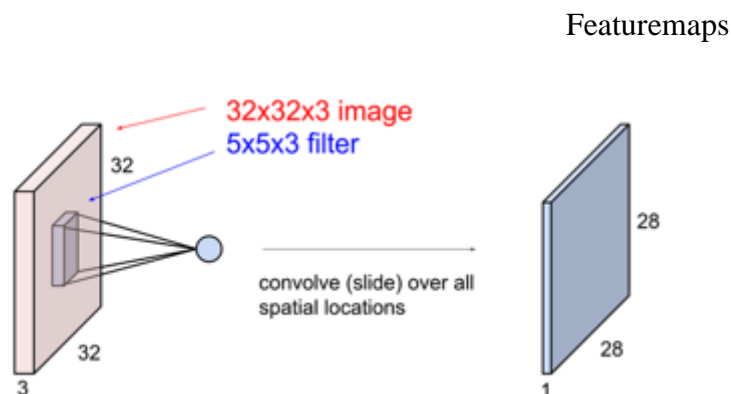


Fig.3.4 : illustration de la convolution d'une image couleur

Une couche de convolution correspond à un filtre (ici de taille $5 \times 5 \times 3$) que l'on va balayer sur l'ensemble de notre image. En sortie on obtient une "image" de taille $28 \times 28 \times 1$, appelée une "featuremap". En général, on n'applique pas qu'un seul filtre, mais un ensemble de k filtres. On obtient alors une pile de k "featuremaps" qui constitue une nouvelle "image multicomposante". La figure (3.5) montre un exemple de 6 featuremaps obtenues par la convolution de 6 filtres.

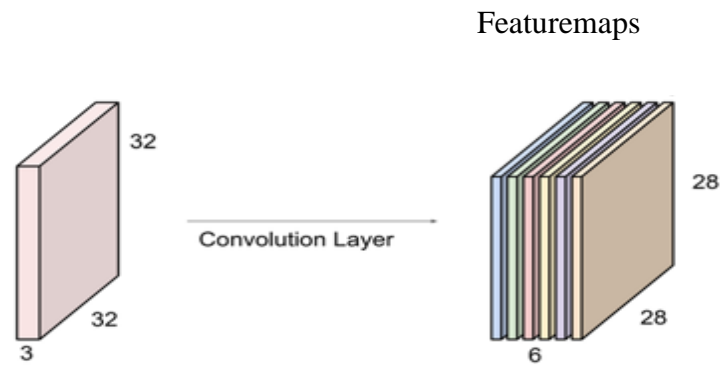


Fig. 3.5: image multi composante de taille 28*28*6

Nous remarquons sur cet exemple que les cartes de caractéristiques ont des dimensions plus réduites que celle de l'image d'entrée. Ceci est dû à la mise à zéro des pixels situés sur les frontières (zéros padding).

Les coefficients des filtres de convolution sont considérés comme des paramètres à apprendre par les réseaux durant la phase d'apprentissage.

3.2.2 Couche de mise en commun (pooling)

Ce type de couche est souvent placé entre deux couches de convolution. Elle reçoit en entrée plusieurs featuremaps, pour ensuite réduire la taille de chacune d'entre elles. Autrement dit; l'opération de pooling consiste à réduire la taille des images caractéristiques, tout en préservant leurs caractéristiques importantes. Elle effectue une sortie de sous échantillonnage.

Il existe plusieurs manières de réaliser le pooling. La méthode la plus utilisée est le 'max-pooling'. Celle-ci consiste à découper l'image en cellules régulières, puis on garde au sein de chaque cellule la valeur maximale. En pratique, on utilise souvent des cellules carrées de petite taille pour ne pas perdre trop d'informations. Les choix les plus communs sont des cellules adjacentes de taille (2*2) ou (3*3) qui ne se chevauchent pas, on obtient en sortie le même nombre de feature maps qu'en entrée, mais de tailles plus petites.

(fig.3.6) illustre cette technique:

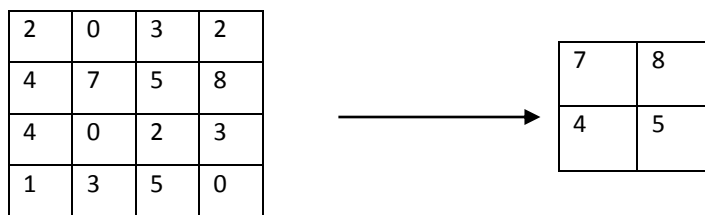


Fig. 3.6 illustration de la méthode maxpolling

Finalement la couche de pooling permet de réduire le nombre de paramètres par conséquent d'accélérer les calculs dans le réseau. Elle permet aussi d'améliorer l'efficacité du réseau et éviter le problème sur-apprentissage.

Aussi, la couche de pooling rend le réseau moins sensible à la position des features : le fait qu'une feature se situe un peu plus en haut ou en bas, ou même qu'elle ait une orientation légèrement différente ne devrait pas provoquer un changement radical dans la classification de l'image.

3.2.3 Couche de correction Relu

Souvent, il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche de correction ReLU (Rectified Linear Units) qui a comme objet le calcul d'une fonction d'activation sur les données de sortie cette fonction (Fig. 3.7), appelée aussi « fonction d'activation non saturante » est une fonction réelle non-linéaire définie par :

$$F(x) = \max(0, x)$$

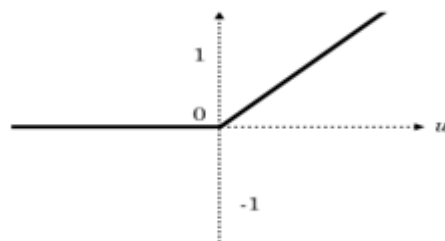


Fig. 3.7: allure de la fonction ReLU

La couche de correction ReLU remplace donc toutes les valeurs négatives reçues en entrées par des zéros.

La fonction Relu est préférable aux autres fonctions d'activation comme la fonction tangente hyperbolique (tanh) et la fonction sigmoïde, car elle permet au réseau d'être plus rapide et d'éviter le problème de "vanishing gradient".

En effet, comme les fonctions tanh et sigmoïde, la fonction ReLu est différentiable et donc bien adaptée à l'algorithme de retro propagation qui est basé sur le calcul du gradient de la fonction d'activation. Cependant, les fonctions tanh et sigmoïde produisent des valeurs entre 0 et 1 et le gradient va tend vers 0 au fur et à mesure qu'on avance dans les couches. La fonction ReLu permet d'amortir ce problème connu sous le nom de "voisinage gradient" car sa valeur est invariante si $x < 0$.

De plus, le calcul des exponentielles présent dans les fonctions tanh et sigmoïde est lent par rapport au calcul du max.

3.2.4 Couches entièrement connectées

L'ensemble successives des cartes de convolutions, pooling et ReLu fournissent au final un ensemble de caractéristiques données sous forme de carte 2D. Ces cartes sont concaténées en un vecteur de caractéristiques, appelé code CNN. Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée d'une ou de plusieurs couches entièrement connectées (fullyconnected ou fc) qu'on peut assimiler au perceptron multicouches. Les neurones dans une couche entièrement connectée ont des connexions vers les sorties de la couche précédente. Leurs fonctions d'activation peuvent donc être calculées avec une multiplication matricielle entre le vecteur d'entrée X et la matrice des poids W auquel on ajoute le vecteur des biais des neurones b :

$$A = WX + b$$

Le nombre de neurones de la dernière couche entièrement connectée est égal au nombre de classes.

Le rôle de cette partie est donc de combiner les caractéristiques du code CNN pour classer une image.....

3.2.5 Couche de perte (LOSS)

Les sorties des neurones de la dernière couche entièrement connectée sont converties en probabilités par l'intermédiaire d'une fonction d'activation "softmax".

La fonction d'activation "softmax" du r ème neurone est définie par :

$$Y_r(x) = \frac{\exp(a_r(x))}{\sum_{j=1}^k \exp(a_j(x))}$$

Où $0 \leq y_r \leq 1$ et $\sum_{j=1}^k y_j = 1$

$Y_r(x)$ est la probabilité que la donnée d'entrée x (images échantillon) appartienne à la r ème classe.

Durant la phase d'apprentissage, une fonction d'erreur (loss) est déterminée afin d'évaluer l'erreur entre les sorties $Y_r(x_i)$ prédites par le réseau pour chaque échantillon x_i et les sorties désirées t_{ij} .

$$t_{ij} = \begin{cases} 1 & \text{si } x_i \in c_j \\ 0 & \text{sinon} \end{cases} \quad i=1, \dots, N \text{ et } j=1, \dots, k$$

La fonction d'erreur est la fonction cross entropie définie comme suit :

$$E = -\sum_{i=1}^N \sum_{j=1}^k t_{ij} \ln y_i(x_i)$$

Pour prédire à partir d'un modèle de réseaux de neurones artificiels, on parcourt le réseau en partant de la couche d'entrée puis, on calcule successivement les valeurs des états internes de chaque neurone dans les couches cachées jusqu'à la couche de sortie.

La prédiction est la classe correspondant au neurone qui possède la plus grande probabilité.

3.3 Paramètres des CNN

En plus de paramètres de réglages habituels (taux d'apprentissage et des constantes de régularisation) utilisés pour l'entraînement des réseaux de neurones perceptron multicouches, nous devons choisir dans les CNNs le nombre de couches de convolution, de correction ReLu, de pooling, et de couches entièrement connectées. Il faut également définir les paramètres pour chaque couche de convolution et de pooling. Trois paramètres permettent de

dimensionner une couche de convolution : le nombre de noyaux de convolution, le pas de chevauchement et la marge à zéro (zéro padding) qui définit le nombre de zéros sur les frontières des cartes, même s'il est parfois souhaitable de conserver la même surface que celle des cartes d'entrée. Concernent la couche de pooling, celle-ci est définie par la taille de la fenêtre de traitement et par le pas de chevauchement. En pratique, une fenêtre de (2*2) avec un pas de 1 sont souvent choisis.

3.4 Exemples de CNN et transfert d'apprentissage

Il est plus usuel d'utiliser des modèles de CNNs déjà entraînés et de les réadapter pour le problème de classification ou autre, c'est ce que l'on appelle le transfert d'apprentissage. Il s'agit de transférer l'apprentissage d'un modèle traitant un problème vers un autre type de problème. On peut effectuer cette opération de deux manières :

-L'extraction de variables du CNN : ici, le CNN est utilisée comme un extracteur, c'est-à-dire qu'un vecteur est extrait à partir d'une certaine couche du modèle sans rien modifier à sa structure ou son poids et le vecteur précédemment extrait est utilisé pour une nouvelle tâche.

-Le réglage fin du modèle de CNN : ici, le nouveau réseau est initialisé avec les poids et la structure du modèle pré-entraîné à utiliser. La structure du modèle pré-entraîné est légèrement modifiée pour la nouvelle tâche, et enfin le nouveau modèle est entraîné pour la nouvelle tâche.

Il existe un grand nombre de réseaux CNN pré entraînés, les plus connus sont :

3.4.1 Google Net

Google Net : a 22 couches de profondeur, et presque 12 fois moins de paramètres (donc plus rapide et moins que Alex Net et beaucoup plus précis. Il réduit le nombre de paramètres de 60 millions (Alex Net) à 4 millions. Leur idée était de créer un modèle qui pourrait également être utilisé sur un Smartphone (conserver un budget de calcul d'environ 1,5 milliard de multiplications par prévision)

Couche de lancement : l'idée de la couche de lancement est de couvrir une plus grande surface, mais aussi de conserver une résolution fine pour les petites informations sur les images. L'idée est donc de convoluer en parallèle différentes tailles, des tailles plus précis

(1*1) à un plus grand (5*5). Le moyen le plus simple d'améliorer les performances en matière d'apprentissage en profondeur consiste à utiliser plus de couches et plus de données.

Google Net utilise 9 modules de démarrage. Le problème est que plus de paramètres signifient également que votre modèle est plus enclin à sur-adapter. Ainsi, pour éviter une explosion de paramètres sur les couches initiales. (Fig.3.8)

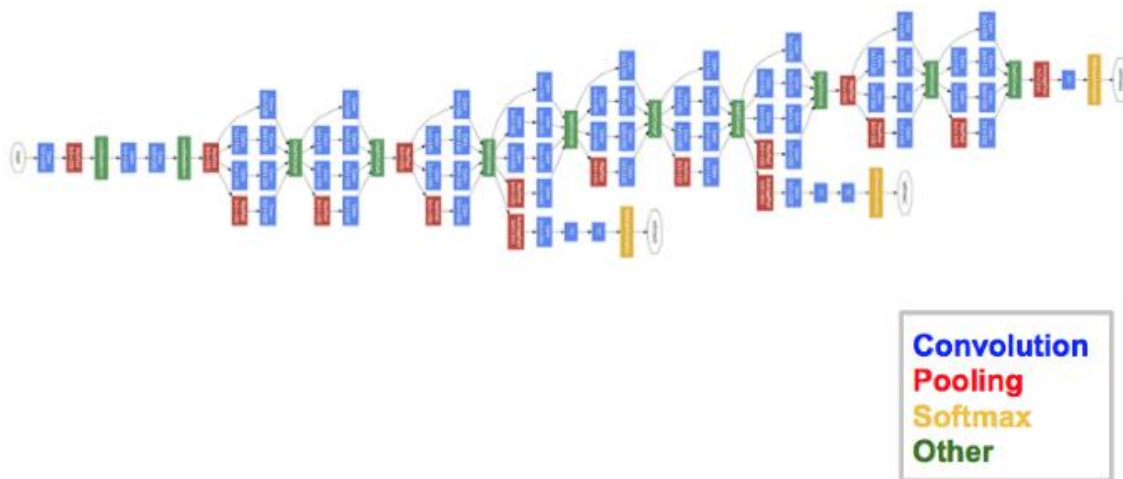


Fig. 3.8 : Architecture de Google Net

3.4.2 VGG

VGG: est un réseau neuronal très performant dans le cadre du challenge de reconnaissance visuelle à grande échelle Image Net. Il a obtenu la première place dans la tâche de localisation des images et la deuxième dans la tâche de classification des images

VGG veut donner une réponse à « comment concevoir la structure du réseau » parmi de nombreux choix. VGG montre également que la profondeur du réseau joue un rôle important et les réseaux plus profonds donnent les meilleurs résultats.

Un inconvénient de VGG est que ce réseau est généralement gros. Il contient environ 160 paramètres. (fig 3.9)

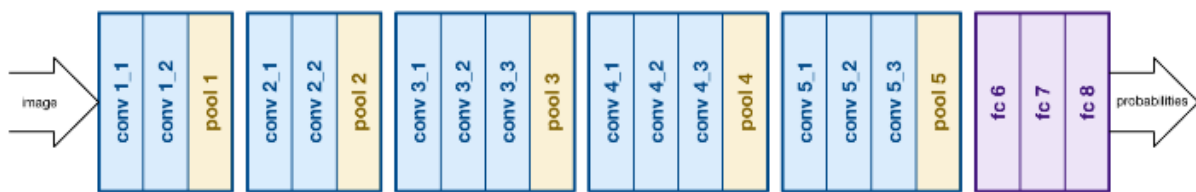


Fig. 3.9: structure de VGG

3.4.3 LeNet

LeNet un réseau de convolution à 7 niveaux pionnier de LeCun et al en 1998, qui classe les chiffres, a été appliqué par plusieurs banques pour reconnaître les numéros manuscrits sur les chèques numérisés en 32x32 pixels. La capacité à traiter des images à plus haute résolution nécessite des couches plus convolutives et plus grandes, cette technique est donc limitée par la disponibilité des ressources informatiques. (fig.3.10)

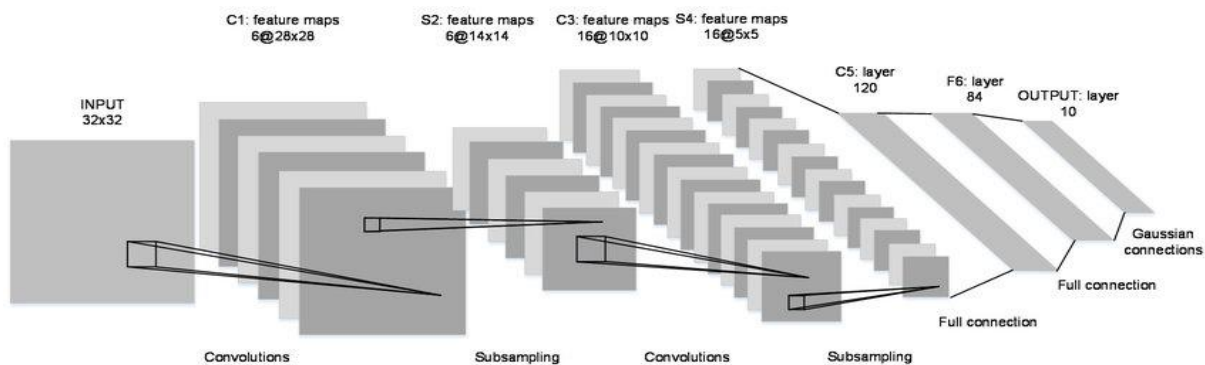


Fig.3.10 : Architecture de Le Net

3.4.4 Alex Net

Alex Net : est le nom d'un réseau neuronal convolutif, initialement écrit avec CUDA pour fonctionner avec le support du GPU, qui a participé au défi de reconnaissance visuelle à grande échelle Image Net en 2012. Alex Net a largement surperformé tous ses concurrents précédents et a remporté le défi en réduisant l'erreur de top 5 de 26% à 15,3% il a été conçu par le groupe Supervision, composé d'Alex krizhevsky. Alex Net a eu un grand impact dans le domaine de l'apprentissage automatique, en particulier dans l'application de l'apprentissage

en profondeur à la vision artificielle. Alex Net contenait huit couches ;les cinq première étaient des couches convolutionnelles et les trois dernières étaient des couches entièrement connectées il a parallèlement deux lignes CNN formées sur deux GPU avec des connexions croisées et Il a utilisé la fonction d'activation non saturante de Relu, qui a montré une amélioration des performances d'entraînement par rapport au Tanh et au sigmoïde. Il a été formé pendant 6 jours simultanément sur deux GPU NvidiaGeforce GTX 580, ce qui explique pour quoi le réseau est divisé en deux pipelines. (fig .3.11)

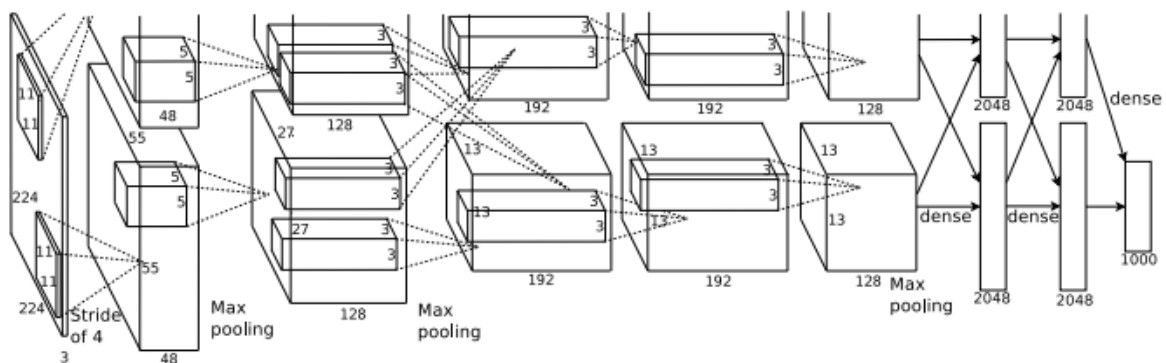


Fig.3.11 Architecture d'Alex Net

3.4.5 ResNet

Le réseau Residual Neural Network (ResNet) a introduit une architecture innovante avec des «sauts de connexions» et une lourde normalisation par lots. Ces connexions sont également appelées unités gated ou unités récurrentes gated et ont une forte similitude avec les éléments récents réussis appliqués dans les RNN. Grâce à cette technique, ils ont pu former un NN avec 152 couches tout en ayant une complexité inférieure à celle de VGGNet. Il atteint un taux d'erreur de 3,57% parmi les cinq premiers, ce qui est supérieur à la performance au niveau humain sur cet ensemble de données. (Fig.3.12)

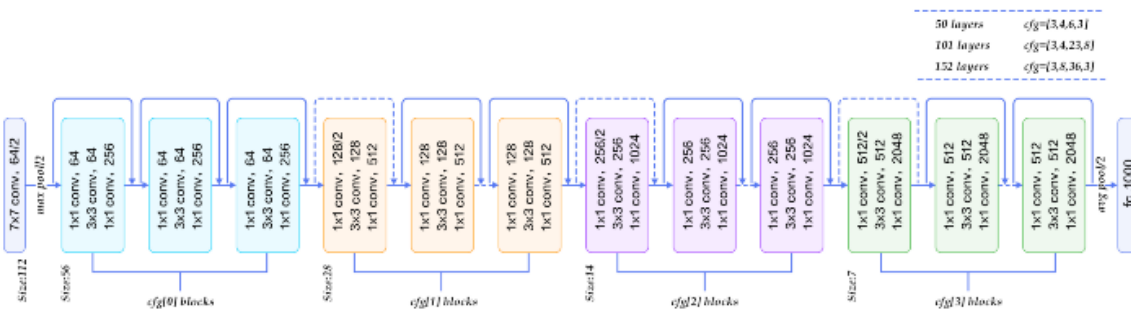


Fig.3.12 : Architecture de ResNet

3.4.6 Overfeat

Overfeat : est un classificateur d'image basé sur un réseau convolutionnel et un extracteur de fonctionnalités. Il a été formé sur le jeu de données Image Net et a participé au concours Image Net 2013 (fig.3.13)

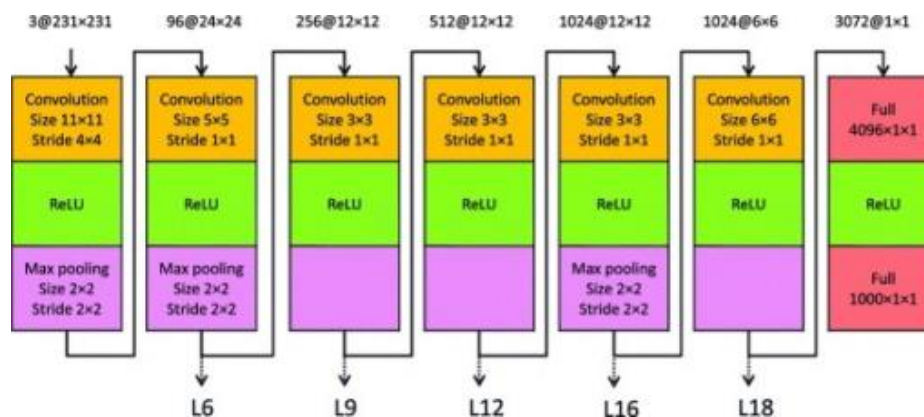


Fig.3.13 : Architecture d'Overfeat

3.5 Classification des textures par un CNN

Nous avons développé un réseau de neurones convolutif pour la classification des textures issues de la base: KTH-TIPS. Rappelons que KTH-TIPS contient 810 images couleur de taille 200x200, réparties en 10 classes de texture. Chaque classe contient 81 images: Pour accélérer les calculs, nous avons réduit la taille de ces images à 32x32. A partir de la base complète, nous avons formé une base pour effectuer l'apprentissage du CNN et une base de test pour évaluer la performance du CNN. 650 images sont utilisées comme base d'apprentissage, soit 65 images par classe et 160 comme base de test soit 16 image par classe.

3.5.1 Architecture du CNN proposé

Nous avons effectué plusieurs tests afin de retenir la meilleure architecture du CNN conduisant au meilleur taux de classification. Pour cela, nous avons varié le nombre de couches, le nombre et la taille des filtres de convolution. Finalement, nous avons retenu le modèle de la figure (3.14). Le modèle ainsi développé est composé de six couches de convolution et trois couches de maxpooling et de trois couches de fullyconnected.

L'image en entrée est de taille 32x32, l'image passe d'abord à la première couche de convolution. Cette couche est composée de 32 filtres de taille 3x3, la fonction d'activation ReLU est utilisé pour forcer les neurones à retourner des valeurs positives. Cette convolution produit 32 featuremaps de taille 32*32 chacune. Ensuite, les 32 featuremaps sont présentées en entrée à la deuxième couche de convolution qui est composée aussi de 32 filtres de taille 3x3. La fonction d'activation ReLU est appliquée sur les couches de convolutions. Le Maxpooling est appliqué après pour réduire la taille de l'image et des paramètres. À la sortie de cette couche, nous obtenons 32 featuremaps de taille 16x16. On répète les mêmes opérations sur les couches de convolutions trois, quatre, cinq et six. Les couches trois et quatre sont composées de 64 filtres de tailles 3X3 et les couches cinq et six sont composées de 128 filtres de tailles 3x3. La fonction d'activation ReLU est appliquée toujours sur chaque convolution. Une couche de Maxpooling est appliquée après les couches de convolutions quatre et six. À la sortie de la dernière couche Maxpooling, nous aurons 128 featuremaps de taille 4*4. Le vecteur de caractéristiques issu des convolutions a une dimension de 2048. Après ces six couches de convolution, nous utilisons un réseau de neurones composé de trois couches fullyconnected. Les deux premières couches ont chacune 1 024 neurones. Le nombre de neurones dans la troisième couche est égal au nombre de classes dans la base d'image KTH-TIPS, soit 10. . Les sorties des neurones de la dernière couche entièrement connectée sont converties en probabilités par l'intermédiaire d'une fonction d'activation "softmax".

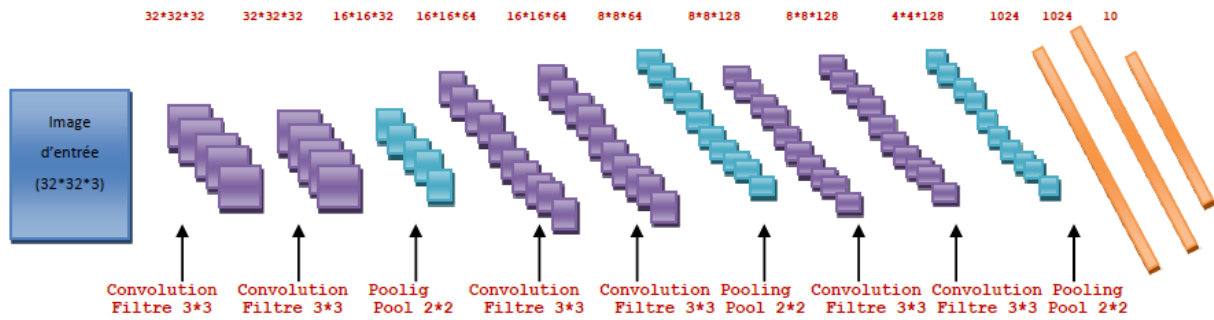


Fig. 3.14: Architecture du CNN utilisé pour la classification des textures

3.5.2 Résultats obtenus

Pour l'évaluation des résultats, nous avons utilisé le taux de classification ou précision (Accuracy), estimé par le rapport entre le nombre d'images bien classées sur le nombre d'images testées. La figure (3.15) montre comment varie le taux de classification (Accuracy) et l'erreur (Loss) obtenus sur la base d'apprentissage en fonction du nombre d'itérations.

On peut constater que la précision augmente au fur et à mesure des itérations. A l'inverse, l'erreur diminue avec les itérations. Le réseau atteint un taux de classification supérieur à 90% sur la base d'apprentissage. Le taux de classification obtenu sur la base de test est de 88.8%, soit 142 images sur un total de 160 ont été correctement classées. Sauf 18 images ont été mal reconnues.

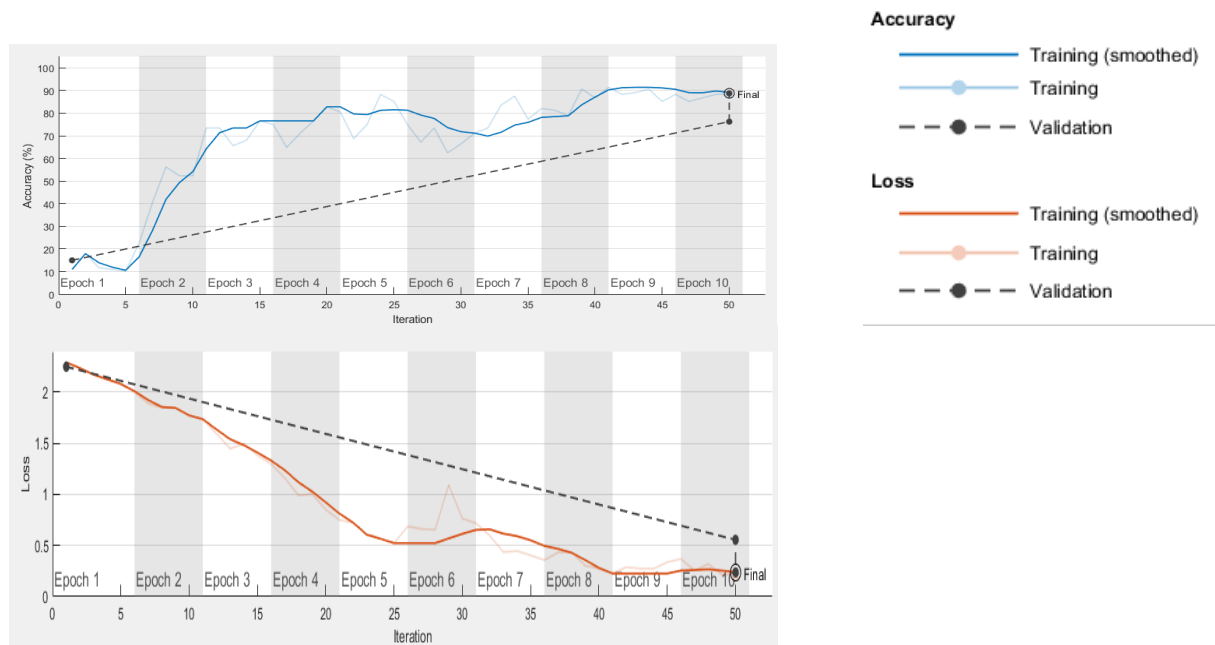


Fig. 3.15: Variation du taux de classification et de l'erreur

La matrice de confusion de la figure (3.16) donne plus de détails sur les résultats de classification obtenus sur la base de test. La diagonal de cette matrice affiche le nombre d'images correctement classées. Ainsi, on peut lire que sur les 10 classes, 6 classes (la première, la cinquième et quatre dernières) ont été correctement identifiées à 100%. Le plus mauvais score est obtenu sur la deuxième et la sixième classes, soit 56.3% et 62.5%, respectivement.

Confusion Matrix

Output Class	aluminium_oil	16	0	0	0	0	0	0	0	0	100%
		10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	brown_read	0	9	0	0	0	0	0	0	0	100%
		0.0%	5.6%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	corduroy	0	2	15	0	0	0	0	0	0	88.2%
		0.0%	1.3%	9.4%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	11.8%
	cotton	0	0	0	12	0	6	0	0	0	66.7%
		0.0%	0.0%	0.0%	7.5%	0.0%	3.8%	0.0%	0.0%	0.0%	33.3%
	cracker	0	0	0	0	16	0	0	0	0	100%
		0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	linen	0	0	0	2	0	10	0	0	0	83.3%
	0.0%	0.0%	0.0%	1.3%	0.0%	6.3%	0.0%	0.0%	0.0%	16.7%	
orange_peel	0	0	0	0	0	0	16	0	0	100%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	
sandpaper	0	0	0	2	0	0	0	16	0	88.9%	
	0.0%	0.0%	0.0%	1.3%	0.0%	0.0%	0.0%	10.0%	0.0%	11.1%	
sponge	0	5	1	0	0	0	0	0	16	72.7%	
	0.0%	3.1%	0.6%	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	27.3%	
styrofoam	0	0	0	0	0	0	0	0	0	16	100%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%
	100%	56.3%	93.8%	75.0%	100%	62.5%	100%	100%	100%	100%	88.8%
	0.0%	43.8%	6.3%	25.0%	0.0%	37.5%	0.0%	0.0%	0.0%	0.0%	11.3%
	aluminium_oil	brown_read	corduroy	cotton	cracker	linen	orange_peel	sandpaper	sponge	styrofoam	
	Target Class										

Fig. 3.16: Matrice de confusion

3.6 Conclusion

Nous avons présenté dans ce chapitre quelques notions sur les réseaux de neurones convolutifs et le rôle de chaque couche qui le constitue.

Un avantage majeur des réseaux convolutifs est l'utilisation d'un poids unique associé aux signaux entrant dans tous les neurones d'un même noyau de convolution. Cette méthode réduit l'empreinte mémoire, améliore les performances et permet une invariance du traitement par translation. C'est le principal avantage du réseau de neurones convolutifs par rapport au perceptron multicouches, qui, lui, considère chaque neurone indépendant et affecte donc un poids différent à chaque signal entrant.

Comparés à d'autres algorithmes de classification d'image, les réseaux de neurones convolutifs utilisent relativement peu de prétraitement. Cela signifie que le réseau est responsable de faire évoluer tout seul ses propres filtres (apprentissage sans supervision), ce qui n'est pas le cas d'autres algorithmes plus traditionnels. L'absence de paramétrage initial et d'intervention humaine est des atouts majeurs des CNNs.

Nous avons également proposé un modèle de réseau CNN pour classer des images de texture. Son application sur la base de texture KTH-TIPS a donné des résultats appréciables.

Conclusion générale

Conclusion générale

Nous avons présenté dans ce mémoire une méthode de classification des textures basée sur les réseaux de neurones convolutifs.

Il offre des outils intéressants dans beaucoup de domaines.

Dans ce projet nous avons discuté des notions fondamentales des réseaux de neurones en générale et des réseaux de neurones convolutionnels en particulier. Nous avons introduit ces réseaux de neurones convolutionnels en présentant les différents types de couches utilisées dans la classification: la couche convolutionnelle, la couche de rectification, la couche de pooling et la couche entièrement connectées, la couche LOSS.

Les réseaux de neurones à convolution éliminent la nécessité d'effectuer une extraction de caractéristiques manuelle, qui est généralement une tâche fastidieuse. Les caractéristiques utilisées ne sont pas pré entraînés, c'est à dire que le réseau les apprend lui-même en s'entraînant sur des images.

Bibliographie

Bibliographie

- <http://lup.lub.lu.se/luur/download?func=downloadFile&recordId=8917826&fileId=8917832>
- <https://ro.uow.edu.au/cgi/viewcontent.cgi?referer=https://www.google.dz/&httpsredir=1&article=1534&context=infopapers>
- <https://medium.com/@CharlesCrouspeyre/comment-les-r%C3%A9seaux-de-neurones-%C3%A0-convolution-fonctionnent-b288519dbcf8>
- <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5082166-quest-ce-quun-reseau-de-neurones-convolutif-ou-cnn>
- <https://openclassrooms.com/fr/courses/966823-developpez-vos-applications-3d-avec-opengl-3-3/963306-les-textures>
- https://scholar.google.com/scholar?q=reseaux+de+neurones&hl=fr&as_sdt=0&as_vis=1&oi=scholar