

**République Algérienne Démocratique et Populaire**

Ministère de l'enseignement supérieur

Et de la recherche scientifique



**Université Mohammed Seddik Ben Yahia – Jijel**

Faculté des Sciences et de la Technologie

Département d'Electronique

Mémoire de fin d'études pour l'obtention du diplôme de

**Master**

En :

**Electronique**

Option :

**Electronique des Systèmes Embarqués**

Thème :

*Reconnaissance des images avec  
les réseaux de neurones artificiels*

Encadré par :

 Dr. Sabrina BOUATMANE

Réalisé par :

 Rafik BOUZELIFA

 Seyfeddine ROUIMEL

Promotion 2019

# ***Remerciements***

*En premier lieu, nous tenons à remercier notre DIEU, Notre créateur pour nous avoir la force pour accomplir ce travail.*

*Nous tenons à remercier tout d'abord notre encadreur : Mme Sabrina BOUATMANE , pour tous ses conseils, son soutien, son aide, et son orientation.*

*Nous remercions les membres de jury d'avoir accepté de juger notre travail.*

*Nous remercions tous les enseignants du département d'électronique et des sciences et de technologies, aussi notre promotion 2018-2019 Master électronique.*

*Nous remercions toutes les personnes qui nous ont soutenues de près ou de loin dans la réalisation de ce travail.*

# *Dédicaces*

*Je dédie ce mémoire :*

*En premier lieu, à mes très chers parents, que nulle dédicace ne puisse exprimer mes sincères sentiments, pour leur patience illimitée, leur encouragement continu, leur aide, en témoignage de mon profond amour et respect pour leurs grands sacrifices*

*« ou Allah me les protèges »*

*Aux personnes dont j'ai bien aimé la présence dans ce jour, à tous mes frères, mes sœurs et toute ma grande famille.*

*A mon binôme seyfeddine et toute sa famille.*

*A mes aimables amis et collègues d'études. Aux personnes qui m'ont toujours aidé et encouragé, qui était toujours à mes cotés, et qui m'ont accompagné durant mon chemin d'études principales et supérieurs.*

*Rafik bouzelifa*

# *Dédicaces*

*Je dédie ce travail :*

*À mes très chers parents pour leur soutien durant toute  
ma vie d'études et sans Lesquels je n'aurais jamais devenu  
ce que je suis.*

*À mon cher frère Hichame, et toute ma famille.*

*À mon binôme rafik et toute sa famille.*

*À tous mes amis*

*À toutes les professeurs et enseignants que j'ai eu durant  
tout mon cursus scolaire et qui m'ont permis de réussir*

*Dans mes études.*

*À mes accompagnants durant mon chemin d'études supérieures, mes  
aimables amis, collègues d'étude.*

*Et à tous ceux qui ont contribué de près ou de loin pour que ce*

*Projet soit possible, je vous dis merci.*

*Seyfeddine rouimel*

# *Sommaire*

Remerciements .....	i
Dédicace .....	ii
Sommaire.....	iv
Liste des figures .....	vii
Liste des tableaux .....	ix
Liste des abréviations .....	x
 Introduction générale.....	 1

## *Chapitre I : Les attributs LBP pour la classification des images de texture*

I.1. Introduction .....	4
I.2. Pourquoi la classification des images ? .....	4
I.3. Architecture d'un système de classification des images .....	5
I.4. Caractéristiques d'une image .....	6
I.4.1. Pixel .....	7
I.4.2. Dimension et Résolution.....	7
I.4.3. Voisinage .....	7
I.4.4. Niveau de gris .....	8
I.5. La texture .....	8
I.5.1. Définition .....	8
I.5.2. Pourquoi la texture ?.....	9
I.5. Les types de textures.....	10
I.6. Analyse de texture.....	12
I.6.1. Les approches structurelles (macrotextures) .....	12

---

I.6.2. Approche statistique des microtextures .....	13
I.7. Analyse de texture par LBP (local binary pattern) .....	14
I.7.1. Le LBP de base .....	14
I.7.2. LBP uniforme .....	16
I.7.3. LBP invariant par rotation .....	18
I.8. Evaluation d'un système de classification .....	19
I.8.1 Courbe ROC (Received Operating Characteristic) .....	21
I.9. Conclusion .....	22

## ***Chapitre II : Les réseaux de neurones artificiels***

II.1. Introduction .....	23
II.2. Notion de classificateur .....	23
II.3. Généralité sur les réseaux de neurones artificiels .....	24
II.3.1. Historique .....	24
II.3.2. Neurones biologiques .....	24
II.4. Les réseaux de neurones artificiels .....	25
II.4.1. Modèle mathématique .....	25
II.4.2. Fonctions d'activation .....	26
II.4.3. Connectivité.....	26
II.5. Modèles de réseaux de neurones .....	27
II.5.1. Réseau de neurone monocouche (Perceptron) .....	28
II.5.2. Perceptron multicouche .....	29
II.5.3. Les réseaux RBF.....	30
II.5.4. Réseaux de Hopfield .....	31
II.5.5. Réseaux de Kohonen (Self Organized Map).....	31

---

II.5.6. Réseaux de Neurones Convolutionnels .....	32
II.6. Apprentissage d'un réseau de neurones .....	33
II.6.1. Apprentissage supervisé .....	33
II.6.2. Apprentissage non supervisé .....	34
II.7. Apprentissage du perceptron multicouche .....	34
II.7.1. Algorithme de rétro propagation du gradient .....	35
II.7.2. Les paramètres d'apprentissage.....	36
II.8. Les avantages et les inconvénients des réseaux de neurones .....	36
II.8.1. Avantages .....	37
II.8.2. Inconvénients.....	37
II.9. Conclusion .....	37

### ***Chapitre III : Tests et résultats***

III.1. Introduction .....	38
III.2. Base de données de texture .....	38
III.3. Implémentation de l'algorithme de reconnaissance de texture.....	39
III.3.1. Extraction des caractéristiques LBP .....	40
III.3.2. Création du réseau de neurone .....	41
III.3.3. Phase d'apprentissage .....	41
III.3.4. Test de performance .....	43
III.4. Conclusion .....	48
Conclusion générale .....	49
Références .....	51

## *Liste des figures*

<b>Figure I.1</b> : Système de classification des images de texture.....	5
<b>Figure I.2</b> : Représentation des notions image et pixel .....	6
<b>Figure I.3</b> : Exemples des images texturées : a) liège, b) bois, c) lin, d) laine, e) laitue, f) feuille d'aluminium et g) pain blanc.....	9
<b>Figure I.4</b> : Exemples de macro (1ère ligne) et microtextures (2ième ligne). .....	11
<b>Figure I.5</b> : Exemples de sables à 2 résolutions différentes.....	12
<b>Figure I.6</b> : Opérateur LBP. ....	15
<b>Figure I.7</b> : Exemples de voisinage utilisés pour le calcul des LBP. ....	15
<b>Figure I.8</b> : Primitives extraites par les motifs binaires locaux. (a) et (b) correspondent à des taches respectivement claires et sombres, (c) est une fin de ligne, (d) une bordure et (e) est un coin.....	16
<b>Figure I.9</b> : (a) : Trois voisinages pour des R et P différents, (b) : Textures particulières détectées par $LBP_{u2}$ .....	17
<b>Figure I.10</b> : Motifs binaires locaux uniformes 2 et invariants en rotation.....	18
<b>Figure I.11</b> : Les 36 modèles binaires uniques invariants en rotation qui peuvent apparaître dans le cercle symétrique ensemble voisin de $LBP_{8,R}^{ri}$ . Les cercles noir et blanc correspondent aux valeurs de bits 0 et 1 dans la sortie 8 bits de l'opérateur. La première rangée contient les neuf modèles « uniformes » et les nombre sà l'intérieur correspondent à leur unique $LBP_{8,R}^{riu2}$ codes. ....	19
<b>Figure I.12</b> : Courbe ROC.....	21
<b>Figure II.1</b> : Classification et modélisation des frontières entre les classes .....	23
<b>Figure II.2</b> : Un neurone avec son arborisation dendritique .....	24
<b>Figure II.3</b> : Modèle d'un neurone artificiel .....	25
<b>Figure II.4</b> : Passage du neurone biologique vers le neurone formel .....	26
<b>Figure II.5</b> : Différents types de réseaux de neurones .....	28
<b>Figure II.6</b> : Perceptron monocouche .....	29
<b>Figure II.7</b> : Perceptron multicouche. ....	30
<b>Figure II.8</b> : Schéma d'un RBF. ....	31
<b>Figure II.9</b> : Architecture standard d'un réseau de neurone convolutionnel .....	32
<b>Figure II.10</b> : Apprentissage supervisé. ....	33

<b>Figure II.11</b> : Apprentissage non supervisé. ....	34
<b>Figure III.1</b> : Echantillons d'images texture de la base UIUCTex. ....	39
<b>Figure III.2</b> : Organigramme d'un système de reconnaissance d'image. ....	40
<b>Figure III.3</b> : L'architecture d'un réseau neurone.....	41
<b>Figure III.4</b> : L'effet de la validation pour la descripteur LBP uniforme (2,16) .....	43
<b>Figure III.5</b> : L'erreur de validation pour LBP uniforme (2,16).....	43
<b>Figure III.6</b> : Les courbes ROC des méthodes LBP différentes. ....	44
<b>Figure III.7</b> : Les courbes des taux d'erreur (%) en fonction de l'apprentissage. ....	45
<b>Figure III.8</b> : Les courbes des taux d'erreur (%) en fonction du nombre des couches.....	45

## *Liste des tableaux*

<b>Tableau I.1</b> : (a) : Voisinage à 4, (b) :Voisinage à 8.....	5
<b>Tableau I.2</b> : Matrice de confusion. ....	17
<b>Tableau II.1</b> : Passage du neurone biologique vers le neurone formel.....	26
<b>Tableau II.2</b> : Fonctions de transfert.....	27
<b>Tableau III.1</b> : Taux d'erreur (%) selon les différentes méthodes LBP. ....	44
<b>Tableau III.2</b> : Taux d'erreur (%) des différentes classes pour les différentes méthodes LBP.....	47

## ***Liste des abréviations***

**IF** : **F**unction **I**mage

**RVB** : **R**ouge, **V**ert, **B**leu

**LBP** : **L**ocal **B**inary **P**attern

**ROC** : **R**eceived **O**perating **C**haracteristic

**RNA** : **R**éseaux **N**euronaux **A**rtificiels

**CNN** : **C**onvolutional **N**eural **N**etwork

**MLP** : **M**ulti **L**ayer **P**erceptron

**RBF** : **R**adial **B**asic **F**unctions

**SOM** : **S**elf –**O**rganising **M**ap

**ART** : **A**daptive **R**esonance **T**heory

**PMC** : **P**erceptron **M**ulti **C**ouches

## ***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 d'images. 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.

Des méthodes générales ont été développées en reconnaissance des formes pour extraire automatiquement des informations des données sensibles afin de caractériser les classes de forme (apprentissage) et d'assigner automatiquement les données à ses classes (reconnaissance ou classification).

Un système de reconnaissance de forme peut être résumé en deux phases principales : L'extraction de caractéristiques consiste en des transformations mathématiques calculées sur les pixels d'une image numérique. Les caractéristiques visuelles permettent généralement de mieux rendre compte de certaines propriétés visuelles de l'image, utilisées pour des traitements ultérieurs entrant dans le cadre d'application telles que la détection d'objets ou la recherche d'images par le contenu.

La classification, se réfère au classement ou groupement d'éléments de données dans des ensembles similaires. L'étape d'extraction de caractéristiques (attributs) réduit donc les données en mesurant certaines propriétés ou caractéristiques des images étiquetées. Ces attributs sont ensuite transmis à un classifieur qui évalue ces caractéristiques et prend une décision relative à la classe de chaque image.

On s'intéresse dans notre travail à la conception et l'évaluation d'un système de classification automatique des images texturées. L'objectif visé dans notre cas par la classification est de pouvoir distinguer entre les classes d'images correspondantes aux différentes textures.

L'analyse des textures est un sujet ancien de l'analyse d'image et de la vision par ordinateur. Les techniques existantes sont utilisées pour identifier une texture quelconque

ou diviser une image texturée en plusieurs régions comportant des caractéristiques identiques. Nous les retrouvons dans plusieurs domaines d'application comme l'analyse de scènes et de vidéos, l'imagerie médicale, l'industrie etc.

L'analyse d'images texturées a pour but d'extraire des caractéristiques ou attributs de texture afin de classer les images. La classification des images est utilisée dans différents domaines tels que la reconnaissance des empreintes, des visages, indexation, reconnaissance de tumeurs etc....

Dans notre cas, nous avons choisi l'opérateur du motif binaire local (Local Binary pattern) pour l'extraction des caractéristiques. L'idée de cet opérateur est d'assigner à chaque pixel un code dépendant des niveaux de gris de son voisinage. Le niveau de gris du pixel central d'un bloc 3 X 3 est comparé à celui de ses voisins suivant certaine équation. Les pixels de ce bloc sont seuillés par sa valeur de pixel central, multiplié par des puissances de deux et ensuite additionné pour obtenir un code pour le pixel central. Ces codes ou leurs statistiques, le plus souvent l'histogramme, sont ensuite utilisées comme vecteur de caractéristiques de l'image.

La deuxième phase est la classification. Nous avons utilisé la méthode de réseaux de neurone. Les réseaux de neurones sont constitués d'un ensemble de neurones artificiels ou nœuds qui sont analogues aux neurones biologiques. Ils sont issus d'une tentative de conception d'un modèle mathématique très simplifié du cerveau humain en se basant sur notre façon d'apprendre et de corriger nos erreurs.

Il existe plusieurs types de réseaux de neurones tels que les "perceptrons", les réseaux à fonctions de base radiales et les réseaux récurrents. Parmi eux, les perceptrons à (feed-forward) et entraînés par rétropropagation (backpropagation) ont eu un succès important dans plusieurs applications. Nous avons choisi le perceptron multicouche pour classer les images de textures.

Pour évaluer les performances de notre système, nous avons utilisé une base de données qui contient 1000 images réelles réparties sur 25 classes de textures différentes (Granit, mur, gravier, ...) L'évaluation de classification est réalisée en termes de taux d'erreur global, le taux d'erreur par classe et la courbe ROC.

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

Dans le premier chapitre on va présenter l'architecture d'un système de classification d'images, la notion de texture dans les images et l'extraction des caractéristiques de texture par la méthode des motifs binaires locaux LBP ainsi que l'évaluation d'un système de classification.

Le deuxième chapitre est consacré à la description des réseaux de neurones, leurs architectures, et leurs fonctionnements dans le domaine de la classification.

Dans le troisième chapitre, nous présenterons la partie expérimentale de notre travail et on discutera des différents résultats obtenus, et à la fin on terminera par une conclusion générale.

# *Chapitre I*

*Les attributs LBP pour la classification des images de texture*

## **I.1. Introduction**

La texture est une caractéristique propre de l'objet ; elle nous permet de le décrire partiellement. Bien sûr, la forme et la couleur sont deux autres caractéristiques très importantes. Pour faire la distinction entre une orange et une cerise, les informations de forme et de couleur sont certainement plus significatives que l'information texture. Mais l'information texture devient primordiale lorsqu'on veut faire la distinction entre deux zones d'une image de même couleur (ou bien de même niveau de gris).

## **I.2. Pourquoi la classification des images ?**

La classification des images consiste à répartir systématiquement des images selon des classes établies au préalable, classer une image est le fait de lui correspondre une classe, marquant ainsi sa parenté avec d'autres images.

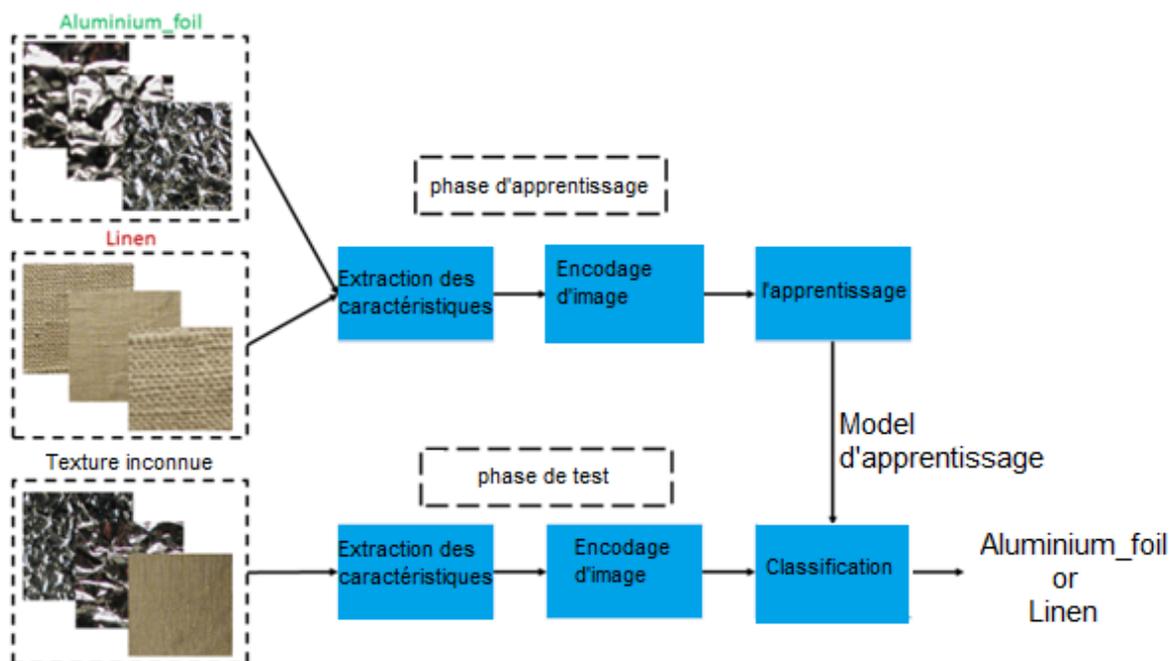
En général reconnaître une image est une tâche aisée pour un humain au fil de son existence, il a acquis des connaissances qui lui permettent de s'adapter aux variations qui résultent de conditions différentes d'acquisition. Il lui est par exemple relativement simple de reconnaître un objet dans plusieurs orientations partiellement cachées par un autre de près ou de loin et selon diverses illuminations.

Toutefois les progrès technologiques en terme d'acquisition d'images (microscopes, caméras, capteurs) et de stockage engendrent des bases de données riche en information et multiplient les domaines d'applications, il devient alors difficile pour l'humain d'analyser le nombre important d'images, le temps requis, le caractère répétitif de la tâche et la concentration nécessaire sont problématiques. Toutefois, celle-ci n'est pas forcément aisée pour un programme informatique pour lequel une image est un ensemble de valeur numérique.

Les applications de la classification automatique d'images sont nombreuses et vont de l'analyse de documents à la médecine en passant par le domaine militaire. Ainsi on retrouve des applications dans le domaine médical comme la reconnaissance de cellules et de tumeurs, la reconnaissance d'écriture manuscrite pour les chèques et les codes postaux. Dans le domaine urbain, comme la reconnaissance de panneaux de signalisation la reconnaissance de piétons, la détection de véhicules, la reconnaissance de bâtiments pour aider à la localisation. Dans le domaine de la biométrie comme la reconnaissance de visage, d'empreintes, d'iris.

### I.3. Architecture d'un système de classification des images

Le point commun à toutes ces applications mentionnées ci-dessus est qu'elles nécessitent la mise en place d'une chaîne de traitement à partir des images disponibles composée de plusieurs étapes afin de fournir en sortie une décision (figure I.1).



**Figure I.1 :** Système de classification des images de textures.

Chaque étape de la mise en place d'un tel système de classification nécessite la recherche de méthodes appropriées pour une performance globale optimale à savoir la phase d'extraction de caractéristiques et la phase d'apprentissage. Typiquement, nous disposons de données images, dont les quelles il faut extraire des informations sous formes de vecteurs numériques. Cette phase d'extraction nous permet de travailler dans un espace numérique. Il s'agit ensuite d'élaborer dans la phase d'apprentissage, à partir de ces données initiales, une fonction de décision pour décider de l'appartenance d'une donnée nouvelle à l'une des classes en présence.

### I.4. Caractéristiques d'une image

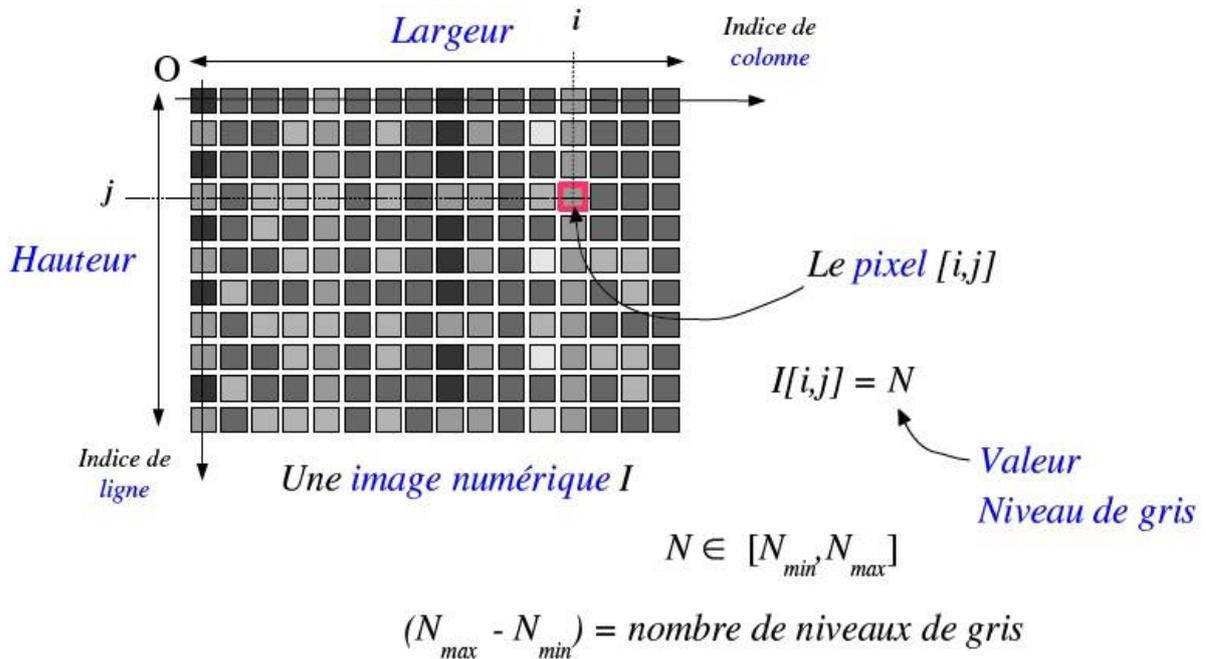
Une image est une représentation planaire d'une scène ou d'un objet situé en général dans un espace tridimensionnel, elle est issue du contact des rayons lumineux provenant des objets formants la scène avec un capteur (caméra, scanner, rayons X, ...). Il ne s'agit en réalité que d'une représentation spatiale de la lumière.

L'image est considérée comme un ensemble de points auquel est affectée une grandeur physique (luminance, couleur). Ces grandeurs peuvent être continues (image analogique) ou bien discrètes (images digitales). Mathématiquement, l'image représente une fonction continue IF, appelée fonction image, de deux variables spatiales représentée par  $IF(i, j)$  mesurant la nuance du niveau de gris de l'image aux coordonnées  $(i, j)$  [01].

La fonction Image peut se représenter sous la forme suivante :

$$IF : \mathbb{R}^2 \rightarrow \mathbb{R} \quad \text{Avec : } \mathbb{R} : \text{ensemble des réelles.}$$

$$(i, j) \rightarrow IF(i, j) \quad i \text{ et } j : \text{Deux variables réelles.}$$



**Figure I.2 :** Représentation des notions image et pixel.

L'image est un ensemble structuré d'information caractérisé par les paramètres suivants :

#### I.4.1. Pixel

Le pixel est l'abréviation du mot « Picture élément », est une unité de surface permettant de définir la base d'une image numérique. Il matérialise un point donné  $(i, j)$  du plan de l'image. L'information présentée par le pixel est le niveau de gris (ou la couleur) prélevée à l'emplacement correspondant dans l'image réelle. La différence entre image

monochrome et image couleur réside dans la quantité d'informations contenue dans chaque pixel, par exemple dans une image couleur (RVB : Rouge, Vert, Bleu) la valeur d'un pixel est représentée sur trois octets pour chaque couleur.

#### **I.4.2. Dimension et Résolution**

La dimension est la taille de l'image. Elle se présente sous forme d'une matrice dont les éléments sont des valeurs numériques qui représentent des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

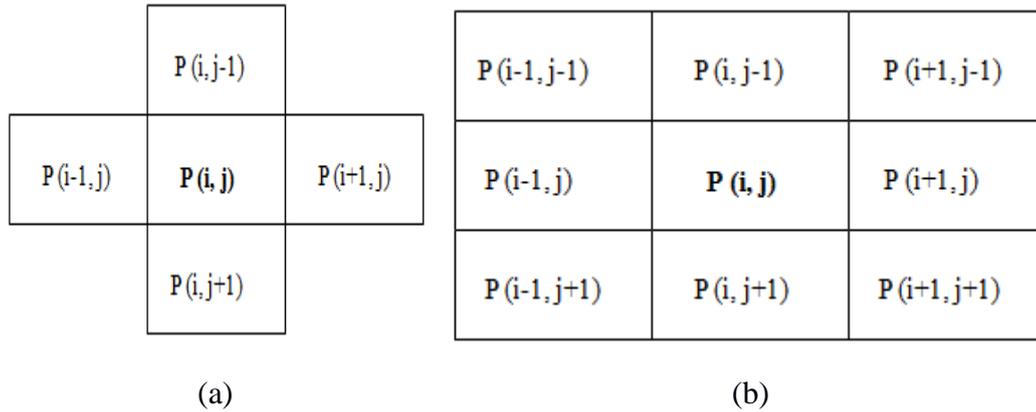
Par contre, la résolution est la clarté atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateur, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels horizontaux et verticaux sur un moniteur. Plus ce nombre est grand, plus la résolution est meilleure.

#### **I.4.3. Voisinage**

Le plan de l'image est divisé en termes de formes rectangulaires ou hexagonales permettant ainsi l'exploitation de la notion de voisinage (voir tableau I.1). Le voisinage d'un pixel est formé par l'ensemble des pixels qui se situent autour de ce même pixel. On définit aussi l'assiette comme étant l'ensemble de pixels définissant le voisinage pris en compte autour d'un pixel.

On distingue deux types de voisinage :

- Voisinage à 4 : On ne prend en considération que les pixels qui ont un côté commun avec le pixel considéré.
- Voisinage à 8 : On prend en compte tous les pixels qui ont au moins un point en liaison avec le pixel considéré.



**Tableau I. 1 :** (a) : Voisinage à 4, (b) : Voisinage à 8.

#### I.4.4. Niveau de gris

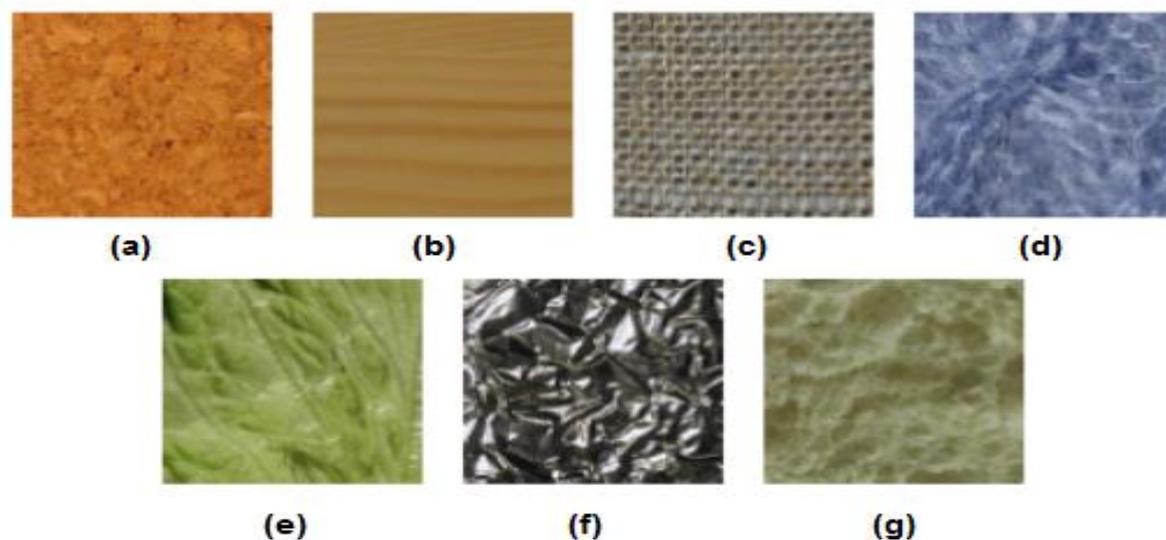
C'est la valeur d'intensité lumineuse d'un pixel. Cette valeur peut aller du noir (0) jusqu'au blanc (255) en passant par les nuances qui sont contenues dans l'intervalle [0, 255]. Elle correspond en fait à la quantité de la lumière réfléchiée. Pour 8 bits, on dispose de 256 niveaux de gris dont 40 sont reconnus à l'œil nue. Plus le nombre de bit est grand plus les niveaux sont nombreux et plus la représentation fiable [01].

### I.5. La texture

#### I.5.1. Définition

Une texture est une région de l'image, pour laquelle on peut définir une fenêtre de dimension minimale, telle qu'une observation au travers de celle-ci se traduit par une perception identique pour toutes les translations possibles de cette fenêtre à l'intérieur de la région considérée. Pour une texture, il existe toujours une telle fenêtre qui correspond à la résolution de la texture [02].

Une texture peut être décrite spatialement ou statistiquement, et d'une manière générale, on qualifiera une texture de fine ou grossière, plus ou moins homogène, rugueuse, plus ou moins régulière, dense, linéaire, isotrope ou directionnelle, tachetée, marbrée, etc. La figure I.3 illustre quelques-uns de ces qualificatifs, dont les significations peuvent d'ailleurs se recouper.



**Figure I.3 :** Exemples des images texturées : a) liège, b) bois, c) lin, d) laine, e) laitue, f) feuille d'aluminium et g) pain blanc [03].

### I.5.2. Pourquoi la texture ?

La grande majorité de la texture naturelle est facilement détectée, segmentée et reconnue par les humains. Le système visuel s'est développé au fil de millions d'années pour percevoir efficacement les textures, car elles contiennent des informations utiles sur une scène observée. Cependant, l'analyse automatique des textures par des algorithmes informatiques reste un problème épineux.

Le défi majeur résulte de la diversité et de la complexité des textures naturelles. Par exemple, si l'on considère uniquement les surfaces en bois, une très grande variété de textures avec différents types de bois, leur âge, leur état et leur coupe ainsi que leurs variations d'éclairage et d'acquisition d'images (par exemple, point de vue, orientation, bruit). En conséquence, les échantillons d'apprentissage peuvent être très différents des images de test, nécessitant une bonne généralisation des modèles de reconnaissance pour éviter les sur-ajustements. C'est aussi une source de grande variation dans les classes, ce qui rend difficile la définition de la règle de discrimination.

De plus, certaines approches sont bien conçues pour un type de textures (par exemple des motifs de texture réguliers, orientés, peu épais, petits ou grands, etc.), mais échouons pour d'autres types, comme expliqué dans la suite de ce chapitre. Développer une méthode pour analyser différents types de la texture avec une généralisation robuste et des invariances multiples est une tâche complexe.

L'analyse de texture est souvent employée comme un moyen pour finaliser la segmentation d'une image. En effet, dans le cas des images naturelles, la seule étude de la distribution des niveaux de gris est insuffisante pour caractériser les zones homogènes. C'est le cas pour les images multispectrales réalisées en télédétection ou les estimateurs de textures permettent de différencier un champ, d'une forêt, d'une ville.... C'est le cas aussi en imagerie médicale ou ces mêmes estimateurs permettent de faire la distinction entre tissus sains et tissus pathologiques (détection de lésions, dépistage de pathologies, segmentation en IRM...).

En compression d'images, les attributs texturaux permettent de représenter l'image par un nombre minimal de paramètres et de ce fait permettent une reconstitution de l'information avec un minimum d'erreur.

La caractérisation de texture à l'aide de paramètres pertinents permet également la restauration d'une partie dégradée ou manquante dans une image en la remplaçant par une version synthétique générée à partir du modèle textural élaboré. Dans le domaine de l'infographie ou de l'audiovisuel, la synthèse de texture découle naturellement de l'analyse et conduit à son utilisation pour le réalisme, l'art, le design....

En contrôle non destructif, l'analyse de texture est largement utilisée pour l'inspection des surfaces en contrôle de qualité (produits industriels, matériaux, produits alimentaires, détection de défauts...).

## **I.5. Les types de textures**

En terme de définition de la notion de texture, on trouve d'abord celle donnée par le dictionnaire, qui précise simplement qu'une texture est la reproduction spatiale d'un motif dimensions :

- ✓ La première concerne la description d'éléments de base ou primitives, à partir desquels est formée la texture.
- ✓ La deuxième est relative à la description des relations spatiales entre ces primitives.

On distingue en fait, selon l'élément de base et les règles de placement, 2 types de textures :

- **Les macrotextures (ou textures structurées)** pour lesquelles il est facile d'extraire visuellement le motif de base et les lois d'assemblage des primitives

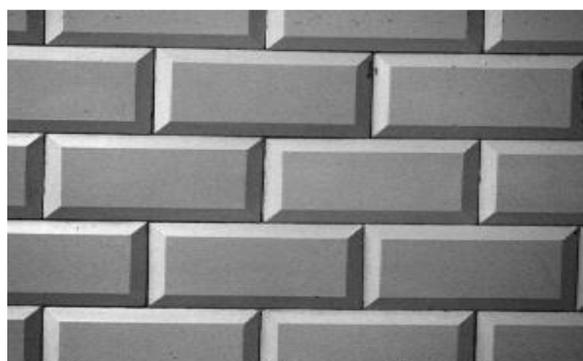
entre elles. Ces textures peuvent même présenter une certaine périodicité ou cyclostationnarité (processus aléatoire plaqué sur un processus périodique). Certains exemples sont représentatifs de ce type de textures, comme la texture d'un mur de brique, de certains tissus ou d'un grillage.

- **Les microtextures (ou textures aléatoires)** qui présentent un aspect plus chaotique et plus désorganisé, mais dont l'impression visuelle reste globalement homogène. Les différentes régions d'une image aérienne, les bois, les champs, etc., représentent des textures microscopiques.

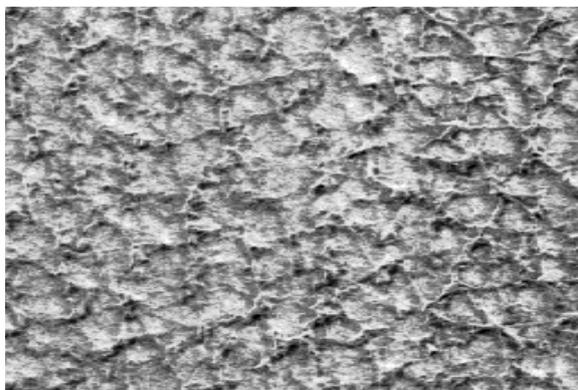
La figure I.4 donne quelques exemples de macro et microtextures.



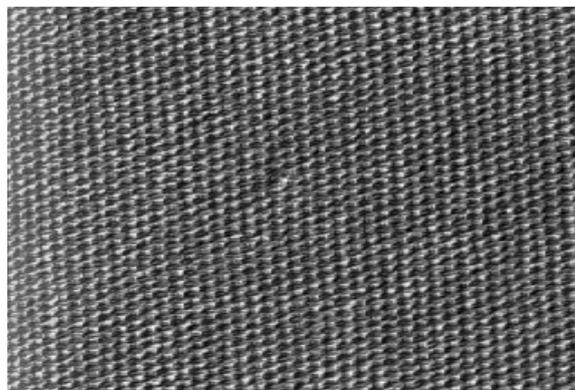
Rotin



Carrelage



Peau de porc



Tresse

**Figure I.4 :** Exemples de macro (1<sup>ère</sup> ligne) et microtextures (2<sup>ième</sup> ligne).

Mais, il est parfois difficile de classer une texture dans l'une ou l'autre des catégories (textures mixtes) d'autant que la résolution de l'image pose un problème supplémentaire, comme en témoigne la figure I.5 où du sable est photographié à 2 résolutions différentes ; la première image laisse à penser que le sable est une microtexture, la seconde, qu'il est plutôt une macrotexture.

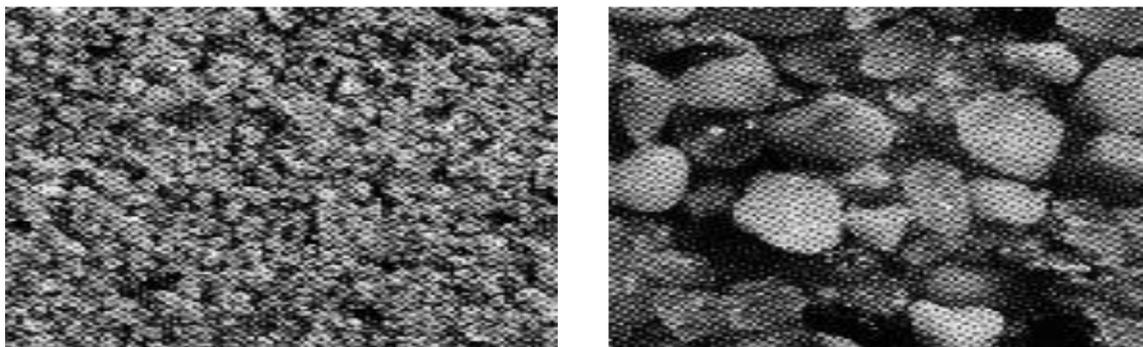


Figure I.5 : Exemples de sables à 2 résolutions différentes.

## I.6. Analyse de texture

### I.6.1. Les approches structurelles (macrotextures)

On considère que les textures sont constituées de motifs de base se répétant dans des directions de l'espace de façon quasi-régulière. Dans les images étudiées, on considère donc l'existence de primitives. Une description structurale d'une texture implique la recherche des motifs élémentaires, leur description, puis, la détermination des règles conditionnant leur position. Plusieurs approches peuvent être regroupées en un ensemble de méthodes heuristiques, dites méthodes structurelles classiques, pour lesquelles on cherche à détecter et caractériser des primitives, puis, à trouver des règles de placement, (méthodes « bottomup ») ou inversement (méthodes « top-down »). Elles s'appuient principalement sur le traitement du signal, la topologie et la géométrie.

L'avantage des méthodes « bottom-up » est que l'on peut utiliser les techniques classiques de segmentation, comme la croissance de région, les seuillages, la détection de contours, etc., pour isoler les primitives, considérées comme des ensembles de pixels ayant des propriétés d'homogénéité communes (en niveau de gris, par exemple) [03].

Ces propriétés (luminance, aire, taille, courbure, directionnalité, etc.) permettent la description d'un certain nombre de classes de primitives. Les règles de placement sont exprimées à l'aide de « vecteurs de régularité », de vecteurs de densités de primitives ou à partir des centres de gravité des classes.

D'autres approches, dites approches syntaxiques, font appel à la théorie des grammaires qui permet d'engendrer des formes en appliquant un ensemble de règles de

placements données à un petit nombre de symboles. Ces symboles peuvent être considérés comme des sous-primitives de base et doivent pouvoir reconstituer toute texture structurale. Mais, le formalisme de ces approches est souvent complexe. De plus, une texture peut être générée ou analysée par plusieurs grammaires. On limite donc leur utilisation à la génération de textures binaires ou à des applications très particulières.

Enfin, il existe une autre catégorie de méthodes structurales dites approches ensemblistes.

Ces approches sont principalement descriptives. Elles s'appuient sur des opérations de morphologie effectuées essentiellement sur des images binaires ; il est donc nécessaire d'appliquer des prétraitements (seuillage, détection de contours) sur les images originales.

La description des primitives sera essentiellement géométrique avec des propriétés comme la taille, la forme, etc. [03].

Ces méthodes structurales se distinguent des méthodes purement statistiques par le fait qu'elles reposent sur une définition explicite des primitives et de leurs règles de placement.

De ce fait, leur application est limitée à des textures simples et régulières ; une approche moins rigide d'analyse de texture structurale est souvent employée, appelée « structurale statistique ». Elle consiste à identifier les éléments de textures (textons), à mesurer leurs propriétés caractéristiques, puis, à décrire, statistiquement, leur distribution spatiale.

### **I.6.2. Approche statistique des microtextures**

De nombreuses études ont porté sur la perception visuelle des textures et ont montré que l'œil était sensible à leurs propriétés statistiques. La vision humaine opère en 2 temps : la vision préattentive qui, quasi-instantanément, perçoit un large champ visuel pour localiser grossièrement les régions de l'image ; et, la vision focale qui effectue des mesures d'organisation spatiale par un lent déplacement de l'ouverture focale et qui fait appel à l'interprétation cognitive (mémoire, connaissances...).

Ceci explique l'émergence et l'importance des méthodes statistiques qui se fondent sur l'hypothèse qu'une texture est une réalisation d'un processus stochastique

bidimensionnel possédant les propriétés de stationnarité et d'ergodicité. L'estimation des paramètres statistiques peut alors se faire sur une réalisation unique du processus (l'image de texture).

Les attributs statistiques sont alors des estimations de moments statistiques ou de densités de probabilité.

Julesz, [04] après avoir postulé que l'on ne pouvait discriminer 2 textures dont les probabilités conjointes jusqu'à l'ordre 2 sont identiques, a donné des contre-exemples à sa propre conjecture, sous forme de textures synthétisées dont les moments jusqu'à l'ordre 3 sont identiques et qui, pourtant, restent identifiables visuellement. Cependant, on a pu montrer que cette conjecture était une bonne approximation des performances visuelles humaines pour la plupart des problèmes de discrimination de textures naturelles.

D'autres études ont confirmé que l'œil ne prend pas en compte les dépendances entre points éloignés. La caractérisation se fait donc à un niveau très local, d'où l'importance de la notion de voisinage dans l'analyse de texture.

De manière générale, ces méthodes statistiques s'appuient sur une description des propriétés locales des textures, ce qui sous-entend que les relations de placement au sein des microtextures se situent à faible échelle. L'importance de la notion de voisinage est, encore ici, mise en évidence dans la compréhension du phénomène de texture.

Nombreuses sont les méthodes qui, dans cette catégorie, utilisent une approche reconnaissance des formes sur des vecteurs d'attributs (statistiques ou non) attachés à chaque pixel de l'image ; avec apprentissage ou sans connaissance a priori des classes existantes, ces méthodes donnent des résultats satisfaisants, si ce n'est qu'elles perdent une part importante de l'information texturale : sa composante spatiale. En effet, une texture est à la fois spatiale et statistique puisqu'elle met en place des relations de regroupements d'éléments de base de la texture.

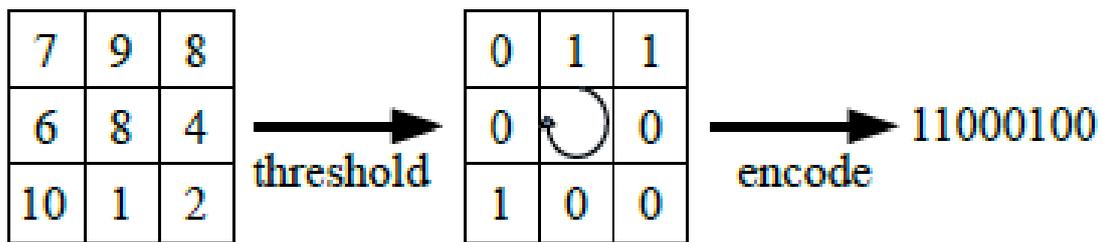
## **I.7. Analyse de texture par LBP (local binary pattern)**

### **I.7.1. Le LBP de base**

L'opérateur LBP a été proposé initialement par Ojala et al. [05] dans le but de caractériser la texture d'une image. Le calcul de la valeur LBP consiste pour chaque pixel à

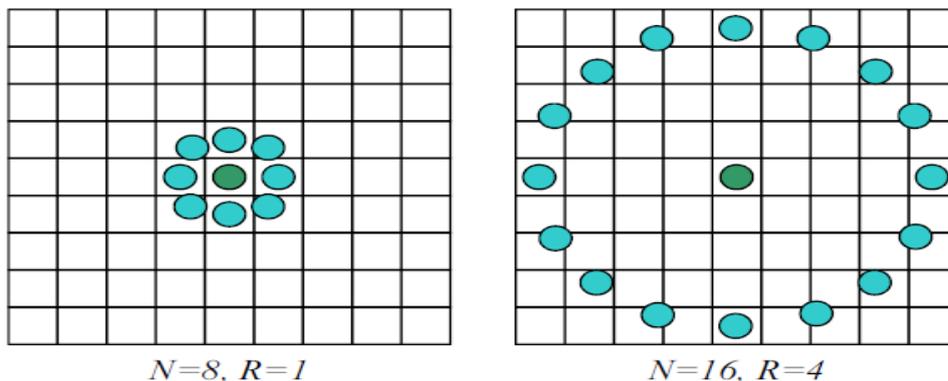
seuiller ses huit voisins directs avec un seuil dont la valeur est le niveau de gris du pixel courant.

Tous les voisins prendront alors une valeur 1 si leur valeur est supérieure ou égale au pixel courant et 0 si leur valeur est inférieure (Figure.I.6). Le code LBP du pixel courant est alors produit en concaténant ces 8 valeurs pour former un code binaire. On obtient donc, comme pour une image à niveaux de gris, une image des valeurs LBP contenant des pixels dont l'intensité se situe entre 0 et 255.



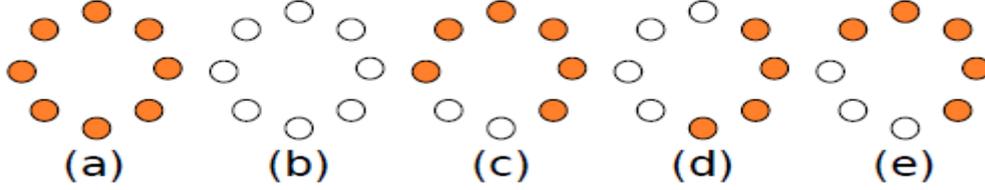
**Figure I.6 :** Opérateur LBP.

La technique LBP a été étendue ultérieurement en utilisant des voisinages de taille déférente. Dans ce cas, un cercle de rayon R autour du pixel central et Les valeurs des P points échantillonnés sur le bord de ce cercle sont prises et comparées avec la valeur du pixel central. Pour obtenir les valeurs des P points échantillonnés dans le voisinage pour tout rayon R, une interpolation est nécessaire. On adopte la notation (P, R) pour définir le voisinage de P points de rayon R d'un pixel. La (Figure.I.7), illustre deux voisinages pour des valeurs de R et P différentes.



**Figure I.7 :** Exemples de voisinages utilisés pour le calcul des LBP.

Les différentes valeurs possibles de motifs peuvent être assimilées à des micro-textons à différentes échelles, au sens où les primitives ainsi mesurées correspondent à des lignes, à des zones uniformes ou encore à des points, voir figure I.8.



**Figure I.8 :** Primitives extraites par les motifs binaires locaux. (a) et (b) correspondent à des taches respectivement claires et sombres, (c) est une fin de ligne, (d) une bordure et (e) est un coin.

Le LBP d'un pixel  $c$  d'une image  $I$ , pour un voisinage de rayon  $R$  avec  $P$  points, est défini comme :

$$\forall c \in \mathbb{N}^2, \forall P \in \mathbb{N}, \forall R \in \mathbb{R}, LBP_{P,R} = \sum_{p=0}^{P-1} s(I_R(p) - I(c)) * 2^p \quad (I.1)$$

Où

$$\forall x \in \mathbb{R}, s(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases}, \quad (I.2)$$

$$\forall (x_c, y_c) \in \mathbb{N}^2, \forall P \in \mathbb{N}, \forall p \in \{1..P\}, \quad (I.3)$$

$$I_R(p) = I\left(x_c + R \cos\left(2\pi \frac{p}{P}\right), y_c - R \sin\left(2\pi \frac{p}{P}\right)\right) \quad (I.4)$$

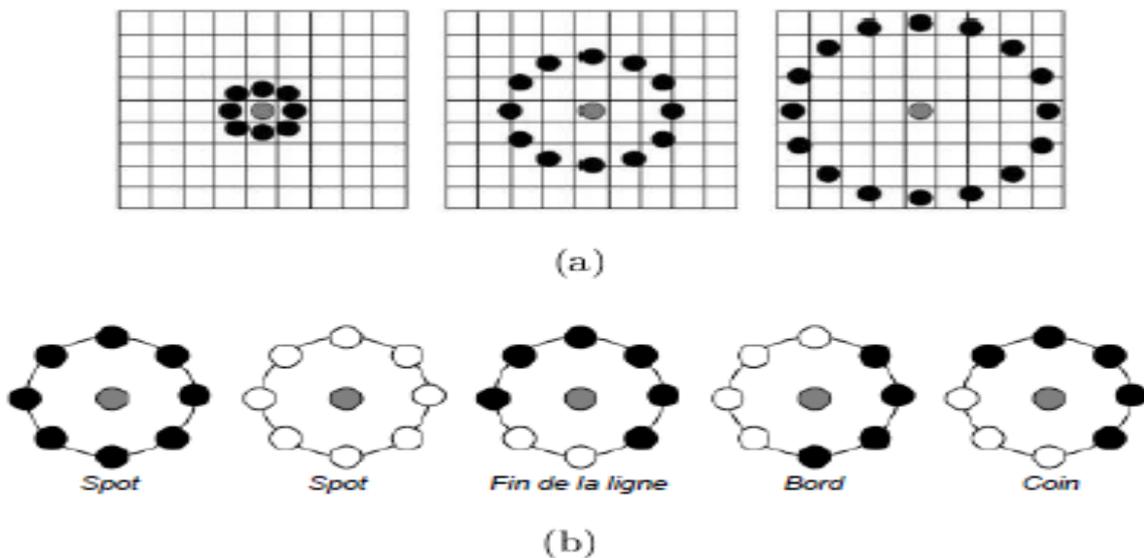
### I.7.2. LBP uniforme

L'extension suivante des LBP concerne la définition des motifs uniformes qui permettent de réduire la taille du vecteur de description et de rendre les LBP invariants par rotation. En effet, le nombre de motifs possibles augmente rapidement avec le nombre de voisins pris en considération. Par exemple, pour 16 voisins, l'histogramme aura une taille de  $2^{16} = 65536$  dimensions, ce qui est inutilisable dans une application réelle. [06] suggèrent de ne considérer que les motifs n'ayant qu'un faible nombre de transitions. Cette modification a été inspirée par le fait que certains motifs apparaissent plus souvent que d'autres et donc concentrent plus d'informations utiles. Le nombre de transitions  $U$  d'un LBP est défini comme :

$$U(LBP_{P,R}) = |s(I_R(p-1) - I(c)) - s(I_R(0) - I(c))| + \sum_{p=1}^{P-1} |s(I_R(p) - I(c)) - s(I_R(p-1) - I(c))| \quad (I.5)$$

Les motifs uniformes les plus utilisés, nommés uniformes 2, correspondent à ceux ayant deux transitions après seuillage ou moins (donc  $U \leq 2$ ). Par exemple, le label 00000111 (deux transitions) est uniforme 2 alors que 01101101 (six transitions) ne l'est pas. Les motifs non uniformes sont tous conservés sous le même label. La taille du vecteur de caractéristiques passe alors de  $2^P$  éléments à  $P(P-1) + 3$ . Par exemple, la description avec un voisinage de huit voisins contient normalement 256 labels différents, parmi lesquels 58 sont uniformes. Cela permet de réduire la taille du vecteur de description à 59 valeurs. Il a été montré par [07] que les motifs uniformes 2 comptabilisent généralement près de 90 % de l'information de texture contenue dans l'image.

L'utilisation d'un code LBP uniforme, noté  $LBP^{u2}$  à deux avantages. Le premier est le gain en mémoire et en temps calcul. Le deuxième est que  $LBP^{u2}$  permet de détecter uniquement les textures locales importantes, comme les spots, les fins de ligne, les bords et les coins (Figure.I.9, b), pour des exemples de ces textures particulières.

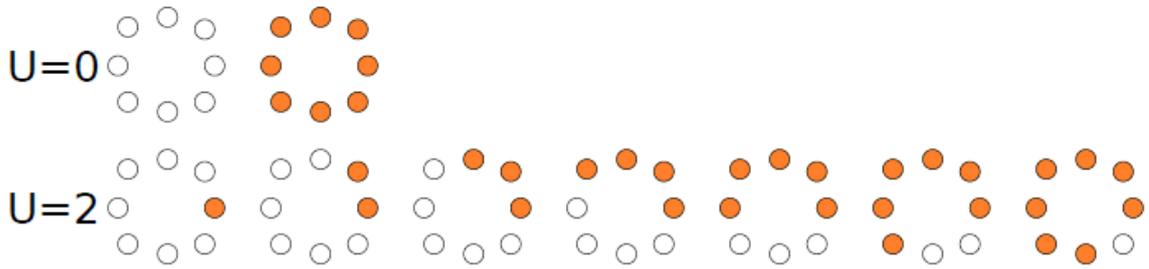


**Figure I.9 :** (a) : Trois voisinages pour des R et P différents, (b) : Textures particulières détectées par  $LBP^{u2}$ .

### I.7.3. LBP invariant par rotation

L'invariance en rotation des motifs est obtenue en mettant les motifs similaires à une rotation près dans un même label, diminuant encore la taille du vecteur de description à  $P + 2$  dimensions. La figure I.10 montre les neuf labels uniformes 2 et invariants en rotations obtenus dans le cas d'un rayon unitaire et de huit voisins. Les LBP uniformes 2 et invariants en rotation peuvent être définis comme :

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} S(I_R(p) - I(c)) & \text{si } U(LBP_{P,R}) \leq 2 \\ P + 1 & \text{sinon} \end{cases} \quad (I.6)$$



**Figure I.10 :** Motifs binaires locaux uniformes 2 et invariants en rotation.

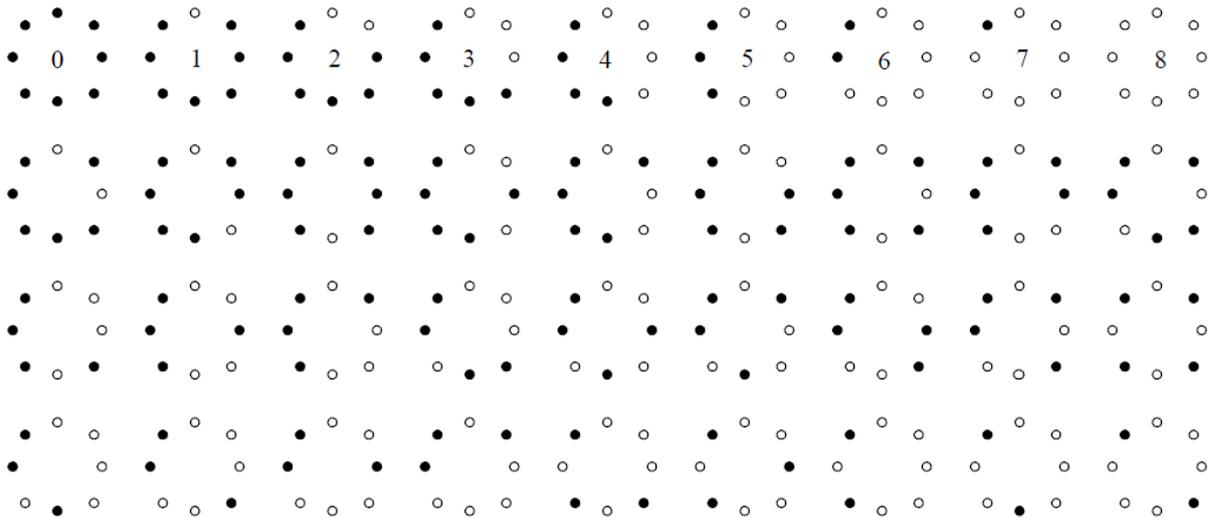
Il est également possible de définir des motifs invariants par rotation sans utiliser de motifs uniformes, ils sont définis comme :

$$LBP_{P,R}^{ri} = \min\{ROR(LBP_{P,R}, i) \mid i = 0, 1, \dots, P - 1\} \quad (I.7)$$

Où l'opérateur ROR ( $x, i$ ) effectue une permutation circulaire du nombre binaire  $x$ , ayant une longueur de  $P$  bits,  $i$  fois vers la droite, avec  $i < |P|$ .

Néanmoins, cette invariance en rotation est à prendre au niveau des motifs et non de l'image. En effet, elle ne prend pas en compte les artefacts issus de l'échantillonnage ni les variations de textures causées par les modifications des sources d'illumination.

La figure I.11 illustre les 36 modèles binaires locaux invariants en rotation uniques pouvant se produire. Dans le cas de  $P = 8$ , c'est-à-dire  $LBP_{8,R}^{ri}$  peut avoir 36 valeurs différentes. Par exemple, le motif n° 0 détecte des points lumineux, points noirs le n°8 et zones plates, et bords le n° 4.



**Figure I.11 :** Les 36 modèles binaires uniques invariants en rotation qui peuvent apparaître dans le cercle symétrique ensemble voisin de  $LBP_{8,R}^{ri}$ . Les cercles noir et blanc correspondent aux valeurs de bits 0 et 1 dans la sortie 8 bits de l'opérateur. La première rangée contient les neuf modèles « uniformes » et les nombre sà l'intérieur correspondent à leur unique  $LBP_{8,R}^{riu2}$  codes.

### I.8. Evaluation d'un système de classification

Évaluer les performances d'un système de classification est un enjeu de grande importance car ces performances peuvent être utilisées pour l'apprentissage en tant que tel ou pour optimiser les valeurs des hyper paramètres du classifieur. Pendant longtemps, le critère retenu pour évaluer ces performances a été le taux de bonne classification, c'est-à-dire le nombre d'éléments d'une base de test correctement classés. Le problème d'un tel critère est qu'il n'est pas adapté à des environnements mal dénis.

Dans de nombreuses situations, toutes les erreurs n'ont pas les mêmes conséquences. Certaines erreurs ont un coût plus important que d'autres, par exemple, pour les diagnostics médicaux.

Un mauvais diagnostic ou traitement peut, en effet, avoir différents coûts ou dangers selon le type d'erreur commise. La matrice de confusion, tableau I.2 permet la différenciation des erreurs selon chaque classe en vue d'évaluer un classifieur.

	Décision Positifs	Décision Négatifs	
Etiquette Positifs	<b>Vrai Positifs, TP</b>	<b>Faux Négatifs, FN</b>	$Pos^a$
Etiquette Négatifs	<b>Faux Positifs, FP</b>	<b>Vrai Négatifs, TN</b>	$Neg^b$
	$PPos^c$	$PNeg^d$	N

$a$  Nombre d'élément étiquetés positifs dans la base.

$b$  Nombre d'élément étiquetés négatifs dans la base.

$c$  Nombre d'élément classés positifs.

$d$  Nombre d'élément classés négatifs.

**Tableau I.2** : Matrice de confusion.

En particulier, si la matrice de confusion est diagonale, le classifieur est parfait. Notons que la matrice de confusion est aussi généralisable lorsqu'il y a plus de 2 classes à prédire [08].

Il est possible de calculer plusieurs indicateurs résumant la matrice de confusion. Par exemple si nous souhaitons rendre compte de la qualité de la prédiction sur la classe 1, on définit :

Le taux de vrais positifs (" True positive rate ") :

$$tpr = \frac{TP}{Pos} = \frac{TP}{TP+FN} \quad (I.8)$$

Le taux de vrais négatifs (" True negative rate ") :

$$tnr = \frac{TN}{Neg} = \frac{TN}{TN+FN} \quad (I.9)$$

Le taux de faux positifs (" False positive rate ") :

$$fpr = \frac{FP}{Neg} = \frac{FP}{FP+TN} \quad (I.10)$$

Le taux de faux négatifs (" False negative rate ") :

$$fnr = \frac{FN}{Pos} = \frac{FN}{FN+TN} \quad (I.11)$$

Le taux de bonne classification ou l'exactitude (" accuracy ") :

$$acc = tbc = Pos * tpr + Neg * ( 1 - fpr ) \quad (I.12)$$

La précision :

$$prec = \frac{TP}{P_{Pos}} = \frac{TP}{TP + FP} \quad (I.13)$$

Le rappel (" recall ") :

$$rec = tpr = \frac{TP}{Pos} = \frac{TP}{TP + FN} \quad (I.14)$$

### I.8.1 Courbe ROC (Received Operating Characteristic)

Dans le cas d'un classifieur binaire, il est possible de visualiser les performances du classifieur sur ce que l'on appelle une courbe ROC (figure I.12). La courbe ROC est une représentation du taux de vrais positifs (tpr) en fonction du taux de faux positifs (fpr).

Son intérêt est de s'affranchir de la taille des données de test dans le cas où les données sont dés équilibrées. Cette méthode robuste permet de s'affranchir de la connaissance des coûts de classification et de la distribution des classes [09].

Cette représentation met en avant un nouvel indicateur qui est l'aire sous la courbe. Plus elle se rapproche de 1, plus le classifieur est performant [08].

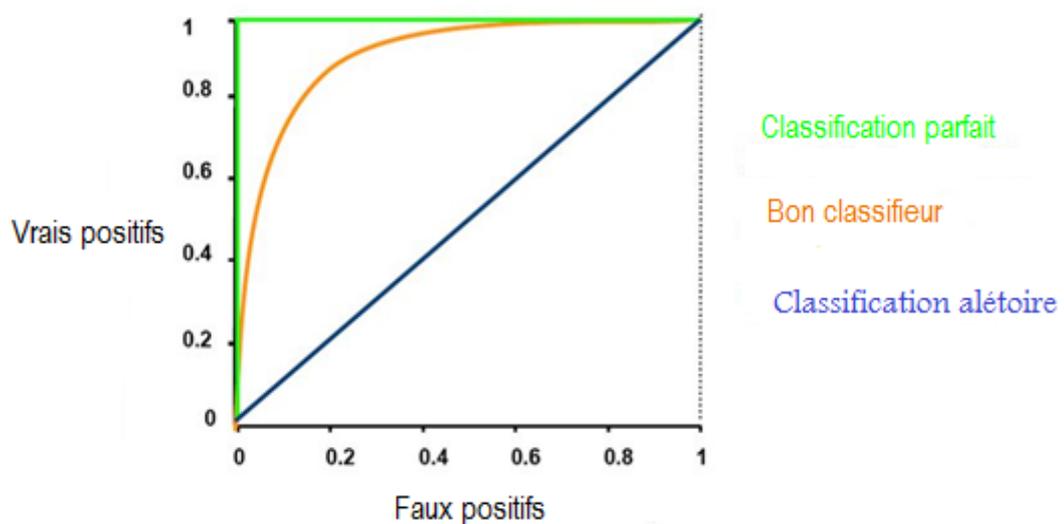


Figure.I.12 : Courbe ROC.

## **I.9. Conclusion**

Dans ce chapitre, après avoir introduit quelques aspects sur la classification des images, nous avons abordé la notion de la texture dans les images, ses différents types et les méthodes d'extraction des caractéristiques liées à la texture. Nous avons ensuite présenté en détail la méthode LBP avec ses différentes variantes qui seront employées lors de la phase d'extraction des caractéristiques dans le système de classification des images étudié. Les descripteurs LBP seront l'entrée du classifieur à réseaux de neurones que nous verrons au chapitre suivant.

# *Chapitre II*

*Les réseaux neurones artificiels*

## II.1. Introduction

Les réseaux neuronaux artificiels (RNA) sont parmi les techniques récemment utilisées dans la reconnaissance et la classification des objets. Ils peuvent modéliser et approximer des systèmes complexes et difficiles, en utilisant une modélisation conventionnelle mathématique.

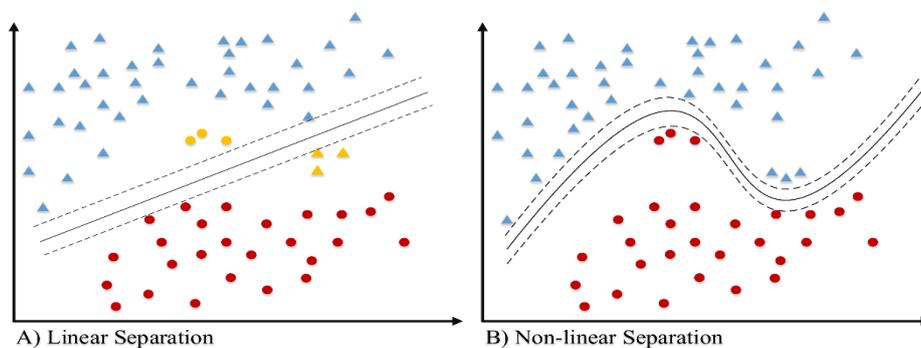
La classification automatique des images consiste à attribuer automatiquement une classe à une image à l'aide d'un système de classification. On retrouve ainsi la classification d'objets, de scènes, de textures, la reconnaissance de visages, d'empreintes digitale et de caractères.

Dans ce chapitre, une description générale des réseaux neuronaux artificiels et des aspects liés à leur fonctionnement sera entamée. Nous présenterons des généralités sur le réseau neuronal, telles que la définition de ses structures, les fonctions d'activation et les différentes architectures de ces réseaux.

## II.2. Notion de classificateur

Soit la représentation d'un objet quelconque au moyen d'un vecteur de caractéristiques  $X=[x_1 \ x_2 \ \dots \ x_d]^T$ . Tous les vecteurs qui représentent l'ensemble des objets peuvent être positionnés dans l'espace Euclidien  $R^d$ , où ils correspondent chacun à un point. Ceux-ci peuvent alors être regroupés en amas, chacun de ces amas étant associé à une classe particulière.

Le rôle d'un classificateur est de déterminer, parmi un ensemble fini de classes, à laquelle appartient un objet donné. Un classificateur doit être capable de modéliser au mieux les frontières qui séparent les classes les unes des autres (figure II.1) [10].



**Figure II.1** : Classification et modélisation des frontières entre les classes.

## II.3. Généralité sur les réseaux de neurones artificiels

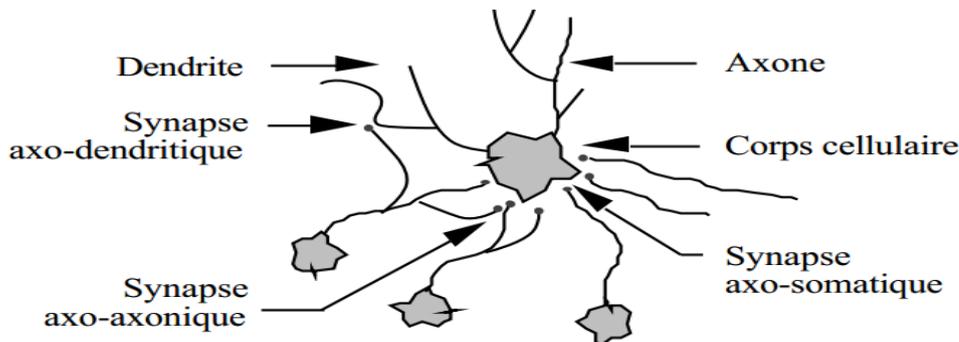
### II.3.1. Historique

Les réseaux de neurones sont des modèles théoriques de traitement de l'information inspirés des observations relatives au fonctionnement des neurones biologiques et du cortex cérébral. Le domaine des réseaux de neurones n'est pas nouveau car il a son origine dans des travaux conduits durant les années 40 (modèle de Hebb pour l'évolution des connexions synaptiques [11]). Ces travaux conduisirent au modèle du perceptron dans les années 60 (modèle qui a principalement été appliqué à la reconnaissance de caractères [12]). Mais ce n'est qu'à partir de 1986 que la recherche dans ce domaine a connu une expansion importante du fait de la publication de modèles de réseaux et d'algorithmes d'apprentissage suffisamment efficaces pour résoudre des problèmes réalistes et complexes [13].

### II.3.2. Neurones biologiques

Le neurone est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire ramifie pour former ce que l'on nomme les dendrites. Celles-ci sont parfois si nombreuses que l'on parle alors de chevelure dendritique ou d'arborisation dendritique.

C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma, corps du neurone. L'information traitée par le neurone chemine ensuite 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 dizaines d'Angströms ( $10^{-9}\text{m}$ ) entre l'axone du neurone afférent et les dendrites (on dit une dendrite) du neurone efférent. La jonction entre deux neurones est appelée la synapse (figure.II.2) [14].



**Figure II.2 :** Un neurone avec son arborisation dendritique.

## II.4. Les réseaux de neurones artificiels

### II.4.1. Modèle mathématique

Les réseaux de neurones proposent une simulation du fonctionnement de la cellule nerveuse à l'aide d'un automate : le neurone formel. Les réseaux neuronaux sont constitués d'un ensemble de neurones (nœuds) connectés entre eux par des liens qui permettent de propager les signaux de neurone à neurone.

Grâce à leur capacité d'apprentissage, les réseaux neuronaux permettent de découvrir des relations complexes non-linéaires entre un grand nombre de variables, sans intervention externe. De ce fait, ils sont largement utilisés dans de nombreux problèmes de classification (ciblage marketing, reconnaissance de formes, traitement de signal, ...).

Les réseaux de neurones sont des outils principalement utilisés pour les tâches de classification : classiquement, on soumet un vecteur  $p$  au réseau de neurones qui retourne en sortie un vecteur  $a$  correspondant à une décision de classification de  $p$  dans une classe donnée (figure.II.3). L'efficacité des réseaux de neurones pour la classification provient de leur capacité à approximer n'importe quelle fonction de l'espace des entrées dans celui des sorties, pour peu l'architecture adoptée soit suffisamment complexe. Dans ce cas, on peut considérer qu'un réseau de neurones donné modélise une famille paramétrée de fonctions dont on va chercher à déterminer les paramètres optimaux pour résoudre le problème considéré. La détermination de ces paramètres se fait grâce à des techniques d'apprentissage [14].

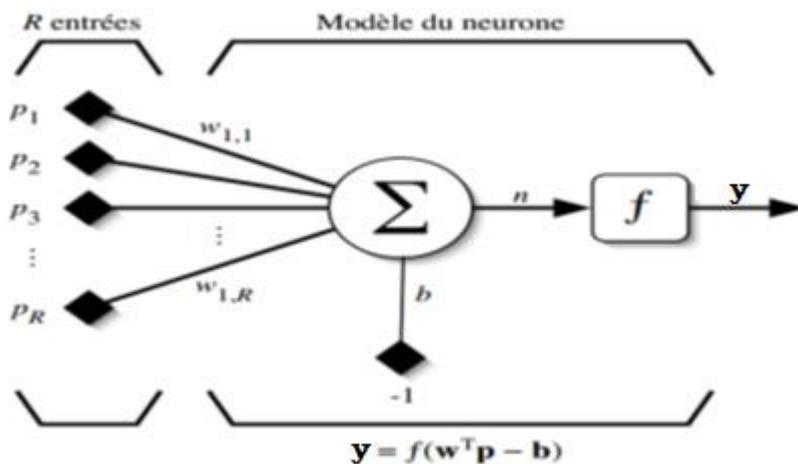
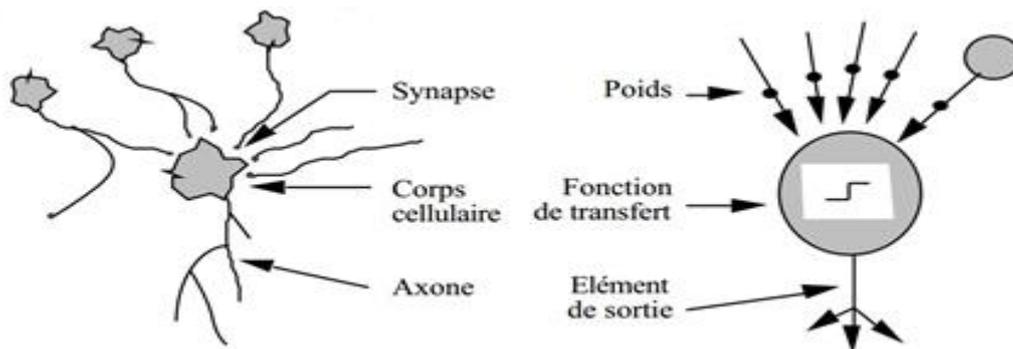


Figure II.3 : Modèle d'un neurone artificiel.

On pourra résumer une modélisation de tel neurone par le tableau.II.1 et la figure.II.4, qui nous permettra de voir clairement le passage du neurone biologique vers le neurone formel.

Neurone biologique	Neurone artificiel
Synapses	Poids de connexions
Axones	Signal de sortie
Dendrite	Signal d'entrée
Somma	Fonction d'activation

**Tableau II.1** : passage du neurone biologique vers le neurone formel.



**Figure II.4** : Passage du neurone biologique vers le neurone formel.

#### II.4.2. Fonctions d'activation

La fonction d'activation (fonction de transfert) joue un rôle très important dans le comportement des neurones, elle contient un paramètre sur la somme des entrées pondérées en plus du seuil d'activation. Il en existe plusieurs types dont la nature est déterminée en fonction du réseau, le tableau II.2 résume ces différents types, les trois les plus utilisées sont les fonctions seuil, linéaire et sigmoïde [14].

#### II.4.3. Connectivité

La connectivité des réseaux c'est à dire la manière dont les neurones sont connectés peut-être totale (tous les neurones sont connectés entre eux) ou par couche (les

neurones d'une couche ne sont connectés qu'aux couches suivantes). Il existe des réseaux monocouches ou multicouches (multi layer networks).

Nom de la fonction	Relation d'entrée/sortie	Icône	Nom Matlab
seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$		hardlim
seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$		hardlims
linéaire	$a = n$		purelin
linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$		satlin
linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$		satlins
linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $n \geq 0$		poslin
sigmoïde	$a = \frac{1}{1 + \exp^{-n}}$		logsig
tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
compétitive	$a = 1$ si $n$ maximum $a = 0$ autrement		compet

Tableau II.2 : Fonctions de transfert [14].

## II.5. Modèles de réseaux de neurones

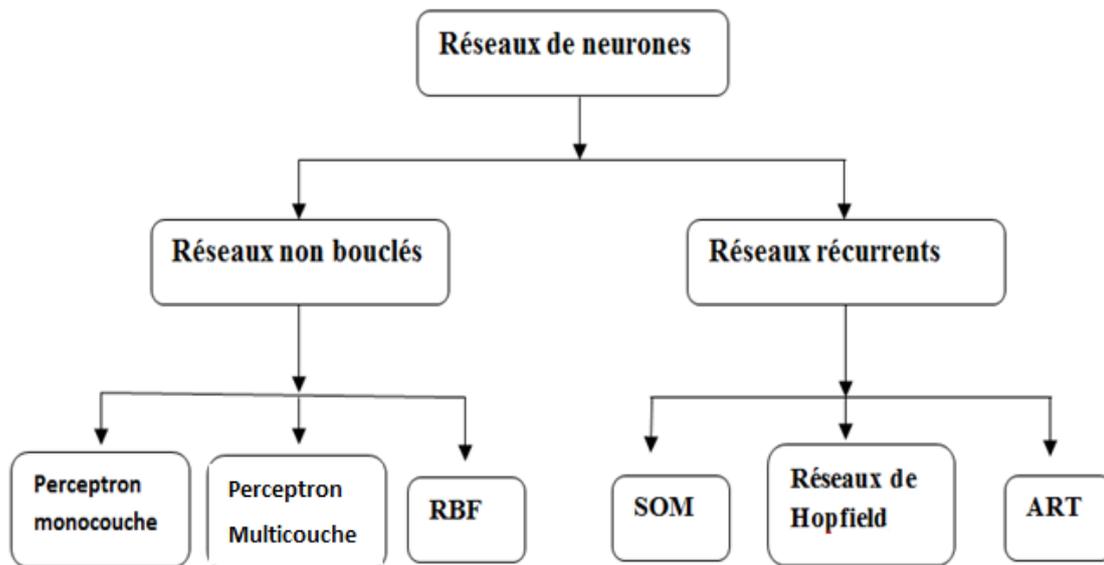
Selon la topologie de connexion des neurones, on peut les classer en deux grandes catégories : réseaux non bouclés (statique ou feedforward) et réseaux bouclés (dynamique, feedback ou récurrent). (Voir figure.II.5)

Un réseau de neurones non bouclé réalise une fonction (ou plusieurs) algébrique sur ses entrées par composition des fonctions réalisées par chacun de ses neurones. Dans un tel réseau le flux d'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 » et les autres qui effectuent des calculs intermédiaires sont des « neurones cachés » [15].

Il existe deux types de réseaux de neurones : les réseaux complètement connectés et les réseaux à couche.

L'architecture la plus générale pour un réseau de neurones est le réseau bouclé (Feedback), dont le graphe des connexions est cyclique : lorsqu'on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ (un tel chemin est appelé cycle). La sortie d'un neurone du réseau peut donc être fonction d'elle-même [15]. Les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche.

Les réseaux de neurones les plus utilisés dans la littérature sont représentés sur la figure.II.5.



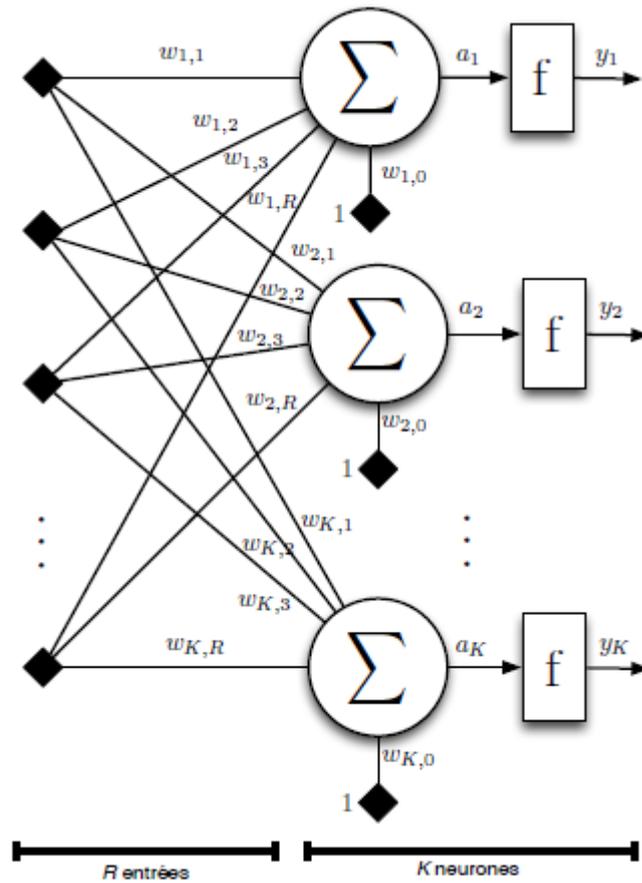
**Figure II.5 :** différents types de réseaux de neurones.

### II.5.1. Réseau de neurone monocouche (Perceptron)

Un réseau de neurones est une structure de réseau constituée d'un nombre de nœuds interconnectés par des liaisons directionnelles. Chaque nœud représente une unité de traitement et les liaisons représentent les relations causales entre les nœuds [16].

Le réseau le plus simple est celui monocouche appelé le perceptron et caractérisé de la manière suivante (figure II.6) :

- Il possède R informations en entrée ;
- Il est composé de K neurones, que l'on représente généralement alignés verticalement. Chacun peut en théorie avoir une fonction d'activation différente. En pratique, ce n'est généralement pas le cas ;
- Chacun des K neurones est connecté aux R informations d'entrée.



**Figure II.6 :** Perceptron monocouche.

Le réseau de neurones possède ainsi  $R$  informations en entrée et  $K$  sorties, chaque neurone renvoyant sa sortie. Une utilisation courante est que chaque neurone de la couche représente une classe. Pour un exemple  $X$  donné, on obtient la classe de cet exemple en prenant la plus grande des  $K$  sorties.

### II.5.2. Perceptron multicouche

Le perceptron multicouche (Multi Layer Perceptron MLP) est un réseau comportant plusieurs couches, chaque neurone d'une couche étant totalement connecté aux neurones de la couche suivante. Chaque neurone est un automate linéaire généralisé dont la fonction de transfert est supposée sigmoïdale (Figure II.7). Une information circule de la couche d'entrée vers la couche de sortie uniquement ; il s'agit donc d'un réseau de type feedforward. Chaque couche est constituée d'un nombre variable de neurones, les neurones de la couche de sortie correspondant toujours aux sorties du système [17].

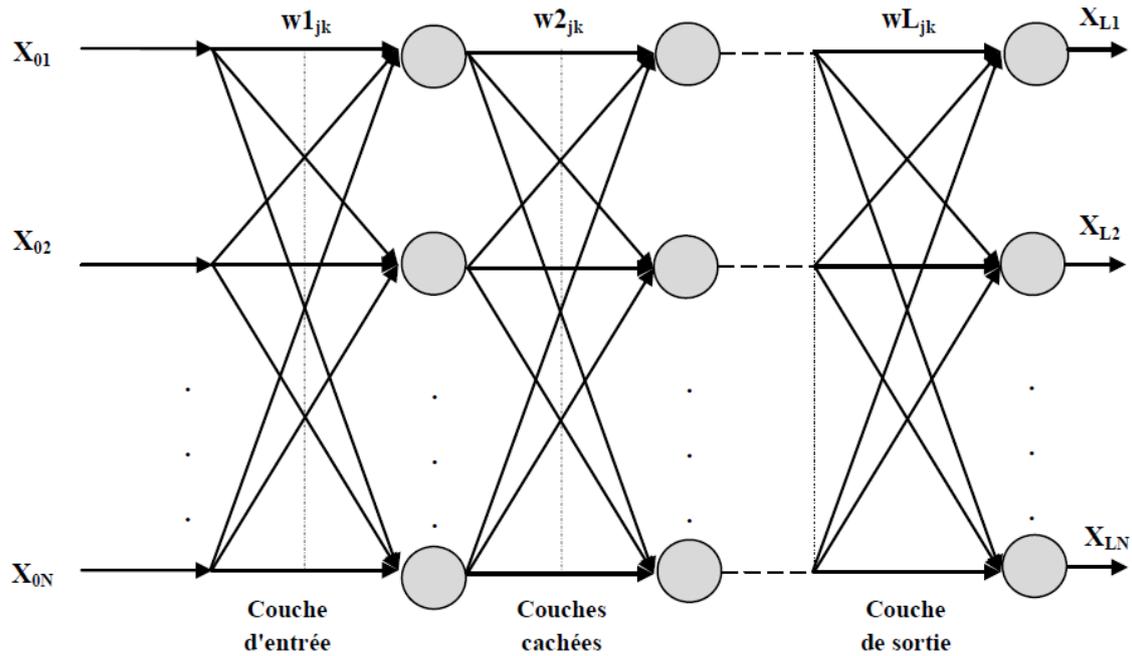


Figure II.7 : Perceptron multicouche.

### II.5.3. Les réseaux RBF

Le réseau RBF est un réseau de neurones supervisé. Il s'agit d'une spécialisation d'un PMC. Un RBF est constitué uniquement de 3 couches (voir figure II.8).

- La couche d'entrée : elle retransmet les entrées sans distorsion.
- La couche RBF : couche cachée qui contient les neurones RBF.
- La couche de sortie : simple couche qui contient une fonction linéaire.

Chaque neurone RBF contient une gaussienne qui est centrée sur un point de l'espace d'entrée. Pour une entrée donnée, la sortie du neurone RBF est la hauteur de la gaussienne en ce point. La fonction gaussienne permet aux neurones de ne répondre qu'à une petite région de l'espace d'entrée, région sur laquelle la gaussienne est centrée. Donc il y a quatre paramètres principaux à régler dans un réseau RBF :

- Le nombre de neurones RBF (nombre de neurones dans l'unique couche cachée).
- La position des centres des gaussiennes de chacun des neurones.
- La largeur de ces gaussiennes.

Le poids des connexions entre les neurones RBF et le(s) neurone(s) de sortie. Toute modification d'un de ces paramètres entraîne directement un changement du comportement du réseau [18].

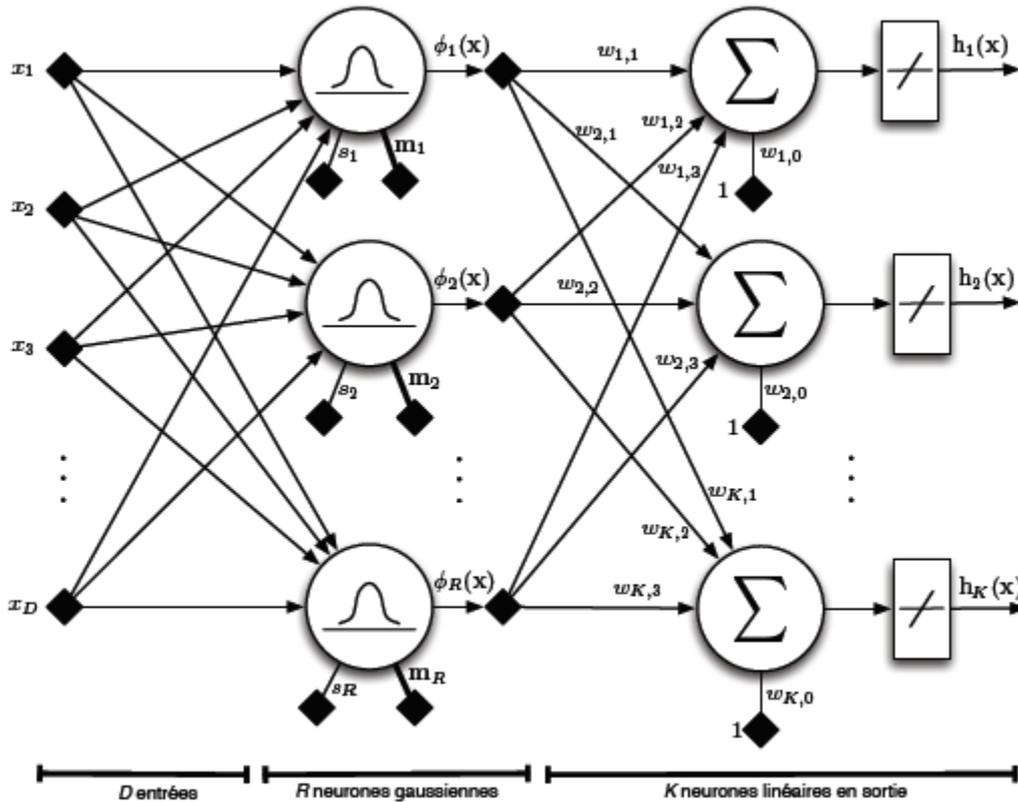


Figure II.8 : schéma d'un RBF.

#### II.5.4. Réseaux de Hopfield

Il s'agit d'un réseau récurrent constitué de neurones à deux états (-1 et 1, ou 0 et 1), dont la loi d'apprentissage est la règle de Hebb (1949), qui veut qu'une synapse améliore son activité si et seulement si l'activité de ses deux neurones est corrélée (c'est-à-dire que le poids d'une connexion entre deux neurones augmente quand les deux neurones sont activés au même temps) [19].

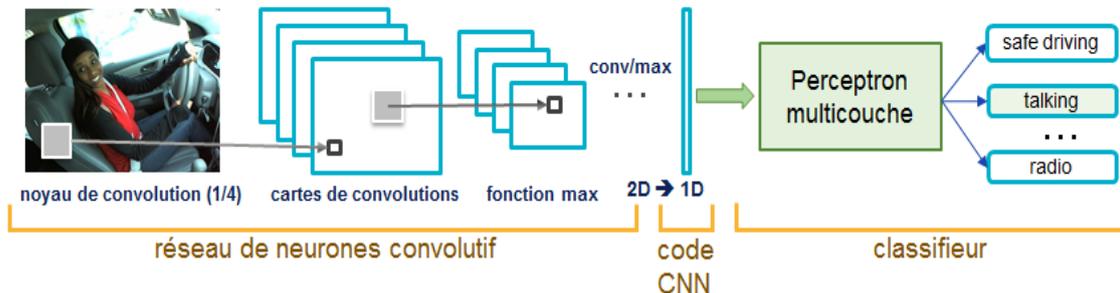
#### II.5.5. Réseaux de Kohonen (Self Organized Map)

Contrairement aux réseaux de Hopfield où les neurones sont modélisés de la façon la plus simple possible, on recherche ici un modèle de neurone plus proche de la réalité. Ces réseaux sont inspirés des observations biologiques du fonctionnement des systèmes nerveux de perception des mammifères. Une loi de Hebb modifiée (tenant compte de l'oubli) est utilisée pour l'apprentissage. La connexion est renforcée dans le cas où les neurones reliés ont une activité simultanée et diminuée dans le cas contraire (alors qu'il ne se passait précédemment rien dans ce cas). Tous ces réseaux ont des applications dans la classification, le traitement d'image, l'aide à la décision et l'optimisation [20].

## II.5.6. Réseaux Neurones Convolutionnels

Les réseaux de neurones convolutifs sont à ce jour les modèles les plus performants pour classer des images. Désignés par l'acronyme CNN, de l'anglais Convolutional Neural Network, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a 2 dimensions pour une image en niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu].

La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN [21] (voir figure II.9)



**Figure II. 9 :** Architecture standard d'un réseau de neurone convolutionnel.

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron multicouche). Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image.

La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories.

## II.6. Apprentissage d'un réseau de neurones

L'apprentissage est la propriété la plus intéressante des réseaux neuronaux. Une caractéristique des réseaux de neurones est leur capacité à apprendre « Par exemple à reconnaître une lettre, un son... ». Mais cette connaissance n'est pas acquise dès le départ.

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

Durant cette phase de fonctionnement, le réseau adapte sa structure « Le plus souvent, les poids des connexions » afin de fournir sur ses neurones de sortie les valeurs désirées. Cet apprentissage nécessite des exemples désignés aussi sous l'appellation d'échantillon d'apprentissage ainsi qu'un algorithme d'apprentissage.

Après initialisation des poids du réseau « En général des valeurs aléatoires », il y a présentation des exemples au réseau et calcul des sorties correspondantes. Une valeur d'erreur ou de correction est calculée et une correction des poids est appliquée.

Le type d'apprentissage est déterminé par la manière dont les changements de paramètres auront lieu. Il existe deux grandes stratégies d'apprentissage.

### II.6.1. Apprentissage supervisé

Dans ce cas, la connaissance à priori de la sortie désirée est nécessaire. On présente au réseau le vecteur d'entrée puis sa sortie est calculée et comparée à la sortie désirée, ensuite les poids sont ajustés de façon à réduire l'écart entre elles. Cette procédure est répétée jusqu'à ce qu'un critère de performance soit satisfait. La figure.II.10 illustre le processus d'apprentissage supervisé [22].

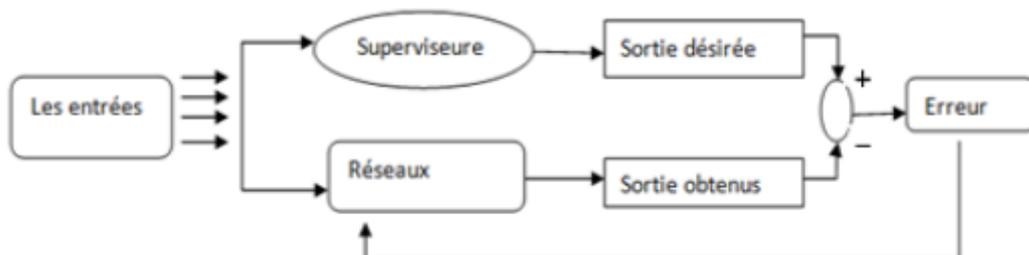


Figure II.10 : Apprentissage supervisé.

## II.6.2. Apprentissage non supervisé

L'apprentissage non supervisé implique la fourniture d'un réseau autonome avec un nombre suffisant d'exemples contenant des répétitions (c'est-à-dire la redondance) de sorte qu'il efface automatiquement les modèles. Ces réseaux sont souvent appelés auto-organiseurs ou apprentissage compétitif.

Dans l'apprentissage non supervisé, les données ne contiennent pas d'informations sur un résultat souhaité, il n'y a pas de superviseur ou d'expert humain. Il s'agit de déterminer les paramètres du réseau neuronal selon un critère défini.

Dans ce cas, les exemples présentés à l'entrée entraînent une auto-adaptation du réseau pour produire des valeurs de sortie proches en réponse à des valeurs d'entrée similaires. Ce type d'apprentissage a souvent moins de complexité dans le calcul par rapport à l'apprentissage supervisé bien sûr, l'architecture réseau, préalablement définie par l'utilisateur, est une forme de non supervision.

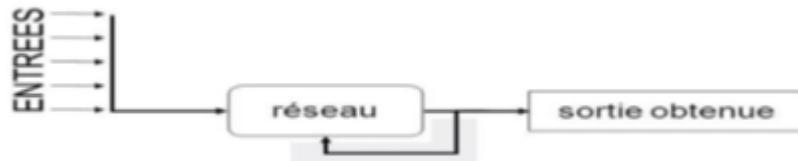


Figure II.11 : Apprentissage non supervisé.

## II.7. Apprentissage du perceptron multicouche

Pour la couche de sortie, on peut appliquer l'apprentissage du perceptron, mais comment modifier les poids pour les connexions qui ne sont pas en relation avec un neurone de sortie ? Le problème est l'obtention d'une estimation de la valeur désirée pour chaque neurone de la couche cachée. Ils utilisent pour modifier leurs poids, un algorithme de rétro-propagation du gradient (back-propagation). Il s'agit de minimiser l'erreur quadratique.

### II.7.1. Algorithme de rétro propagation du gradient

Le réseau MLP contient  $N_0$  entrées,  $L-1$  couche cachées de  $N_i$  neurones chacune ( $1 < i < L-1$ ) et une couche de sortie contenant  $N_L$  neurones. [23] L'indice de la couche est  $i$ ,

$X_{ik}$  est la sortie du neurone K de la couche i ,  $W_{ijk}$  est le poids reliant la sortie  $x_{i-1j}$  au neurone K de la couche i ,  $N_i$  est le nombre de neurones dans la couche i (voir figure II.7). Avec ces notions, la sortie  $X_{ik}$  du neurone (i ,k )sera donnée par :

$$y_{ik} = \sum_{j=1}^{N(i-1)} W_{ijk} x_{i-1j} + b_{ik} \quad (\text{II.1})$$

$$X_{ik} = f(y_{ik}) \quad (\text{II.2})$$

C'est un algorithme supervisé dont l'objectif est d'ajuster les poids du réseau MLP de façon à minimiser une fonction de coût différentiable, telle que l'erreur quadratique entre la sortie du réseau et la sortie désirée :

$$E(n) = \|d(n) - Y_L(n)\|^2 \quad (\text{II.3})$$

Où  $Y_L(n)$  est la sortie du réseau au temps  $n$  et  $d(n)$  la sortie désirée. L'algorithme de BP assure la descente du gradient sur le critère d'erreur pour atteindre le minimum. Le gradient  $E(n)$  d'est calculé pour tous les poids de la manière suivante :

$$\frac{\partial E(n)}{\partial w_{ijk}(n)} = \frac{\partial E(n)}{\partial y_{ik}} \cdot \frac{\partial y_{ik}}{\partial w_{ijk}} = \frac{\partial E(n)}{\partial y_{ik}} \cdot x_{i-1j} \quad (\text{II.4})$$

Dans le cas de la couche de sortie ( $i = L$ ) on évalue le terme d'erreur de sortie  $\delta_{Lk}$  comme suite :

$$\delta_{lk} = \frac{\partial E(n)}{\partial y_{lk}} = 2f'(y_{lk})(d_k - x_{L,k}) \quad (\text{II.5})$$

Ou  $f'()$  est la dérivée de la fonction d'activation :

$$f'(x) = \frac{df(x)}{dx} \quad (\text{II.6})$$

Pour les couches cachées, le terme  $\delta_{ik}$  d'erreur du neurone ( $i, k$ ) est donné par :

$$\delta_{ik} = f'(y_{ik}) \sum_{j=1}^{N(j+1)} \delta_{i+1j} w_{i+1kj} \quad (\text{II.7})$$

La modification des poids et biais est obtenue selon les équations suivantes :

$$w_{ijk}(n+1) = w_{ijk}(n) + \eta \delta_{ik} x_{i-1j} + \Omega(w_{ijk}(n) - w_{ijk}(n-1)) \quad (\text{II.8})$$

$$b_{ik}(n+1) = b_{ik}(n) + \eta \delta_{ik} \quad (\text{II.9})$$

$\eta$  est le pas d'apprentissage et  $\Omega$  le terme d'inertie (momentum).

### II.7.2. Les paramètres d'apprentissage

Tous les modèles de réseaux de neurones requièrent un apprentissage. Plusieurs types d'apprentissages peuvent être adaptés à un même type de réseau de neurones. Les critères de choix sont souvent la rapidité de convergence ou les performances de généralisation. L'efficacité d'apprentissage dépend de plusieurs paramètres :

- **Taux d'apprentissage  $\eta$**

Ce paramètre détermine la vitesse de convergence. Si la valeur de démarrage de  $\eta$  est grande, alors on aura un apprentissage très rapide mais au prix de la création d'oscillations dans l'erreur totale moyenne qui empêcheront l'algorithme de converger vers le minimum désiré. Le réseau devient instable. Dans la plupart des cas, si la fonction d'erreur possède plusieurs minimums locaux, le réseau subira un blocage dans l'un d'eux. Toutes ces conditions nous obligent à commencer l'apprentissage avec une petite valeur de  $\eta$ , si on veut atteindre un minimum global, même si l'apprentissage est long.

- **Momentum  $\Omega$**

Ce paramètre permet l'introduction de l'ancien poids dans le calcul de sa nouvelle valeur. Le momentum empêche l'algorithme de rester bloqué dans un minimum local et il augmente encore la vitesse de convergence. La valeur de  $\Omega$  est toujours inférieure à 1 (valeur typique  $\Omega = 0.9$ ).

- **Seuil de tolérance**

Ce paramètre critique détermine la précision dans la réponse du réseau de neurones. La phase d'apprentissage est souvent arrêtée lorsque l'erreur calculée sur l'ensemble de la base d'apprentissage est inférieure à un seuil déterminé par l'utilisateur. Il est possible d'arrêter l'apprentissage en fixant une limite au nombre d'itérations.

### II.8. Les avantages et les inconvénients des réseaux de neurones

Les RNA sont une formulation mathématique simplifiée des neurones biologiques. Ils ont la capacité de mémorisation, de généralisation et surtout d'apprentissage qui est le phénomène le plus important. Dans cette partie, nous allons résumer les avantages et les inconvénients de l'utilisation des réseaux de neurones.

### **II.8.1. Avantages**

Les principales qualités des réseaux de neurones sont leur capacité d'adaptabilité et d'auto organisation et la possibilité de résoudre des problèmes non-linéaires avec une bonne approximation. Ils ont une bonne immunité aux bruits et se prêtent bien à une implantation parallèle. La rapidité d'exécution est une qualité importante et elle justifie souvent à elle seule le choix d'implanter un réseau de neurones. Ces qualités ont permis de réaliser avec succès, plusieurs applications : classification, filtrage, compression de données, contrôleur, etc....

### **II.8.2. Inconvénients**

La difficulté d'interpréter le comportement d'un réseau de neurone est un inconvénient pour la mise au point d'une application. Il est souvent impossible d'utiliser les résultats obtenus pour améliorer ce comportement. Il est également hasardeux de généraliser à partir d'expériences antérieures et de conclure ou de créer des règles sur le fonctionnement et le comportement des réseaux de neurones. Beaucoup d'heuristiques sont utilisées, mais elles se contredisent parfois et elles ne permettent pas toujours de trouver des valeurs optimales.

## **II.9. Conclusion**

Dans ce chapitre, nous avons présenté un aperçu général sur les réseaux de neurones, les architectures, les différents modèles, l'apprentissage et l'apprentissage du perceptron multicouche ont été discuté. Nous avons aussi parlé sur les avantages et les inconvénients des réseaux de neurones. Dans le chapitre suivant, on va choisir un modèle des RN qui est le MLP pour l'appliquer sur la reconnaissance des images texturées.

# *Chapitre III*

*Tests et résultats*

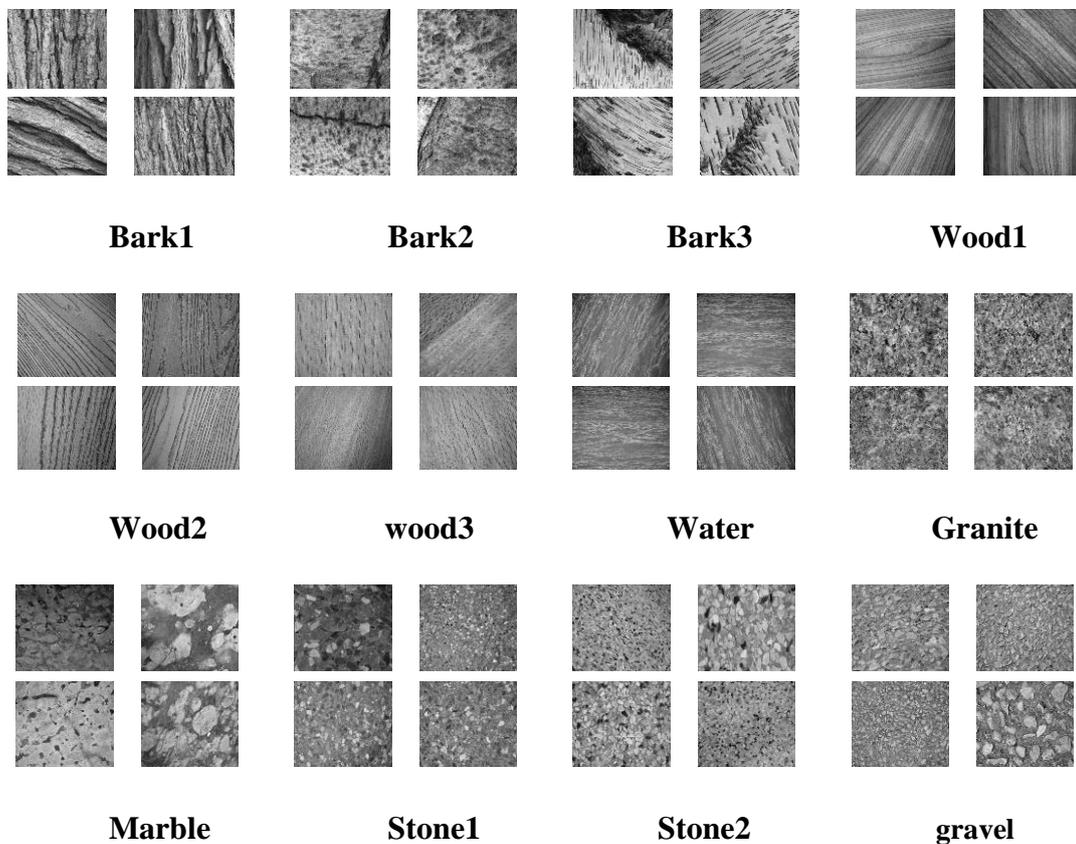
### III.1. Introduction

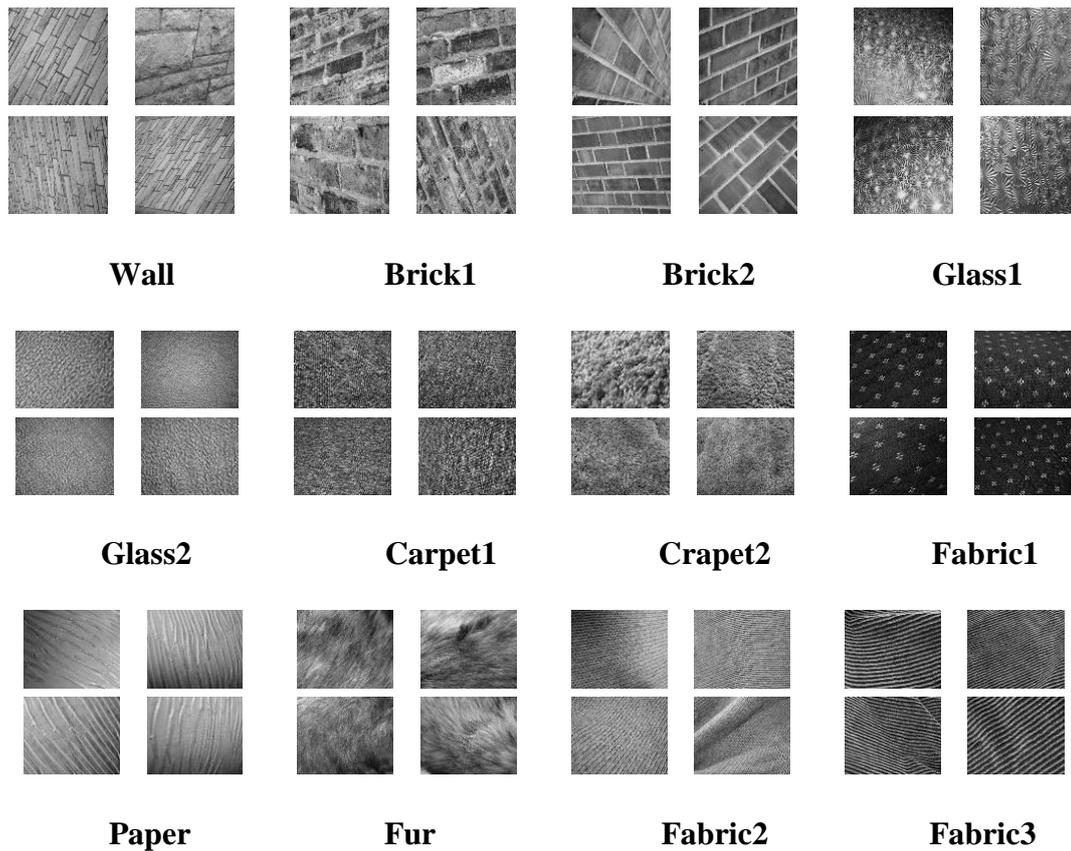
Après avoir décrits les réseaux de neurones et le problème de la reconnaissance des images texturées ainsi que l'extraction des caractéristiques LBP dans les deux chapitres précédents, ce chapitre sera consacré à la conception de notre réseau neurone et aux résultats expérimentaux que donne l'implémentation de ces méthodes.

Puisque l'apprentissage, le test et la définition des paramètres sont souvent considérés comme l'un des problèmes du réseau neuronal prototype, donc nous concentrons sur cette question en premier lieu. Plusieurs expériences et tests seront menés pour entrainer et tester le réseau afin d'atteindre les performances voulues. Ensuite, nous utiliserons le réseau entrainé pour la reconnaissance des images texturées avec des résultats expérimentaux.

### III.2. Base de données de texture

Dans notre travail, nous exploitons la base d'image de texture UIUCTex. Elle se compose de 1000 images de taille 640x480 réparties sur 25 classes de texture avec 40 images par classe (figure III.1).





**Figure III.1 :** Echantillons d'images texture de la base UIUCTex [24].

### III.3. Implémentation de l'algorithme de reconnaissance de texture

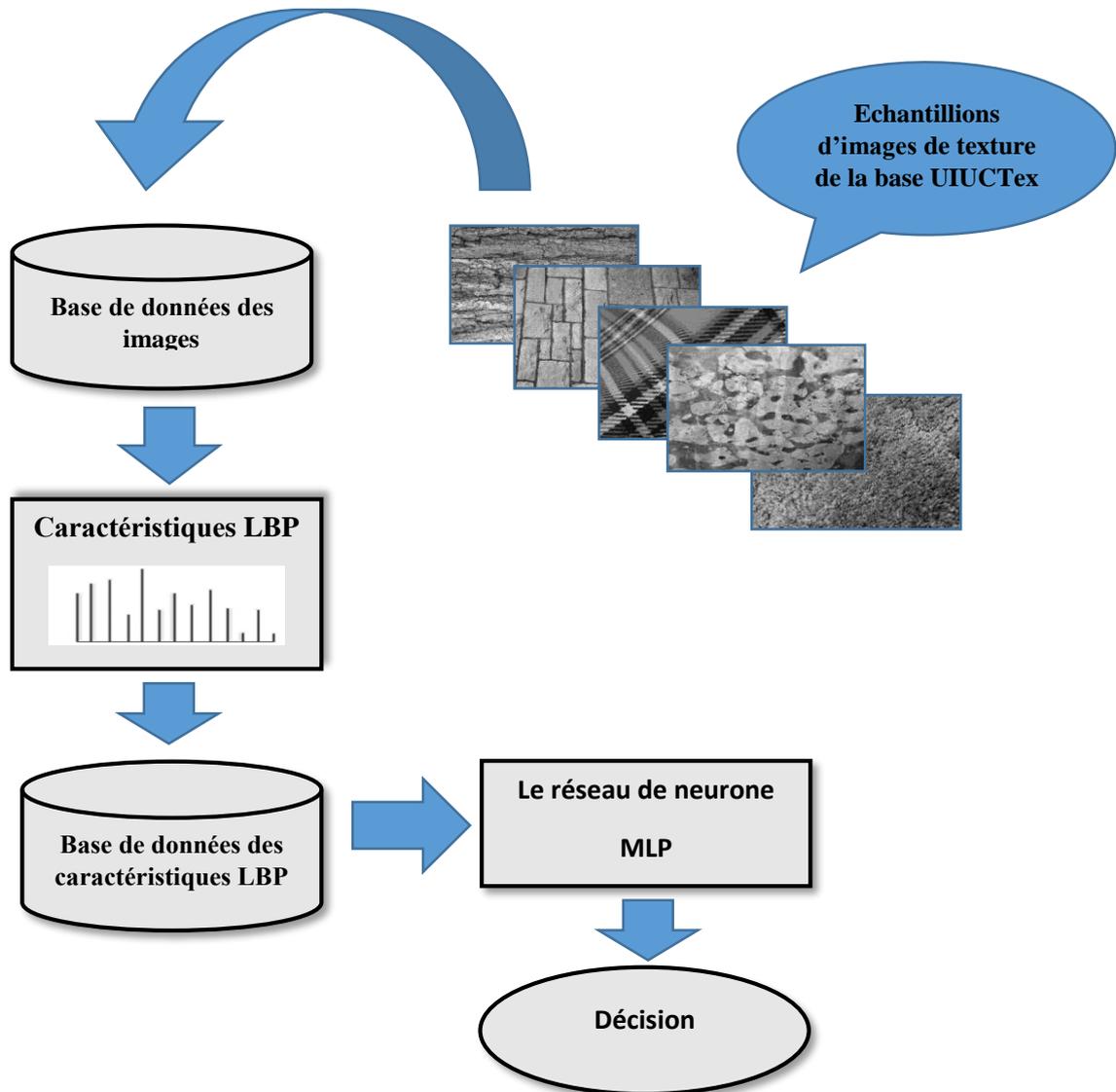
Pour réaliser notre travail, nous avons utilisé un PC portable SAMSUNG avec les caractéristiques suivantes :

- Processeur : Intel(R) Core (TM) i3-2370M CPU.
- Vitesse d'horloge : 2.40 GHz.
- Capacité Mémoire (RAM) :8.00 Go.
- Carte graphique : Intel HD 3000.
- Capacité disque dur : 500 Go.
- Système d'exploitation : Windows 7 professionnel.

Pour la réalisation de notre système nous avons choisi le langage de programmation MATLAB Version 9.4 (R2018a). MATLAB est un environnement de calcul scientifique et de visualisation de données. Sa facilité d'apprentissage et d'utilisation (due à une syntaxe très claire) en ont fait un standard adapté pour les divers problèmes l'ingénierie. Parmi les raisons qui nous ont poussés à l'utiliser, on trouve :

- Ses très nombreuses fonctions prédéfinies et prêtes à l'emploi.
- Sa simplicité à l'implémentation et rapidité de calculs.
- Sa fiabilité et sa robustesse.

La figure III.2 représente l'organigramme du système étudié et implémenté dans notre travail dans le but de classer les images de textures.



**Figure III.2 :** Organigramme d'un système de reconnaissance d'image.

### III.3.1. Extraction des caractéristiques LBP

L'extraction des caractéristiques de textures est la phase élémentaire de la classification de texture. De bons résultats à cette étape facilitent les étapes suivantes. Nous avons utilisé le descripteur LPB (voir chapitre1) pour des paramètres R, P différents (varie entre (R=1, P=8) et (R=2, P=16)), ainsi que ses variantes LBP uniforme, LBP invariant par rotation et LBP uniforme et invariante par rotation.

### III.3.2. Création du réseau de neurone

Pour obtenir des meilleurs résultats nous avons choisi le réseau de neurone à rétropropagation MLP (*Multi Layer Perceptron*). Dans notre cas :

- Le réseau est un réseau à rétropropagation avec la fonction d'activation *sigmoïde* dans la couche masquée (*Hidden Layer*) et une fonction d'activation *softMax* dans la couche en sortie (*Output Layer*). Nous avons affecté 10 neurones à la couche masquée de la figure III.3.
- Le réseau a 25 neurones de sortie (*Output*), car il existe 25 valeurs cibles (classes) associées à chaque vecteur d'entrée. Chaque neurone de sortie représente une catégorie.
- Lorsqu'un vecteur d'entrée (*Input*) de la classe appropriée est appliqué au réseau, le neurone correspondant doit produire un 1 et les autres neurones doivent fournir un 0.
- La matrice d'entrée des caractéristiques LBP de base est constituée de 256 éléments (varie en fonction de R et P).
- Les vecteurs d'entrée et les vecteurs cibles (classes) seront divisés de manière aléatoire, 70% étant utilisés pour l'apprentissage, 15% pour la validation et 15% pour les tests.

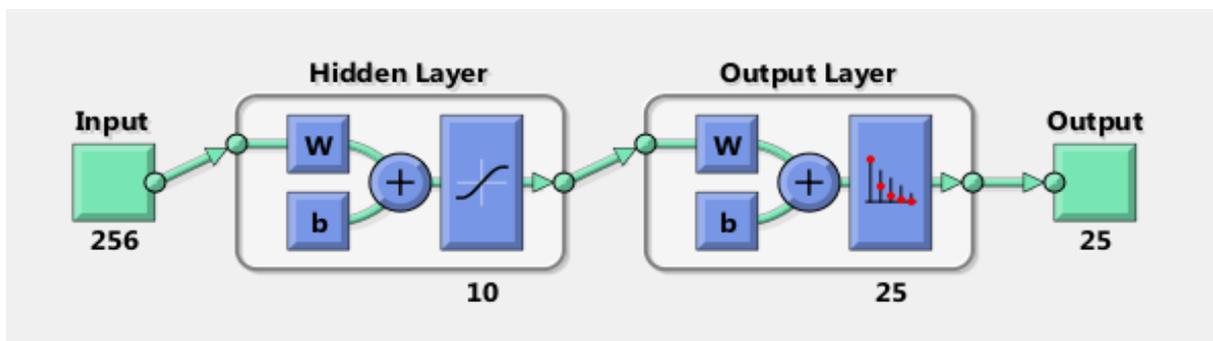


Figure III.3 : L'architecture d'un réseau neurone.

### III.3.3. Phase d'apprentissage

MLP utilise une technique d'apprentissage supervisé appelée rétropropagation pour l'apprentissage. La rétropropagation est une méthode utilisée dans les réseaux de neurones artificiels pour calculer le gradient nécessaire au calcul des poids à utiliser dans le réseau (voir le chapitre 2). Dans le contexte de l'apprentissage, l'algorithme d'optimisation de la descente du gradient utilise la rétropropagation pour ajuster le poids des neurones en calculant le gradient de la fonction de coût.

Le réseau de neurones produit pour l'entrée  $x^{(i)}$  une sortie  $\hat{y}^{(i)}$ . On aimerait savoir à quel point cette sortie diffère de la classe target  $y^{(i)}$ . Pour cela, on utilise une fonction de cout (loss) adaptée à notre problème. La majorité du temps, la fonction de cout globale  $L(X, Y)$  est la moyenne d'une fonction de cout unitaire  $l(y^{(i)}, \hat{y}^{(i)})$  entre les prédictions et les targets. Dans notre cas l'entropie croisée (cross-entropy) plutôt adaptée aux problèmes de classification :

$$l(y^{(i)}, \hat{y}^{(i)}) = -\sum_i y^{(i)} \log \hat{y}^{(i)} \quad (\text{III.1})$$

Grace à la fonction de cout, nous disposons donc d'une mesure de l'erreur de notre réseau de neurones. Nous devons maintenant apprendre ses paramètres (les matrices et vecteurs  $W$  et  $b$  pour le MLP) afin de minimiser cette erreur sur l'ensemble des exemples d'apprentissage. Pour ce faire, nous allons utiliser l'algorithme de *scaled conjugate gradient*. Il consiste à calculer la dérivée de la fonction de cout par rapport à un paramètre  $w$ , et à faire un pas dans l'opposé de la direction du gradient (voir le chapitre II).

Cependant, il est dangereux de poursuivre indéfiniment la phase d'apprentissage sans contrôle : la capacité d'ajustement d'un réseau de neurones est tellement forte qu'au bout d'un certain nombre d'itérations, les poids synaptiques parviennent à prédire quasiment sans erreur les données.

Cette idée est illustrée sur la figure III.4, l'erreur d'apprentissage continue la diminution au cours des itérations (courbe bleu : *train*). On parle alors d'apprentissage par cœur ou de sur-apprentissage (*overfitting*). On prévient le sur-apprentissage par l'ensemble de validation. Les données de validation sont neutres dans la détermination des poids ; elles ne servent qu'à arrêter l'apprentissage à une itération précédant le début du sur-apprentissage. Sur la figure III.4, l'erreur sur le groupe de validation augmente après l'itération (*epoch*) 339, donc le processus d'apprentissage se termine à cette itération et les poids sont retenus pour les servir dans la phase de test. Notons que pour la figure III.5, la vérification de l'augmentation de l'erreur de validation (*validation checks*) est effectuée 6 fois avant d'arrêter l'apprentissage.

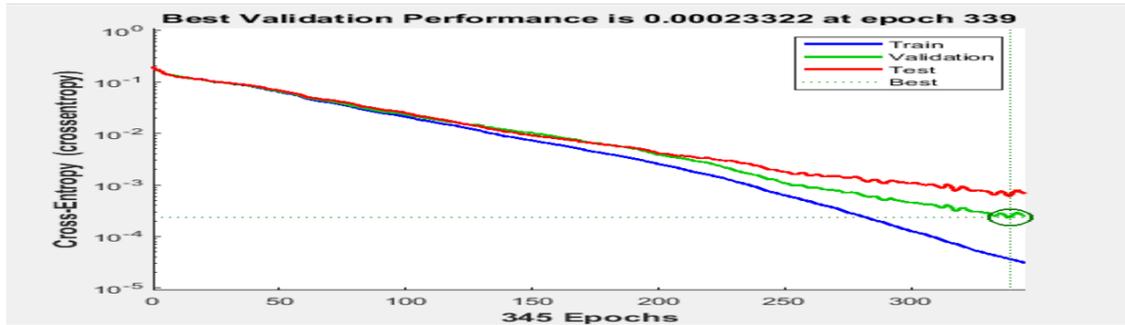


Figure III.4 : L'effet de la validation pour la descripteur LBP uniforme (2,16).

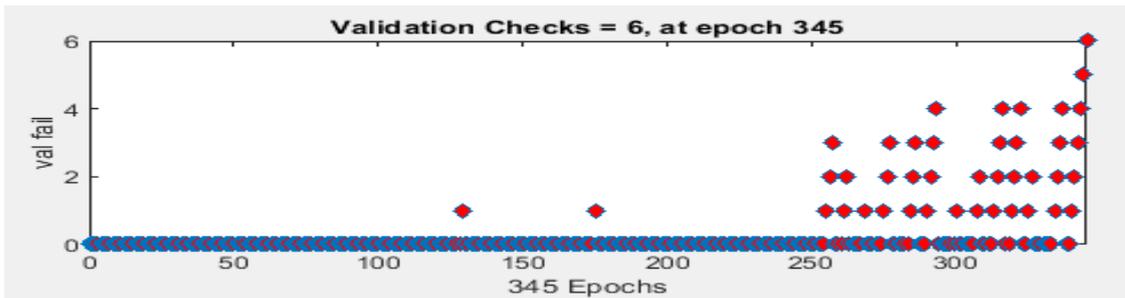


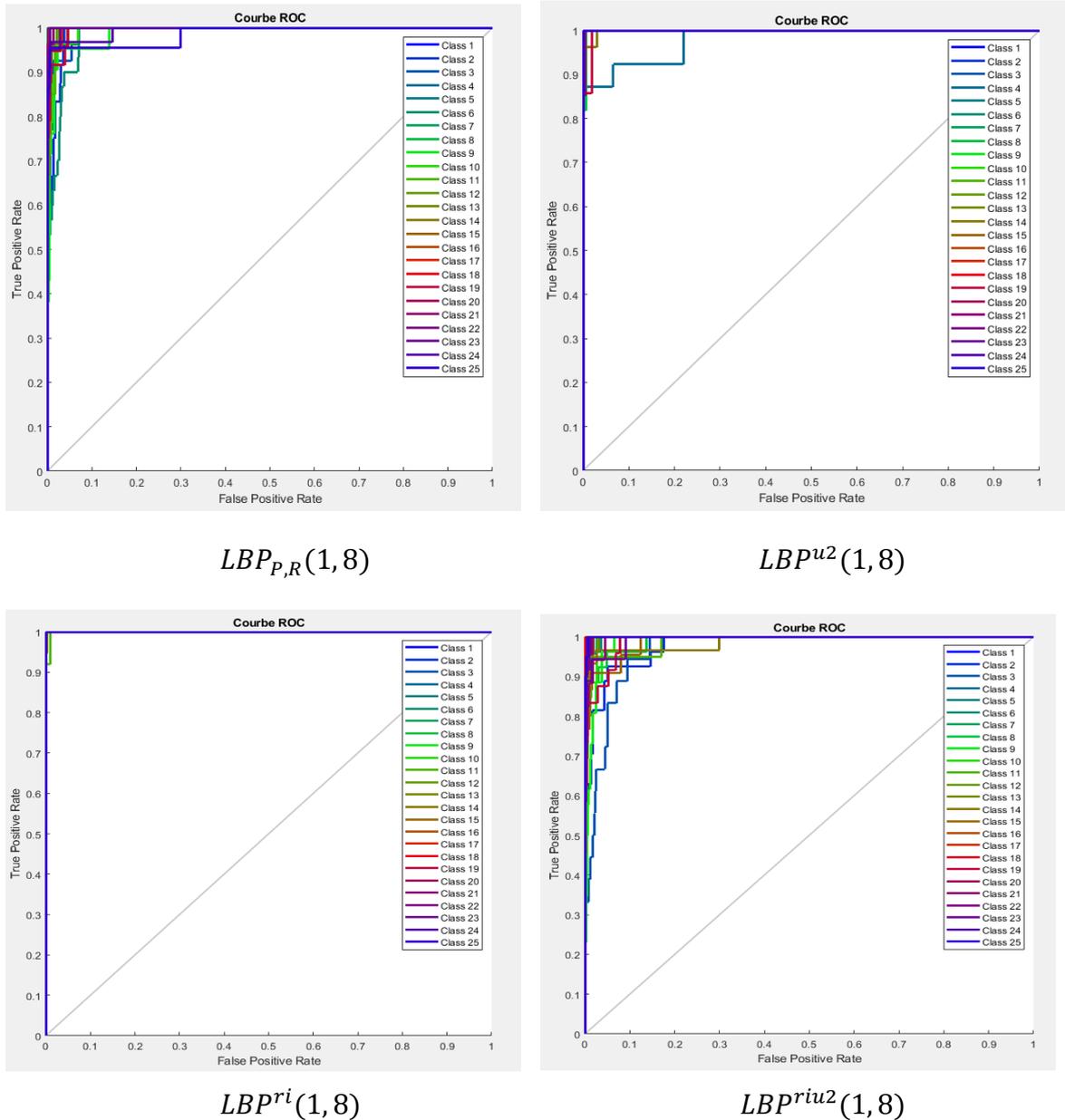
Figure III.5 : L'erreur de validation pour LBP uniforme (2,16).

### III.3.4. Test de performance

Une fois le réseau est entraîné, nous pouvons l'utiliser pour un groupe de test indépendant des groupes d'apprentissage et de validation. Dans le tableau ci-dessous tous les résultats de taux d'erreur pour des méthodes LBP différentes avec des différentes valeurs de R et P, ici le nombre de couches dans le réseau de neurones est 20 :

Méthodes		Nombre des couches	
		10	20
LBP base	R=1, P=8	19.72	14.05
LBP uniforme	R=1, P=8	4.10	1.27
	R=2, P=16	0.27	0.38
LBP invariant par rotation	R=1, P=8	1.05	1.38
	R=2, P=16	16.62	7.05
LBP invariant par rotation et uniforme	R=1, P=8	30.90	16.21
	R=2, P=16	18.53	13.88

Tableau III.1 : Taux d'erreur (%) selon les différentes méthodes LBP.



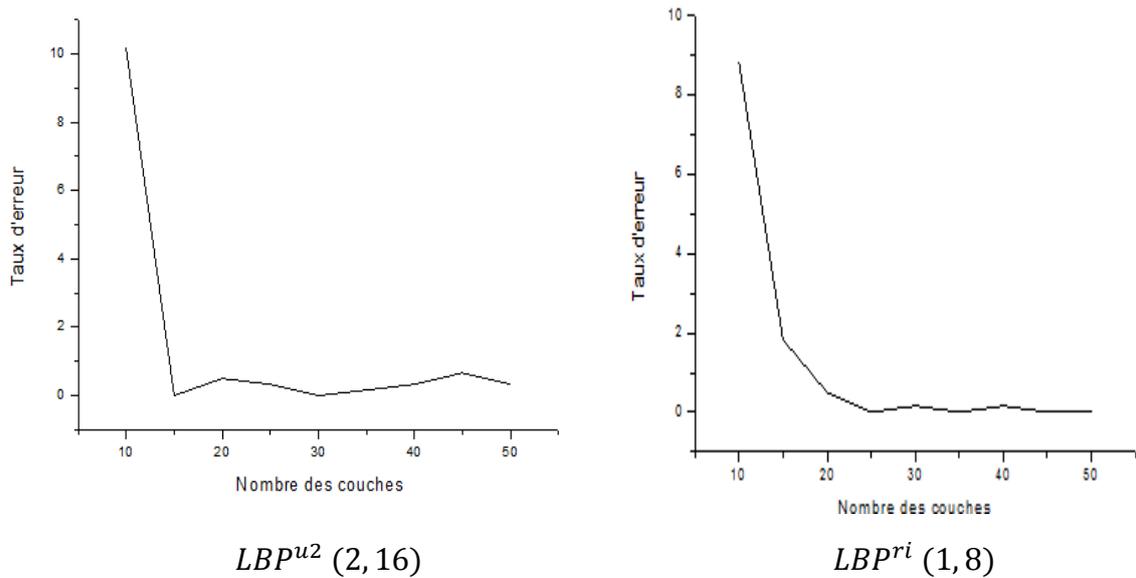
**Figure III.6 :** Les courbes ROC des méthodes LBP différentes.

La caractéristique ROC (Receiver Operating Characteristic) permet de voir si le réseau fonctionne normalement ou non. Les lignes colorées sur chaque axe représentent les courbes « ROC ». Cette courbe est un graphique du vrai taux positif (sensibilité) par rapport au faux taux positif (spécificité) lorsque le seuil est modifié. La figure.III.6 montre des bons résultats en terme des taux vrais positif pour les différentes classes.

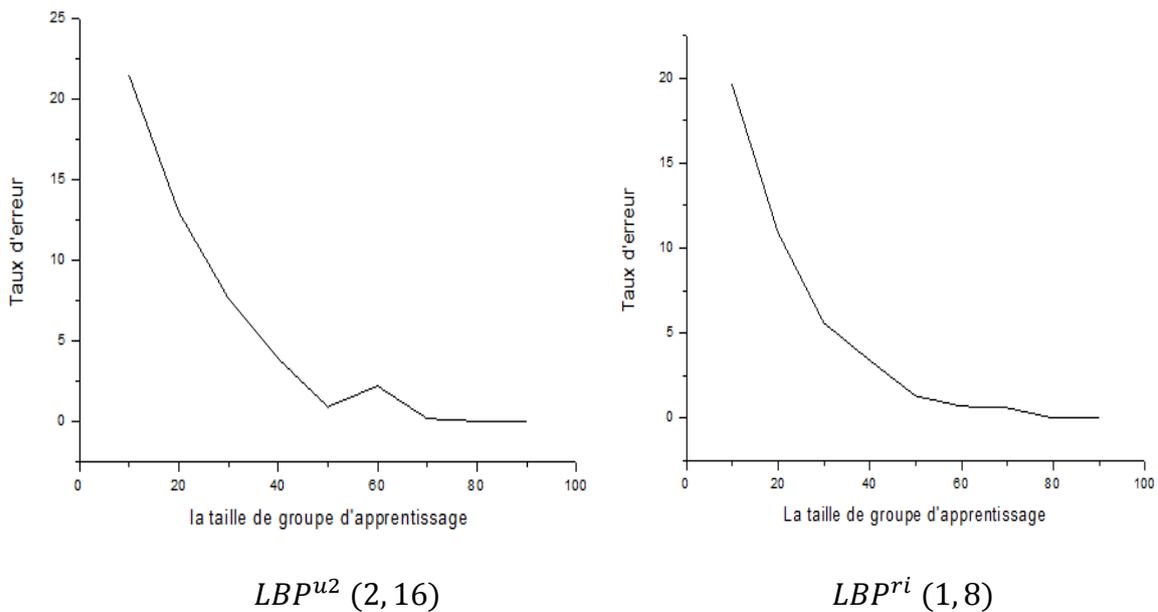
Les taux d'erreur sont affectés par différents paramètres, nous examinons : le nombre d'échantillons d'apprentissage et le nombre de couche dans le réseau de neurone.

Dans la figure III.7 on donne la variation du taux d'erreur par rapport à l'augmentation des nombres de couches cachées. L'augmentation de la complexité du réseau de neurones peut améliorer les performances du système de la reconnaissance, mais juste pour une limite, après le taux soit reste inchangé ou il augmente.

Dans la figure III.8 qui montre la diminution du taux d'erreur avec l'augmentation de la dimension de la base d'apprentissage. Nous remarquons aussi, qu'en atteignant un taux limite, le changement de la dimension de la base n'a aucun effet sur ce dernier.



**Figure III.7 :** Les courbes des taux d'erreur (%) en fonction de l'apprentissage.



**Figure III.8 :** Les courbes des taux d'erreur (%) en fonction du nombre des couches.

Examinons de près les résultats de la classification pour les différentes classes de texture. Nous avons choisi les méthodes LBP dont les résultats sont les meilleures avec le nombre de couche du réseau de neurone égale 20.

Classes	LBP base (1,8)	LBP uniforme (2,16)	LBP invariant par rotation (1,8)	LBP uniforme et invariant rotation (2,16)
<b>Bark1</b> 	9,26	0	0	4,17
<b>Bark2</b> 	34,02	0	0	23,92
<b>Bark3</b> 	5,04	0	6,89	12,31
<b>Wood1</b> 	24,29	14,28	8,73	23,85
<b>Wood2</b> 	5,89	0,34	0	20,86
<b>Wood3</b> 	29,70	8	23,42	11,73
<b>Water</b> 	0	0	0	0
<b>Granite</b> 	12,17	0	0,34	18,69
<b>Marble</b> 	4	3,57	16,83	21,05
<b>Stone1</b> 	17,01	0,17	0,17	33,85
<b>Stone2</b> 	11,88	0	0	33,85
<b>Gravel</b> 	30,87	0	0	13,80
<b>Wall</b> 	19,39	0	0,34	10,20

<b>Brick1</b> 	13,91	0	0	16,69
<b>Brick2</b> 	5	0,68	3,70	5,09
<b>Glass1</b> 	0,17	8,33	0	17,81
<b>Glass 2</b> 	3,84	0	0	11,80
<b>Carpet1</b> 	3,57	0	0	4,52
<b>Crapet2</b> 	7,17	0	8,33	6,12
<b>Fabric1</b> 	3,85	0,34	2,77	7,24
<b>Paper</b> 	19,39	8,33	0,51	39,74
<b>Fur</b> 	5,73	0	5,26	10,87
<b>Fabric2</b> 	0,17	0	0,17	4,86
<b>Fabric3</b> 	4,34	0	0	7,93
<b>Fabric4</b> 	1,37	0	0	0,34

**Tableau III.2 :** Taux d'erreur (%) des différentes classes pour les différentes méthodes LBP.

Le réseau de neurones employé dans notre travail est un outil puissant de classification des données avec des taux d'erreur très petits avec la méthode d'extraction de caractéristiques LBP uniforme et invariante par rotation. Cependant, nous avons soulevé un taux d'erreur élevé dans certains types de texture à cause de leurs similitudes et de leurs structures.

### **III.4. Conclusion :**

Dans ce chapitre nous avons créé un réseau neurone artificiel de type MLP pour faire une classification des images texturées. Nous avons créé, entraîné et testé un réseau de neurones avec la base UIUCTex dont le but d'améliorer le taux de reconnaissance c'est-à-dire minimiser le taux d'erreur.

Les résultats trouvés étaient en amélioration continue en termes de la performance et de la convergence du gradient vers les faibles valeurs. L'élaboration de ce chapitre nous a permis d'apprendre les règles de fonctionnement d'un réseau de neurones artificiels multicouche à apprentissage supervisé et de montrer l'efficacité de la méthode d'extraction des caractéristiques LBP ce qui nous a permis de classer des images texturées d'une grande base de données.

## ***Conclusion générale***

La reconnaissance des images est une tâche très importante dans plusieurs domaines de la vie, car elle permet l'identification de n'importe quelle image de manière automatique sans intervention humaine avec le moins d'erreur possible.

Notre objectif c'est une contribution modeste dans l'amélioration de la classification, dans ce travail nous avons choisi les images texturées, la grande majorité de la texture naturelle est facilement détectée, segmentée et reconnue par les humains pour cela nous avons utilisé un système de classification simule le fonctionnement humain qui est les réseaux neurones artificielles.

Dans notre travail, nous utilisons la méthode de motif binaire local LBP pour l'extraction des caractéristiques de chaque image de texture, la méthode LBP contient quatre variantes qui sont LBP simple, LBP uniforme, LBP invariante par rotation, LBP uniforme et invariant par rotation et pour chaque type on trouve deux paramètres qui sont le voisinage de points P et le rayon R d'un pixel, nous avons changé ces paramètres (P, R) pour couvrir plusieurs types de texture (microtexture et macrotecture), pour enfin obtenir une matrice qui est utilisée comme entrée dans l'étape de la classification.

L'étape de la classification vise à déterminer la classe de l'image, pour faire cette étape nous avons créé un réseau neurone multicouches de type MLP. Le choix de ce dernier comme algorithme de classification a été essentiellement motivé par les résultats encourageants obtenus avec cette technique dans plusieurs applications de la reconnaissance de forme. La précision de ces résultats était, dans la plupart des cas, supérieure à celle obtenue avec les méthodes conventionnelles de classification [25]. En effet, la supériorité des réseaux de neurones est essentiellement due à son aspect non paramétrique qui évite l'utilisateur de faire des suppositions sur la distribution statistique des données. En plus, les réseaux de neurones ont l'avantage de considérer différemment l'apport de chaque source d'information pour la prise de décision d'une classe donnée. Ce dernier avantage permet de renforcer automatiquement le poids des informations qui ont une contribution majeure dans la discrimination de certaines classes.

Nous avons appliqué le réseau MLP pour classer les images de la base de données UIUCTex, celle-ci se compose de 1000 images de taille 640x480 réparties sur 25 types de texture (bois, marbre, tissu, eau, verre...) avec 40 images par classe.

Les résultats obtenus dans ce mémoire ont montré que le réseau de neurones est un puissant outil de classification des données avec des taux d'erreur très petits avec la méthode d'extraction de caractéristiques LBP uniforme et invariante par rotation. Cependant, nous avons soulevé un taux d'erreur élevé dans certains types de texture à cause de leurs similitudes et de leurs structures.

### **Perspectives**

Nous proposons comme perspectives, l'utilisation des réseaux de neurones convolutionnels pour la reconnaissance des images de texture.

---

## Références

- [01] R.C. Gonzalez, R.E. Woods, " Digital image processing " , Upper Saddle River, N.J. , Prentice Hall, 2002.
- [02] M. Petrou, P.G. Sevilla, " Image Processing : Dealing With Texture " , England : John Wiley and Sons Ltd, ISBN : 978-0470026281, 2006.
- [03] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh, " On the significance of real-world conditions for material classification " , In Computer Vision-ECCV 2004, pp. 253–266, Springer, 2004.
- [04] B. Julesz, "Experiments in the Visual Perception of Texture," Scientific American, pp. 34-43, 1975.
- [05] T. Ojala, M. Pietikainen, and T. Maenpaa, " Multiresolution gray-scale and rotation invariant texture classification with local binary patterns " , IEEE Trans. PAMI,24(7).
- [06] T. Maenpaa, T. Ojala, M. Pietikainen, and M. Soriano, " Robust texture classification by subsets of local binary patterns " . In: Proc. ICPR. Vol. 3. IEEE, pp. 935–938, 2000.
- [07] T. Ojala, M. Pietikainen, T. Maenpaa, " Multiresolution gray-scale and rotation invariant texture classification with local binary patterns " . IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7),971–987, 2002.
- [08] [https://msdn.microsoft.com/big\\_data\\_france/2014/06/17/evaluer-un-modle-en-apprentissage-automatique/](https://msdn.microsoft.com/big_data_france/2014/06/17/evaluer-un-modle-en-apprentissage-automatique/)
- [09] T. Fawcett, " An introduction to roc analysis " . Pattern recognition Letters, 27 :861–874, 2006.
- [10] O. Duda Richard, E. Hart Peter, " Pattern Classification " , J. Wiley and Sons, First Edition, pp. 482, 1973.
- [11] D.O. Hebb, " The organization of behavior " , J. Wiley and Sons, 1949.
- [12] F. Rosenblait, " Principles of neurodynamics " , Spartan, New York, 1962.
- [13] G. Burel et al., " Réseaux De Neurones En Traitement De L'image Et Du Signal " , Flux, No. 151-152, pp 18-27, juin 1993.

- [14] T. M. Hagan, B. H. Demuth, M. H. Beale, O. De Jesús " Neural Network Design ", 2<sup>nd</sup> Edition, septembre 2014.
- [15] A.K. Jain, J. Mao, and K.M. Mohiuddin, " Artificial neural networks : a tutorial ", Computer 29(3), pp. 31-44, 1996.
- [16] M. Minsky and S. Papert, "Perceptrons : An Introduction to Computational Geomerty", MIT Press, Cambridge, Mass., 1969.
- [17] S. Haykin, " Neural Networks : A Comprehensive Foundation " , Macmillan College Publishing Company, New York, 1994.
- [18] J. Park and I.W. Sandberg, " Universal Approximation Using Radial-Basis-Function Networks", Neural Computation, Vol. 5, pp. 305-316, 1993.
- [19] J.J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," in Proc.Nat'l Academy of Sciences, USA 79, pp. 2554-2558, 1982.
- [20] T. Kohonen, Self Organization and Associative Memory Third Edition, Springer-Verlag, New York, 1989.
- [21] M.Z. Mokri, " Classification des images avec les réseaux neurones convolutionnels " , Mémoire de PFE, Université de Tlemcen, 2017.
- [22] N. Zaarour, " Modélisation d'un canal minier ultra large bande (UWB) en utilisant les réseaux de neurones artificiels RBF " , Québec, 2013.
- [23] Y. Le Cun et al., "Back-Propagation Applied to Handwritten Zip Code Recognition, Neural Computation, Vol. 1, pp. 541-551, 1989.
- [24] [http://www-cvr.ai.uiuc.edu/ponce\\_grp/data/texture database](http://www-cvr.ai.uiuc.edu/ponce_grp/data/texture_database).
- [25] M. Benhabiles, C. Boudjaoui, " Classification des images par les motifs binaires locaux (lbp) " , Mémoire de PFE, Université de jjel, 2017.

## ملخص

في عملنا هذا، تناولنا موضوع التعرف على الصور النسيجية، للقيام بهذا قمنا بخطوتين مهمتين وضروريتين:

الخطوة الأولى: استخراج خصائص كل صورة باستخدام الأنماط الثنائية المحلية LBP ، إن فكرة واصف النسيج هذه هي تعيين رمز لكل بكسل وفقاً للمستويات الرمادية لمنطقة الجوار. بناءً على حجم الحي المستخدم، يمكن اكتشاف بعض المناطق المهمة مثل الزوايا أو الحواف بواسطة هذا الوصف

الخطوة الثانية: تتمثل في استخدام خصائص LBP الصور النسيجية لتعليم شبكة عصبية ذات انتشار رجعي حسي متعدد الطبقات هدفها الحصول على نظام تلقائي لتصنيف صور نسيج غير معروفة بدقة جيدة جداً.

## Résumé

Dans notre travail, nous avons abordé le sujet de l'identification des images de textures, pour ce faire nous avons effectué deux étapes importantes et nécessaires :

Première étape : l'extraction des caractéristiques de chaque image à l'aide de motifs binaires locaux LBP. L'idée de ce descripteur de texture est d'assigner à chaque pixel un code dépendant des niveaux de gris de son voisinage. Selon l'échelle du voisinage utilisé, certaines zones d'intérêts tels des coins ou des bords peuvent être détectés par ce descripteur.

La deuxième étape : consiste à utiliser les caractéristiques LBP des images texturées pour entraîner un réseau de neurones à retro-propagation de type perceptron multicouches dont le but est d'obtenir un système automatique pour classer les images de texture inconnues avec une très bonne précision.

## Abstract

In our work, we have discussed the subject of the identification of texture images, to do this we have made two important and necessary steps:

First step: extracting the characteristics of each image using local binary patterns LBP. The idea of this texture descriptor is to assign each pixel a code depending on the gray levels of its neighborhood. Depending on the scale of the neighborhood used, certain areas of interest such as corners or edges can be detected by this descriptor.

The second step consists of using the LBP characteristics of the textured images to create a multi-layered perceptual retro-propagation neuron network whose goal is to obtain an automatic system for classifying unknown texture images with very good precision.