

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

UNIVERSITÉBADJI MOKHTAR-ANNABA

BADJI MOKHTAR– ANNABAUNIVERSITY



جامعة باجي مختار – عنابة

Faculté : science de l'ingénieur

Département : informatique

Domaine : mathématique-informatique

Filière : Informatique

Spécialité : IATI

## Mémoire

Présenté en vue d'obtenir le diplôme de MASTER

Thème :

**CLASSIFICATIONDES MASSES MAMMAIREUTILISANT  
LOCAL TERNARY PATTERN**

Présenté par : *M. Chared achraf eddine*

Encadrant : *Mme. Sarah Benmazou*

**Jury de Soutenance :**

|                |     |      |           |
|----------------|-----|------|-----------|
| Koubsi Noha    | MAB | UBMA | Président |
| Benmazou Sarah | MCB | UBMA | Encadrant |
| Djebbar Akila  | MCB | UBMA | Examineur |

Année universitaire 2020/2021

# TABLE DE MATIERE

|  |    |
|--|----|
| Table de matière.....                              | 1  |
| Liste de figures.....                              | 4  |
| Liste des tableaux.....                            | 5  |
| Remerciements.....                                 | 6  |
| Dédicaces.....                                     | 7  |
| Résumé.....  | 8  |
| Introduction générale.....                         | 14 |
| Chapitre 1 : description de texture.....           | 16 |
| 1.1 Introduction.....                              | 17 |
| 1.2 Définition de texteur.....                     | 17 |
| 1.3 Analyse de texture.....                        | 18 |
| 1.3.1 Approche statistique .....                   | 18 |
| 1.3.2 Approche structurelle.....                   | 19 |
| 1.4 Les attributs statistiques de texture.....     | 19 |
| 1.4.1 Attributs statistiques du premier ordre..... | 19 |
| 1.4.1.1 La moyenne.....                            | 20 |
| 1.4.1.2 La variance.....                           | 20 |
| 1.4.1.3 Le moment d'ordre 3.....                   | 20 |
| 1.4.1.4 Le moment d'ordre 4 .....                  | 20 |
| 1.5.2 Attributs statistique du second ordre .....  | 21 |

|   |    |
|---|----|
| 1.5.1 Attributs statistiques d'ordre supérieur.....   | 23 |
| 1.5.2 Motifs binaires locaux.....                     | 25 |
| 1.5.2.1 Etape d'algorithme.....                       | 25 |
| 1.5.2 Les motifs ternaires locaux .....               | 29 |
| 1. conclusion.....                                    | 31 |
| Chapitre 2 la classification.....                     | 32 |
| 2.1 Introduction.....                                 | 33 |
| 2.2 Définition.....                                   | 33 |
| 2.3 Méthodes de classification.....                   | 33 |
| 2.3.1 Méthodes de classification non supervisée ..... | 34 |
| 2.3.1.1 Méthode de K-Means .....                      | 34 |
| 2.3.1.2 Méthode Ascendante Hiérarchique .....         | 35 |
| 2.3.2.3 Méthode des modes de l'histogramme.....       | 35 |
| 2.3.2 Méthodes de classification supervisée.....      | 35 |
| 2.3.2.1 K plus proche voisins(KNN).....               | 36 |
| 2.3.2.2 Distance minimale.....                        | 36 |
| 2.3.2.3 Analyse factorielle discriminantes.....       | 36 |
| 2.3.2.4 Les réseaux de neurones.....                  | 37 |
| 2.3.2.5 Classification supervisée par SVM.....        | 37 |
| 2.4 Validation d'une classification .....             | 46 |
| 2.3.1 Matrice de confusion.....                       | 48 |
| 2.5 Conclusion.....                                   | 50 |
| Chapitre 3 : conception et implémentation.....        | 51 |
| 3.1 Conception générale.....                          | 52 |

|   |    |
|---|----|
| .2 Architecture du système proposé .....        | 52 |
| 3.3 Description générale du système.....        | 53 |
| 3.4 Base de données.....                        | 54 |
| 3.5 Extraction de caractéristiques.....         | 54 |
| 3.6 Généralité sur LBP.....                     | 54 |
| 3.7 Généralités sur LTP.....                    | 54 |
| 3.8 Apprentissage.....                          | 55 |
| 3.9 Classification par SVM.....                 | 55 |
| 3.10 Classification par KNN.....                | 56 |
| 3.10 Évaluation de système.....                 | 57 |
| 3.11 Définition de la Base DDSM.....            | 57 |
| 3.12.2 Langages de programmations utilisés..... | 57 |
| 3.13 Conclusion générale.....                   | 60 |
| Référence.....                                  | 61 |

# Liste de figure

---

|  |    |
|--|----|
| Figure 1.1 : prendre le quartier binaire 8 bits du pixel central et de convertir en une représentation décimale.....     | 25 |
| Figure 1.2 : exemple de résultat obtenue par LBP.....  | 26 |
| Figure 1.3 : capteur d'écran sur un exemple d'image résultat de LBP.....   | 26 |
| Figure 1.4 : histogramme LBP en peut considérer les valeurs de l'histogramme comme des caractéristique d'image .....     | 27 |
| Figure 1.5 : Trois exemples de quartier avec différents p et r utilisés pour construire des modèles binaires locaux..... | 28 |
| Figure 1.6 : étapes d'extraire le pattern code.....  | 29 |
| Figure 1.7 : montrer la 2 <sup>ème</sup> étape d'algorithme LTP.....   | 30 |
| Figure 1.8: transformation de code binaire en décimale.....  | 30 |
| Figure 2.1 : principe SVM.....   | 38 |
| Figure 2.2 : hyperplan sépare deux classes.....  | 39 |
| Figure 2.3 : hyperplan optimal, vecteurs de support et marge maximal.....  | 39 |
| Figure 2.4 : meilleur hyperplan séparateur.....  | 40 |
| Figure 2.5 : exemple de linéarité et non linéarité.....  | 41 |
| Figure 2.6 : exemple de transformation d'un problème de séparation non linéaire.....                                     | 41 |
| Figure 2.7 : exemple graphique des données linéairement séparable.....   | 42 |
| Figure 2.8 : espace de projection des données non séparable.....   | 43 |
| Figure 3.1 : architecture de système avec LTP  |    |
| Figure 3.2 : architecture de système avec LBP  |    |
| Figure 3.3 : implémentation d'algorithme LBP sous python   |    |
| Figure 3.4 : implémentation d'algorithme LTP sous python   |    |
| Figure 3.5 : implémentation de SVM et KNN sous python  |    |
| Figure 3.6: image extraire de la base DDSM   |    |
| Figure 3.7 : logo d'IDE spyder.  |    |

# Listes des tableaux

---

|   |    |
|---|----|
| Table 1:Exemple de construction d'une matrice de cooccurrence.....              | 21 |
| Table 2:Exemple de construction de la matrice de plages de niveaux de gris..... | 23 |
| Table 3:les fonctions des noyaux la plus utilise avec SVM.....                  | 46 |
| Table 4:Matrice de confusion.....   | 50 |

---

## Résumé

---

Le cancer du sein est c'est le cancer auquel la plupart des femmes sont exposées. L'incidence du cancer du sein progresse dans le monde. On sait tous que La mammographie est l'examen radiographique accrédité. Elle a pour but de détecter d'éventuelles anomalies au niveau des tissus, c'est pour cela le nombre de mammographie est en augmentation croissante, le diagnostic par ordinateur devient à nos jours l'un d'impératif pour la facilité de détection du tumeur "précoce et correcte".

Dans ce cadre, nous proposons un système d'aide pour diagnostiquer le cancer du sein en utilisant le textureur, pour extraire des caractéristiques avec LTP et LBP et puis on fait une classification avec SVM et KNN.

**Mot clés : Texture, Mammographie, LTP, LBP, SVM, KNN.**

# Abstract

---

Breast cancer is cancer that most women exposed to. The incidence of breast cancer is on the rise worldwide. We all know that Mammography is the accredited x-ray examination. Its purpose is to detect any abnormalities in the tissues, which is why the number of mammograms is increasing; computer diagnosis is nowadays one of the imperative for the ease of detection of the tumor. "Early and correct".

In this framework, we propose an aid system to diagnose breast cancer using the texture, to extract features with LTP and LBP and then we do a classification with SVM and KNN.

Keywords: Texture, Mammography, LTP, LBP, SVM, KNN.



# ملخص

---

سرطان الثدي هو السرطان الذي تتعرض له معظم النساء. إن معدل الإصابة بسرطان الثدي أخذ في الارتفاع في جميع أنحاء العالم. نعلم جميعاً أن التصوير الشعاعي للثدي هو فحص الأشعة السينية المعتمد. والغرض منه هو الكشف عن أي تشوهات في الأنسجة ، ولهذا السبب يتزايد عدد صور الثدي بالأشعة السينية ، يعد التشخيص الحاسوبي في الوقت الحاضر أحد الأمور الضرورية لسهولة الكشف عن الورم بشكل "مبكر وصحيح".

في هذا السياق ، نقترح نظاماً مساعداً لتشخيص سرطان الثدي باستخدام شكل نسيج الثدي ، لاستخراج الميزات نستخدم

الخوارزميات LBP و LTP ونقوم بتصنيف باستخدام KNN و SVM

**الكلمات الرئيسية:** الملمس ، التصوير الشعاعي للثدي SVM , KNN, LTP, LBP



# Remerciements

---

*Je remercie tout d'abord Dieu pour m'avoir accordé force et patience tout au long de mon travail*

*Je tiens aussi à remercier encadrante Mme Benmazou de m'avoir soutenue jusqu'au bout, à mes très chers parents qui m'ont sans cesse encouragé et qui sont tout le temps étaient présent pour moi, à l'examineur qui va nous assister durant l'examination et à la présidente de jury.*

*Et enfin à tous ceux qui ont contribué à l'aboutissement de notre modeste projet.*

# Dédicaces

---

*Je dédie ce projet à toute notre promotion Master IATI 2020/2021 en leur souhaitons à tous succès et réussite.*

*À tous nos enseignants du département d'informatique qui nous ont transmis leur savoir et leur connaissance durant ces deux ans d'études.*

*À mes très chers parents qui m'ont inculqué les bonnes valeurs, qui m'ont soutenue et toujours crus en moi ainsi qu'à toute ma famille.*

*Et pour finir à tous ceux qui nous ont aidés et encouragés tout le long de notre cursus*



## **Introduction générale:**

L'une des nouvelles les plus choquantes de nos jours est la croissance maligne des cellules mammaires et c'est une raison attendue de la mort chez les femmes. De façon constante, le taux de disparition a augmenté en raison de la croissance maligne des cellules mammaires. Nous sommes en mesure de diagnostiquer le cancer du sein à un stade précoce, cela améliore les chances de traitement et contribue également à sauver la vie d'un patient. Un regroupement plus précis des tumeurs aimables peut empêcher les patients de subir des médicaments inutiles. De cette manière, la bonne analyse du cancer du sein et la catégorisation des patients en groupe malin et non malin font l'objet de nombreux examens.

Il ya plusieurs manières pour extraire les caractéristiques de l'image dont on a choisi comme solution la caractérisation des mammographies.

Les techniques d'apprentissage automatique et de l'IA sont des méthodes convaincantes pour caractériser les informations. Ces techniques et méthodes peuvent être utilisées dans la détermination et l'examen en particulier dans le domaine clinique et peuvent également être utilisées dans la prise de décision qui est d'une importance primordiale. En outre, l'objectif majeur et le rôle de ces algorithmes dans les soins de santé, en particulier pour la détection de la malignité, peuvent être réalisés à l'aide de modèles tels que **SVM**, **KNN**, etc. Ici, notre objectif principal est la précision dans l'organisation et l'évaluation des informations de chaque calcul en ce qui concerne l'exactitude, la précision, l'affectibilité et l'explicitation. Donc nous allons viser à étudier le concept et la procédure d'application des algorithmes d'apprentissage automatique pour la classification du cancer du sein...

## **Diagnostiquer le cancer :**

Le diagnostic de cancer du sein peut être évoqué soit devant des signes radiologiques frustes, mis en évidence lors d'un cliché de dépistage, soit devant des signes cliniques souvent découverts par la patiente elle-même. Quelles que soient les circonstances, ce diagnostic reposera sur une confrontation clinico-radiologique et surtout sur une confirmation pathologique obtenue par biopsie.

D'autres investigations sont ensuite nécessaires pour:

-évaluer l'extension de la maladie ;

-évaluer les facteurs pronostiques et les facteurs prédictifs ;

-évaluer le terrain.

Tous ces paramètres sont essentiels pour déterminer la stratégie optimale décidée au mieux lors d'une confrontation multidisciplinaire. De nombreux << guidelines >> sont actuellement proposées pour la prise en charge de ce type de cancer. Une liste non exhaustive de ces guidelines est proposée en annexe. Il est prudent avant de s'y référer de vérifier la date de validation compte tenu de la rapidité actuelle d'évolution de la thérapeutique en oncologie mammaire.

Ce mémoire est reparti comme suit :

### **Chapitre 1** :description de texture

Dans ce chapitre on se base sur la définition d'une texture de différent type et voire de différentes méthodes ainsi la manière d'analyse. On explique un ensemble des algorithmes tel que le LBP et LTP.

### **Chapitre 2** : la classification

Dans ce chapitre on a expliqué non seulement le type de classification, mais aussi quelques algorithmes de classification comme le SVM et le KNN.

### **Chapitre 3** : conception et implémentation

Dans ce dernier chapitre on a expliqué notre travaille de classification de masse mammaire en deux type maligne et bénigne, en expliquantla manière d'implémentation de notre système grâce à des outil python.

# **Chapitre 1**

## **Description de texture**

### **1.1 Introduction**



Dans les systèmes de vision par ordinateur, un objet est reconnaissable soit par sa couleur, par sa forme, ou bien par sa texture. Il n'existe pas de définition universelle pour la notion de texture, ce qui fait de l'analyse de texture une des problématiques les plus difficiles et d'actualité en traitement d'images. Intuitivement, la notion de texture est liée à l'aspect homogène d'une surface. Une des caractéristiques importantes de la texture, est son invariance à la translation. L'analyse de texture est utilisée dans plusieurs applications : détection des objets, restauration des images, reconnaissance des manuscrits, compression d'image, segmentation, ... etc. [Haralick, R.M.1973]

## 1.2 Définition detexteur

La définition littéraire de la texture est la suivante : "La répétition spatiale du même motif dans différentes directions spatiales". Cette définition est limitée car elle caractérise les objets indépendamment des observateurs humains. Le concept de texture est utilisé pour transformer l'aspect uniforme de la surface de l'objet sur l'image. Par conséquent, la texture s'exprime à travers des informations visuelles, qui peuvent être décrites qualitativement à l'aide des adjectifs suivants : rugueux, fin, lisse, tacheté, granuleux, marbré, régulier ou irrégulier. De nombreuses recherches psycho visuelles ont été menées sur la reconnaissance des textures du système visuel humain. Dans de nombreux cas, une conjecture importante et valable est que l'œil humain ne peut pas immédiatement distinguer deux textures avec les mêmes statistiques de second ordre. Cependant, dans certains cas, les textures avec les mêmes statistiques de second ordre peuvent toujours être distinguées en fonction des attributs locaux.

[Haralick 79] élargit la définition en décrivant une texture comme un phénomène à deux dimensions: la première concernant la description d'éléments de base ou primitives (le motif) à partir desquels est formée la texture; la deuxième dimension est relative à la description de l'organisation spatiale de ces primitives.

Unser présente la texture comme une structure disposant de certaines propriétés spatiales homogènes et invariantes par translation. Une autre approche serait encore de définir la texture à partir de deux types d'informations essentielles que comporte l'image:

- 1) les contours, de type monodimensionnel, qui marquent les frontières entre régions homogènes

2) l'aspect de surface, de type bidimensionnel, qui définit les régions homogènes. Cependant, la description d'une texture peut s'avérer erronée à une autre échelle d'observation, c'est à dire en changeant la résolution. [Unser, M. ;1995]

En pratique, on distingue deux grandes classes de textures, qui correspondent à deux niveaux de perception:

- 1) les macro-textures qui présentent un aspect régulier, sous formes de motifs répétitifs spatialement placés selon une règle précise (ex: peau de lézard, mur de brique) donc une approche structurelle déterministe.
- 2) les micro-textures présentant des primitives "microscopiques" distribuées de manière aléatoire (ex: sable, laine tissée, herbe) d'où une approche probabiliste cherchant à caractériser l'aspect anarchique et homogène.

Propose une synthèse des deux approches en considérant la texture comme "une structure spatiale constituée de l'organisation de primitives ayant chacune un aspect aléatoire, donc une structure hiérarchique à deux niveaux». [Pratt, William et al].

### **1.3 Analyse de texture:**

Dans le traitement d'image, l'analyse de texture consiste à calculer une série de mesures dans le but de définir une texture perçue sur une image. L'analyse de texture renvoie des informations sur l'arrangement spatial des couleurs ou des intensités dans tout ou partie de cette image.

Donc le but de l'analyse de texture est de formaliser les descriptifs de la texture par des paramètres mathématiques qui serviraient à l'identifier, il existe deux approches possibles pour analyser une texture l'approche structurée et l'approche statistique.

[Robert, M. Haralick et al 1973]

#### **1.3.1 Approche statistique :**

Ils étudient la relation entre les pixels et leurs voisins. Ils sont utilisés pour caractériser des structures fines et n'ont pas de régularité évidente. La recherche a été menée sur des structures hétérogènes complètement aléatoires et plus courantes, c'est pourquoi ce type de méthode est le mieux utilisé. Dans ce cas, la texture est décrite par les données statistiques de ces distributions de niveaux de gris (ou d'intensité). [2]

### **1.3.2 Approche structurelle :**

Ces méthodes prennent en compte la structure de la forme et les informations contextuelles, et sont particulièrement adaptées aux macrostructures. Ils permettent la description des textures en définissant des primitives et les règles d'arrangement qui les relient. En fait, les textures ordonnées ont des primitives qui répètent leurs positions dans l'image selon certaines règles.

### **1.4 Les attributs statistiques de texture :**

L'œil humain distingue facilement plusieurs textures est une tâche difficile en vision par ordinateur, car le nombre de textures est illimité, et chaque texture a ses propres caractéristiques. Il n'y a aucun moyen de classer toutes les textures comme un observateur humain. Cependant, certains attributs qui définissent l'impression visuelle peuvent être extraits, de sorte que la classification de texture effectuée par un observateur humain peut être mieux reproduite. Ces caractéristiques sont à la base de nombreuses méthodes d'analyse de texture. Ils permettent de définir un grand nombre d'attributs qui caractérisent la texture présente dans l'image. Ces attributs peuvent être géométriques, basés sur la modélisation de la texture spatiale, la fréquence spatiale et les statistiques. Les attributs statistiques sont utilisés pour caractériser tous types de textures, même les textures fines sans régularité évidente. Ces attributs peuvent être divisés en plusieurs catégories selon leur « ordre ». L'ordre des attributs dépend du type d'interaction spatiale entre les pixels considérés. Il est donné par le nombre de pixels impliqués dans le calcul des paramètres. Par exemple, pour les histogrammes d'images, nous ne nous intéressons qu'aux pixels eux-mêmes, ce paramètre appartient donc à la catégorie des attributs statistiques de premier ordre. Comme pour le calcul de la matrice de cooccurrence, on considère des paires de pixels, ce sont des attributs de second ordre. [3]

#### **1.4.1 Attributs statistiques du premier ordre :**

De nombreux attributs peuvent être extraits directement de l'image pour caractériser les caractéristiques de l'image. la texture qu'il contient. Les attributs statistiques de premier ordre fournissent des informations générales Caractériser l'image en fonction des changements d'intensité des pixels qui composent l'image. [4]

**Les plus utilise sont :**

1. La moyenne
2. La variance

3. Le moment d'ordre 3
4. Le moment d'ordre 4

#### **1.4.1.1 La moyenne :**

La valeur moyenne (ou intensité moyenne) des niveaux de gris de tous les pixels est donnée par :

$$MOY = \frac{1}{N} \sum_{x,y} I(x,y) \quad (1.1)$$

Avec  $I(x,y)$  représente la valeur du niveau de gris du pixel  $(x, y)$  et  $N$  correspond au nombre total des pixels.

#### **1.4.1.2 La variance :**

Elle correspond au moment d'ordre 2. Elle mesure la répartition des niveaux de gris autour de la valeur moyenne. Sa valeur est donnée par :

$$VAR = \frac{1}{N} \sum_{x,y} (I(x,y) - moy)^2$$

#### **1.4.1.3 Le moment d'ordre 3 :**

Le moment d'ordre 3 (en anglais skewness) centré autour de la moyenne, mesure la déviation de la distribution des niveaux de gris par rapport à une distribution symétrique. Il est donné par :

$$M3 = \frac{1}{N} \sum_{x,y} (I(x,y) - MOY)$$

Il mesure le niveau (degré d'asymétrie) par rapport à la moyenne. Dévier pour les valeurs élevées, le moment d'ordre 3 est positif, est pour les valeurs faibles, il est négatif.

#### **1.4.1.4 Le moment d'ordre 4 :**

Le moment d'ordre 4 centré autour de la moyenne (en anglais kurtosis) est donné par :

$$KURT = \frac{1}{N} \sum_{x,y} (I(x,y) - MOY)^4$$

Par rapport à la moyenne, de paramètre estime le niveau de concavité ou de convexité.

#### **1.4.2 Attributs statistique du second ordre :**

Dans les attributs de premier ordre, il n'y a aucune information sur les positions des pixels.

Il par conséquent, il est nécessaire d'utiliser des méthodes d'ordre supérieur pour une analyse plus précise. En particulier, nous conserverons la méthode de matrice de cooccurrence, également appelée dépendance spatiale des niveaux de gris.

Elle permet de déterminer la fréquence d'apparition d'un motif formé de deux pixels séparés par une certaine distance « d » dans une direction particulière  $\theta$ . Afin de limiter le nombre de calculs, on prend généralement comme valeurs de la direction  $\theta$ :  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ,  $180^\circ$  et 1 pour la valeur de « d ». Ces indices bien que corrélés, réduisent l'information contenue dans la matrice de cooccurrence et permettent une meilleure discrimination des textures.

Ces indices bien que corrélés, réduisent l'information contenue dans la matrice de cooccurrence et permettent une meilleure discrimination des textures.<sup>38</sup> Le choix du vecteur de déplacement et de la taille de la fenêtre du voisinage sur laquelle s'effectue la mesure, sont les paramètres sur lesquels reposent la réussite de la méthode. La difficulté à surmonter lors de l'application de cette technique réside justement dans le choix de ces paramètres car ils varient selon le type d'images et de textures.

A chaque direction et pour chaque valeur de d correspond une matrice de cooccurrence  $\phi(d, \theta)$ .

La figure (1.1) illustre un exemple de construction de la matrice de cooccurrence pour  $\theta=0^\circ$  et  $d=1$ .

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 1 | 5 | 6 | 8 |
| 2 | 3 | 5 | 7 | 1 |
| 4 | 5 | 7 | 1 | 2 |
| 8 | 5 | 1 | 2 | 5 |

Image initiale

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 2 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

matrice de cooccurrence ( $d=1, \theta=0^\circ$ )

**Table 1.** Exemple de construction d'une matrice de cooccurrence

On définit généralement les matrices symétriques de cooccurrence. Elles sont construites à partir des constatations suivantes :

$$\phi(d, 0^\circ) = \phi^t(d, 180^\circ)$$

$$\phi(d, 45^\circ) = \phi^t(d, 225^\circ)$$

$$\phi(d, 90^\circ) = \phi^t(d, 270^\circ)$$

$$\phi(d, 135^\circ) = \phi^t(d, 315^\circ)$$

Par exemple, la matrice symétrique associée à la direction  $0^\circ$  sera de la forme :

$$S_0(d) = \frac{1}{2} [\phi(d, 0^\circ) + (d, 180^\circ)]$$

Une fois la matrice symétrique générée, une quinzaine de paramètres peuvent être extraits, qui contiennent des informations sur la douceur et la granularité de la texture. Pour les textures rugueuses, les valeurs de la matrice sont concentrées sur la diagonale principale. Au contraire, pour les textures fines, les valeurs de la matrice seront éparées : en effet, pour de telles textures, il existe de nombreuses transitions en niveaux de gris. De cette matrice de cooccurrence, nous pouvons extraire la moyenne, la variance, l'énergie, le contraste, la corrélation et l'homogénéité. La valeur moyenne est donnée par la relation suivante :

$$MOY = \frac{1}{Ng + Ng} \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} p(i, j)^2$$

$P(i, j)$ : correspond aux éléments de la matrice de cooccurrence, c'est-à-dire à la probabilité de passer d'un pixel de niveau de gris  $i$  à un pixel de niveau de gris  $j$ .

$I, j$ : Deux niveaux de gris.

$Ng$  : correspond au maximum des niveaux de gris de l'image.

La variance est définie par :

$$VAR = \frac{1}{Ng + Ng} \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} (i - MOY)^2 p(i, j)$$

Elle caractérise la distribution des niveaux de gris autour de la valeur moyenne  $MOY$  calculée précédemment.

L'énergie ou moment angulaire d'ordre deux mesure l'homogénéité de l'image. L'énergie a une valeur d'autant plus faible qu'il y a peu de zones homogènes : dans ce cas, il existe beaucoup de transitions de niveaux de gris

$$E = \frac{1}{Ng + Ng} \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} p(i, j)^2$$

Le contraste ou (l'inertie) mesure les variations locales des niveaux de gris. Si elles sont importantes (c'est-à-dire s'il existe peu de régions homogènes), alors le contraste sera élevé.

Ce paramètre permet aussi de caractériser la dispersion des valeurs de la matrice par rapport à sa diagonale principale.

$$\text{CONT} = \frac{1}{N_g + N_g} \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i, j)^2 p(i, j)$$

La corrélation est donnée par:

$$\text{COR} = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} ijp(i, j) - uiuj}{oi oj}$$

### 1.4.3 Attributs statistiques d'ordre supérieur:

L'interaction entre plusieurs pixels est étudiée pour extraire des propriétés statistiques d'ordre supérieur. La méthode de la longueur de la plage de gris est l'une des méthodes les plus connues. Cela comprend le calcul du nombre de plages de niveaux de gris  $i$  d'une certaine longueur  $j$  dans une direction donnée  $\theta$  (généralement  $\theta$  est  $(0^\circ, 45^\circ, 90^\circ$  ou  $135^\circ)$ . Ainsi, la plage de niveaux de gris  $i$  correspond au même Tous les pixels de l'image avec des valeurs de gris. La longueur de la plage correspond au nombre de pixels appartenant à la plage. Par conséquent, chaque direction correspondra à une matrice  $R(\theta)$ , qui est donnée par la formule suivante :

$$R(\theta) = [r(i, j)]$$

Avec : élément de la matrice  $R(\theta)$ , représente le nombre de plages de niveau de gris  $i$ , de longueur  $j$  dans la direction. La figure suivante donne un exemple de construction de la matrice de longueur de plages de niveau de gris.

|   |   |   |   |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 0 | 2 | 3 | 3 |
| 2 | 1 | 1 | 1 |
| 3 | 0 | 3 | 0 |

Image initiale

|   |   |   |   |   |   |
|---|---|---|---|---|---|
|   | J | 1 | 2 | 3 | 4 |
| I | 0 | 4 | 0 | 0 | 0 |
|   | 1 | 4 | 2 | 1 | 0 |
|   | 2 | 3 | 0 | 0 | 0 |
|   | 3 | 5 | 1 | 0 | 0 |

Matrice  $R(0^\circ)$  de longueur de plage associées

**Table2** : exemple de construction de la matrice de plages de niveaux de gris

Les principaux paramètres issus de la matrice de longueur de plages sont le « poids » des plages courtes, le poids des plages longues et la distribution des niveaux de gris. Le poids des plages courtes caractérise la finesse de la texture. Plus la valeur de cet attribut est grande, plus les zones ayant le même niveau sont petites. Il est défini par la relation :

$$SRE = \frac{1}{N} \sum_{i=0}^{M-1} \sum_{j=1}^N \frac{r(i,j)}{j^2}$$

Avec  $nr = \sum_{i=0}^{M-1} \sum_{j=1}^N r(i,j)$  : représente le nombre total des plages de l'image.

M correspond au nombre de niveaux de gris dans l'image et N à la longueur de la plage maximale.

Le « poids » des plages longues caractérise les textures grossières. Plus la valeur de cet attribut est grande plus il y'a des zones étendues ayant le même niveau. Il est donné par :

$$LRE = \frac{1}{nr} \sum_{i=0}^{M-1} \sum_{j=0}^N j^2 r(i,j)$$

### 1.3.2 Motifs binaires locaux :

Les **motifs binaires locaux (LBP)**, (*Local Binary Pattern* en anglais) est un opérateur de texture simple mais très efficace qui étiquette les pixels d'une image en seuil le voisinage de chaque pixel et considère le résultat comme un nombre binaire. Sont un descripteur de texture rendu populaire par le travail d'Ojala et coll. dans leur article de 2002, Multi-résolution Grays-scale and Rotation Invariant Texture Classification avec Local Binary Patterns (bien que le concept de LBP ait été introduit dès 1993). En raison de sa puissance discriminante et de sa simplicité de calcul, l'opérateur de texture LBP est devenu une approche populaire dans diverses applications. Il peut être considéré comme une approche unificatrice des modèles statistiques et structurels traditionnellement divergents de l'analyse de la texture. Peut-être la propriété la plus importante de l'opérateur LBP dans les applications réelles est sa robustesse aux changements mono-toniques à l'échelle grise causés, par exemple, par des variations

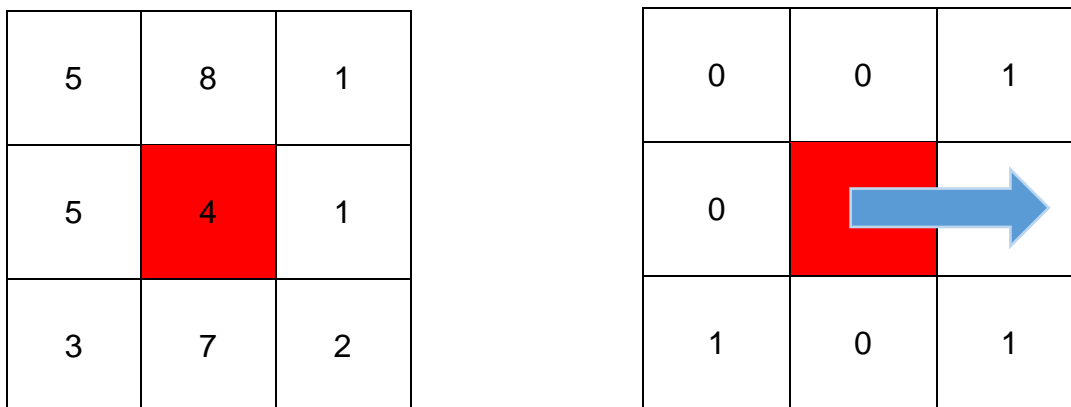


d'éclairage. Une autre propriété importante est sa simplicité de calcul, qui permet d'analyser des images dans des environnements difficiles en temps réel. [Ojala, Timo, et al 1996]

### 1.3.2.1 Etape d'algorithme :

La première étape dans la construction du descripteur de texture LBP est de convertir l'image en échelle grise. Pour chaque pixel de l'image à échelle grise, nous sélectionnons un quartier de taille  $r$  entourant le pixel central. Une valeur LBP est ensuite calculée pour ce pixel central et stockée dans le tableau 2D de sortie avec la même largeur et la même hauteur que l'image d'entrée.

Par exemple, jetons un coup d'œil au descripteur LBP original qui fonctionne sur un *quartier fixe de 3 x 3 pixels* comme ceci :

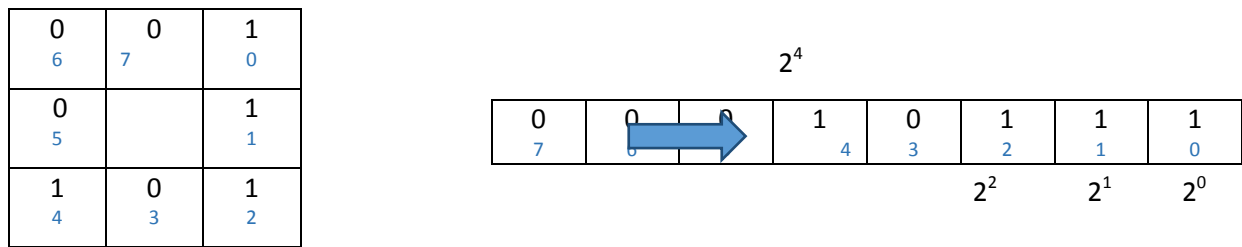


**Figure 1.1:** la première étape dans la construction d'un LBP est de prendre le quartier de 8 pixel entourant un pixel central et le seuil pour construire un ensemble de 8 chiffres binaires

Dans la figure ci-dessus, nous prenons le pixel central (mis en surbrillance en rouge) et le seuil contre son quartier de 8 pixels. Si l'intensité du pixel central est supérieure ou égale à celle de son voisin, alors nous définissons la valeur à 1; sinon, nous l'avons mis à 0. Avec 8 pixels environnants, nous avons un total de  $2^8 = 256$  combinaisons possibles de codes LBP.

À partir de là, nous devons calculer la valeur LBP pour le pixel central. Nous pouvons commencer à partir de n'importe quel pixel voisin et travailler notre chemin dans le sens des aiguilles d'une montre ou dans le sens inverse des aiguilles d'une montre, mais notre commande doit être maintenue cohérente pour tous les pixels de notre image et toutes les images de notre ensemble de données. Compte tenu d'un quartier 3 x 3, nous avons donc 8

voisins que nous devons effectuer un test binaire sur. Les résultats de ce test binaire sont stockés dans un tableau 8 bits, que nous convertissons ensuite en décimale, comme ceci:

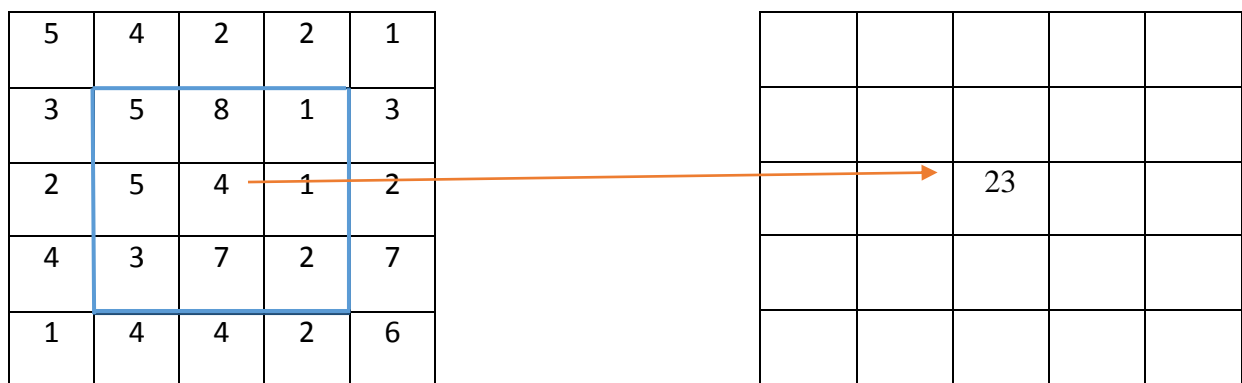


$$16 + 4 + 2 + 1 = 23$$

**figure 1.2** : prendre le quartier binaire 8 bits du pixel central et de convertir en une représentation décimale.

Dans cet exemple, nous commençons au point en haut à droite et travaillons notre chemin dans le sens des aiguilles d'une montre en accumulant la chaîne binaire au fur et à mesure. Nous pouvons ensuite convertir cette chaîne binaire en décimale, donnant une valeur de 23.

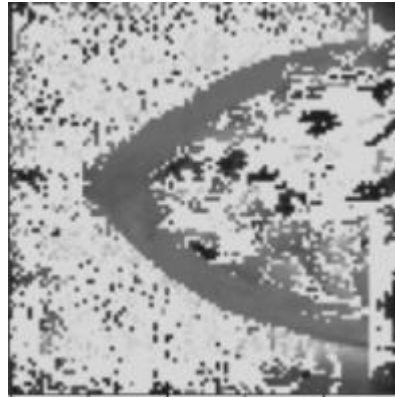
Cette valeur est stockée dans le tableau de sortie LBP 2D, que nous pouvons ensuite visualiser ci-dessous :



**Figure 1.3** : exemple de résultat obtenue par LBP.

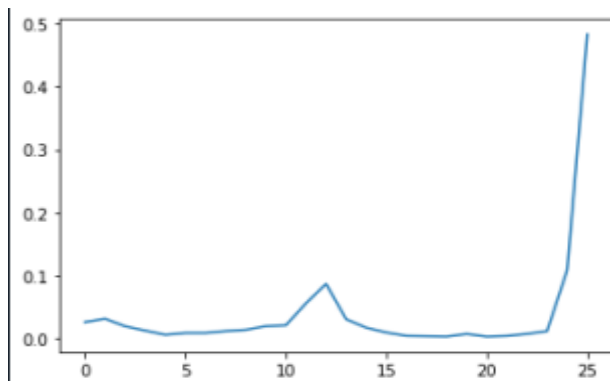
Ce processus de seuil, d'accumulation de chaînes binaires et de stockage de la valeur décimale de sortie dans le tableau LBP est ensuite répété pour chaque pixel de l'image d'entrée.

Voici un exemple de calcul et de visualisation d'un tableau LBP 2D complet :



**Figure 1.4:** capteur d'écran sur un exemple d'image résultat de LBP

La dernière étape consiste à calculer un histogramme sur le tableau de sortie LBP. Depuis un *quartier*  $3 \times 3$  à  $2^8 = 256$  modèles possibles, notre tableau LBP 2D a donc une *valeur minimale de 0* et une valeur maximale de 255, ce qui nous permet de construire un histogramme de 256 bacs de codes LBP comme notre vecteur de fonctionnalité finale:



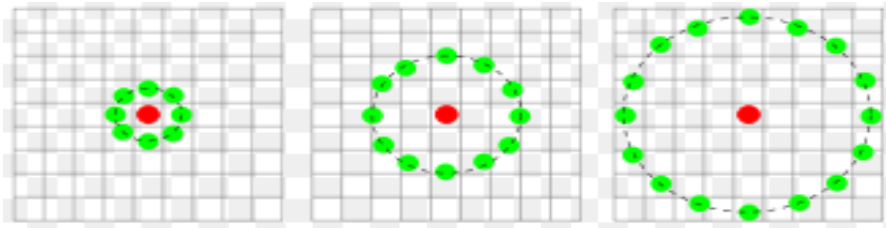
**Figure 1.5:** histogramme LBP en peut considère les valeurs du l'histogramme comme des caractéristique d'image (capteur d'écran).

Un des principaux avantages de cette implémentation LBP originale est que nous pouvons capturer des détails extrêmement fins dans l'image. Cependant, être capable de capturer des détails à une si petite échelle est également le plus grand inconvénient de l'algorithme - nous ne pouvons pas capturer des détails à différentes échelles, seulement l'échelle fixe  $3 \times 3$ !

Pour ce faire, Ojala et coll. ont proposé une extension de la mise en œuvre originale du LBP pour gérer les tailles variables des quartiers. Pour tenir compte des tailles variables des quartiers, deux paramètres ont été introduits :

1. Le nombre de points  $p$  dans un quartier circulairement symétrique à considérer (supprimant ainsi en s'appuyant sur un quartier carré).
2. Le rayon du cercle  $r$ , ce qui nous permet de tenir compte de différentes échelles.

Voici une visualisation de ces paramètres:



**Figure 1.6:** Trois exemples de quartier avec différents  $p$  et  $r$  utilisés pour construire des modèles binaires locaux.

Enfin, il est important que nous considérons le concept d'uniformité LBP. Un LBP est considéré comme uniforme s'il a tout au plus deux transitions 0-1 ou 1-0. Par exemple, le modèle 00001000 (2 transitions) et 10000000 (1 transition) sont tous deux considérés comme des modèles uniformes puisqu'ils contiennent tout au plus deux transitions 0-1 et 1-0. Le modèle 01010010) d'autre part n'est pas considéré comme un modèle uniforme car il a six transitions 0-1 ou 1-0.

Le nombre de prototypes uniformes dans un modèle binaire local dépend complètement du nombre de points  $p$ . Comme la valeur de  $p$  augmente, de même que la dimensionnalité de votre histogramme résultant. Veuillez consulter le document original d'Ojala et coll. pour obtenir l'explication complète sur le nombre de motifs et de modèles uniformes basés sur cette valeur. Toutefois, pour l'instant il suffit de garder à l'esprit que compte tenu du nombre de points  $p$  dans le LBP il y a  $p + 1$  modèles uniformes. La dimensionnalité finale de l'histogramme est donc  $p + 2$ , où l'entrée ajoutée compile tous les modèles qui ne sont pas uniformes.

Alors pourquoi les modèles uniformes de LBP sont-ils si intéressants ? En d'autres termes: ils ajoutent un niveau supplémentaire de rotation et d'invariance à l'échelle grise, d'où ils sont couramment utilisés lors de l'extraction des vecteurs de fonctionnalité LBP à partir d'images.

### 1.3.3 Les motifs ternaires locaux (LTP)

Sont une extension des motifs binaires locaux (LBP). Contrairement à LBP, il ne limite pas

|    |   |    |
|----|---|----|
| -1 | 1 | -1 |
|----|---|----|

les pixels en 0 et 1, il utilise plutôt une constante de seuil pour permuter les pixels en trois valeurs. En considérant k comme

constante de seuil, c comme valeur du pixel central, un pixel voisin p, le résultat de seuil est :

$$\begin{cases} 1 & \text{si } p > c + k \\ 0 & \text{si } p > c - k \text{ et } p < c + k \\ -1 & \text{si } p < c - k \end{cases}$$

De cette façon, chaque pixel possède une des trois valeurs. Les pixels voisins sont combinés après seuillage en un motif ternaire. Le calcul d'un histogramme de ces valeurs ternaires entraînera une large plage, de sorte que le motif ternaire est divisé en deux motifs binaires. Les histogrammes sont concaténés pour générer un descripteur double de la taille de LBP.

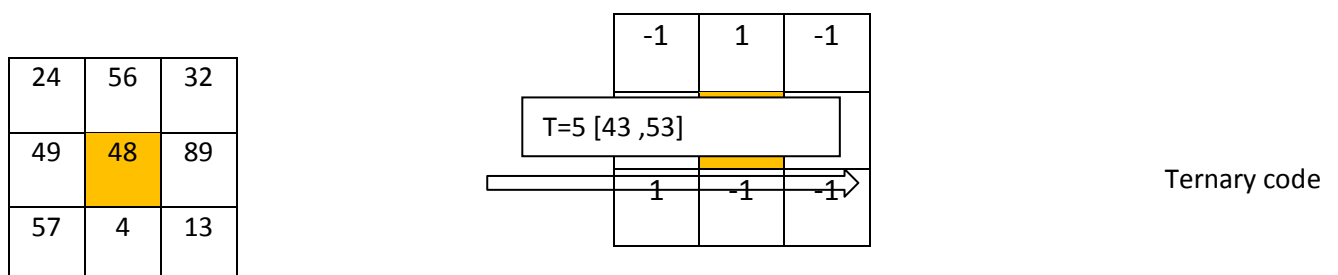
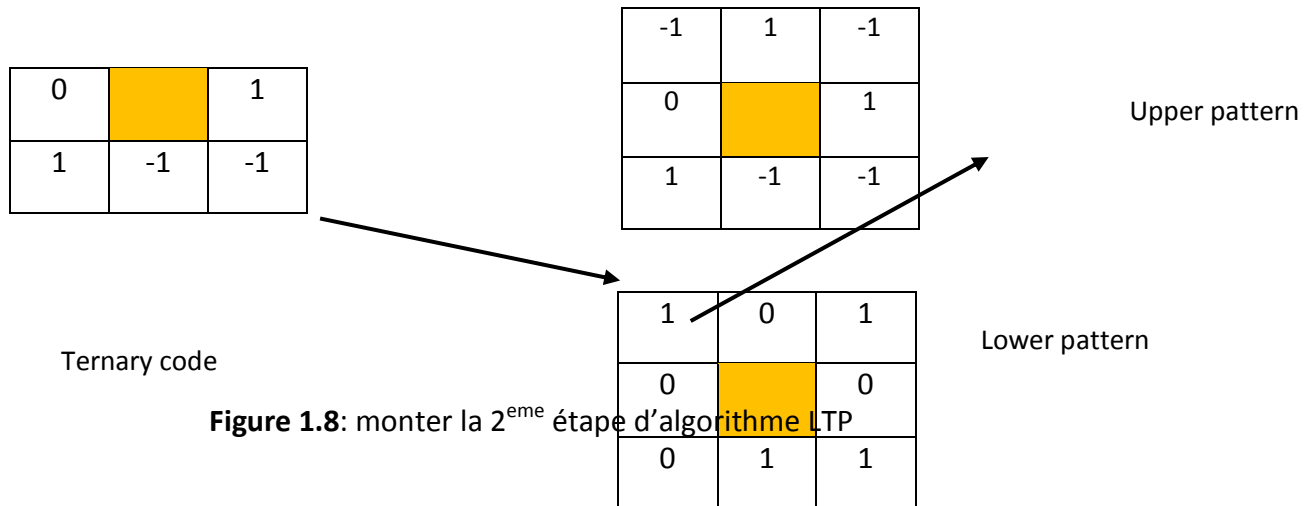


Figure 1.7: étapes d'extraire le pattern code

Dans la figure ci-dessus, nous prenons le pixel central (mis en surbrillance en orange) et le seuil contre son quartier de 8 pixels. Si l'intensité du pixel central moins la constante de seuil est supérieure à celle de son voisin, alors nous définissons la valeur à -1; sinon, si l'intensité du pixel central plus la constante de seuil est moins à celle de son voisin, alors nous

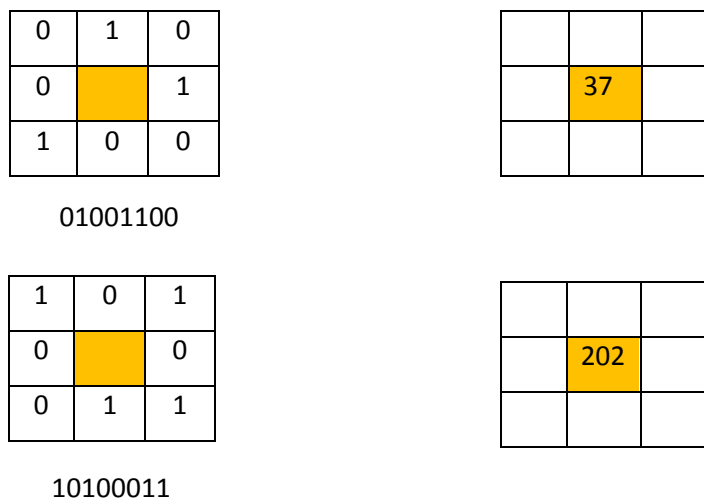
©ZÉdéfinissons la valeur à 1; sinon nous allons la mettre à 0.



Dans cette étape d'algorithme on remplit deux tableaux de même taille 3\*3, **upper pattern** et **lowerpattern** comme ce suit :

Upper pattern (positive code) : on remplace les valeurs de -1 dans le ternary code avec 0 en upper pattern (positive code).

Lower pattern (négative code) : en remplace la valeur -1 dans le ternary pattern avec 1 dans lower pattern enremplissantles autres valeurs par 0.



**Figure 1.9:** transformation de code binaire en décimale

Dans cet exemple, nous commençons au point en haut à droite et travaillons notre chemin dans le sens des aiguilles d'une montre en accumulant la chaîne binaire au fur et à mesure. On convertit la chaîne binaire possédée en décimale ; par la suite elle donnera la valeur 37 dans le upper pattern et 202 dans le lower pattern. Ces valeurs se stockent dans les tableaux upper pattern et lower pattern de même taille d'image ; la dernière étape d'algorithme LTP c'est de calculer un histogramme de chacun des tableaux résultants ; et faire la concaténation de ces histogrammes. L'histogramme résultat peut être considéré comme un vecteur de caractéristique de cette image. [Naghashi, Vahid, 2018]

#### **1.4 Conclusion :**

De nos jours, les bases d'imagerie sont devenues l'un des piliers importants dans le domaine de la vision par ordinateur et de l'IA (intelligence artificielle). Cela est dû à son importance où peut être utilisé dans de nombreuses applications. La clé principale de la reconnaissance d'image est de savoir comment extraire des caractéristiques distinctives de l'image pour obtenir une précision de reconnaissance élevée, dans notre cas nous avons utilisées des mammographies pour aider les médecins à détecter la tumeur du sein à un temps précoce. Motif binaire local (LBP) et plusieurs de ses variantes utilisées comme caractéristiques de texture dans de nombreux systèmes de reconnaissance faciale.

Bien que la LBP ait bien fonctionné dans de nombreux domaines, elle est sensible au bruit et différents modèles de LBP peuvent être classés dans la même classe, ce qui réduit sa propriété discriminante. Le motif ternaire local (LTP) est l'une des nouvelles fonctionnalités de texture proposées pour surmonter les inconvénients du LBP.

Dans cette étude, nous étudions les performances de l'opérateur LTP et LBP pour la tâche de classification d'image. La base de données DDSM est utilisée dans les expériences.

## Chapitre 2

# La classification



## **2.1 Introduction :**

La classification automatique est une méthode d'analyse, qui simplifie les données en les regroupant en catégories possédant des caractéristiques communes. La masse grandissante des données et la facilité de leur stockage, a fait que les applications de la classification soient très nombreuses. Elles concernent les problèmes liés à la prise de décision automatisée, à savoir : la reconnaissance des formes, chiffres, caractères manuscrits, images, paroles, signaux temporels ainsi que l'économie, les finances, la sociologie, la biologie... etc. dans la domaine médicale la classification aide a diagnostic les anomalies telle que le cancer sien de peau...etc.

## **2.2 Définition :**

La classification consiste à construire les règles qui déterminent l'affectation de chaque donnée, à une des classes existantes. Les éléments appartenant à chacune des classes, ont la particularité d'avoir des caractéristiques communes. Une distinction doit être faite entre les termes : classification et classement. Le classement est le fait d'affecter des données à un ensemble de classes suivant des règles préétablies par la classification. La classification automatique prend une place importante dans l'analyse des données. L'automatisation est due au surcroît du volume des données, et au grand flux d'informations généré dans plusieurs domaines, ce qui fait de l'automatisation un moyen incontournable pour la résolution du problème.

Dans le contexte des images satellites, les données à classifier sont représentées par les valeurs radiométriques des pixels et/ou leurs propriétés spatiales, dans une ou plusieurs bandes. L'intervention d'un spécialiste dans un processus de classification reste importante dans:

- La spécification des classes thématiques qui seront prises en considération dans la classification, ainsi que les paramètres à utiliser: descripteurs de couleur, forme, texture.
- Dans le cas d'une classification supervisée, l'étiquetage d'un échantillon de données est requis, pour effectuer l'apprentissage

## **2.3 Méthodes de classification :**

Les méthodes de classification ont pour but d'identifier les classes auxquelles appartiennent des objets à partir de certains traits descriptifs.

Les techniques de classification peuvent être classées en deux catégories: le mode supervisé et le non supervisé. Si l'on dispose d'un échantillon de données étiqueté, on parle de classification supervisée. Dans le cas contraire, il s'agit de classification non supervisée, appelée aussi catégorisation.

### **2.3.1 Méthodes de classification non supervisée :**

Le principe des méthodes de classification non supervisées est de chercher des partitions d'un ensemble de données, en fonction de critères de proximité de leurs vecteurs d'attributs, dans l'espace de représentation.

Elles représentent l'action de découper une image en classes sans savoir à quoi correspondent ces classes thématiquement. C'est une classification en aveugle, une segmentation sans connaissance à priori.

Parmi les méthodes de classification non supervisée, on trouve:

#### **2.3.1.1 Méthode de K-Means :**

La plus célèbre méthode de classification non supervisée, appelée aussi algorithme des centres mobiles, où chaque classe est représentée par son centre de gravité.

Au début  $K$  centres  $c_1, c_2, c_3, \dots, c_K$  sont définis arbitrairement, chaque  $C_i$  représente le centre d'une classe  $c_i$   $1 \leq i \leq k$  dont les éléments sont le plus proches possible du centre  $C_i$ , que tout autre centre. Une nouvelle itération consiste à regrouper les éléments autour des nouveaux centres de gravités  $m_j$  qui remplacent  $C_i$ , calculés à partir des classes obtenus dans l'itération précédente. Le processus est réitéré jusqu'à atteindre un état de stabilité ou aucune amélioration n'est possible.

Cette méthode présente l'avantage d'être convergente et rapide. Les expériences ont montré que le nombre d'itérations nécessaire à la convergence est inférieur à 20. L'inconvénient de la méthode des  $K$ -means, est que le résultat dépend de la partition initiale. C'est pour cela, qu'il est conseillé d'exécuter l'algorithme plusieurs fois et de comparer les résultats obtenus.

Elle a aussi tendance à regrouper des classes de mêmes tailles, ce qui peut fausser la classification, en cas de présence de classes de tailles différentes

#### **2.3.1.2 Méthode Ascendante Hiérarchique :**

Il s'agit d'une méthode qui calcule une hiérarchie de partitions. Les différentes partitions de la hiérarchie sont déterminées d'une manière itérative, en commençant par la partition la plus fine qui est composée de l'ensemble des singletons. La classification hiérarchique utilise cette partition initiale et procède à des regroupements successifs en fusionnant à chaque itération deux sous-ensembles d'individus. Le choix des deux sous-ensembles qui sont appelés à se fusionner, à une étape donnée, est effectué à l'aide d'une mesure de similitude, définie entre deux sous-ensembles.

Le nombre de sous-ensembles générés est variable car le processus de regroupement peut être arrêté à tout moment.

#### **2.3.2.3 Méthode des modes de l'histogramme :**

Cette méthode est basée sur les modes de l'histogramme de l'image. Le mode correspond à un maximum local de l'histogramme. Il indique un regroupement de pixels et sert à détecter les classes.

L'algorithme de la classification par modes d'histogramme se déroule comme suit :

- Construire l'histogramme de l'image.
- Détecter K mode à partir de l'histogramme ; K étant le nombre de classes souhaité.
- Considérer les K modes comme centres des classes. - Regrouper les pixels de l'image selon le critère de distance minimale par rapport aux différents centres de classes. Chaque pixel est placé dans la classe dont le centre lui est le plus proche.

#### **2.3.2 Méthodes de classification supervisée**

La classification supervisée, ou assistée, consiste à établir un modèle à partir d'un échantillon de données étiquetées, appelés l'ensemble d'apprentissage. Le model construit sera en mesure de prédire les classes d'appartenance du reste des éléments.

Les principales méthodes de classification supervisée sont les suivantes:

### **2.3.2.1 K plus proche voisins(KNN) :**

En intelligence artificielle, plus précisément en apprentissage automatique, la méthode des k plus proches voisins est une méthode d'apprentissage supervisé. En abrégé k-NN ou KNN, de l'anglais k-nearest neighbors.

La méthode k-NN est basée sur l'apprentissage préalable, ou l'apprentissage faible, où la fonction est évaluée localement, le calcul définitif étant effectué à l'issue de la classification. L'algorithme k-NN est parmi les plus simples des algorithmes de Machines Learning.

Les voisins sont pris depuis un ensemble d'objets pour lesquels la classe (en classification k-NN) ou la valeur (pour une régression k-NN) est connue. Ceci peut être considéré comme l'ensemble d'entraînement pour l'algorithme, bien qu'un entraînement explicite ne soit pas particulièrement requis.

Une particularité des algorithmes KNN est d'être particulièrement sensible à la structure locale des données.

### **2.3.2.2 Distance minimale :**

La classification par distance minimale nécessite le calcul au préalable des vecteurs moyens de toutes les classes à partir des données d'apprentissage. La classification d'un nouveau point, revient à calculer les distances entre celui-ci et chacun des centres de classes. Le point est affecté à la classe correspondant à la distance minimale.

Elle peut donner de bon résultats, et similaire à d'autres algorithmes dont coût de calcul est important.

L'inconvénient de cette méthode est que l'utilisateur doit fournir les centres de gravités des données d'apprentissage. Le résultat de la classification dépend du type de distance utilisée.

### **2.3.2.3Analyse factorielle discriminantes :**

L'analyse factorielle discriminante a pour but de trouver les axes factoriels qui discriminent au mieux les classes de l'ensemble d'apprentissage, tel que : les projections des individus d'une même classe soient les plus proches possible, alors que celles des autres classes sont les plus éloignées possible.

#### **2.3.2.4 Les réseaux de neurones :**

Les réseaux de neurones ont été largement utilisés dans la classification des images satellites. Leur succès est dû à leurs capacité d'apprentissage et de généralisation. C'est une méthode non paramétrique, ce qui explique sa large utilisation pour les images de télédétection, où les données sont issues de différentes sources et peuvent avoir des distributions statistiques différentes. Le modèle neuronal perceptron multi couche, entraîné avec l'algorithme de rétro-propagation a été utilisé avec succès dans la classification des images satellites.

Le choix du nombre de neurones par couche et le nombre de couche cachée utilisé, sont des paramètres qui influent sur le résultat de la classification et doivent être fournis au préalable.

#### **2.3.2.5 Classification supervisée par SVM (Séparateur à Vaste Marge) :**

Introduction L'apprentissage artificiel a connu un essor considérable ces dernières années ; dû essentiellement au besoin croissant des utilisateurs en matière de fouille automatique de grandes bases de données. Les techniques de classification supervisée ont suscité beaucoup d'intérêt par la communauté scientifique. Les travaux de Vladimir Vapnik (1995), ont bouleversé le domaine de l'apprentissage artificiel. Depuis, SVM est devenu un moyen de référence pour la classification automatique des données, et souvent cité dans des études comparatives avec d'autres méthodes. La méthode a montré son efficacité dans plusieurs domaines, tels que : la vision par ordinateur, la catégorisation de textes, la biométrie, bio-informatique, la finance...etc. SVM doit sa popularité à sa robustesse mathématique ainsi qu'à sa performance face à un ensemble de données de grande dimension. L'utilisation de S.V.M pour la classification des images aériennes et satellites par Maddoulayne (2009), Riffi (2011), Maiza (2012), a donné d'excellents résultats.

##### **2.3.2.5.1 Définition :**

Les SVM (Support Vecteurs Machines ou Séparateur à Vaste Marge) sont des techniques d'apprentissage supervisées initiées par Vladimir Vapnik en 1995, destinées à résoudre des problèmes de discrimination et de régression. La méthode SVM est fondée sur des bases théoriques robustes. Elle a aussi montré sa performance même en présence d'une grande base de données, ce qui explique son grand succès.

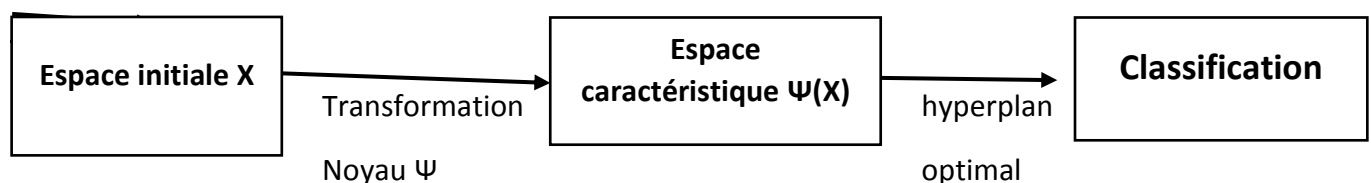
### 2.3.2.5.2 Historique :

Les SVM sont fondés sur deux notions importantes: la notion de marge maximale et celle de fonction noyau. La notion de marge maximale fut explorée en 1963 par Vapnik. La théorie d'apprentissage et la théorie de Vapnik-Chervonenkis élaborée en 1970, constituent les bases théoriques de SVM. Les fonctions noyaux ont été traitées dans le théorème de Mercer en 1909. Leur application à l'apprentissage artificiel fut montrée en 1967 par Aizermann, Bravermann et Rozoener. Ces travaux ont fait l'objet d'une publication, par les chercheurs Boser, Guyon et Vapnik, considérée comme le point de départ de SVM [Bose & al.92]. En 1995, Vapnik, introduit un classificateur à marge souple [Vapn.95]. Vladimir Vapnik a détenu un brevet américain en 1997.

### 2.3.2.5.3 Principe de SVM:

Le principe de SVM est de projeter en premier, les données initiales, non linéairement séparables, dans un espace de grande dimension, appelé espace des caractéristiques. Ensuite, il s'agit de trouver l'équation d'un hyperplan dans cet espace des caractéristiques, séparant les classes des données de façon optimale telles que:

1. Les données appartenant à deux classes différentes soient nettement séparées par l'hyperplan, et se trouvent dans deux cotés différents.
2. Maximiser la plus petite distance entre les données et l'hyperplan.

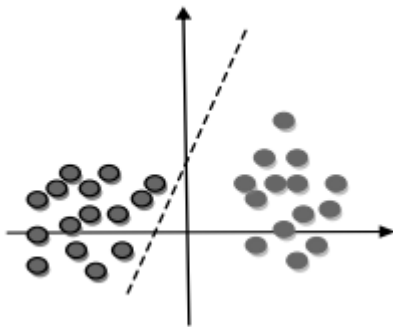


**Figure 2.1** : principe SVM

**2.3.2.5.4 Notation de base :**

**Hyperplan :**

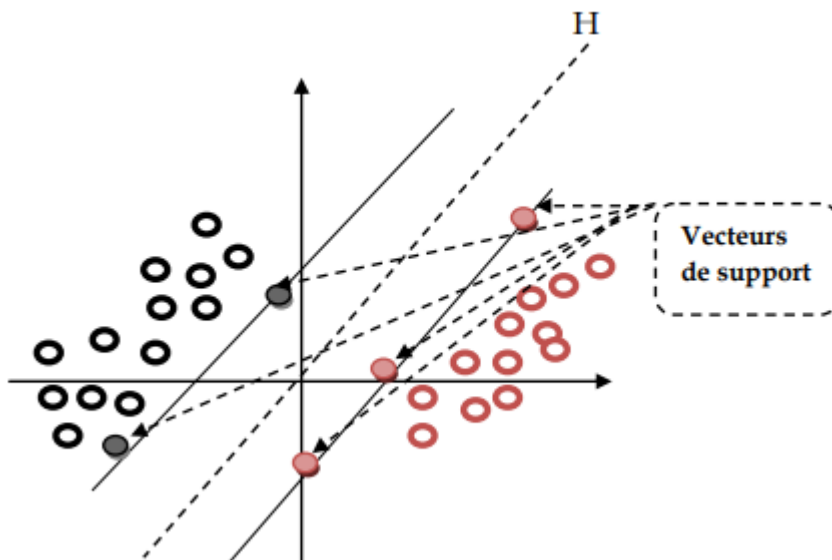
Plaçons-nous dans le cas d'une classification binaire (i.e. les exemples à classifier réparties en 2 classes). On appelle hyperplan séparateur un hyperplan qui sépare deux classes figure.1, en particulier il sépare leurs points d'apprentissage. Comme il n'est en générale pas possible d'en trouver un, on se contentera donc de chercher un hyperplan discriminant qui est une approximation au sens d'un critère à fixer (maximiser la distance entre ces deux classes).



**Figure 2.2:** hyperplan sépare deux classes [5]

**Vecteurs de support :**

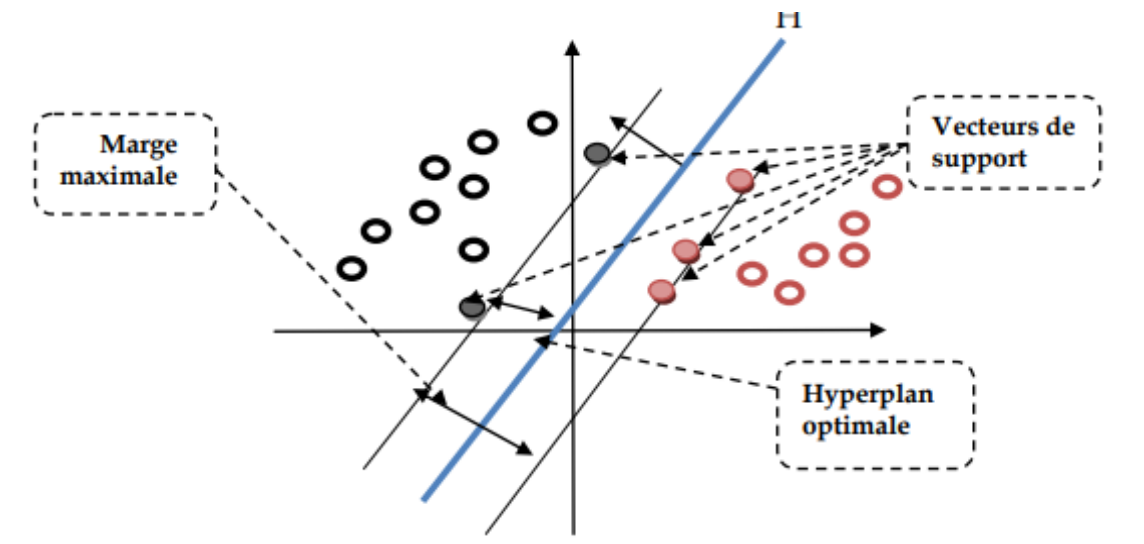
Pour une tâche de détermination de l'hyperplan séparable des SVM est d'utiliser seulement les points les plus proches (i.e. les points de la frontière entre les deux classes des données) parmi l'ensemble total d'apprentissage, ces points sont appelés vecteurs de support.



**Figure 2.3:** les vecteurs de support.[5]

**Marge :**

Il existe une infinité d'hyperplans capable de séparer parfaitement les deux classes d'exemples. Le principe des SVM est de choisir celui qui va maximiser la distance minimale entre l'hyperplan et les exemples d'apprentissage (i.e. la distance entre l'hyperplan et les vecteurs de support), cette distance est appelée la marge.



**Figure2.4 :** hyperplan optimal, vecteurs de support et marge maximal [5]

**2.3.2.5.5 Propriétés fondamentale :**

Intuitivement, le fait d'avoir une marge plus large procure plus de sécurité lorsqu'on classe un nouvel exemple. De plus, si l'on trouve le classificateur qui se comporte le mieux vis-à-vis des données d'apprentissage, il est clair qu'il sera aussi celui qui permettra au mieux de classer les nouveaux exemples. Dans le schéma figure.4, la partie droite nous montre qu'avec un hyperplan optimal, un nouvel exemple reste bien classé alors qu'il tombe dans la marge. On constate sur la partie gauche qu'avec une plus petite marge, l'exemple se voit mal classé.[B. Taachouche, O. Douak ;2009]



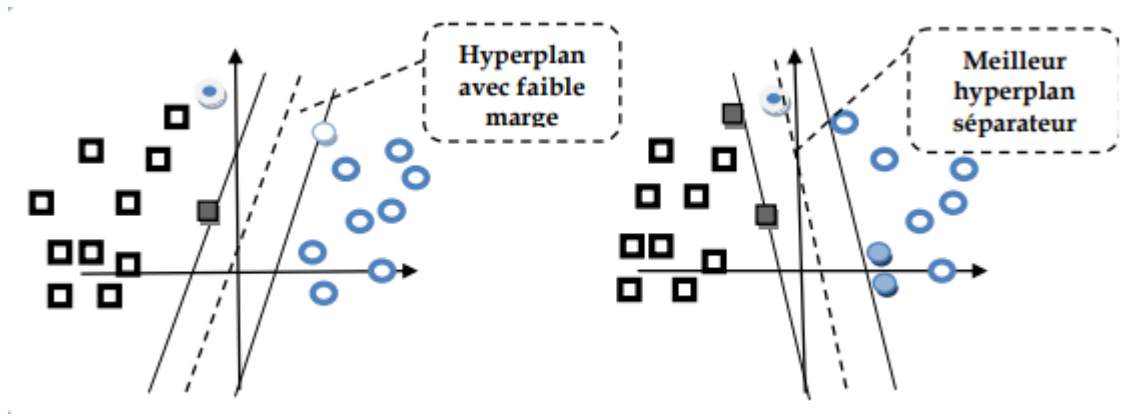


Figure 2.5 : meilleur hyperplan séparateur [5]

### Linéarité et non-linéarité :

Parmi les modèles des SVM, on constate les cas linéairement séparables et les cas non linéairement séparables. Les premiers sont les plus simples des SVM car ils permettent de trouver facilement le classificateur linéaire. Dans la plupart des problèmes réels il n'y a pas de séparation linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables.

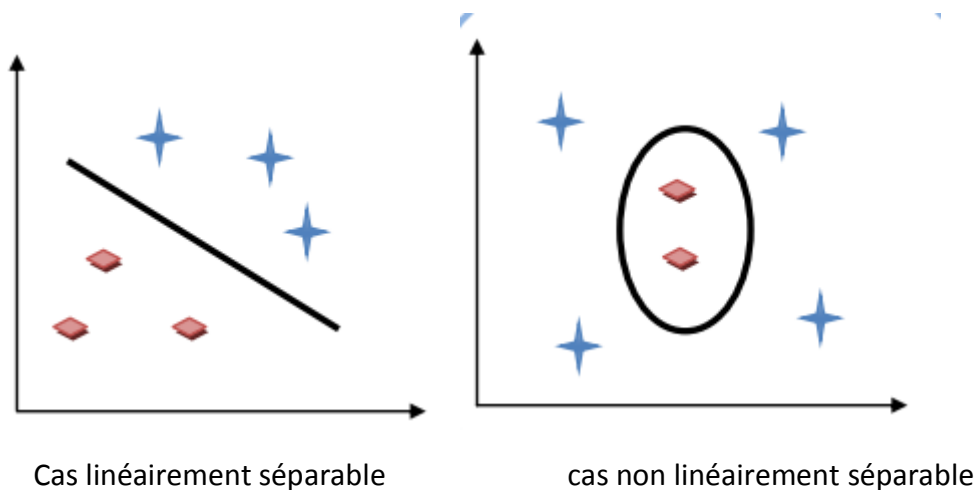
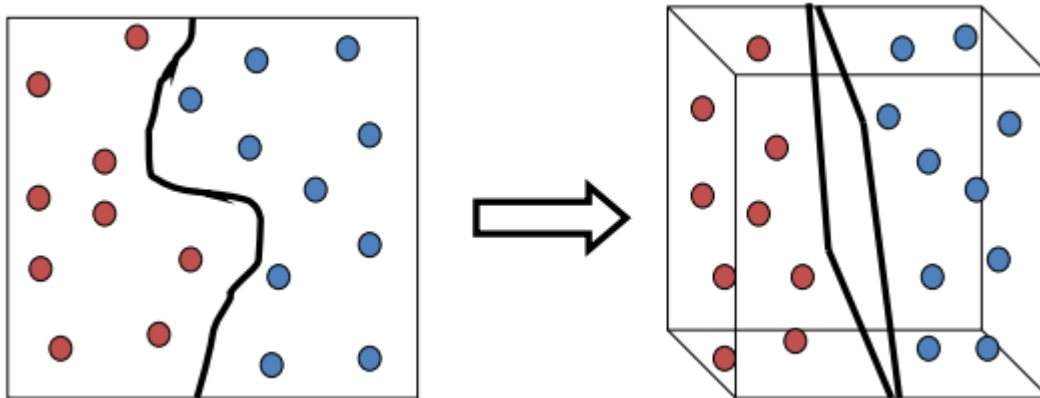


Figure 2.6 : exemple de linéarité et non linéarité [5]

### Cas non linéaire :

Pour surmonter les inconvénients des cas non linéairement séparable, l'idée des SVM est de changer l'espace des données. La transformation non linéaire des données peut permettre une séparation linéaire des exemples dans un nouvel espace. On va donc avoir un changement de

dimension. Ce nouvel espace est appelé « espace de ré-description ». En effet, intuitivement, plus la dimension de l'espace de ré-description est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples est élevée. Ceci est illustré par le schéma suivant :



**Figure 2.7 :** exemple de transformation d'un problème de séparation non linéaire [5]

On a donc une transformation d'un problème de séparation non linéaire dans l'espace de représentation en un problème de séparation linéaire dans un espace de ré-description de plus grande dimension. Cette transformation non linéaire est réalisée via une fonction noyau. En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. On peut citer les exemples de noyaux suivants : polynomiale, gaussien, sigmoïde et laplacien.

### **Fondement mathématique :**

Si les données sont linéairement séparables, alors il existe un hyperplan d'équation tel que :

$\langle W \cdot X \rangle + b = 0$  Tel que :

$$\left\{ \begin{array}{ll} W \cdot X + b \geq 1 & \text{si } y_i = +1 \\ W \cdot X + b \leq -1 & \text{si } y_i = -1 \end{array} \right\} \text{ et } \left\{ \begin{array}{ll} \text{si } y_i = +1 & \\ \text{si } y_i = -1 & \end{array} \right\}$$

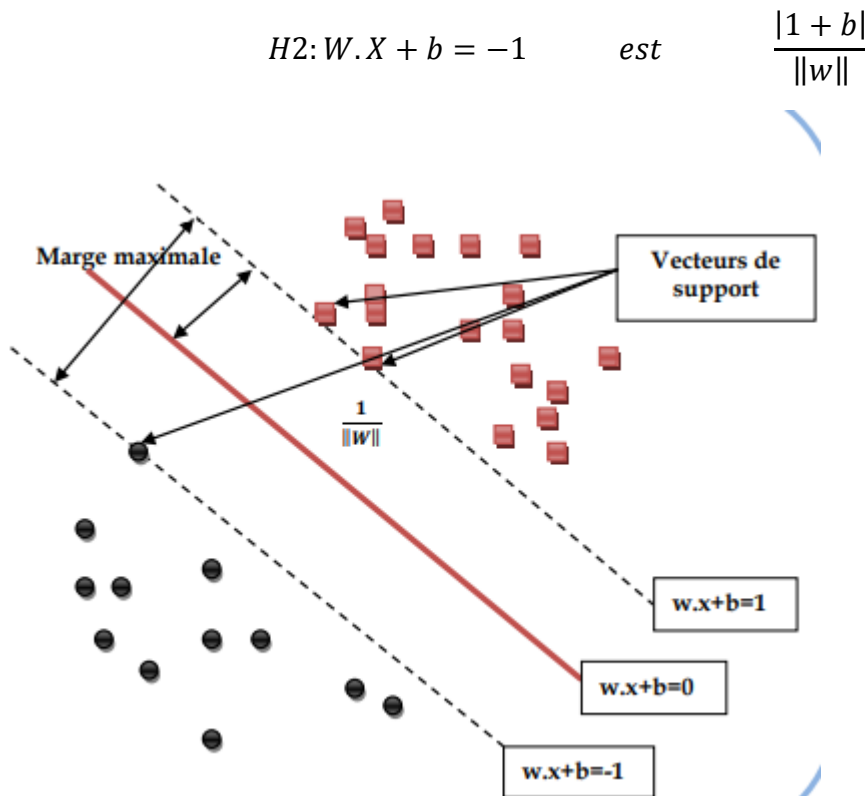
On peut combiner ces deux inéquations en une seule :

$$y_i(W \cdot X + b) \geq +1$$

La distance perpendiculaire de l'origine à l'hyperplan :

$$H1 : W \cdot X + b = 1 \quad \text{est} \quad \frac{|1-b|}{\|W\|}$$

De même pour :



**Figure 2.8:** exemple graphique des données linéairement séparable

**Calcul de marge :**

Rappelons que (en deux dimensions) la distance entre un point  $(x_0, y_0)$  et une droite

$Ax + By + c = 0$  est donnée par la relation suivante :

$$\frac{|Ax_0 + By_0 + c|}{\sqrt{A^2 + B^2}}$$

De façon similaire, la distance entre un point situé sur  $H_1$  et l'hyperplan  $H_0$  est donnée par :

$$\frac{|w \cdot x + b|}{\|w\|} = \frac{1}{\|w\|}$$

Donc la marge (la distance entre les deux hyperplans  $H_1$  et  $H_0$ ) est :  $\frac{2}{\|w\|}$  La maximisation de

cette quantité revient à minimiser l'inverse  $\frac{\|w\|}{2}$ , Donc finalement notre problème peut être

formulé comme suit :  $\min \frac{1}{2} w^t w$  toujours en restant dans le cadre de la condition initiale qui est :

$$y_i(w \cdot x + b) \geq +1$$

On a :

$$\left\{ \begin{array}{l} \min \left( \frac{1}{2} w^t w \right) \\ \text{sous la condition} \\ y_i(w \cdot x + b) \geq +1 \end{array} \right.$$

Ce genre de problème d'optimisation peut être résolu en associant un multiplicateur de

Lagrange  $\alpha_i$  à chaque contrainte ( $\alpha_i \geq 0$ ). Le lagrangien est donné par :

$$L(w, b, \alpha) = \frac{1}{2} w^t w - \sum_{i=1}^N \alpha_i y_i (w \cdot x + b) + \sum_{i=1}^N \alpha_i$$

En passant à la formulation duale, le problème devient : maximiser le Lagrangien, cela revient à dire, de trouver les  $\alpha_i$  et  $w$  qui annulent ses dérivées partielles :

$$\frac{\partial L(w, b, \alpha)}{\partial w} = 0, \frac{\partial L(w, b, \alpha)}{\partial b} = 0 \text{ et } \alpha_i \geq 0$$

On trouve :

$$w = \sum_{i=1}^N \alpha_i y_i x_i \text{ et } \sum_{i=1}^N \alpha_i y_i = 0$$

Et en les réinjectant dans le Lagrangien on obtient le Lagrangien dual :

$$L(w, b, \alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j x_i x_j$$

La résolution des  $\alpha_i$  donne la valeur du vecteur  $w = \sum_{i=1}^N \alpha_i y_i x_i$  et peut classer une nouvelle cible suivant son vecteur de caractéristique  $x$  selon la fonction :

$$f(x) = \text{signe}(w \cdot x + b) = \text{signe} \left( \sum_{i=1}^N \alpha_i y_i x_i \cdot x + b \right)$$

### Le cas linéairement non séparable :

Dans le cas non linéairement séparable, on introduit des variables d'écart  $\varepsilon_i$  ( $i = 1 \dots N$ ) avec  $\varepsilon < 0$  dans les contraintes, qui deviennent :

$$\begin{cases} w \cdot x + b \geq 1 - \varepsilon_i \\ w \cdot x + b \geq -1 + \varepsilon_i \end{cases} \quad \text{et} \quad \begin{cases} \text{si } y_i = 1 \\ \text{si } y_i = -1 \end{cases}$$

Un moyen naturel de donner un coût aux erreurs est de remplacer la fonction à minimiser précédente par :

$$\frac{\|w\|}{2} + C \left( \sum_{i=1}^N \varepsilon_i \right)^k$$

Dans le cas où les données sont non linéairement séparables, c'est-à-dire la surface séparatrice est non linéaire, on transpose le problème dans un autre espace  $F$  de dimension

plus élevé pour rendre les nuages de points des deux classes linéairement séparable au moyen d'une transformation  $\Phi$  tel que :

$$\Phi: x \rightarrow \Phi \in F$$

Le calcul de la surface de séparation revient alors à chercher l'hyperplan optimal dans ce nouvel espace F.

La fonction de décision peut être représentée par le produit scalaire :

$$\Phi^T(x_i) * \Phi(x_i)$$

Cette dernière quantité peut être remplacée par une fonction de la forme  $K(x_i, y_i)$  (Les fonctions scalaires symétriques et définies positives, que l'on désigne souvent simplement par "noyaux", sont plus précisément des "noyaux de Mercer"), c'est ce qu'on appelle le noyau.

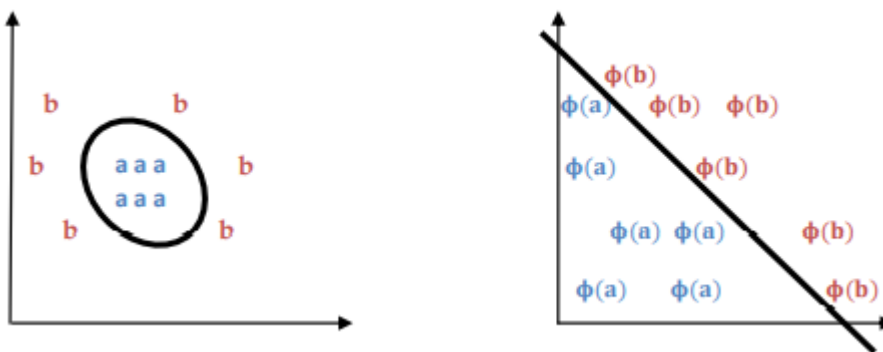
Donc :

$$k(x_i, x_j) = \Phi^T(x_i) * \Phi(x_j)$$

Le lagrangien devient alors :

$$L(w, b, \alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j - k(x_i, x_j)$$

A ce stade, le problème se situe dans le choix de la transformation  $\Phi$  ou plus généralement à la fonction noyau  $K$ . Ils existent peu de noyaux régulièrement utilisés avec les SVM.



**Figure 2.9** : espace de projection des données non séparable [5]

### Les fonctions Noyau (Kernel) :

| Type de noyau    | Expression analytique                                |
|------------------|--|
| Noyau linéaire   | $K(x, z) = (x, z)$                                   |
| Noyau polynomial | $K(x, z) = (a * (x, z) + b)^d$                       |
| Noyau gaussien   | $K(x, z) = e^{-\left(\frac{\ x-z\ ^2}{2a^2}\right)}$ |

**Table 3 :** les fonctions des noyaux la plus utilise avec SVM

### **SVM a plusieurs classes :**

Le principe du SVM expliqué dans la partie précédente se résume dans la résolution des problèmes de classification binaire, or les problèmes rencontrés dans la réalité, sont de type multi classes. D'où l'importance d'étendre le principe du SVM aux problèmes de plus de deux classes, il y a eu plusieurs tentatives pour combiner des classificateurs binaires pour cerner ce problème (multi classes), il y a également des essais pour incorporer la classification de plusieurs classes dans le processus du SVM de sorte que toutes les classes soient traitées simultanément. Nous allons par suite expliquer brièvement quelques méthodes parmi les plus utilisées :

### **Un conter tout :**

La méthode la plus intuitive pour la gestion de la multi-classe consiste à construire autant de classifieur SVM que de classe. Chaque classifieur renvoie 1 si la forme à reconnaître appartient à la classe, -1 sinon. Il faut donc pour reconnaître une forme, le soumettre à tous les classifieur, le meilleur remportant la décision. Il est évident qu'avec un nombre de classe élevé, la combinatoire peut devenir énorme. Cette méthode est appelée en anglais One Versus-All (1-vs-A) et suppose donc la construction de N classifieur et N comparaisons pour la décision.

### **Un conter un :**

Il est également possible de concevoir des classifieurs spécialisés dans la comparaison classe à classe (méthode One-versus-One (1-vs-1) en anglais). Pour un problème à N classes, On a classifié. On soumet la forme à reconnaître à tous ces classifieurs 1-vs-1, la classe remportant le plus de suffrage remporte la décision. Cette méthode possède un gros inconvénient : sa complexité augmente rapidement avec N puisqu'elle nécessite comparaisons.

### **Les avantages de SVM :**

SVM est une méthode de classification intéressante car le champ de ses applications est large, parmi ses avantages nous avons :

1. Un grand taux de classification et de généralisation par rapport aux méthodes classiques.
2. Elle nécessite moins d'effort pour designer l'architecture adéquate (petit nombre de paramètres à régler ou à estimer).
3. La résolution du problème est convertie en résolution d'un problème quadratique convexe dont la solution est unique et donnée par des méthodes mathématiques classiques de programmation quadratique.

### **Les inconvénients de SVM :**

L'inconvénient majeur du classificateur SVM est qu'il est désigné ou conçu pour la classification binaire (la séparation entre deux classes).

## **2.4 Validation d'une classification :**

Lorsqu'une classification automatique est effectuée, on est souvent confronté au problème de la validation des classes résultantes par rapport aux réalités. Pour détecter d'éventuelles failles, il est alors possible de se servir de la matrice de confusion.

### **2.3.1 Matrice de confusion :**

La matrice de confusion appelé aussi tableau de contingence sert à évaluer la qualité d'une classification. Elle est obtenue en comparant les données classées avec des données de référence.

|        |            |                     |                     |
|--------|------------|---------------------|---------------------|
| Actual | Prédiction |                     |                     |
|        |            | positive            | négative            |
|        | positive   | True positive (TP)  | False négative (FN) |
|        | négative   | False positive (FP) | True négative (TN)  |

**Table 4** : Matrice de confusion [6]

On compte le nombre d'observations que le modèle a correctement ou incorrectement prédit en distinguant les prédictions égales à 1 (positives) et nulles (négatives).

On a 2 types d'erreurs :

- On prédit vrai alors que c'est faux (FP) = erreur de type 1
- On prédit faux alors que c'est vrai (FN) = erreur de type 2

Plusieurs métriques sont calculées à partir de la matrice de confusion :

#### 2.4.2 Hamming loss :

Le **hamming\_loss** calcule la perte de Hamming moyenne ou la distance de Hamming entre deux ensembles d'échantillons.

Si  $\hat{y}_i$  est la valeur prédite pour la  $i$ -ème étiquette d'un échantillon donné,  $y_i$  est la valeur vraie correspondante et  $n$  est le nombre de classes ou d'étiquettes, alors la perte de Hamming entre deux échantillons est définie comme suit : [4]

$$L_{Hamming}(x, y) = \frac{1}{n_{labels}} \sum_{j=0}^{n_{labels}-1} 1(x_j \neq y_j)$$

#### 2.4.3 Accuracy score :

Score de classification de précision.



Dans la classification, cette fonction calcule la précision du sous-ensemble : l'ensemble d'étiquettes prédit pour un échantillon doit correspondre exactement à l'ensemble d'étiquettes correspondant dans `y_true`.

#### **2.4.4 Recall score :**

Le rappel est le rapport  $tp / (tp + fn)$  où `tp` est le nombre de vrais positifs et `fn` le nombre de faux négatifs. Le rappel est intuitivement la capacité du classificateur à trouver tous les échantillons positifs.

La meilleure valeur est 1 et la pire valeur est 0.

Lorsque `vrai positif + faux négatif == 0`, le rappel renvoie 0 et déclenche `UndefinedMetricWarning`. Ce comportement peut être modifié avec `zero_division`.

#### **2.4.5 Précision score :**

Calculer la précision.

La précision est le rapport  $TP / (TP + FP)$  où `TP` est le nombre de vrais positifs et `FP` le nombre de faux positifs. La précision est intuitivement la capacité du classificateur à ne pas étiqueter comme positif un échantillon qui est négatif.

La meilleure valeur est 1 et la pire valeur est 0.

Lorsque `vrai positif + faux positif == 0`, la précision renvoie 0 et déclenche `UndefinedMetricWarning`. Ce comportement peut être modifié avec `zero_division`.

#### **2.4.6 F1 score :**

Calculez le score F1, également appelé score F équilibré ou mesure F.

Le score F1 peut être interprété comme une moyenne pondérée de la précision et du rappel, où un score F1 atteint sa meilleure valeur à 1 et le pire score à 0. La contribution relative de la précision et du rappel au score F1 est égale. La formule du score F1 est :

$$F1 = 2 * (\text{précision} * \text{rappel}) / (\text{précision} + \text{rappel})$$

## **2.5 Conclusion :**

La classification est la méthode la plus importante dans le domaine de vision machine qui contient nombreuse méthode au-dessous tel que :

- La classification supervise
- La classification non supervise

Les classifieur SVM et KNN est l'une de classifieur supervisée la plus performant, qui présente des alternatif utile aux différentes méthodes de classification classique.

## Chapitre 3

# Conception et Implémentation

### 3.1 Conception générale :

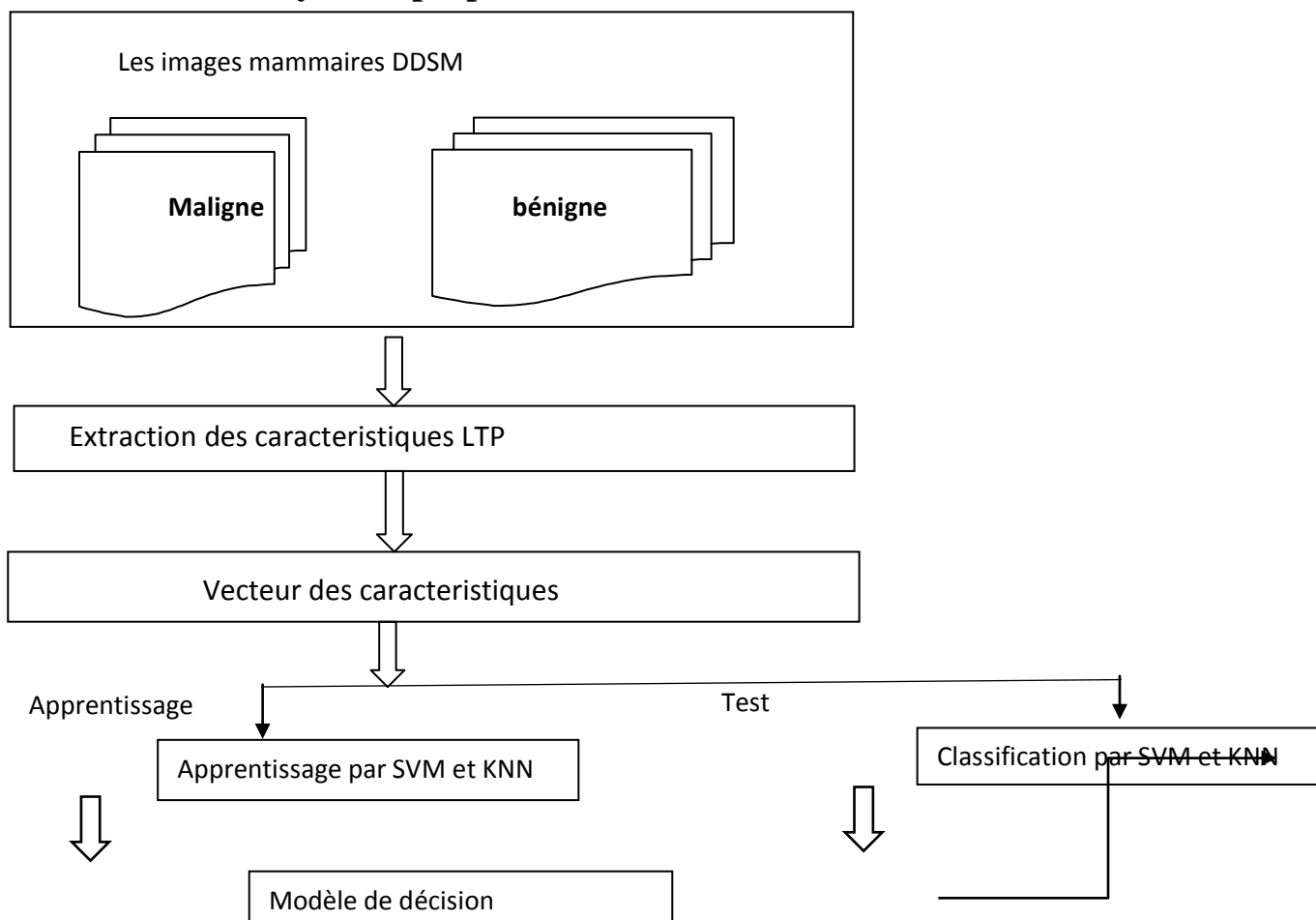
A nos jours les scientifiques essayent de trouver une solution optimale pour plusieurs maladies dont le cancer du sein est la pièce maîtresse de ses dernières.

On sait tous qu'ils y avait plusieurs recherches effectuées à propos de ce sujet afin de protégerle maximum possible des malades atteints par cette pathologie et en parallèle on ne peut pas négliger le rôle de la mammographie comme elle aide bien à définir le type du cancer \* maligne ou bien bénigne \*.

Donc j'ai fait une extraction de données d'une BDD de suites des mammographies pour bien cibler l'emplacement et le degré de danger.

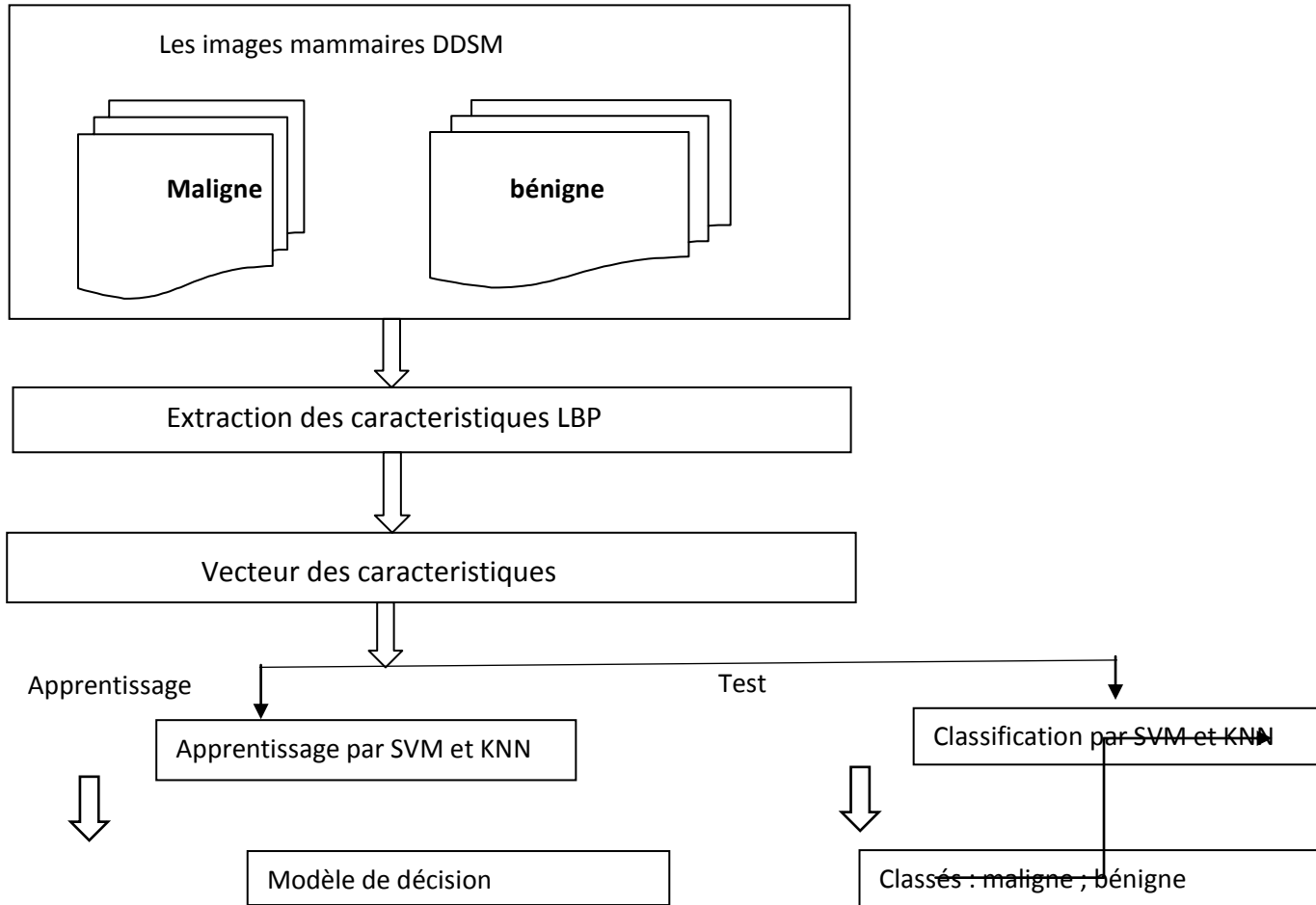
Dans ce chapitre on va parler de la classification effectuée par la décomposition par LBP \* Local Binary Pattern \* et LTP \* Local Ternary Pattern \* et bien sur cette étape fait appel à un outil de classification supervisée SVM et le KNN, et cela ça va nous guider vers une décision finale sur le type de la tumeur.

### 3.2 Architecture du système proposé :



Classés : maligne ; bénigne

**Figure3.1** : architecteur de système avec LTP



**Figure 3.2** : architecteur de système avec LBP

### 3.3 Description générale du système :

L'architecture de la méthode proposée pour résoudre le problème de détection des tumeurs maligne et bénigne:

On commence par le processus de détection qui est divisé en deux phases :

Apprentissage 0.7 et partie test 0.3

Les étapes d'extraction des caracteristiques par LBP et LTP sont communes entre les deux phases. Ce système aide à récupérer les images de mammographies pour l'utiliser.

En phase d'apprentissage ; le logiciel reçoit une suite des images de la BDD, et comme chaque image est traitée ; leurs caractéristiques sont extraites et stockées sur une table de vecteurs libellé \*0 maligne ,1 bénigne \*

### 3.4 Base de données :

Dans cette étude j'ai utilisé 200 images extraites de la base DDMS \* data base for screening mammaire \* ; et cette dernière est classée de deux groupe : maligne et bénigne.

### 3.5 Extraction de caractéristiques :

On fait l'extraction des caractéristiques de l'image et après on fait l'enregistrement de ces caractéristiques avec classes \*0 ou bien 1 \* qui correspond dans une BDD.

On utilise donc la transformation dans LBP et LTP.

### 3.6 Généralité sur LBP :

Local Binary Patterns (LBP) est un modèle non paramétrique descripteur dont le but est de résumer efficacement les structures locales des images. Ces dernières années, il a suscité un intérêt croissant dans de nombreux domaines du traitement d'images et de la vision par ordinateur, et a démontré son efficacité dans de nombreuses applications.

```
class LocalBinaryPatterns:
    def __init__(self, numPoints, radius):
        self.numPoints = numPoints
        self.radius = radius
    def describe(self, image, eps=1e-7):
        LOBP = lbp(image, self.numPoints, self.radius, method="uniform")
        (hist, _) = np.histogram(LOBP.ravel(),
                                bins=np.arange(0, self.numPoints + 3),
                                range=(0, self.numPoints + 2))
        image=plt.imshow(LOBP/np.max(LOBP))
        hist = hist.astype("float")
        hist /= (hist.sum()+ eps)
        return hist
```

Figure 3.3 : implémentation d'algorithme LBP sous python (capteur d'écran)

### 3.7 Généralités sur LTP :

Le LTP sont plus Résistant au bruit, mais n'est plus strictement invariant au niveau de gris métamorphoses. Un schéma de codage est utilisé pour diviser chaque ternaire modèle en deux parties : la positive et la négative, comme illustré dans le chapitre au-dessous. Un problème de LTP est que le seuil t doit être réglé, ce qui n'est pas simple.

```

@jit(nogil=True)
def local_ternrey_pattern(image,t):
    coffition=[128,64,32,16,8,4,2,1]
    upper_pattern=np.zeros((image.shape[0],image.shape[1]))
    lower_pattern=np.zeros((image.shape[0],image.shape[1]))
    tab_upper=[]
    tab_lower=[]
    pattern=np.zeros((3,3))
    for i in range(1,image.shape[0]-1):
        for j in range(1,image.shape[1]-1):
            tab=image[i-1:i+2,j-1:j+2]
            center=tab[1,1]
            for x in range(tab.shape[0]):
                for y in range (tab.shape[1]):
                    if tab[x,y]>=center+t:
                        pattern[x,y]=1
                    elif tab[x,y]>=center-t and tab[x,y]<=center+t:
                        pattern[x,y]=0
                    elif tab[x,y]<=center+t:
                        pattern[x,y]=-1
            pattern_l=list(pattern.ravel())
            del pattern_l[4]
            tab_upper=[0 if elem== -1 else elem for elem in pattern_l]
            tab_upper=[int(f) for f in tab_upper]
            tab_lower=[0 if elem== 1 else elem for elem in pattern_l]
            tab_lower=[ abs(r) for r in tab_lower]
            resulta_upper=sum([n*m for n,m in zip(tab_upper,coffition)])

```

**Figure 3.4** : implémentation d'algorithme LTP sous python (capteur d'écran)

### 3.8 Apprentissage :

Nous avons défini un système d'apprentissage qui a pour but d'extraire d'un modèle performant pour la production de fonction de classification avec des « SVM et KNN ».

Dans notre étude nous avons utilisé 200 images provenant de la base digitale (data base for screening mammaire) « DDMS ». Cette base est classée en deux groupes : Bénigne et maligne.

Le système d'apprentissage doit alors, à partir de cet ensemble d'image, extraire une procédure de classification, il s'agit en effet d'extraire une règle générale à partir de données observées.

La procédure devra classer correctement les images de l'échantillon pour avoir un bon taux de classification et une bonne prédiction.

### 3.9 Classification par SVM:

Le classifieur SVM travaille en se basant sur un fichier qui contient les caractéristiques des Images pour l'apprentissage et essaye de trouver un modèle de décision.

Étiquette 1 : masse maligne et étiquette 0: masse bénigne

D'après l'apprentissage, SVM génère alors un modèle décision à partir de la base des Caractéristiques normalisées.

On va tester par le noyau « polynomial » pour obtenir un bon taux de décision.

### 3.10 Classification par KNN:

Le classifieur KNN travaille au même fichier des caractéristiques que le SVM pour générer un modèle de décision étiquette 1 : masse maligne et 0 : masse bénigne

Pour améliorer la performance de classifieur KNN on utilise la technique de cross-validation (CV en abrégé). Un ensemble de test doit être toujours conservé pour l'évaluation finale, mais l'ensemble de validation n'est plus nécessaire lors de l'élaboration d'un CV. Dans l'approche de base, appelée **k-fold CV**, l'ensemble d'apprentissage est divisé en  $k$  ensembles plus petits (d'autres approches sont décrites ci-dessous, mais suivent généralement les mêmes principes). La procédure suivante est suivie pour chacun des  $k$  « plis » :

- Un modèle est entraîné en utilisant  $k-1$  des plis en tant que données d'apprentissage.
- Le modèle résultant est validé sur la partie restante des données (c'est-à-dire qu'il est utilisé comme ensemble de test pour calculer une mesure de performance telle que la précision).

La mesure de performance rapportée par la validation croisée **k fois** est alors la moyenne des valeurs calculées dans la boucle. Cette approche peut être coûteuse en calcul, mais ne gaspille pas trop de données (comme c'est le cas lors de la fixation d'un ensemble de validation arbitraire).

```
labels.append(label)
xtrain,xtest,ytrain,ytest=train_test_split(features, labels,test_size=0.2)
print(len(xtrain))
print(len(ytrain))
model=SVC(kernel='poly')
model.fit(xtrain,ytrain)
#prediction=model.predict(xtest)
accuracy=model.score(xtest,ytest)
print(accuracy)
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import validation_curve
model1=KNeighborsClassifier()
i =np.arange(1,50)
train_score,val_score=validation_curve(model1,xtrain,ytrain,'n_neighbors',i,cv=5)
#plt.plot(i,val_score.mean(axis=1))

model=KNeighborsClassifier(n_neighbors=5)
model.fit(xtrain,ytrain)
model.score(xtest,ytest)

prediction=model.predict(xtest)
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(ytest, prediction))
print(classification_report(ytest, prediction))
```



**Figure 3.5** : implémentation de SVM et KNN sous python (capteur d'écran)

### 3.10 Évaluation de système :

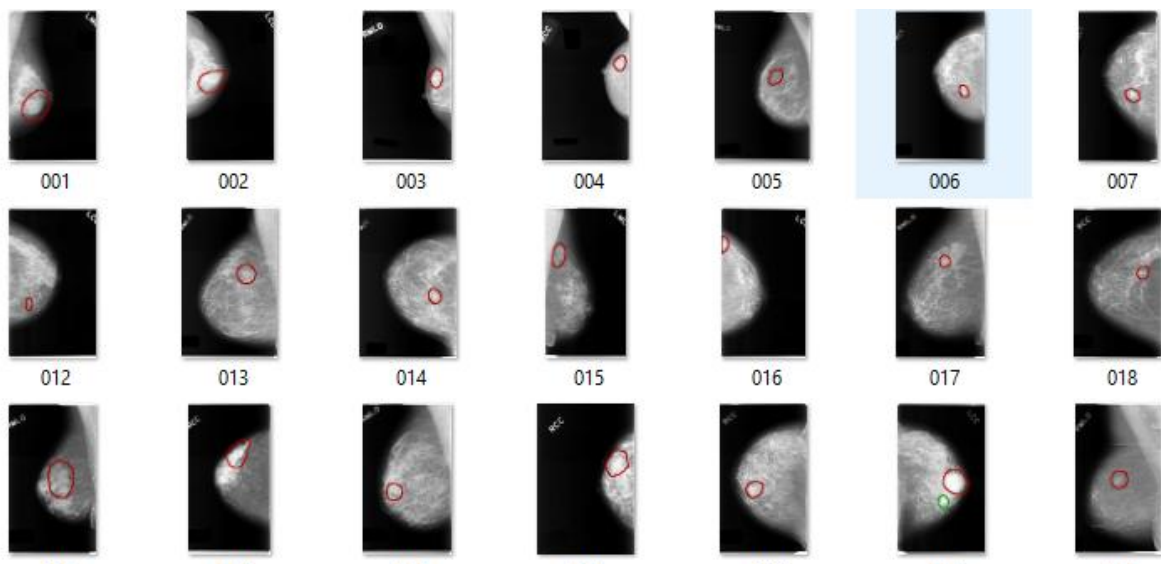
Pour fait une évaluation a un système de classification en doit calcule des mesures de performance de ce système pour prédire quelle est l'algorithme la plus servie pour résoudre un problème donne l'un la plus utilise si le accuracy qui en utiliser dans cet exemple :

|     | SVM | KNN | SVM AVEC CV | KNN AVEC CV |
|-----|-----|-----|-------------|-------------|
| LBP | 75  | 70  | 85          | 80          |
| LTP | 80  | 74  | 87          | 82          |

**Table 6** : tableau exprime les taux de performance des algorithmes avec et sans le cross\_validation

### 3.11 Définition de la Base DDSM :

La base DDSM a été rassemblée par un groupe de chercheurs de l'Université sud de Floride en collaboration avec plusieurs hôpitaux \* de massachusetts, de washington \* et l'Université \* wake forest \*. Cette BDD contient 2620 cas correspondant chacun a une patiente différente, chaque cas contient deux images en vue CC et MLO pour le sein gauche et deux autres pour le sein droit.



**Figure 3. 6:** image extraire de la base DDSM (capteur d'écran)

### **3.12 Environnement de développement :**

#### **3.12.1 Spyder :**



**Figure 3.7 :** logo d'IDE spyder.

Spyder est un environnement scientifique gratuit et open source écrit en Python, pour Python, et conçu par et pour les scientifiques, les ingénieurs et les analystes de données. Il dispose d'une combinaison unique des fonctionnalités avancées d'édition, d'analyse, de débogage et de profilage d'un outil de développement complet avec l'exploration de données, l'exécution interactive, l'inspection approfondie et les belles capacités de visualisation d'un package scientifique. [4]

#### **3.12.2 Langages de programmations utilisés**

##### **Python :**

Python est un langage de programmation puissant et facile à apprendre. Il possède des structures de données de haut niveau efficaces et une approche simple mais efficace de la programmation orientée objet. La syntaxe élégante et le typage dynamique de Python, ainsi que sa nature interprétée, en font un langage idéal pour les scripts et le développement rapide d'applications dans de nombreux domaines sur la plupart des plates-formes.

L'interpréteur Python et la vaste bibliothèque standard sont disponibles gratuitement sous forme source ou binaire pour toutes les principales plates-formes à partir du site Web Python <http://www.python.org/>, et peuvent être distribués gratuitement. Le même site contient également des distributions et des pointeurs vers de nombreux modules, programmes et outils Python tiers gratuits, ainsi qu'une documentation supplémentaire.

L'interpréteur Python est facilement étendu avec de nouvelles fonctions et types de données implémentés en C ou C++ (ou d'autres langages appelables à partir de C).

Python convient également comme langage d'extension pour les applications personnalisables.

Ce didacticiel présente au lecteur de manière informelle les concepts et fonctionnalités de base du langage et du système Python. Il est utile d'avoir un interpréteur Python à portée de main pour une expérience pratique, mais tous les exemples sont autonomes, de sorte que le didacticiel peut également être lu hors ligne.

### **Importation biblio nécessaire :**

La première partie du programme est l'importation des bibliothèques nécessaires pour faire les différentes tâches, et les bibliothèques que j'ai utilisées sont comme suit :

**Numpy** : est une bibliothèque pour le langage de programmation Python, ajoutant la prise en charge de grands tableaux et matrices multidimensionnels, ainsi qu'une grande collection de fonctions mathématiques de haut niveau pour opérer sur ces tableaux. Numpy est un logiciel open source et compte de nombreux contributeurs.

**Matplotlib** : est une bibliothèque complète pour créer des visualisations statiques, animées et interactives en Python.

**Import plot** : Pyplot est une collection de fonctions dans le package de visualisation populaire Matplotlib. Ses fonctions manipulent des éléments d'une figure, comme la création d'une figure, la création d'une zone de traçage, le traçage de lignes, l'ajout d'étiquettes de tracé...etc.

**Import skimage** : scikit-image est un ensemble d'algorithmes de traitement d'images. Il est disponible gratuitement et sans restriction. Nous sommes fiers d'un code de haute qualité, évalué par des paires, écrit par une communauté active de bénévoles.

### **Import accuracy\_score : génère le score assuré**

Dans un problème de classification multi-classes ou binaire en une seule étiquette, la précision absolue est donnée par le rapport (nombre d'instances de données correctement classées / nombre total d'instances de données).

### **Import sklearn: génère de model d'apprentissage**

Scikit-learn est une bibliothèque libre Python destinée à l'apprentissage automatique.

Elle est développée par de nombreux contributeurs notamment dans le monde académique par des instituts français d'enseignement supérieur et de recherche comme L'Institut national de recherche en informatique et en automatique (Inria).

### **3.13 Conclusion générale :**

Nous nous sommes intéressés dans ce travail à réaliser une application pour la détection le type de tumeur mammaire dans le cadre d'aide au diagnostic et décision médical.

Dans ce mémoire, nous avons modélisé les connaissances médicales du cancer du sein basé sur l'analyse de texture par LBP et LTP et par les classifieur SVM et KNN. Notre objectif est de montrer l'efficacité de classification automatique à détecter le type de tumeur (Bénigne, Maligne) sur une mammographie.

Ce travail a été accompli dans le but de :

Aider au diagnostic pour le radiologue.

Construire un deuxième lecteur potentiel après le radiologue et réduction des erreurs commises par le médecin.

Dans un premier temps, la méthode LBP et LTP a été utilisée sur les images mammographies afin d'extraire les caractéristiques de l'image.

Par la suite, nous avons classifié la base de données mammographies obtenue pour bénéficier du résultat des différentes tumeurs malignes ou bénignes.

Par rapport aux méthodes proposées dans littérature, le grand avantage de notre application est qu'elle conserve les caractéristiques de couleurs et permet de voir la tumeur sur deux angles différents ainsi l'information sur le contour, qui sont des notions importantes dans le diagnostic des tumeurs mammaire ainsi que leur caractérisation. et comme perspective je pense a amélioré mon Project par l'utilisation d'un algorithme du Deep-Learning « GAN », ce type d'algorithme plus performante en générale a des algorithme de Machine Learning , et développer un interface plus utilisable.

# Références

1. [Haralick, R.M]. “Statistical and Structural Approaches to Texture.” Proceedings of the IEEE, vol. 67, no. 5, 1979, pp. 786–804, 10.1109/proc.1979.11328.
2. [Pratt, William et al].Pratt, William K., et al. “Visual Discrimination of Stochastic Texture Fields.” IEEE Transactions on Systems, Man, and Cybernetics, vol. 8, no. 11, 1983, pp. 796–804, 10.1109/tsmc.1978.4309867.
3. J.P. COCQUEREZ et Sylvie Philippe, « Analyse d'images: Filtrage et Segmentation »,Masson, 1995.
4. S. ATMANI et M. HALLI, « Extraxion rapide des attributs de texture basés sur les matrices de cooccurrence », Mémoire d'ingénieur en électronique, UMMTO, 2006.
5. [Ojala, Timo, et al 1996] Ojala, Timo, et al. “A Comparative Study of Texture Measures with Classification Based on Featured Distributions.” Pattern Recognition, vol. 29, no. 1, Jan. 1996, pp. 51–59
6. [Xiaoyang Tan et Bill Triggs , 2010] Xiaoyang Tan et Bill Triggs, Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions, IEEE Transactions on Image Processing, 19(6), pp. 1635-1650, 2010
7. [Robert, M. Haralik et al 1973] Haralik, Robert M., et al. “Textural Features for Image Classification.” IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, no. 6, Nov. 1973, pp. 610–621, [ieeexplore.ieee.org/document/4309314/](http://ieeexplore.ieee.org/document/4309314/), 10.1109/tsmc.1973.4309314. Accessed 27 Nov. 2019.

8. [Unser, M. ;1995] Unser, M. "Texture Classification and Segmentation Using Wavelet Frames." IEEE Transactions on Image Processing, vol. 4, no. 11, 1995, pp. 1549–1560, 10.1109/83.469936.
9. [B.Taachouche,O. Douak ;2009] B. Taachouche&O. Douak, "La reconnaissance automatique du locuteur en mode independent du texte en utilisant les méthodes à noyaux (Application du classifieur SVM) «, Pour l'obtention du diplôme d'ingénieur, ecole militaire polytechnique,2009.
10. – [1] [http://www.univ-usto.dz/theses\\_en\\_ligne/doc\\_num.php?explnum\\_id=2141](http://www.univ-usto.dz/theses_en_ligne/doc_num.php?explnum_id=2141)
11. – [2] <https://hal.archives-ouvertes.fr/hal-00706168/file/Analyse%20d'images-Filtrage-et-segmentation>
12. – [3] [http://murphylab.web.cmu.edu/publications/boland/boland\\_node26.html](http://murphylab.web.cmu.edu/publications/boland/boland_node26.html)
13. – [4] <https://www.spyder-ide.org/>
14. – [https://scikit-learn.org/stable/modules/model\\_evaluation.html#hamming-loss](https://scikit-learn.org/stable/modules/model_evaluation.html#hamming-loss)
15. – [Naghashi, Vahid] Naghashi, Vahid. "Co-Occurrence of Adjacent Sparse Local Ternary Patterns: A Feature Descriptor for Texture and Face Image Retrieval." Optik, vol. 157, Mar. 2018, pp. 877–889, 10.1016/j.ijleo.2017.11.160.
16. [5] García-Gonzalo, Esperanza, et al. "Hard-Rock Stability Analysis for Span Design in Entry-Type Excavations with Learning Classifiers." Materials, vol. 9, no. 7, 1 July 2016, p. 531, [www.mdpi.com/1996-1944/9/7/531](http://www.mdpi.com/1996-1944/9/7/531), 10.3390/ma9070531. Accessed 3 May 2021.
17. [6][https://docs.oracle.com/cd/E11882\\_01/datamine.112/e16808/img\\_text/confusion\\_matrix.htm](https://docs.oracle.com/cd/E11882_01/datamine.112/e16808/img_text/confusion_matrix.htm).
18. [Figure1.7][https://upload.wikimedia.org/wikipedia/commons/thumb/c/c2/Lbp\\_neigh\\_hbors.svg/358px-Lbp\\_neighbors.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/c/c2/Lbp_neigh_hbors.svg/358px-Lbp_neighbors.svg.png)
19. [Figure 2.1] <https://www.lebigdata.fr/confusion-matrix-definition>

