

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université Mouloud MAMMERY, Tizi-Ouzou**



**Faculté de Génie Electrique et d'Informatique**  
**Département d'Automatique**

**Mémoire de Fin d'Etudes**

En vue de l'obtention du diplôme

*Master académique en automatique*  
*Option commande des systèmes*

**Thème**

***Extraction des caractéristiques pour la lecture  
automatique des chiffres manuscrits***

Proposé et dirigé par :

☞ M<sup>lle</sup>. AIT AIDER M

Présenté par :

☞ HAMDAD Idir  
☞ MERDAOUI Amar

Soutenu le : 30 / 09 / 2012

***Promotion 2012***

Ce travail a été préparé à : Université Mouloud MAMMERY, Tizi-Ouzou

## Remerciements

On tient, au terme de ce travail, à présenter nos vifs remerciements à toutes les personnes qui ont contribué, de près ou de loin, à son bon déroulement.

Notre gratitude s'adresse également à M<sup>lle</sup> AIT AIDER pour son encadrement et pour l'aide qu'elle nous a prodigué durant ce rapport, ainsi qu'à tous mes formateurs.

On remercie également les membres des jurys qui ont accepté d'évaluer notre travail.

# Dédicace

A

**Mon très cher père et ma très chère mère**

En témoignage de ma reconnaissance envers le soutiens, les sacrifices  
et tous les  
efforts qu'ils ont fait pour mon éducation ainsi que ma formation

A

**Ma très chère grand-mère**

A

**Mon cher frère, et mes chères sœurs**

Pour leurs affections, compréhension et patience

A

Tous ceux qui ont une relation de proche ou de loin avec la réalisation du  
présent rapport.

**Que dieu leur accorde santé et prospérité**

# Dédicace

A

**Mon très cher père et ma très chère mère**

En témoignage de ma reconnaissance envers le soutiens, les sacrifices  
et tous les  
efforts qu'ils ont fait pour mon éducation ainsi que ma formation

A

**Mes défunts grands parent  
mon défunt oncle**

A

**Mon cher frère, et mes chères sœurs**  
Pour leurs affections, compréhension et patience

A

Tous ceux qui ont une relation de proche ou de loin avec la réalisation du  
présent rapport.

**Que dieu leur accorde santé et prospérité**

# Sommaire

<b>Introduction générale.....</b>	<b>1</b>
-----------------------------------	----------

## **Chapitre 1 : Généralités sur la reconnaissance de formes**

1.1 Introduction.....	3
1.2 Prétraitements.....	3
1.2.1 Normalisation de l'image.....	4
1.2.2 Binarisation de l'image.....	4
1.2.2.1 Le seuillage simple.....	4
1.2.2.2 Le seuillage automatique.....	5
1.2.2.2.1 La méthode d'Otsu.....	5
1.2.2.2.2 Le calcul du seuil optimum s.....	5
1.2.3 Amincissement.....	5
1.2.4 Extraction de primitives.....	7
1.2.5 Classification.....	7
1.2.5.1 Approche statistique.....	8
1.2.5.1.1 Méthode paramétrique.....	9
1.2.5.1.2 Méthode non paramétrique.....	9
1.2.5.2 Approche structurelle (syntaxique).....	10
1.2.6 Domaine d'application.....	10
1.2.7 Conclusion.....	10

## **Chapitre 2 : Extracteurs de caractéristiques**

2.1 Introduction.....	11
2.2 Extraction de caractéristiques.....	11
2.2.1 Masques de Kirsch.....	11
2.2.1.1 La version intuitive.....	12
2.2.1.2 La version formelle.....	12
2.2.1.3 Caractéristiques extraites des masques de Kirsch.....	14
2.2.2 Filtres de Gabor.....	15
2.2.2.1 Rappels théoriques.....	15
2.2.2.1.1 Transformée de Fourier.....	15
2.2.2.1.2 Produit de convolution.....	16
2.2.2.1.3 Filtrage d'image.....	16
2.2.2.2 Filtres de Gabor.....	17
2.2.2.3 Caractéristiques extraites à partir des filtres de Gabor.....	22
2.2.2.3.1 Calcul d'énergie.....	22
2.2.2.3.2 Calcul de la moyenne et la variance.....	23
2.2.3 Les moments de Zernike.....	23
2.2.3.1 Les moments géométriques.....	23
2.2.3.2 Les moments géométriques centrés.....	24
2.2.3.3 Les moments géométriques centrés normalisés.....	25

2.2.3.4 Les moments géométriques centrés normalisés de Hu.....	25
2.2.3.5 Les moments orthogonaux.....	26
2.2.3.6 Les moments de Zernike orthogonaux.....	26
2.2.3.6.1 Calcul des moments de Zernike.....	26
2.2.3.7 Caractéristiques extraites à partir des moments de Zernike.....	28
2.2.4 La transformée en ondelettes.....	28
2.2.4.1 Les aspects théoriques de la transformée en ondelettes.....	29
2.2.4.1.1 Ondelette mère.....	29
2.2.4.1.2 Propriétés des ondelettes.....	30
2.2.4.2 La transformée en ondelettes continue.....	30
2.2.4.3 La transformée en ondelettes discrète.....	31
2.2.4.4 Analyse multi-résolution.....	31
2.2.4.5 La décomposition dyadique.....	32
2.2.4.6 Transformée en ondelettes à deux dimensions.....	33
2.2.4.7 Les familles d'ondelettes.....	35
2.2.4.7.1 Ondelette de Haar.....	35
2.2.4.7.2 Ondelette de Daubechies.....	36
2.2.4.7.3 Ondelette de Symmlets.....	36
2.2.4.7.4 Ondelette de coiflets.....	36
2.2.4.8 Conclusion.....	37

### **Chapitre 3 : Théorie sur les SVM**

3.1 Introduction.....	38
3.2 Théorie sur les machines à vecteurs de support.....	38
3.2.1 Classification de données binaire.....	38
3.2.1.1 L'hyperplan optimal.....	38
3.2.1.2 La notion du noyau.....	44
3.2.1.3 Les hyper paramètres.....	45
3.2.2 Classification de données multi classes.....	46
3.2.2.1 Approche un contre tous.....	46
3.2.2.2 Approche un contre un.....	47
3.3 Applications des SVM.....	47
3.4 Conclusion.....	47

### **Chapitre 4 : Tests et résultats**

4.1 Introduction.....	48
4.2 La base de données USPS.....	48
4.3 Le classifieur utilisé.....	49
4.4 Tests et résultats.....	49
4.4.1 Méthode basée sur les masques de Kirsch.....	50
4.4.1.1 Implémentation de l'image compressée en taille 8x8.....	50
4.4.1.2 Implémentation de l'image compressée en taille 4x4.....	50
4.4.2 Méthode basée sur les filtres de Gabor.....	51

4.4.2.1 Implémentation de l'image filtrée.....	51
4.4.2.2 Implémentation de la moyenne et la variance.....	54
4.4.2.3 Implémentation de la moyenne et l'énergie.....	56
4.4.3 Méthode basée sur les moments de Zernike.....	58
4.4.3.1 Image en niveau de gris.....	58
4.4.3.1.1 Implémentation du module.....	58
4.4.3.1.2 Implémentation du module et phase.....	59
4.4.3.2 Image binarisée et amincie.....	60
4.4.3.2.1 Implémentation du module.....	60
4.4.3.2.2 Implémentation du module et phase.....	61
4.4.4 Méthode basée sur la transformée ondelettes.....	62
4.4.4.1 Ondelettes de Daubechies.....	62
4.4.4.1.1 Haar (db1).....	62
4.4.4.1.2 Daubechies (db2).....	64
4.4.4.1.3 Daubechies (db3).....	66
4.4.4.1.4 Daubechies (db4).....	68
4.4.4.2 Ondelettes de Symmlets .....	70
4.4.4.2.1 Symmlets (sym2).....	70
4.4.4.2.2 Symmlets (sym3).....	72
4.4.4.2.3 Symmlets (sym4).....	74
4.4.4.3 Ondelettes de Coiflets.....	76
4.4.4.3.1 Coiflets (coif1).....	76
4.4.4.3.2 Coiflets (coif2).....	78
4.4.4.3.3 Coiflets (coif3).....	80
4.4.4.3.4 Coiflets (coif4).....	82
4.5 Interprétation des résultats.....	84
4.6 Conclusion.....	85
<b>Conclusion générale.....</b>	<b>86</b>

## **Bibliographie**

# Introduction générale

### Introduction générale

La reconnaissance de l'écriture manuscrite n'est plus un sujet nouveau. Il remonte à plus d'une trentaine d'années et s'avère un problème extrêmement complexe qui n'a pas de solution satisfaisante à ce jour. Aujourd'hui, il existe plusieurs domaines dans lesquels la reconnaissance de l'écriture manuscrite est attendue avec impatience. Parmi celles-ci on cite : le tri automatique du courrier postal, le traitement automatique de dossiers administratifs, des formulaires d'enquêtes, ou encore l'enregistrement des chèques bancaires.

Dans ce mémoire, nous nous intéressons particulièrement à la reconnaissance de chiffres manuscrits isolés. Deux disciplines peuvent être dégagées suivant le mode de saisie. Si le mode de saisie est dynamique, on parle de reconnaissance en temps réel appelée en ligne. Dans le cas de la saisie statique on parle alors de reconnaissance en temps différé appelée hors-ligne.

La reconnaissance hors-ligne à laquelle on s'est intéressée s'applique une fois que l'écriture est sur un support papier qui est numérisé puis enregistré sur un format d'image. Cette image contient des pixels soit de type binaire (pixels noirs et blancs) soit de type entier (image niveaux de gris).

Le fait que le nombre de classes naturelles soit réduit à dix classes (chiffres allant de 0 à 9), on trouve à l'intérieur de chacune d'entre elles, une très grande variabilité de l'écriture dû aux conditions souvent relativement précaires dans lesquelles sont écrits les chiffres (chèques écrits rapidement sur un coin de table), la variabilité du matériel utilisé (utilisation de divers stylos, de différentes qualités de papier) et la variation du style d'écriture tendent à rendre la tâche de reconnaissance très complexe. D'où la nécessité de développer un système de reconnaissance qui se rapproche le plus de l'être humain dans sa capacité de lire. Cependant, la performance de ce système est fortement liée à la qualité des primitives à extraire d'une forme inconnue.

Dans le domaine de la reconnaissance de l'écriture, les primitives peuvent être décrites comme un moyen permettant de distinguer un objet (mot, lettre, chiffre) d'une classe d'un autre objet (mot, lettre, chiffre) d'une autre classe. Donc il est nécessaire de définir des primitives significatives lors du développement d'un système de reconnaissance. Les primitives sont généralement définies par expérience ou par intuition. Plusieurs primitives

peuvent être extraites. La représentation des primitives utilisée est une représentation vectorielle. La taille du vecteur peut être large si un grand nombre de primitives est extrait. Il a été observé que certaines primitives extraites sont non pertinentes ou redondantes au système de reconnaissance.

Notre manuscrit est organisé de la façon suivante :

Le chapitre 1 présente des généralités sur la reconnaissance de formes tout en décrivant les étapes nécessaires au développement du système.

Dans le chapitre 2 sont décrites certaines méthodes permettant l'extraction de caractéristiques à savoir :

- Les masques de Kirsch
- Les filtres de Gabor
- Les moments de Zernike
- Les ondelettes

Le chapitre 3 présente les fondements théoriques sur lesquels reposent les machines à vecteurs de support, à savoir les notions d'hyperplan optimal et d'astuce du noyau. Deux principales méthodes de décomposition de problèmes multi classes utilisant les SVM y sont présentées également.

Le chapitre 4 exposera en détail notre méthode proposée présentant les différents tests effectué aussi les résultats obtenus lors de l'étape de reconnaissance, et nous terminerons par une conclusion générale.

# Chapitre 1

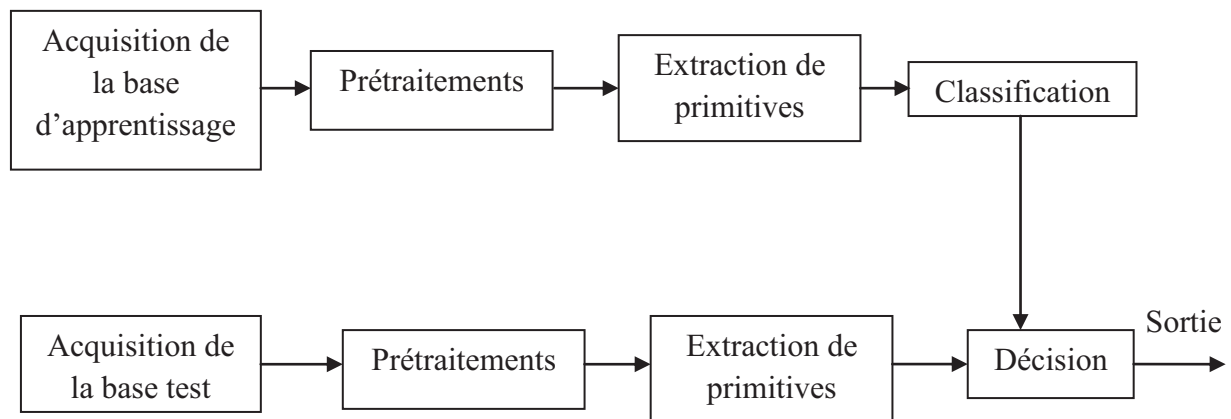
**Généralités sur la reconnaissance de formes**

## 1.1 Introduction

La reconnaissance de formes et particulièrement la reconnaissance de caractères manuscrits a pour objet de convertir des images qui sont compréhensibles par l'homme, en un code interprétable par un ordinateur. Deux disciplines peuvent être dégagées : la reconnaissance en-ligne et la reconnaissance hors-ligne.

Dans la reconnaissance en-ligne, les caractères sont reconnus au moment même où ils sont écrits. Ainsi, l'information est dynamique et monodimensionnelle, et consiste en une séquence de traits qui suivent une échelle temporelle. Dans la reconnaissance hors-ligne, au contraire, le processus d'écriture n'est pas accessible. L'information ne possède pas de nature temporelle; elle est bidimensionnelle, et consiste en l'image digitalisée des caractères [1].

La figure 1.1 nous montre les différentes étapes de la reconnaissance de formes.



**Figure 1.1** Schéma synoptique de la reconnaissance de formes

## 1.2 Prétraitements

L'objectif des prétraitements est de faciliter la caractérisation de la forme (caractère, chiffre) selon la qualité des images obtenues par l'organe d'acquisition, il est parfois nécessaire d'utiliser des algorithmes préliminaires de traitement de l'image.

Ces algorithmes ont pour but d'améliorer la qualité visuelle de l'image ou de permettre l'extraction de certaines informations particulières qui seraient difficile à retrouver autrement.

La réduction de la quantité d'information à traiter en gardant que les informations les plus significatives, réduit considérablement le temps d'exécution.

Nous allons introduire quelques étapes de prétraitements :

### **1.2.1 Normalisation de l'image**

Pour faciliter l'étape de reconnaissance, les chiffres segmentés doivent être normalisés à une taille fixée. La contrainte principale réside dans le choix de cette taille. Trop petite, il y a un risque d'une perte d'information. Trop grande, l'étape de reconnaissance va opérer lentement.

### **1.2.2 Binarisation de l'image**

La binarisation consiste à réduire la quantité d'informations à traiter lorsque l'image est en niveaux de gris. L'image entrée est représentée en niveau de gris et les algorithmes de reconnaissance travaillent souvent sur des images binaires. La binarisation nécessite l'application d'un seuillage.

Pour le seuillage, il existe deux types : seuillage simple et seuillage automatique.

#### **1.2.2.1 Le seuillage simple**

Le seuillage simple signifie que les pixels ayant une valeur de niveau de gris supérieure (ou inférieure selon le choix de l'utilisateur) au seuil vont être codés à 1 (objets sur l'image binaire) et les autres à 0 (fond de l'image binaire). Ne seront sélectionnés dans l'image que les endroits où la valeur des pixels est supérieure au seuil.

#### **1.2.2.2 Le seuillage automatique**

Le seuillage n'est pas nécessairement une opération qui s'effectue manuellement : la valeur de seuil peut être obtenue de manière automatique. Le seuillage automatique est basé sur l'analyse de la distribution en fréquence (cumulée ou histogramme) des niveaux de gris associée à l'image. L'objectif des méthodes de recherche de seuil automatique est de détecter 2 "populations" sur l'histogramme, et de proposer la valeur qui permet de les séparer au mieux.

##### **1.2.2.2.1 La méthode d'Otsu**

La méthode d'Otsu permet de fixer le seuil de binarisation le plus efficacement possible. Pour appliquer la méthode d'Otsu sur une image, nous supposons que cette dernière

est divisée en deux classes, l'une correspondant au fond et l'autre correspondant aux objets d'intérêt. On commence par calculer l'histogramme de l'image. Cette méthode nécessite une normalisation de l'histogramme de manière à ce qu'elle soit indépendante du nombre de pixels dans l'image.

### 1.2.2.2 Le calcul du seuil optimum $s^*$

Deux classes sont distinguées en fonction du seuil  $s$ .

$$A_0 = \{0, 1, \dots, s\} \quad \text{et} \quad A_1 = \{s + 1, \dots, L - 1\} \quad (1.1)$$

On définit  $\sigma_w^2$  la variance d'une classe,  $\sigma_b^2$  la variance interclasse et  $\sigma_T^2$  la variance totale.

Le seuil optimum  $s^*$  peut être déterminé en maximisant le critère suivant :

$$\eta(s) = \frac{\sigma_b^2}{\sigma_T^2} \quad (1.2)$$

$$\text{avec} \quad \sigma_T^2 = \sum_{i=0}^{L-1} (i - \mu_t)^2 p_i ; \quad \mu_t = \sum_{i=0}^{L-1} i p_i$$

$$\sigma_b^2 = p_t q (\mu_1 - \mu_2)^2 ; \quad p_t = \sum_{i=0}^t \frac{h(i)}{N} ; \quad q = 1 - p_t$$

$$\mu_1 = \frac{\mu_t - \mu_s}{1 - p_t} ; \quad \mu_2 = \frac{\mu_s}{p_t} ; \quad \mu_s = \sum_{i=0}^t i p_i ;$$

Et  $p_i = \frac{h(i)}{N}$  : correspond à la probabilité d'apparition du niveau de gris  $i$ .

Le seuil optimum  $s^*$  est défini comme suit :

$$s^* = \text{Arg max}_{s \in L} \eta(s) \quad (1.3)$$

### 1.2.3 Amincissement [2]

Les caractères acquis sont d'une épaisseur variable, ce qui peut produire des instabilités dans les paramètres caractérisant un objet (chiffre par exemple). Il est donc nécessaire de rendre le caractère d'épaisseur à 1 (c.à.d. un pixel) avant d'extraire ses attributs. Cette technique constitue l'étape d'amincissement du caractère.

Les deux critères retenus pour les méthodes d'amincissement sont les suivants :

- l'épaisseur du caractère aminci doit être de 1.
- le caractère aminci doit conserver les propriétés topologiques de la forme comme le nombre de parties, le nombre de trous et la connexité.

Nous présentons dans ce paragraphe, un algorithme d'amincissement d'un objet binaire. Cet algorithme est un algorithme itératif. Dans cette méthode on passe successivement par deux étapes appliquées sur chaque point de l'objet.

### Première étape

Dans la première étape, chaque point  $P_1$  de l'objet sera mis à 0 (c.à.d. effacé) si les quatre conditions suivantes sont satisfaites :

- a)  $2 \leq N(P_1) \leq 6$  ;
- b)  $S(P_1) = 1$  ;
- c)  $P_2 P_4 P_6 = 0$  ;
- d)  $P_4 P_6 P_8 = 0$  ;

où les  $P_i$  sont égaux soit à 0 soit à 1 et :

$N(P_1)$  = nombre voisins de  $P_1$  qui sont non nuls

$N(P_1) = P_2 + P_3 + P_4 + P_5 + P_6 + P_7 + P_8 + P_9$

Le tableau 1.1 présente les huit-voisins d'un point  $P_1$ .

$P_9$	$P_2$	$P_3$
$P_8$	$P_1$	$P_4$
$P_7$	$P_6$	$P_5$

Tableau 1.1 Les huit-voisins du point  $P_1$

$S(P_1)$  : nombre de transitions de 0 à 1 dans la séquence  $P_2, P_3, P_4, P_5, P_6, P_7, P_8$  et  $P_9$

### Deuxième étape

Sur l'image obtenue après l'application de l'étape 1, on applique l'étape 2, qui consiste à supprimer tout point  $P_1$  vérifiant les quatre conditions suivantes :

- a1)  $2 \leq N(P_1) \leq 6$  ;
- b1)  $S(P_1) = 1$  ;
- c1)  $P_2 P_4 P_8 = 0$  ;
- d1)  $P_2 P_6 P_8 = 0$  ;

Une itération dans cet algorithme consiste donc à :

- Appliquer l'étape 1 à chaque point de l'image.
- Effacer tout point vérifiant les 4 conditions a), b), c), d) de la première étape.
- Appliquer l'étape 2 à l'image obtenue suite à l'étape 1.
- Effacer tout point vérifiant les 4 conditions a1), b1), c1), d1) de la deuxième étape.

Cette itération de base sera appliquée itérativement jusqu'à qu'une itération n'apporte pas de variation à l'image que l'on appelle idempotence.

### **1.2.4 Extraction des primitives**

L'objectif de l'extraction des caractéristiques dans le domaine de la reconnaissance de formes consiste à exprimer les primitives sous une forme numérique ou symbolique appelée codage. Selon le cas, les valeurs de ces primitives peuvent être réelles, entières ou binaires. Cette étape de la reconnaissance consiste à extraire des caractéristiques permettant de décrire de façon non équivoque les formes appartenant à une même classe de caractères tout en les différenciant des autres classes. Les primitives qui en découlent de cette étape doivent être discriminantes et seront utilisées dans la phase de classification.

Cette étape consiste à calculer des informations pertinentes pour la classification de l'objet à partir des valeurs de pixels qui le composent [3].

### **1.2.5 Classification**

La classification est l'élaboration d'une règle de décision qui transforme les attributs caractérisant les formes en appartenance à une classe (passage de l'espace de codage vers l'espace de décision). Avant qu'un modèle de décision ne soit intégré dans un système de reconnaissance de l'écriture, il faut avoir procédé auparavant à deux étapes : l'étape d'apprentissage et l'étape de test.

L'étape d'apprentissage consiste à caractériser les classes de formes de manière à bien distinguer les familles homogènes de formes. C'est une étape clé dans le système de reconnaissance. On distingue deux types d'apprentissage : apprentissage supervisé et apprentissage non supervisé [3].

L'apprentissage supervisé utilise une base de données d'apprentissage qui présente des exemples dont la classe est déjà identifiée (elle aura été identifiée préalablement par un opérateur humain). Dans ce cas, le classifieur doit arriver à générer une fonction ou un ensemble de règles qui permettra d'obtenir la meilleure performance de classification.

Éventuellement, le système pourra être éprouvé avec une base de données de test qui contient des patrons qui n'ont pas encore été observés, afin d'évaluer sa performance éventuelle sur des données inconnues. Cette dernière mesure est appelée performance de généralisation.

Dans le cas de l'apprentissage non supervisé, ce type d'apprentissage n'utilise pas d'exemples préalablement identifiés. Les échantillons des bases de données ne font donc pas à priori partie d'une classe. Un système de reconnaissance de forme obtenu sans supervision doit lui-même classifié les nouveaux échantillons présentés en fonction des classes qu'il aura élaborées à partir des patrons observés. Dans certains cas, le système de classification non supervisé devra aussi déterminer par lui-même le nombre de classes.

Dans le domaine de la reconnaissance des caractères manuscrits, ce sont les méthodes basées sur un apprentissage supervisé qui sont le plus souvent utilisées; et plus particulièrement pour les chiffres manuscrits isolés car les classes naturelles sont connues et en nombre limité.

L'étape de test permet d'évaluer la performance du classifieur pour un apprentissage donné. C'est une étape importante car elle peut mettre en cause le choix des primitives ou le choix de la méthode d'apprentissage.

Ces phases d'apprentissage et de test sont réalisées préalablement à l'intégration du module de décision dans le système de reconnaissance. Dans tous les cas, on peut permettre au système de reconnaissance d'itérer les phases d'apprentissage et de test tant que l'on n'a pas atteint les performances désirées. Le calcul de cette performance est le résultat du classifieur utilisé. Pour construire un classifieur, il existe plusieurs approches. On définit l'approche statistique et l'approche structurelle (aussi appelé approche syntaxique).

### 1.2.5.1 Approche statistique

En reconnaissance de formes statistique, une forme est caractérisée par  $n$  attributs numériques extraite de données, et représentée par un vecteur d'un espace numérique de

dimension  $n$  pour être caractériser d'une manière statistique. Elle a besoin d'un nombre élevé d'exemples afin de réaliser un apprentissage correct des lois de probabilité des différentes classes. La construction du classifieur consiste donc en la partition de l'espace des attributs en différentes régions, chacune affectée à une classe.

Toute la difficulté de cette approche réside dans le choix des attributs : de bons attributs doivent permettre de discriminer le plus possible les classes c'est-dire représenter toutes les formes d'une même classe dans une région la plus compacte possible de l'espace, et situer chaque classe dans une région la plus éloignée possible des régions ou se situent chacune des autres classes.

Cependant certains types de formes sont difficilement descriptibles par le biais de vecteurs d'attributs numériques, rendant leur extraction inefficace [4].

Les deux principales familles de méthodes utilisées sont les méthodes paramétriques et les méthodes non paramétriques [5] :

#### **1.2.5.1.1 Méthode paramétrique**

Cette méthode opère sur l'hypothèse que les classes étudiées suivent une distribution de probabilités d'une certaine forme connue a priori. Autrement dit on fait appel à une famille paramétrable de densité ou de surface et on optimise les paramètres pour minimiser la probabilité d'erreur.

La phase de décision consiste à déterminer la classe pour laquelle la forme inconnue présente la probabilité d'appartenance maximale. Elle exige des bases d'apprentissage assez importantes pour une estimation correcte des paramètres de la distribution supposée.

#### **1.2.5.1.2 Méthodes non paramétriques**

Dans le cas de méthodes non paramétriques, les lois de probabilités sont inconnues pour une des classes, cela revient à développer un algorithme qui converge vers la densité ou la surface idéale quelle qu'elle soit (moyennant certaines hypothèses). Le problème revient à établir une frontière de décision entre les classes.

### 1.2.5.2 Approche structurelle (syntaxique)

En approche structurelle, les formes sont représentées par des structures complexes composées de primitives simples et de relations topologiques entre les primitives. Cette approche cherche à structurer l'information en décrivant l'organisation topologique (la structure) de la forme à partir de ses composantes les plus élémentaires.

La notion de structure quoique sujette à de nombreuses définitions, peut se ramener à l'existence d'un tout (la structure, c'est-à-dire la forme) décomposables en parties et de relations entre ces parties.

Dans le cadre de notre projet, la méthode de classification de chiffres manuscrits isolés hors ligne, repose sur la théorie des SVM qui est une approche statistique non paramétrique supervisée [4].

### 1.2.6 Domaine d'application

La reconnaissance de forme s'est propagée à un nombre important de domaines tels que :

**Industrie** : contrôle qualité, diagnostic de pannes, fusion de données fournies par des capteurs.

**Télécommunications** : Analyse du signal, élimination du bruit, compression de données.

**Environnement**: Analyse chimique prévisions météorologiques.

**Transport** : Péage automatique.

**Traitement d'image** : la reconnaissance de caractères manuscrits, tri-postal, bureautique, chèque bancaire, saisie de texte, diagnostique médical ...etc.

### 1.2.7 Conclusion

Nous avons évoqué dans ce chapitre les généralités concernant la reconnaissance de formes. Nous nous sommes arrêtés sur la reconnaissance de l'écriture manuscrite et précisément, les chiffres manuscrits.

Dans le deuxième chapitre, nous nous focaliserons sur l'étape d'extraction de caractéristiques où nous allons développer certains descripteurs les plus connus.

# Chapitre 2

**Extracteurs de caractéristiques**

## 2.1 Introduction

L'objectif de l'extraction des caractéristiques dans le domaine de la reconnaissance de formes consiste à exprimer les primitives sous une forme numérique ou symbolique appelée codage. Selon le cas, les valeurs de ces primitives peuvent être réelles, entières ou binaires. Le vecteur des  $n$  primitives constituées représente un point dans le nouvel espace à  $n$  dimensions. Cette étape de la reconnaissance permet d'extraire des caractéristiques décrivant façon non équivoque les formes appartenant à une même classe de caractères tout en les différenciant des autres classes [5].

Le choix des primitives lors de l'étape d'extraction de caractéristiques est la partie la plus importante dans le système de reconnaissance, car ses performances dépendent de la qualité des descripteurs utilisés.

Il existe différentes méthodes d'extraction de primitives citées dans la littérature, nous allons voir par la suite les différentes méthodes utilisées dans ce travail pour l'extraction de primitives.

## 2.2 Extractions de caractéristiques

### 2.2.1 Masques de Kirsch

Le principe des masques de Kirsch est d'extraire les composantes directionnelles d'une image. Ces composantes directionnelles (Verticales; Horizontales; Obliques gauches; Obliques droites) utilisées conjointement avec l'image de départ nous donnent des cartes de caractéristiques utilisables par le classifieur.

La création de ces cartes de caractéristiques nous fournit un vecteur de caractéristiques de grande dimension. Ainsi, il est nécessaire d'effectuer une compression des informations de l'image initiale et des cartes pour en extraire respectivement l'information globale et locale de l'image d'origine.

Dans un premier temps nous allons voir comment obtenir des cartes de caractéristiques à l'aide des masques de Kirsch, puis nous verrons la méthode de compression des images que nous utiliserons par la suite.

### 2.2.1.1 La version intuitive

Le but de l'utilisation des masques de Kirsch est de caractériser les pixels de l'image de départ suivant une direction précise. On distingue quatre orientations différentes : la composante verticale de l'image, la composante horizontale, la composante oblique gauche et la composante oblique droite. Pour extraire ces pixels, on travaille sur les huit voisins de chaque pixel de l'image. L'idée générale est que pour savoir si un pixel appartient à une composante, on calcule son gradient  $G(i, j)$  en soustrayant les pixels de part et d'autre de la direction de la composante.

Intuitivement, les pixels qui auront un gradient élevé seront ceux situés sur les contours de la forme de l'image, orientés dans la direction de la caractéristique.

### 2.2.1.2 La version formelle

Kirsch a défini un algorithme de détection de contours donné par l'équation suivante:

$$G(i, j) = \max \left\{ 1, \max_{k=0}^7 [|5S_k - 3T_k|] \right\} \quad (2.1)$$

Tel que :

$$S_k = A_k + A_{k+1} + A_{k+2} \quad (2.2)$$

$$T_k = A_{k+3} + A_{k+4} + A_{k+5} + A_{k+6} + A_{k+7} \quad (2.3)$$

$S_k$  : qui est le potentiel des trois voisins supérieurs du pixel (par rapport à la direction recherchée).

$T_k$  : comme le potentiel des trois voisins inférieurs du pixel (par rapport à la direction recherchée).

A0	A1	A2
A7	(i, j)	A3
A6	A5	A4

**Figure 2.1** Définition des huit voisins du pixel  $(i, j)$

$A_k$  ( $k = 0, \dots, 7$ ) est le voisin du pixel  $(i, j)$ .

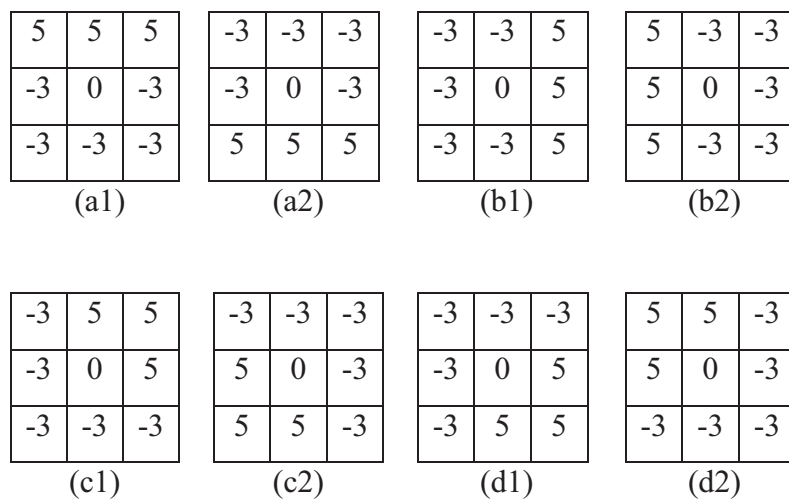
Les  $A_k$  étant les huit voisins du pixel  $(i, j)$ , ( $k = 0, 1, \dots, 7$ ). Ils sont pris modulo 8 pour faire le tour du pixel.

Dans l'équation (2.1),  $G(i, j)$  est le gradient du pixel  $(i, j)$ , les indices de  $A$  sont évaluées modulo 8, défini comme le montre la figure 2.1.

L'image d'entrée est de taille 16 x 16, puis les vecteurs de caractéristiques directionnelles pour l'horizontale (H), verticale (V), oblique droit (R) et oblique gauche (L) sont calculées à partir de l'image d'entrée comme suit:

$$\begin{aligned}
 G(i, j)H &= \max(|5S_0 - 3T_0|, |5S_4 - 3T_4|) \\
 G(i, j)V &= \max(|5S_2 - 3T_2|, |5S_6 - 3T_6|) \\
 G(i, j)R &= \max(|5S_1 - 3T_1|, |5S_5 - 3T_5|) \\
 G(i, j)L &= \max(|5S_3 - 3T_3|, |5S_7 - 3T_7|)
 \end{aligned} \tag{2.4}$$

La figure 2.2 montre les masques de Kirsch utilisés pour le calcul des ces vecteurs.



**Figure 2.2** Les différents masques de Kirsch utilisés pour l'extraction des quatre caractéristiques directionnelles.

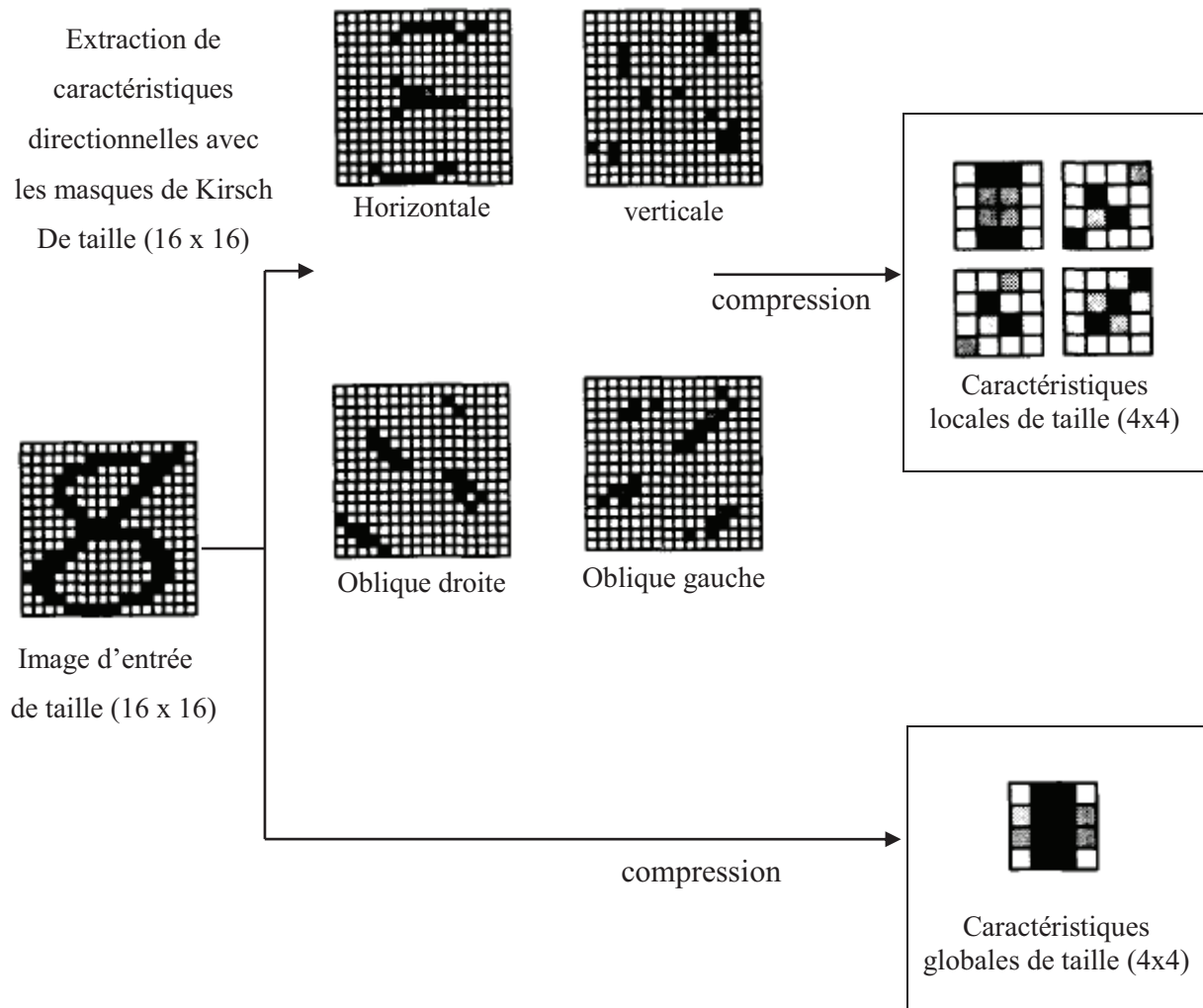
(a1, a2) : masques de Kirsch pour l'extraction de caractéristiques horizontales.

(b1, b2) : masques de Kirsch pour l'extraction de caractéristiques verticales.

(c1, c2) : masques de Kirsch pour l'extraction de caractéristiques oblique droite.

(d1, d2) : masques de Kirsch pour l'extraction de caractéristiques oblique gauche.

Dans notre cas nous avons utilisé les masques (a1, b1, c1 et d1) pour l'extraction de ces quatre caractéristiques.



**Figure 2.3** Aperçu de la méthode d'extraction de caractéristiques globales et locales de taille (4x4)

### 2.2.1.3 Caractéristiques extraites à partir des masques de Kirsch

Pour réduire la dimension des cartes directionnelles obtenues, comme une étape finale dans l'extraction de caractéristiques directionnelles, chaque vecteur de caractéristique directionnelle de taille (16 x 16) est comprimé à un vecteur de taille ( $N \times N$ ).

Le principe de compression utilisé ici, consiste à diviser l'image d'origine de taille  $M \times M$  en blocs et à calculer la moyenne au niveau de chaque bloc carré de dimension  $N$  qui sera prise comme intensité d'un pixel de la nouvelle image qui a donc pour dimension  $M/N$ .

En conséquence, les caractéristiques finales se composent de cinq images de taille  $(N \times N)$ , avec 4x  $(N \times N)$  caractéristiques locales et 1x  $(N \times N)$  caractéristiques globales. La figure 2.3 montre l'aperçu de l'étape d'extraction de caractéristiques proposée en taille  $(4 \times 4)$ .

## 2.2.2 Filtres de Gabor

Le filtre de Gabor a été utilisé avec succès dans de nombreuses applications, telles que la segmentation de texture [7], la détection de cible, la reconnaissance d'empreintes digitales, la reconnaissance du visage, la compression etc.

Il doit leur nom au physicien anglais d'origine hongroise Dennis Gabor. C'est un filtre linéaire. Il est défini par le produit entre un filtre Gaussien et une sinusoïdale orientée. C'est donc une fonction sinusoïdale à laquelle on a rajouté une enveloppe gaussienne. La fonction sinusoïdale est caractérisée par sa fréquence et par son orientation. Ainsi appliqué sur une image, les filtres de Gabor sont considérés comme des filtres précurseurs dans le domaine des méthodes de filtrage spatio-fréquentiel. Ils présentent d'excellentes propriétés de localisation fréquentielles et spatiales dont le fonctionnement est proche des traitements visuels humains. Ils ont l'avantage d'être paramétrables en fréquence, et en orientation.

### 2.2.2.1 Rappels théoriques

#### 2.2.2.1.1 Transformée de Fourier

La transformation de Fourier est une opération mathématique qui permet de représenter un signal dans le domaine fréquentiel, donnant accès à l'information portée par cette fonction sous forme d'un spectre fréquentiel. Elle est définie comme suit [6]:

- Pour un signal 1-D

$$F(v) = \int_{-\infty}^{+\infty} f(t) e^{-j2\pi vt} dt \quad (2.5)$$

- Pour un signal 2-D

$$F(v_1, v_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{-j2\pi v_1 x} e^{-j2\pi v_2 y} dx dy \quad (2.6)$$

La transformée de Fourier discrète d'une image  $f(x, y)$  de taille  $(NxM)$  est définie par la formule suivante :

$$F(n, m) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) e^{-j2\pi nx/M} e^{-j2\pi my/N} \quad (2.7)$$

$$0 \leq n \leq N-1, 0 \leq m \leq M-1.$$

### 2.2.2.1.2 Produit de convolution

Le produit de convolution définit la relation entrée/sortie dans un filtrage linéaire. Il permet de déduire le signal de sortie d'un système linéaire défini par sa réponse impulsionnelle  $h(t)$  et son entrée  $e(t)$ . Défini comme suit [6]:

$$S(t) = e(t) \otimes h(t) \quad (2.8)$$

- Pour un signal 1-D

$$S(t) = \int_{-\infty}^{+\infty} e(u) h(t-u) du \quad (2.9)$$

- Pour un signal 2-D

$$S(t_1, t_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} e(u, v) h(t_1 - u, t_2 - v) dt_1 dt_2 \quad (2.10)$$

La convolution discrète d'une image de taille  $(NxM)$  est définie comme suit :

$$S(n, m) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} e(u, v) h(n-u, m-v) \quad (2.11)$$

$$0 \leq n \leq N-1 \text{ et } 0 \leq m \leq M-1$$

### 2.2.2.1.3 Filtrage d'image

Un filtre est une transformation mathématique permettant, de modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect. En pratique, il s'agit de

créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine. Il y a deux catégories de filtrage :

- **Filtrage global** : Dans le filtrage global, chaque pixel de la nouvelle image est calculé en prenant en compte la totalité des pixels de l'image de départ.
- **Filtrage local** : Dans le filtrage local, chaque pixel de la nouvelle image est calculé en prenant en compte seulement un voisinage du pixel correspondant dans l'image d'origine.

Un filtre local est dit linéaire si la valeur du nouveau pixel est une combinaison linéaire des valeurs des pixels du voisinage. Pratiquement, le filtrage d'une image est une convolution entre sa fonction d'intensité  $I(x,y)$  et une fonction  $h(x,y)$  réponse impulsionnelle du filtre.

Il existe plusieurs types de filtre parmi eux on trouve :

- **Filtre passe-bas** : A un rôle de lissage du signal image, il élimine le bruit et rend l'image floue mais, en gardant intacte l'intensité moyenne de l'image.
- **Filtre passe-haut** : A pour but de rehausser les frontières : il met en valeur les points isolés ou les limites des objets.

### 2.2.2.2 Filtres de Gabor

La réponse impulsionnelle d'un filtre de Gabor dans le domaine spatial est le produit d'une gaussienne et d'une exponentielle complexe. En 1D, l'expression d'un filtre de Gabor est donnée par [6]:

$$h(x, \omega, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{\left(\frac{-x^2}{2\sigma^2} + j\omega x\right)} \quad (2.12)$$

$\omega = 2\pi f$ , avec  $f$ : fréquence de la sinusoïde.

$\sigma$ : l'écart type.

Cette fonction complexe peut être décomposée en partie imaginaire et partie réelle.

La partie réelle du filtre 1-D (Figure 2.4) est donné par la formule suivante [6]:

$$h(x, \omega, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{\left(\frac{-x^2}{2\sigma^2}\right)} \cos(\omega x) \quad (2.13)$$

Tandis que la partie imaginaire 1-D, dont l'allure est donnée par la figure 2.4, est [6] :

$$h(x, \omega, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{\left(\frac{-x^2}{2\sigma^2}\right)} \sin(\omega x) \quad (2.14)$$

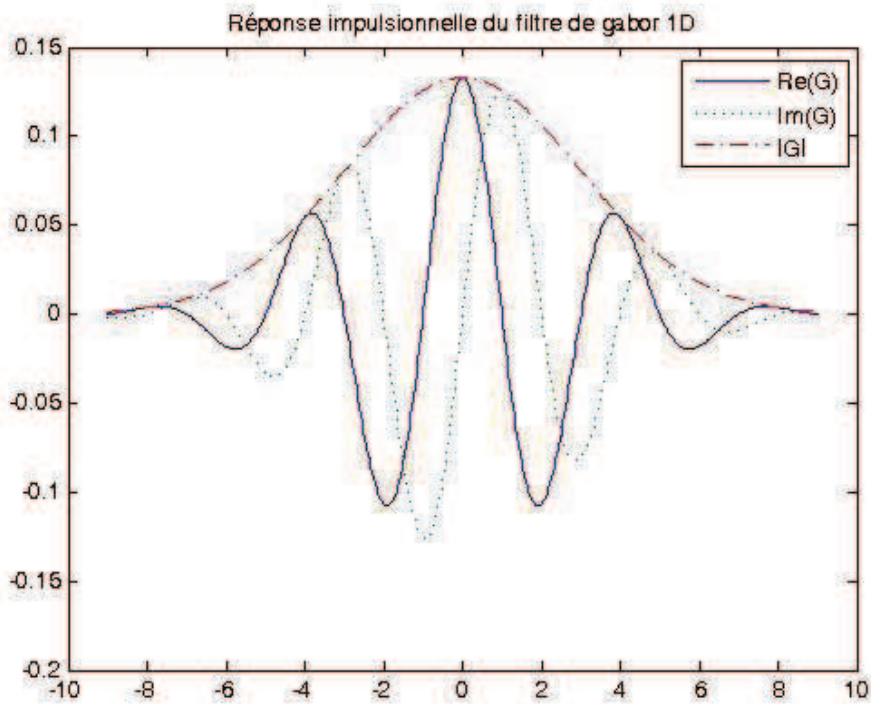


Figure 2.4 Réponse impulsionnelle d'un filtre de Gabor 1-D.

En 2D, le filtre de Gabor est défini par une gaussienne bidimensionnelle modulée par une fonction sinusoïdale complexe orientée d'un angle  $\theta$  par rapport à l'axe des abscisses  $X$ . L'expression du filtre de Gabor 2D dans le domaine spatial est donnée par [6]:

$$h(x', y', \omega, \theta, \sigma_x, \sigma_y) = \frac{1}{\sqrt{2\pi\sigma_x\sigma_y}} e^{-\frac{1}{2}\left[\left(\frac{x'}{\sigma_x}\right)^2 + \left(\frac{y'}{\sigma_y}\right)^2\right] + j\omega x'} \quad (2.15)$$

Avec :

$$x' = x \cos \theta + y \sin \theta$$

$$y' = -x \sin \theta + y \cos \theta$$

$\theta$  : l'orientation de la sinusoïde,  $f$  sa fréquence.

$\sigma_x, \sigma_y$  sont les variances de la gaussienne respectivement le long des axes  $X$  et  $Y$ , qui déterminent la largeur du filtre de Gabor.

Pour un filtre orienté à  $0^\circ$  ( $\theta = 0$ ) à symétrie paire et a partie réelle, on aura [6]:

$$h(x, y, \omega, \sigma_x, \sigma_y) = e^{-\frac{1}{2}\left[\left(\frac{x}{\sigma_x}\right)^2 + \left(\frac{y}{\sigma_y}\right)^2\right]} \cos(\omega x) \quad (2.16)$$

Pour une orientation non nulle ( $\theta \neq 0$ ) à symétrie paire c'est cette fonction de Gabor qui est utilisé dans notre travail, donné comme suit [6]:

$$h(x', y', \omega, \theta) = e^{-\frac{1}{2}\left[\left(\frac{x'}{\sigma_x}\right)^2 + \left(\frac{y'}{\sigma_y}\right)^2\right]} \cos(\omega x') \quad (2.17)$$

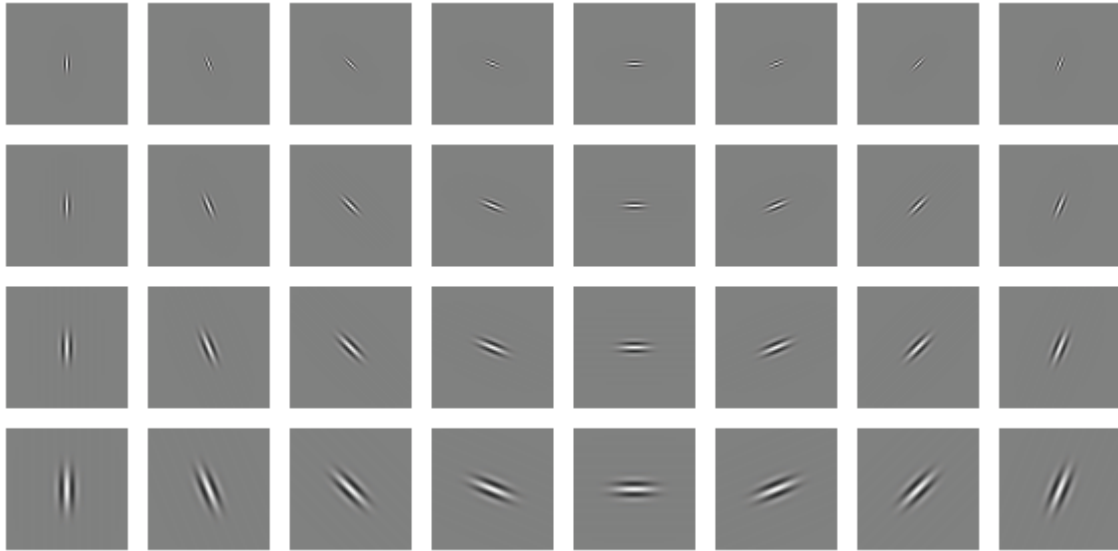
Il y a trois paramètres qui contrôlent le filtre de Gabor [7]:

$\theta$ : spécifie l'orientation du filtre, ce paramètre fait tourner le filtre de Gabor autour de son centre (origine).  $\theta$  dicte l'angle pour laquelle le filtre répondra. Dans la plupart des cas  $\theta$  est un ensemble de valeurs entre  $[0, \pi]$ , les valeurs de  $[\pi, 2\pi]$  sont redondantes en raison de la symétrie du filtre.

$f$ : la fréquence du filtre, choisi en fonction de l'application, les motifs les plus étroits peuvent être détectés par des fréquences importantes, et vice versa.

$\sigma_x$  et  $\sigma_y$ : indiquent les rayons de la gaussienne, la taille de gausse détermine la région qui doit être affectée par la convolution.

La figure 2.5 montre l'allure des filtres de Gabor bidimensionnels :



**Figure 2.5** Partie réelle de fonctions de Gabor dans le domaine spatial suivant la variation des écarts types  $(\sigma_x, \sigma_y)$  ainsi que de l'orientation  $\theta$ .

La transformée de Fourier du filtre de Gabor, se résume donc à la formule suivante [6]:

$$H(u, v) = A \left( \exp \left\{ -\frac{1}{2} \left[ \frac{(u-f)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2} \right] \right\} + \exp \left\{ -\frac{1}{2} \left[ \frac{(u+f)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2} \right] \right\} \right) \quad (2.18)$$

Tels que :

$$\sigma_u = \frac{1}{2} \pi \sigma_x, \quad \sigma_v = \frac{1}{2} \pi \sigma_y \quad \text{et} \quad A = 2\pi\sigma_y\sigma_x$$

La figure 2.6 donne exemple de filtrage d'image par les filtres de Gabor pour 3 orientations avec  $\sigma_x = \sigma_y = 6$ ,  $f = \frac{\sqrt{2}}{10}$

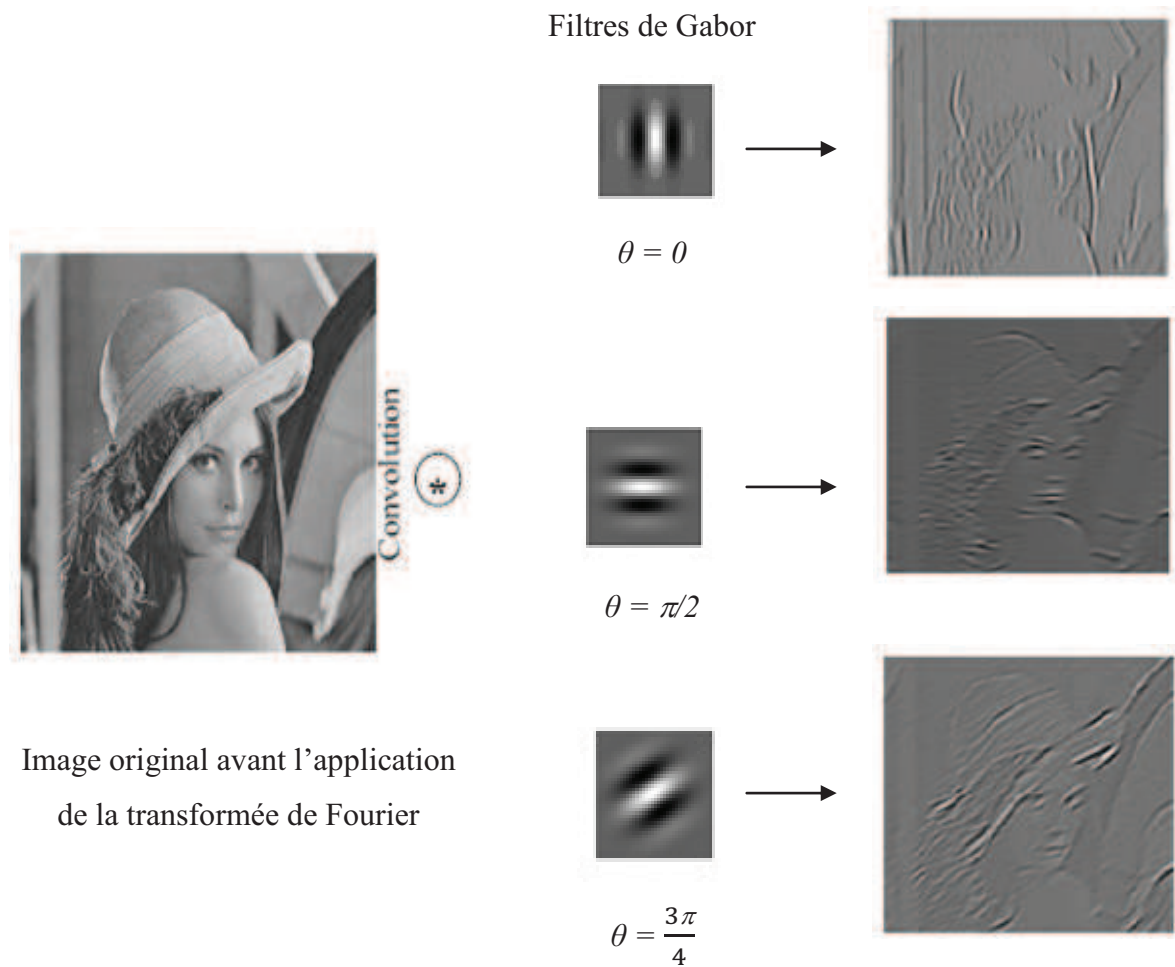


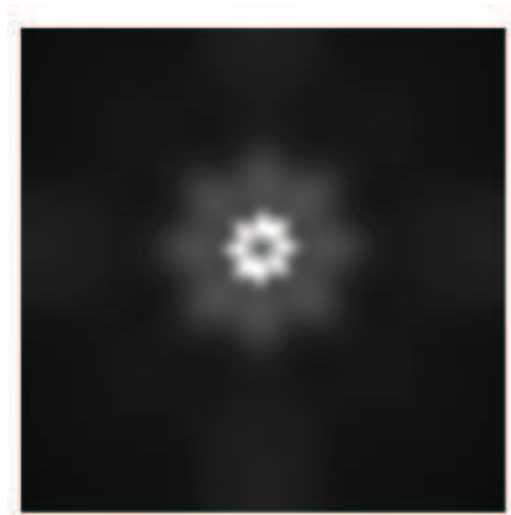
Image original avant l'application de la transformée de Fourier

**Figure 2.6** Exemple d'image filtrée par un filtre de Gabor dans le domaine spatial avec

$$\sigma_x = \sigma_y = 6, f = \frac{\sqrt{2}}{10} \text{ et trois orientations}$$

Comme nous on l'a vu dans la (figure 2.6), les filtres de Gabor permettent d'isoler les contours d'une image d'orientation perpendiculaire à  $\theta$  et répondant à une certaine épaisseur, qui dépend de  $f$ . Ceci justifie le fait que pour détecter l'ensemble des contours d'une image, on lui applique généralement un ensemble de filtres de Gabor que nous appelons banc ou famille de Gabor. Pour cela, on utilise plusieurs filtres de caractéristiques fréquentielles différentes grâce aux quatre paramètres  $(f, \theta, \sigma_x, \sigma_y)$ .

Ainsi l'association d'un ensemble de filtres de Gabor permet de couvrir plus largement l'espace fréquentiel et de détecter un plus grand nombre d'orientations, donc d'extraire tous les contours de l'image. Dans ce cas le filtre final est défini comme suit :



**Figure 2.7** Les filtres de Gabor dans le domaine fréquentiel, pour 3 échelles (de 0.4 à 0.05) et 4 orientations.

Pour pouvoir couvrir au maximum l'espace fréquentiel, on peut utiliser plusieurs orientations  $\theta$  tel que :

$$\theta_k = (k - 1)\pi/N; \quad k=1, 2, \dots, N, \text{ où } N \text{ est le nombre d'orientations.}$$

Ainsi pour 4 orientations nous aurons :  $N = 4, k = 1, 2, 3, 4$ .

$$\theta_1 = 0, \quad \theta_2 = \frac{\pi}{4}, \quad \theta_3 = \frac{\pi}{2}, \quad \theta_4 = \frac{3\pi}{4}$$

### 2.2.2.3 Caractéristiques extraites à partir des filtres de Gabor [8]

L'idée d'approche de Gabor est alors de concevoir un filtrage particulièrement sélectif en fréquence et orientation dans le but de caractériser au détail près l'image. Chaque filtre est alors appliqué à l'image d'origine, et une analyse éventuelle permettra de créer un simple vecteur de caractéristiques de ces images sont la moyenne, la variance et l'énergie des sorties de chacun des filtres (banc).

#### 2.2.2.3.1 Calcul d'énergie

L'énergie correspond à la somme des niveaux de gris de l'ensemble des éléments primitifs (pixels). Après application des filtres de Gabor sur l'image avec différentes

orientations et échelles (fréquences), nous obtenons un ensemble de grandeurs (magnitude) qui mesure l'homogénéité sous la forme suivante:

$$E_i = \sum_x \sum_y [G_i(x, y)]^2 \quad (2.19)$$

$i=1, \dots, N$  ( $N$  étant le nombre de filtre)

$G_i(x, y)$  : image filtrée

Ces grandeurs représentent la teneur en énergie à différentes balances (échelles) et orientations de l'image.

### 2.2.2.3.2 Calcul de la moyenne et la variance

A partir de l'image filtrée, on peut extraire deux paramètres, la moyenne  $\mu$  qui caractérise l'intensité lumineuse de l'image et qui correspond au niveau de gris moyen des éléments primitifs (pixels) et la variance  $\sigma$  qui caractérise le contraste de l'image. Ces paramètres sont calculés de la manière suivante :

$$\mu_i = \sum_x \sum_y \frac{G_i(x, y)}{M \times M} \quad (2.20)$$

$$\sigma_i = \frac{(\sum_x \sum_y |G_i(x, y) - \mu_i|)^2}{M \times M} \quad (2.21)$$

$i = 0, \dots, N$  ( $N$  étant le nombre de filtre)

$M \times M$ : taille de l'image

## 2.2.3 Les moments de Zernike

### 2.2.3.1 Les moments géométriques

D'une manière générale, les moments sont une suite de valeurs calculées par des formules mathématiques qui permettent de caractériser un objet.

Les moments sont souvent utilisés en physique pour décrire la répartition des masses dans un corps. En associant le niveau de gris d'un point de l'image à la masse d'un corps en chaque point, on peut reprendre le même formalisme pour décrire la répartition des niveaux

de gris dans une image [9]. Ils représentent les propriétés spatiales de la distribution des pixels dans l'image. Ils sont facilement calculés et implémentés. Par contre, cette approche est très sensible aux bruits et aux déformations et le temps de calcul de ces moments est très lent [10].

La formule générale des moments géométriques est donnée par la relation suivante [10]:

$$M_{p,q} = \sum_{p=0}^m \sum_{q=0}^n x^p y^q f(x, y) \quad (2.22)$$

Avec  $p + q$  est l'ordre du moment et  $f(x, y)$  le niveau de gris de l'image à traiter.

L'ensemble complet de moment d'ordre  $n$  :  $\{M_{i,j}, i + j \leq n\}$  est  $(n + 1)(n + 2)/2$ , ou les moments d'ordre  $n$  représentent une propriété caractéristique [9] :

- **Ordre 0** : Représente le poids total de l'image  $M_{0,0}$  (surface de l'objet c.à.d. nombre de pixels)
- **Ordre 1** : permet de déterminer les coordonnées du centre de gravité de l'image  $(x_g, y_g)$ :

$$x_g = \frac{M_{1,0}}{M_{0,0}} \quad y_g = \frac{M_{0,1}}{M_{0,0}} \quad (2.23)$$

### 2.2.3.2 Les moments géométriques centrés

Les moments centrés peuvent être calculés en fonction des moments non centrés par la relation suivante [9]:

$$\mu_{ij} = \sum_{r=0}^i \sum_{s=0}^j C_r^i C_s^j (-x_g)^{i-r} (-y_g)^{j-s} M_{rs} \quad (2.24)$$

Avec :  $C_b^a = \frac{a!}{b!(a-b)!}$

Voici l'expression des neuf premiers moments centrés [9]:

$$\mu_{00} = M_{00} \quad (2.25)$$

$$\mu_{10} = \mu_{01} = 0 \quad (2.26)$$

$$\mu_{11} = M_{11} - (M_{00}x_g y_g) \quad (2.27)$$

$$\mu_{20} = M_{20} - (M_{00}x_g^2) \quad (2.28)$$

$$\mu_{02} = M_{02} - (M_{00}y_g^2) \quad (2.29)$$

$$\mu_{21} = M_{21} - (M_{20}y_g) - (2 M_{11}x_g) - (2 M_{00}x_g^2y_g) \quad (2.30)$$

$$\mu_{12} = M_{12} - (M_{02}x_g) - (2M_{11}y_g) - (2M_{00}y_g^2x_g) \quad (2.31)$$

$$\mu_{30} = M_{30} - (3M_{20}x_g) + (2M_{00}x_g^3) \quad (2.32)$$

$$\mu_{03} = M_{03} - (3M_{02}y_g) + (2M_{00}y_g^3) \quad (2.33)$$

### 2.2.3.3 Les moments géométriques centrés normalisés

Les moments centrés  $\mu_{ij}$  sont utilisés pour le calcul des moments centrés normalisés  $\eta_{ij}$  par la relation suivante [9]:

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{(i+j+2)/2}} \quad (2.34)$$

### 2.2.3.4 Les moments géométriques centrés normalisés de Hu

A partir des moments géométriques, Hu a proposé un ensemble de sept moments invariants aux translations, rotations et au changement d'échelle. Ils sont très utilisés dans la littérature pour la description de formes en vue d'une classification, mais sont assez sensibles aux bruits. Par ailleurs, cette famille de descripteurs n'est pas orthogonale [10].

$$\phi_{p,q} = \sum_i \sum_j (i-x_g)^p (j-y_g)^q f(i,j) \quad (2.35)$$

Voici les sept moments invariants du second et troisième ordre de Hu [9]

#### Invariant du second ordre

$$\phi_1 = \eta_{20} + \eta_{02} \quad (2.36)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4 \eta_{11}^2 \quad (2.37)$$

#### Invariant du troisième ordre

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (2.38)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (2.39)$$

$$\begin{aligned} \phi_5 = & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\ & [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (2.40)$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] 4 \eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (2.41)$$

$$\begin{aligned} \phi_7 = & (3 \eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3 \eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) \\ & [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (2.42)$$

### 2.2.3.5 Les moments orthogonaux

Par opposition aux moments géométriques qui sont définis par une base quelconque  $(x^p y^q)$ , les moments orthogonaux, comme leur nom l'indique, sont définis dans une base orthogonale, ce qui évite la redondance des informations portées par chacun des moments. Les moments orthogonaux les plus utilisés sont les moments de Zernike, pour ce dernier on précisera la définition suivante [10].

### 2.2.3.6 Les moments de Zernike orthogonaux

Par opposition aux moments géométriques qui sont définis par une base quelconque  $(x^p, y^q)$ . Ce dernier inconvénient est surmonté par les moments de Zernike qui possèdent une propriété d'orthogonalité, ces moments sont définis dans une base orthogonale, ce qui évite la redondance des informations portées par chacun des moments.

L'intérêt du calcul des moments de Zernike repose sur l'invariance par translation, par changement d'échelle et par rotation. Ils constituent un espace vectoriel dans lequel l'image de la forme est projetée. Cela permet de s'affranchir du problème de normalisation rencontré dans le cas des moments de Hu, mais aussi d'aboutir à une description plus précise des formes, par utilisation d'ordres plus élevés, en maintenant les propriétés d'invariance [10].

#### 2.2.3.6.1 Calcul des moments de Zernike

Ce type de moment a été initialement introduit par Teague et qui sont définis comme des polynômes complexes et forme un ensemble orthogonal complet définie sur le disque de cercle unité  $(x^2 + y^2 \leq 1)$ . Ils correspondent à la projection de la forme  $f(x, y)$  décrivant une image sur un espace de polynômes  $V_{nm}(\rho, \theta)$  complexes orthogonaux comme suit [9]:

$$V_{nm}(\rho, \theta) = R_{nm}(\rho) \exp(jm\theta) \quad (2.43)$$

$n$ : nombre entier positif ou nul qui signifie l'ordre du moment.

$m$ : nombre entier positif ou négatif correspond au nombre de répétition sujet à ses conditions :  $n - |m|$  pair,  $|m| \leq n$

$\rho$ : longueur du vecteur à partir de l'origine au pixel  $(x, y)$

$\theta$ : angle entre le vecteur  $\rho$  et l'axe X dans le sens inverse de l'aiguille d'une montre

$$V_{nm}(\rho, \theta) = R_{nm}(\rho) \exp[jm \tan^{-1}(y/x)] \quad (2.44)$$

$R_{nm}(\rho)$  est le polynôme radial orthogonal défini par:

$$R_{nm}(x, y) = \sum_{s=0}^{n-|m|} (-1)^s \frac{(m-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} (x^2 + y^2)^{\frac{n-2s}{2}} \quad (2.45)$$

Soit  $\rho = \sqrt{x^2 + y^2}$  la longueur du vecteur à partir de l'origine du pixel  $(x, y)$  et  $\theta = \tan^{-1}(y/x)$  l'angle entre l'axe X et ce vecteur.  $R_{nm}(\rho)$  la représentation en coordonnées polaires ( $x = \rho \cos \theta, y = \rho \sin \theta$ ) de  $R_{nm}(x, y)$  qui est un polynôme de degré  $n$  [11] définit comme suit:

$$R_{nm}(\rho) = \sum_{s=0}^{n-|m|} (-1)^s \frac{(m-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} (\rho)^{n-2s} \quad (2.46)$$

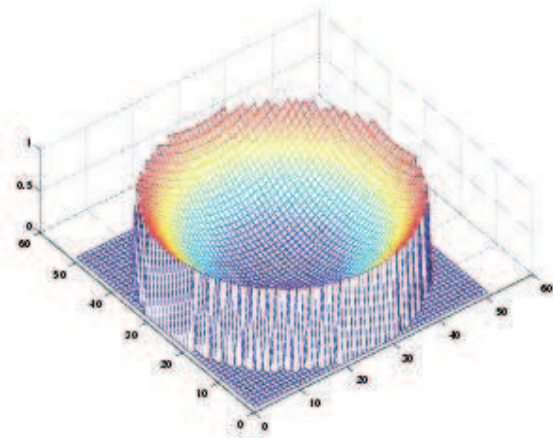
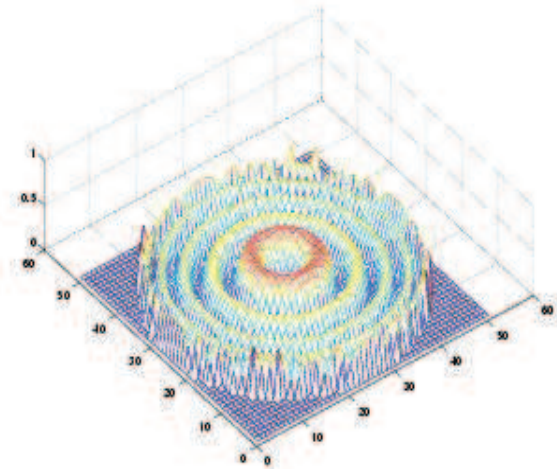
Les moments de Zernike d'ordre  $n$  et de répétition  $m$ , est un nombre complexe défini comme suit :

$$A_{nm} = \frac{n+1}{\pi} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) [V_{nm}(x, y)]^* \quad (2.47)$$

Où  $x^2 + y^2 \leq 1$  et  $*$  défini le complexe conjugué

$f(x, y)$ : est le niveau de gris d'un pixel de l'image  $f$  sur laquelle on calcule le moment.

La figure 2.8 montre un exemple de tracé module des moments d'ordre 2 avec différentes répétitions.

Figure A:  $n=2, m=2$ Figure B:  $n=2, m=12$ 

**Figure 2.8** Tracé des modules des polynômes de Zernike d'ordre 2 avec différentes répétitions

### 2.2.3.7 Caractéristiques extraites à partir des moments de Zernike

Les moments de Zernike sont calculés à partir des images en niveau de gris, ces derniers sont des nombres complexes et les paramètres qu'on a choisis comme caractéristiques sont le module et la phase.

### 2.2.4 La transformée en ondelettes

L'analyse par ondelettes a été introduite au début des années 1980. Il s'agit de donner une représentation des signaux permettant la mise en valeur simultanée des informations temporelles et fréquentielles (localisation temps-fréquence).

L'idée originale sur laquelle sont basées les ondelettes est apparue vers les années 1940 grâce au physicien Denis Gabor, qui a introduit la notion de la transformée de Fourier (TF) à fenêtre glissante dans le but de remédier au problème de localisation temps-fréquence en proposant de multiplier le signal par une fonction localisée dans le temps (fenêtre), et ensuite appliquer la TF. Il s'agit de segmenter en tranches de temps fixe le signal à analyser pour appliquer par la suite la TF à chaque tranche [12].

L'inconvénient de cette méthode est que la taille de la fenêtre reste inchangée au cours de l'analyse. Or pour étudier un signal qui a en général une allure irrégulière, il est potentiellement intéressant de pouvoir changer la taille de la fenêtre analysante en temps.

C'est à partir de là que sont nées les ondelettes qui s'adaptent d'elles mêmes à la taille et aux caractéristiques recherchées [12].

### 2.2.4.1 Les aspects théoriques de la transformée en ondelettes

#### 2.2.4.1.1 Ondelette Mère

L'analyse en ondelette adopte une fonction prototype d'ondelettes connue sous le nom d'ondelette mère. On aboutira à représenter n'importe quel signal de  $L^2$  par une base d'ondelettes qui ne sont que des versions dilatées et translatées de l'ondelette mère. La représentation d'un signal continu  $x(t)$  par une base d'ondelettes est donnée par :

$$x(t) = \sum C(s, \tau) \varphi_{s\tau}(t) \quad (2.48)$$

Où  $C(s, \tau)$  est le poids de la contribution de l'ondelette  $\varphi_{s\tau}$  liée à l'ondelette mère par :

$$\varphi_{s\tau}(t) = \frac{1}{\sqrt{s}} \varphi\left(\frac{t-\tau}{s}\right) \quad (2.49)$$

Où :  $\tau$  : facteur de translation,  $s$  : facteur d'échelle,  $\varphi$ : ondelette mère;

$\frac{1}{\sqrt{s}}$  : facteur de normalisation de l'énergie afin que le signal transformé ait la même énergie à toutes les échelles.

Grâce au changement d'échelle (figure 2.9), les fonctions de base d'ondelette peuvent être étirées ou comprimées dans la fenêtre d'analyse. Ceci aide à isoler des discontinuités du signal parce que de courtes fonctions de base sont employées pour obtenir l'information détaillée de fréquence tandis que de longues fonctions de base sont employées pour rechercher l'information approximative de fréquence du signal.

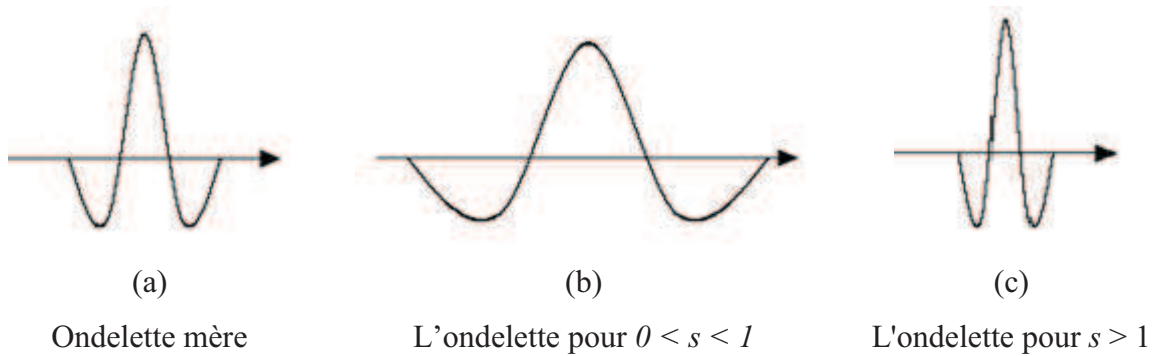


Figure 2.9 Illustration de la variation du facteur d'échelle.

### 2.2.4.1.2 Propriétés des ondelettes [13]

Toute fonction oscillante d'énergie finie et de moyenne nulle représente une ondelette.

La transformée de Fourier de l'ondelette  $\varphi$  est donnée par :

$$\Psi(w) \Big|_{w=0} = \int_{-\infty}^{\infty} \varphi(t) e^{-j\omega t} dt = \int_{-\infty}^{\infty} \varphi(t) dt = 0 \quad (2.50)$$

$\Psi$  est la TF de l'ondelette  $\varphi$ .

La condition nécessaire pour que  $\varphi$  puisse être utilisée comme une ondelette analysante est qu'elle vérifie la condition d'admissibilité suivante:

$$\int_{-\infty}^{\infty} \frac{|\psi(w)|^2}{|w|} dw < \infty \quad (2.51)$$

### 2.2.4.2 La transformée en ondelettes continue

La transformée en ondelette continue (TOC) utilise des translations et des dilatations de la fonction ondelette mère durant tout l'intervalle temporel de manière continue. Elle est définie par [12]:

$$w[x(t)] = C(s, \tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} x(t) \varphi\left(\frac{t-\tau}{s}\right) dt \quad (2.52)$$

Ainsi, la pondération  $C(s, \tau)$ , appelée aussi coefficients en ondelette et définie par (2.48), n'est autre que la transformée en ondelette de la fonction  $x(t)$ . Cette transformation est en théorie infiniment redondante puisque l'ondelette est translatée de manière continue.

### 2.2.4.3 La transformée en ondelettes discrète

Pour plier au problème de redondance d'information et au calcul des intégral qui demande beaucoup d'espace mémoire, il est donc nécessaire de discrétiser la transformée. Dans le cas de la TOC, on peut utiliser la variation de l'échelle pour réduire la fréquence d'échantillonnage. La transformée en ondelettes discrète (TOD) est produite pour surmonter le problème de redondance de la TOC. Cette redondance mobilise une grande quantité de ressources de calculs. La TOD, au contraire, fournit suffisamment d'information, tant pour l'analyse que pour la reconstruction du signal original, en un temps de calcul notablement réduit. La TOD est considérablement plus simple à implémenter que la TOC.

La TOD translate et dilate l'ondelette selon des valeurs discrètes des facteurs d'échelle et de translation telles que [12]:

$$\begin{aligned} s &= s_0^j; & \tau &= k \tau_0 s_0^j \\ s_0 &> 1; & \tau_0 &> 0; & k, j &\in Z \end{aligned} \quad (2.53)$$

La TOD peut alors être exprimée par :

$$w_d[x(t)] = C(s_0, \tau_0) = \frac{1}{\sqrt{s_0^j}} \int_{-\infty}^{\infty} x(t) \varphi\left(\frac{t - k s_0^j \tau_0}{s_0^j}\right) dt \quad (2.54)$$

Dans la pratique, la transformée en ondelettes discrète est calculée par application du filtre passe-bas et passe-haut à l'image, suivis du sous-échantillonnage par deux, c'est-à-dire en omettant une valeur sur deux. Ceci est effectué sur les lignes et les colonnes. L'opération de sous-échantillonnage est également appelée la décimation.

### 2.2.4.4 Analyse multi-résolution

L'analyse multi-résolution (figure 2.13) de  $L^2(\mathbb{R})$ , développée par Mallat et Meyer, permet la représentation d'un signal par des approximations successives, où chaque approximation est une version lissée de la précédente qui n'ont pas la même résolution. Ces

échelles sont définies par des espaces imbriqués  $\{V_j\}_{j \in \mathbb{Z}}$  qui sont soumis aux conditions suivantes [14]:

$$\forall (j, k) \in \mathbb{Z}^2, f \in V_j \Leftrightarrow f(t - 2^j k) \in V_j \quad (2.55)$$

La condition 2.55 indique la stabilité de l'espace  $V_j$  pour n'importe quelle translation de l'ordre  $2^j k$

$$\forall j \in \mathbb{Z}, V_{j+1} \subset V_j \quad (2.56)$$

La condition 2.56 exprime que l'espace du niveau de résolution  $j + 1$  découle de l'espace du niveau de résolution  $j$

$$\forall j \in \mathbb{Z}, f \in V_j \Leftrightarrow f\left(\frac{1}{2}\right) \in V_{j+1} \quad (2.57)$$

La condition 2.57 indique l'approximation de la fonction  $f$  dans l'espace  $V_{j+1}$  est obtenue à partir de la dilatation par 2 de son approximation dans l'espace précédent  $V_j$

$$\lim_{j \rightarrow \infty} V_j = \bigcap_{-\infty}^{+\infty} V_j = \{0\} \quad (2.58)$$

La condition 2.58 exprime qu'on n'a pas l'approximation du signal quand le niveau de la résolution  $j$  tend vers  $+\infty$

$$\lim_{j \rightarrow -\infty} V_j = \bigcup_{-\infty}^{+\infty} V_j = L^2(\mathbb{R}) \quad (2.59)$$

La condition 2.59 exprime que l'approximation du signal reproduit le signal original quand le niveau de la résolution  $j$  tend vers  $-\infty$

$$\exists \phi \in V_0 \text{ tel que } \{\phi(t - n)\}_{n \in \mathbb{Z}} \text{ est une base orthonormée de } V_0 \quad (2.60)$$

La condition 2.60 indique qu'il y a une fonction de l'espace  $V_0$  qui forme une base orthonormée de cet espace. Cette fonction est nommée fonction d'échelle  $\phi$

### 2.2.4.5 La décomposition dyadique

La décomposition dyadique est obtenue en décomposant le signal avec des bancs de filtres en coefficients d'approximations et en coefficients de détails pour le premier niveau. Pour les autres niveaux, les décompositions sont obtenues en décomposant de nouveau les coefficients d'approximations des niveaux précédents en coefficients d'approximations  $x_0^{j+1}$  et

en coefficients de détails  $x_1^{j+1}$ . Les coefficients de détails ne sont pas décomposés dans la décomposition dyadique. La figure 2.10 illustre une décomposition dyadique [14]:

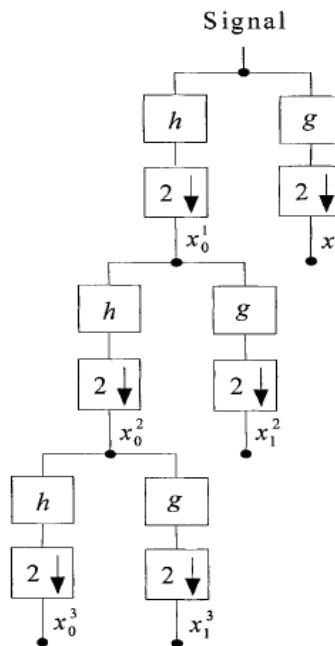


Figure 2.10 La décomposition Dyadique

Où h : filtre passe bas ; g : filtre passe haut

### 2.2.4.6 Transformée en ondelettes à deux dimensions

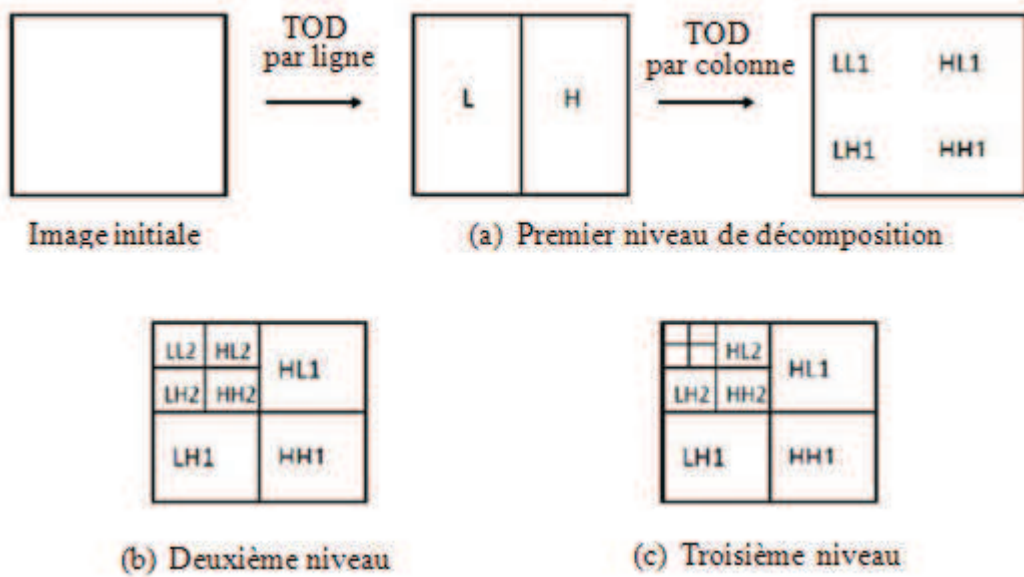
L'extension à deux dimensions de la TOD (TOD-2D) est essentielle pour la transformation des signaux à deux dimensions (2D), comme une image numérique. L'approche simple pour la mise en œuvre de la TOD-2D est d'effectuer la TOD unidimensionnelle (TOD-1D) ligne par ligne pour produire un résultat intermédiaire et ensuite effectuer la même TOD-1D en colonne sur ce résultat intermédiaire pour obtenir le résultat final représenté par la figure 2.11.

La TOD-2D s'effectue de manière naturelle par séparation des variables. Le sous espace  $V_2^j(x, y)$  correspondant à la résolution  $j$ , est le produit tensoriel des sous espaces  $A_2^j(x)$  et  $A_2^j(y)$ :

$$V_2^j(x, y) = V_2^j(x) \otimes V_2^j(y) \tag{2.61}$$

La base d'interpolation est sous la forme :

$$\phi(x, y) = \phi(x) \otimes \phi(y) \tag{2.62}$$



**Figure 2.11** Représentation de la TOD d'une image.

L'information de différence entre deux approximations de l'image originale se calcule à partir de trois ondelettes bidimensionnelles séparables, définies par:

$$\phi^H(x, y) = \phi(x) \phi(y) \tag{2.63}$$

$$\phi^V(x, y) = \phi(x) \phi(y) \tag{2.64}$$

$$\phi^D(x, y) = \phi(x) \phi(y) \tag{2.65}$$

La TOD-2D fournit à chaque échelle les quatre sous-images suivantes :

Une image de basse résolution :  $A^{j+1}$

Une image de détails horizontaux :  $D^H$

Une image de détails verticaux :  $D^V$

Une image de détails diagonaux :  $D^D$

La figure 2.12 illustre un niveau de représentation multi-résolution d'une image.

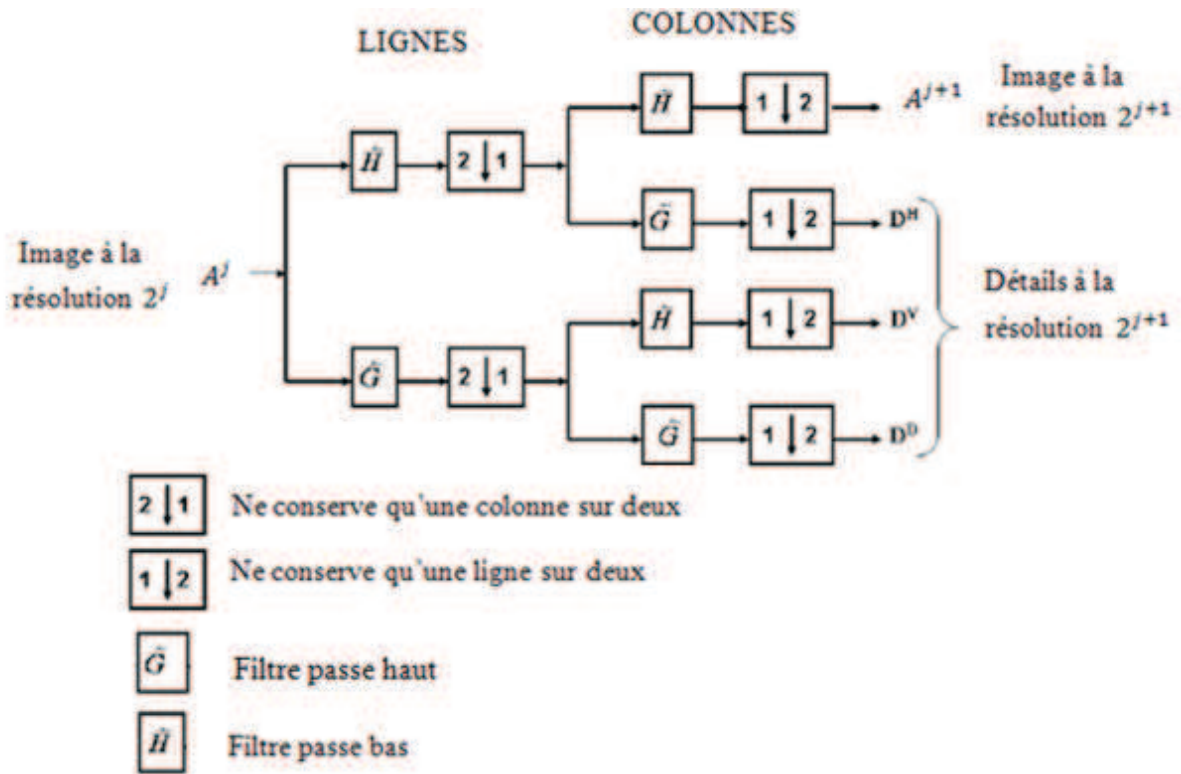


Figure 2.12 Un niveau de représentation multi-résolution d'une image.

2.2.4.7 Les familles d'ondelettes

Il existe une infinité de fonctions d'ondelettes parce que toute fonction oscillante localisée est une ondelette mère possible. Toutefois, elles ne possèdent pas toutes des propriétés intéressantes. Aussi, de nombreux spécialistes des ondelettes ont construit des familles d'ondelettes possédant certaines propriétés remarquables. Parmi ces familles, les ondelettes de Haar sont les plus simples, mais elles ne sont pas bien localisées. Ingrid Daubechies a construit des ondelettes à support compact qui permettent d'utiliser des filtres de taille finie. Une autre famille d'ondelettes est la famille des splines dont la réponse fréquentielle est bien localisée. Les différentes familles d'ondelettes sont utilisées selon leurs propriétés en fonction du problème à résoudre.

2.2.4.7.1 Ondelette de Haar

L'ondelette de Haar, est une ondelette créée par Alfréd Haar en 1909. On considère que c'est la première ondelette connue. Elle est la plus simple à comprendre et à implémenter. C'est une fonction dilatée et/ou translatée de la fonction mère  $\varphi$  (figure 2.13 (a)) qui vaut :

$$\varphi(t) = \begin{cases} 1 & \text{pour } 0 \leq t < \frac{1}{2} \\ -1 & \text{pour } \frac{1}{2} \leq t < 1 \\ 0 & \text{sinon} \end{cases} \quad (2.66)$$

L'inconvénient majeur de cette analyse est sa très mauvaise résolution fréquentielle. Son intérêt majeur est la grande simplicité des filtres et donc de l'algorithme de calcul [13].

#### 2.2.4.7.2 Ondelette de Daubechies

Nommées d'après leur créatrice Ingrid Daubechies, les ondelettes de Daubechies sont une famille d'ondelettes orthogonales définissant une transformée en ondelettes discrète, caractérisées par un nombre maximal de moments dissipants pour un support donné. Pour chaque type d'ondelette de cette classe, il existe une fonction d'échelle (appelée aussi ondelette mère) qui génère une analyse multi-résolution orthogonale.

Inconvénient : l'abandon de la linéarité en phase ou plus simplement de la symétrie droite-gauche très recherchée en traitement d'image.

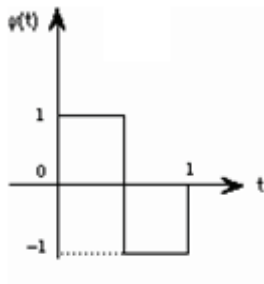
L'avantage de compacité des fonctions de base et surtout des filtres associés est l'économie en cout de calcul qu'elle procure. Le filtrage peut être effectué directement par l'application de masque de convolution. La figure 2.13(b) nous montre la représentation graphique de l'ondelette daubechies 4.

#### 2.2.4.7.3 Ondelette Symmlets

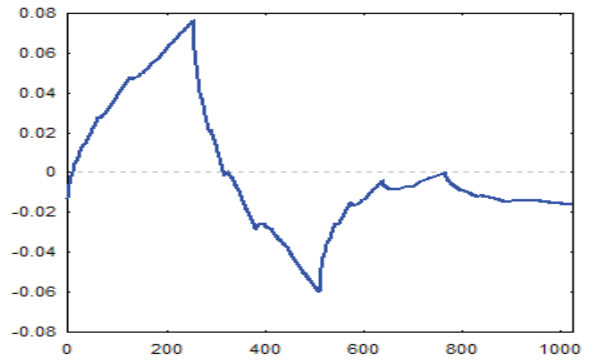
Daubechies a construit des ondelettes à support compact les plus symétriques possible. En effet, il n'existe pas d'ondelettes à support compact dans une analyse multi-résolution orthogonale qui soit symétriques, exceptée l'ondelette de Haar qui est anti-symétrique. La figure 2.13 (c) nous montre la représentation graphique de l'ondelette symmlet 4.

#### 2.2.4.7.4 Ondelettes de Coiflets

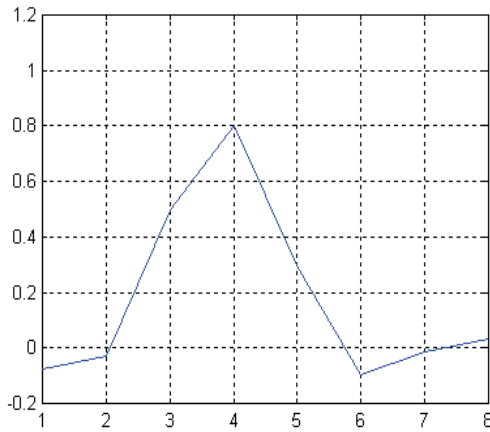
Coifman a suggéré à daubechies de construire une famille telle que la fonction d'échelle ait, elle aussi, des moment nuls. la figure 2.13 (d) nous montre la représentation graphique de l'ondelette coiflet 2.



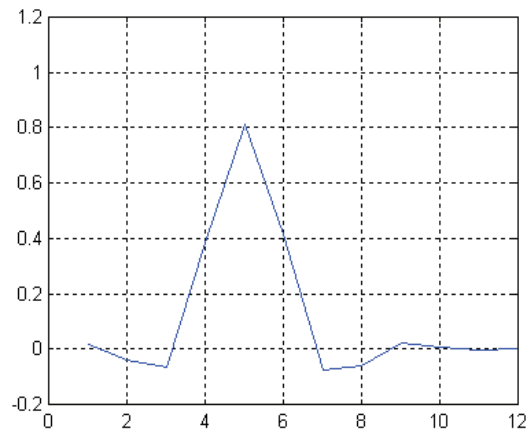
(a) Ondelette de Haar



(b) Ondelette de Daubechies 4



(c) Ondelette de Symmlet4



(d) Ondelette de Coiflet2

**Figure 2.13** Quelques familles d'ondelettes.

**2.2.4.8 Conclusion**

Nous avons abordé dans ce chapitre brièvement des notions de base portant sur quatre parties différentes à savoir les détecteurs de directions basés sur les masques de Kirsch, les moments de Zernike, les filtres de Gabor et la théorie des ondelettes. Dans ce qui suit, nous allons donner une introduction aux machines à vecteurs de support (SVM).

# Chapitre 3

**Théorie sur les SVM**

### 3.1 Introduction

Les machines à vecteurs de support (SVM) sont de nouvelles méthodes non paramétriques qui permettent de modéliser des formes complexes en se basant sur la théorie de l'apprentissage statistique [15]. Depuis, les SVM ont prouvé à maintes reprises leur efficacité pour résoudre des problèmes de classification dans des espaces de grandes dimensions.

Dans la suite de ce chapitre, nous présenterons les fondements théoriques sur lesquels reposent les machines à vecteurs de support, à savoir les notions d'hyperplan optimal et la fonction du noyau.

### 3.2 Théorie sur les machines à vecteurs de support

#### 3.2.1 Classification de données binaire

L'origine des machines à vecteurs de support (en anglais Support Vector Machine, SVM) remonte à 1975 lorsque Vapnik et Chervonenkis [16] proposèrent le principe du risque structurel et la dimension VC pour caractériser la capacité d'une machine d'apprentissage. Depuis, les SVM n'ont cessé de susciter l'intérêt de plusieurs communautés de chercheurs de différents domaines d'expertise.

Parmi les problèmes que les machines à vecteurs de support permettent de résoudre efficacement, on trouve des problèmes de classification, ou de régression et ce, en exploitant les résultats de la théorie de l'apprentissage statistique développée en grande partie par Vladimir Vapnik [17]. Par exemple, l'identification/vérification du locuteur [18], la catégorisation de textes [19], la reconnaissance des visages [20] et récemment la reconnaissance des instruments de musique [21].

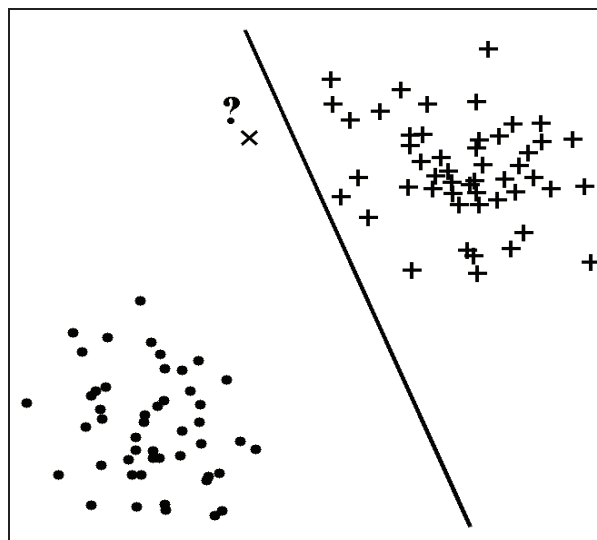
Les machines à vecteurs de support sont avant tout une approche de classification linéaire à deux classes. Elles tentent de séparer des objets issus de deux classes (+1, -1) en cherchant l'hyperplan optimal qui sépare les deux ensembles, en garantissant une grande marge entre les deux classes. Un nombre réduit d'exemples pour la recherche de l'hyperplan est suffisant pour la description de cet hyperplan.

##### 3.2.1.1 L'hyperplan optimal

La notion d'optimalité implique un critère d'évaluation. Dans le cas d'un problème de classification, ce critère pourrait donc être le nombre d'erreurs de classification parmi les

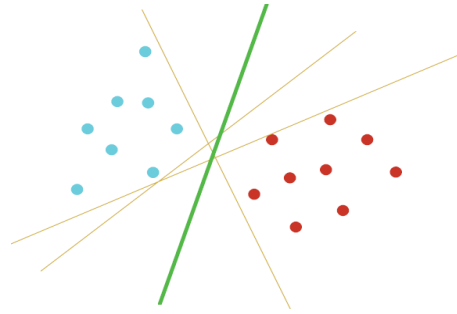
données d'apprentissage. On parle alors de minimisation du risque empirique. Cependant, la minimisation de ce type de critère ne garantit pas la minimisation du risque réel qui correspond à l'espérance des coûts sur les données à venir.

En effet, dans le cas de données linéairement séparables : s'il existe généralement une infinité d'hyperplans qui permettent de séparer les données d'apprentissage, tous ne permettent pas de bien généraliser, c'est-à-dire de bien classer les données à venir. Un exemple simple est illustré à la figure 1. La droite permet effectivement de séparer les données de la première classe, représentées par des plus (+), de celles de la seconde classe, représentées par des points (•), mais cette frontière de décision conduirait à classer la donnée de test, représentée par une croix (×), comme appartenant à la seconde classe, alors que cette donnée appartient probablement à la première classe.



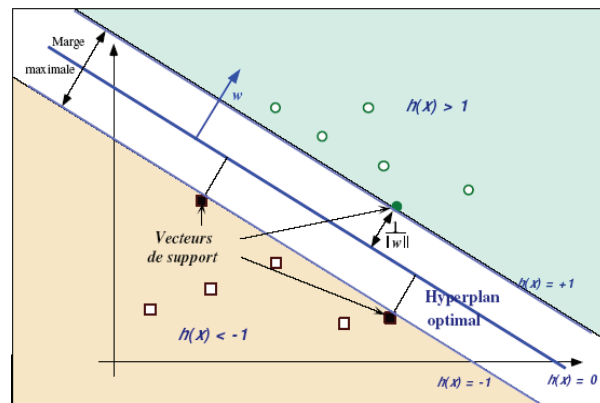
**Figure 3.1** Exemple de frontière de décision sous optimale

Il est évident qu'il existe une multitude d'hyperplans valides mais la propriété remarquable des SVM que seul l'hyperplan optimal maximise la marge (figure 3.2). Intuitivement, le fait d'avoir une marge plus large procure plus de sécurité lorsque l'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. Le schéma qui suit nous montre qu'avec un hyperplan optimal, un nouvel exemple reste bien classé.



**Figure 3.2** Exemple de frontière de décision optimale

Cet hyperplan peut être considéré comme optimal dans le sens où il maximise la marge de séparation, c'est-à-dire la distance entre lui-même et les données d'apprentissage les plus proches de lui qui sont appelées vecteurs de support (voir figure 3.3).



**Figure 3.3** Hyperplan optimal, marge de séparation et vecteurs de support

En pratique, un hyperplan est constitué de l'ensemble des points  $x \in \mathfrak{R}^d$  vérifiant :

$$w \cdot x + b = 0 \quad (3.1)$$

qui est donc caractérisé par un vecteur de poids  $w \in \mathfrak{R}^d$  et une valeur de biais  $b \in \mathfrak{R}$ .

La fonction de décision associée à un hyperplan est donc :

$$h(x) = w \cdot x + b \quad (3.2)$$

La règle de décision consiste alors à classer la donnée  $x$  dans la première classe, caractérisée par le label  $y = 1$ , si  $h(x) \geq 0$ , ou dans la seconde classe caractérisée par le label  $y = -1$ , si  $h(x) < 0$ .

Ainsi, étant donné un ensemble de données d'apprentissage  $\{x_i \in \mathfrak{R}^d; i = 1, \dots, n\}$  et de labels  $\{y_i \in \{1, -1\}; i = 1, \dots, n\}$ , si les données des deux classes sont linéairement séparables, il existe au moins un hyperplan qui vérifie :

$$h(x_i) \cdot y_i > 0, (i = 1, \dots, n) \quad (3.3)$$

La recherche de l'hyperplan optimal, tel que défini précédemment, consiste donc à déterminer les valeurs de  $w$  et  $b$  qui vérifient les contraintes (3.3) et maximisent la marge de séparation qui correspond à la distance minimale entre les données d'apprentissage et leurs projections sur cet hyperplan et qui est égale à :

$$\min_{i=1, \dots, n} \left\{ \frac{|w \cdot x_i + b|}{\|w\|} \right\} \quad (3.4)$$

La marge de séparation est alors égale à  $\frac{1}{\|w\|}$ .

Déterminer l'hyperplan optimal revient donc à résoudre un problème d'optimisation qui dans sa formulation primale consiste à minimiser :

$$\frac{1}{2} \|w\|^2 \quad (3.5)$$

sous les contraintes :

$$y_i (w \cdot x_i + b) \geq 1; (i = 1, \dots, n) \quad (3.6)$$

Notons que le «  $> 0$  » des contraintes (3.3) a été remplacé par «  $\geq 1$  » qui correspond au choix de l'hyperplan canonique et permet de garantir l'unicité de la solution. Une manière de résoudre ce type de problème d'optimisation sous contraintes est d'introduire les multiplicateurs de Lagrange tels que :

$$\alpha_i \geq 0; (i = 1, \dots, n) \quad (3.7)$$

et le Lagrangien :

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i (w \cdot x_i + b) - 1) \quad (3.8)$$

Le Lagrangien  $L$  doit alors être minimisé par rapport à  $w$  et  $b$  et maximisé par rapport aux  $\alpha_i$ . Par conséquent, au point d'équilibre, nous devons obtenir :

$$\frac{\partial L(w, b, \alpha)}{\partial w} = 0 \text{ et } \frac{\partial L(w, b, \alpha)}{\partial b} = 0 \quad (3.9)$$

ce qui entraîne :

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (3.10)$$

et

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad (3.11)$$

La formulation duale du problème d'optimisation revient ainsi à maximiser :

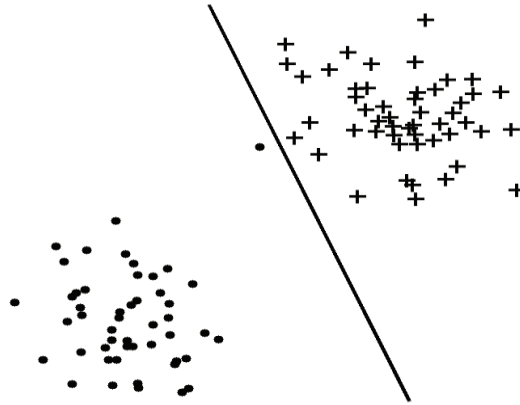
$$\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (3.12)$$

sous les contraintes (3.7) et (3.10)

Notons par ailleurs que seuls les vecteurs de support ont des  $\alpha_i \neq 0$  et que l'hyperplan optimal est donc caractérisé par ce sous-ensemble des données d'apprentissage, ainsi que par les multiplicateurs de Lagrange associés à chacune des données de ce sous ensemble.

Le vecteur de poids  $w$  peut ainsi être calculé à l'aide de l'équation (3.11).

Cependant, dans le cas de données non linéairement séparables, la méthode décrite précédemment ne fonctionne pas. De plus, même si les données d'apprentissage sont linéairement séparables, en présence de données aberrantes comme le montre la figure 3.4, la maximisation de la marge de séparation peut conduire à l'obtention d'un hyperplan qui n'est plus du tout optimal.



**Figure 3.4** Hyperplan sous optimal lié à la présence d'une donnée aberrante

Pour contourner ce problème, Cortes et Vapnik [22] ont introduit la notion de classifieurs à marge molle (« soft margin » en anglais) qui consiste à tolérer que certaines données d'apprentissage violent les contraintes de séparation (3.6). Ils proposent ainsi d'introduire des variables d'écart (« slack variables » en anglais) telle que:

$$\xi_i \geq 0, (i = 1, \dots, n) \quad (3.13)$$

et les contraintes deviennent :

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \quad (i = 1, \dots, n) \quad (3.14)$$

La formulation primale du problème d'optimisation revient alors à minimiser sous les contraintes (3.13) et (3.14) la fonction objective suivante :

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad (3.15)$$

où  $C$  est une constante positive, dite de régularisation. Ainsi, plus la valeur de  $C$  est grande, moins l'algorithme tolère que les données violent les contraintes de séparation (3.6).

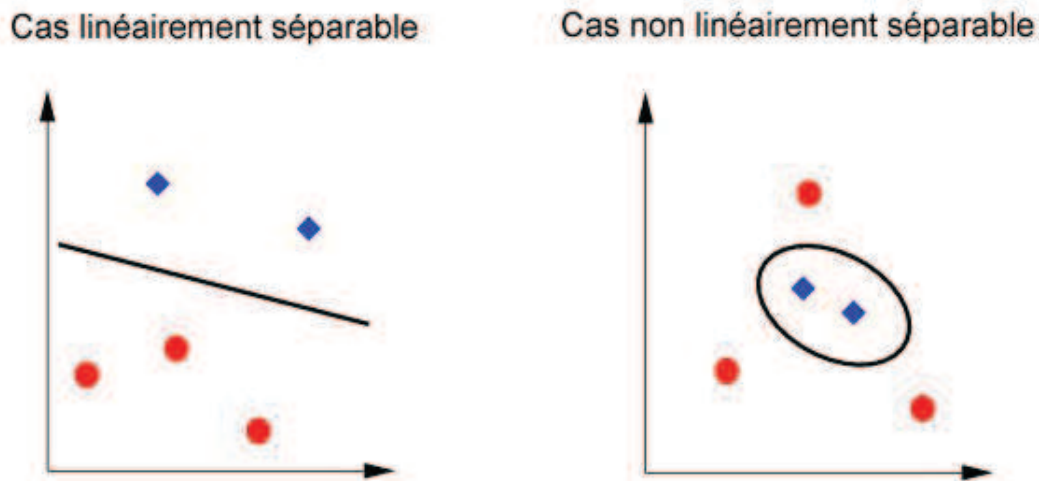
Dans le cas de la formulation duale, on maximise toujours l'expression (3.12) sous les contraintes (3.7) et (3.10), aux quelles on ajoute les contraintes :

$$\alpha_i \leq C \quad (i = 1, \dots, n) \quad (3.16)$$

Notons enfin que les données qui violent les contraintes de séparation (3.6) ont des multiplicateurs de Lagrange différents de zéro ( $\alpha_i \neq 0$ ) et sont donc sélectionnées comme vecteurs de support.

### 3.2.1.2 La notion du noyau

Dans la plupart des problèmes réels il n'y a pas de séparation linéaire possible entre les données, ainsi, 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 3.5** Représentation des cas de données linéairement et non linéairement séparables

Pour surmonter le problème de cas de données non linéairement séparables, l'idée des SVM est basée sur l'utilisation de fonctions dites noyau ou kernel. Etant donné que la formulation duale du problème d'optimisation est uniquement basée sur le calcul de produits scalaires, l'idée consiste alors à trouver une fonction, que l'on nomme noyau, qui correspond à un produit scalaire dans l'espace de re-description de plus grande dimension, mais qui ne nécessite pas de re-décrire les données dans cet espace pour en calculer sa valeur. En effet, cette transformation non linéaire des données peut permettre une séparation linéaire des données dans un nouvel espace.

Ainsi, en remplaçant dans la formulation duale du problème d'optimisation, le produit scalaire entre deux données  $(x_i, x_j)$  par l'expression du noyau :

$$k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (3.17)$$

Finalement, l'apprentissage d'un SVM consiste donc à maximiser la fonction objective :

$$\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j k(x_i, x_j) \quad (3.18)$$

sous les contraintes :

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad \text{et} \quad 0 \leq \alpha_i \leq C \quad (i = 1, \dots, n) \quad (3.19)$$

Après l'apprentissage, la fonction de décision du SVM est égale à :

$$h(x) = \sum_{i=1}^n \alpha_i y_i k(x, x_i) + b \quad (3.20)$$

Et la règle de décision consiste, comme dans le cas de l'hyperplan, à classer la donnée  $x$  dans la classe caractérisée par le label  $y = 1$ , si  $h(x) \geq 0$ , ou dans la classe caractérisée par le label  $y = -1$ , si  $h(x) < 0$ . En d'autres termes, le label d'une donnée inconnue  $x$  est égal à :

$$y = \text{sign}(h(x)) \quad (3.21)$$

Ainsi, le SVM est caractérisé par les données d'apprentissage pour lesquelles  $\alpha_i \neq 0$ , par les coefficients  $\alpha_i$  associés à chacune de ces données que l'on nomme vecteurs de support et par un biais  $b$ . La valeur de ce biais peut être calculée en effectuant la moyenne, pour tous les vecteurs de support pour lesquels  $\alpha_i < C$ , des valeurs de :

$$b_i = y_i - \sum_{j=1}^n y_j \alpha_j k(x_i, x_j) \quad (3.22)$$

### 3.2.1.3 Les hyper paramètres

Comme pour la plupart des algorithmes d'apprentissage, l'utilisation de SVM nécessite de fixer au préalable certains paramètres à savoir la constante de régularisation  $C$  et le type du noyau qui ne peuvent pas être modifiés lors de l'apprentissage. Toutefois, non seulement il existe différents types de noyaux, mais chacun d'entre eux est généralement caractérisé par une ou plusieurs constantes. Le noyau polynomial en est un exemple particulier où seul le paramètre lié à ce noyau représentant le degré  $p$  du polynôme est optimisé.

$$k(x, x_j) = (x \cdot x_j + 1)^p \quad (3.23)$$

Notons que d'autres types de noyaux existent tels que le noyau linéaire dont l'expression est donnée respectivement comme suit :

$$k(x, x_j) = (x \cdot x_j) \quad (3.24)$$

Un autre type de noyau très utilisé est le noyau Gaussien dont l'expression courante est :

$$k(x, x_j) = \exp\left(-\frac{\|x - x_j\|^2}{\sigma^2}\right) \quad (3.25)$$

Ce noyau nécessite donc l'optimisation d'un seul hyper paramètre qui est égal à :  $\lambda = \frac{1}{\sigma^2}$ . De plus, si ce paramètre est plus délicat à ajuster que le degré  $p$  du polynôme, une optimisation fine de celui-ci permet généralement d'obtenir de meilleurs résultats. Il est donc particulièrement important de trouver les bons hyper paramètres. De plus ces hyper paramètres ne sont pas indépendants et doivent donc être optimisés conjointement. La solution la plus simple est d'utiliser une grille de recherche [23]. La méthode consiste alors à échantillonner les valeurs de  $C$  et de  $\lambda$ , à entraîner un SVM pour chaque combinaison et à évaluer la capacité à généraliser de chaque SVM sur une base de validation.

Comme on vient de le voir, dans sa formulation originale, un SVM ne permet de traiter que des problèmes à deux classes. Pour résoudre des problèmes à plusieurs classes comme dans le cas de notre application qui présente dix classes de chiffres manuscrits, plusieurs approches ont été proposées pour étendre l'utilisation de SVM à ces cas.

### 3.2.2 Classification de données multi classes

Pour aborder les problèmes multi classes à l'aide de SVM, plusieurs approches ont été proposées et parmi elles on trouve :

#### 3.2.2.1 Approche un contre tous

L'approche "un contre tous" est la plus simple et la plus ancienne des méthodes de décomposition d'un problème multi classes en un ensemble de sous problèmes binaires et à construire indépendamment un SVM à valeurs réelles pour chacun d'entre eux. Autrement dit, cette stratégie consiste à construire autant de SVM qu'il y a de classes. Ainsi, chaque SVM est alors entraîné à séparer les données d'une classe qui seront étiquetées +1, de celles de toutes les autres classes qui seront étiquetées -1. Le k-ième classifieur est destiné à distinguer la classe d'indice k de toutes les autres. Pour classer une nouvelle donnée, on la présente donc aux  $K$  classifieurs, et la décision s'obtient en application du principe "winner-

takes-all" : l'étiquette retenue est celle associée au classifieur ayant renvoyé la valeur la plus élevée.

### 3.2.2.2 Approche un contre un

Une autre méthode de décomposition très naturelle est la méthode "un contre un". Elle consiste à construire un SVM pour chaque paire de classes, soit  $\frac{K(K-1)}{2}$  SVM pour un problème à  $K$  classes. Chaque classifieur est donc entraîné à séparer les données d'une classe de celles d'une autre classe. Pour affecter une donnée inconnue, on la présente donc à  $C_K^2$  classifieurs, et la décision s'obtient habituellement en effectuant un vote majoritaire ("max-wins voting"). Cependant, il est possible que plusieurs classes reçoivent le même nombre de votes. Dans ce cas, il s'agit de prendre la décision soit d'une manière aléatoire ou bien d'associer la donnée inconnue à la classe la plus proche, qui est déterminée en considérant parmi les classes à départager la valeur de sortie négative de plus faible valeur absolue.

## 3.3 Applications des SVM

SVM est une méthode de classification qui montre de bonnes performances dans la résolution de problèmes variés. Cette méthode a montré son efficacité dans de nombreux domaines d'applications tels que le traitement d'image, la catégorisation de textes ou le diagnostic médicale et ce même sur des ensembles de données de très grandes dimensions.

## 3.4 Conclusion

Pour conclure cette brève présentation des machines à vecteurs de support, nous pouvons dire que cet algorithme d'apprentissage permet de résoudre efficacement non seulement des problèmes de classification de données binaires linéairement séparables mais également des données non linéairement séparables, et ce, en proposant l'utilisation la fonction de kernel. Cependant, dans le cadre de la reconnaissance de chiffres manuscrits où le nombre de classes à traiter est supérieur à deux, il est nécessaire de décomposer un problème multi classes en sous problèmes binaires pour permettre l'utilisation de SVM dans la tâche de classification. Ainsi, nous avons proposé dans ce manuscrit deux méthodes de décomposition, à savoir la méthode un contre un et celle de un contre tous, seule la première est considérée ici, dans notre travail.

# Chapitre 4

**Tests et résultats**

#### 4.1 Introduction

Dans ce chapitre nous allons présenter les résultats sur la base de données de chiffres manuscrits, composés de dix classes.

#### 4.2 La base de données USPS

La base de données d'USPS (de l'anglais «US postal Service») est un benchmark (banc d'essai) bien connu au sein de la communauté de reconnaissance de formes. Cet ensemble contient 9298 images de chiffres manuscrits dont 7291 images d'apprentissage et 2007 images de test. Ces images ont été saisies à partir d'images d'enveloppes collectées au centre CEDAR à Buffalo (États Unis) [24]. Chaque image de chiffre est représentée par 16×16 pixels de niveau de gris allant de 0 à 255 (Figure 4.1). Il est connu que l'ensemble de test de USPS est plutôt difficile - l'erreur humaine se situe autour de 2.5% [25].



**Figure 4.1** Echantillons représentant la première centaine d'images binarisées et leurs étiquettes dans la base de données USPS

Il est noté que les chiffres peuvent présenter les défauts suivants :

- Présence de chiffres fragmentés
- Forme provenant des cas de chevauchement des chiffres
- Présence de formes de chiffres inconnus
- Cas de confusion entre le chiffre 1 et le 7, le chiffre 4 et 9, 2 et 9 et le chiffre 3 et 5.

La figure 4.2 montre certaines imperfections qui se trouvent dans les bases de données.

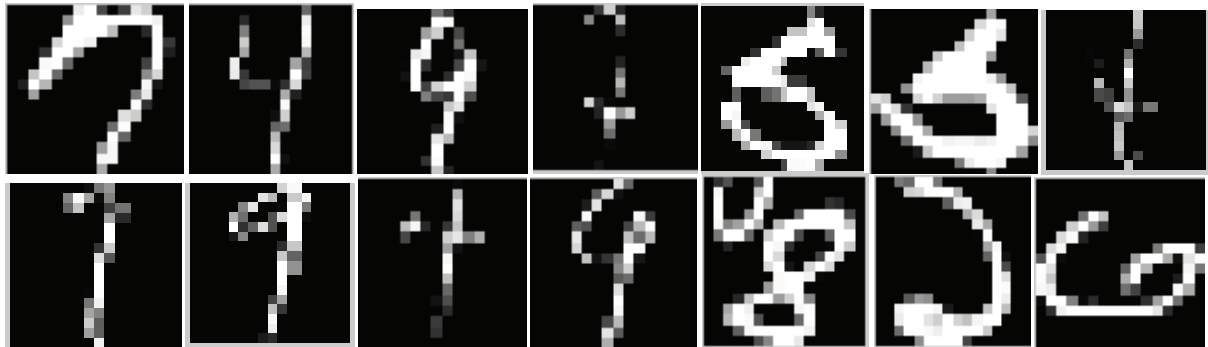


Figure 4.2 Les défauts de chiffres dans les deux bases

### 4.3 Le classifieur utilisé

Nous avons opté pour le classifieur SVM pour ses performances qu'il a montré dans différentes applications.

Nous avons choisi l'approche un-contre-un, où chaque classifieur  $c(i, j)$  tente d'approximer la fonction de décision séparant les classes  $i$  et  $j$  avec la condition ( $i \neq j$ ). La technique des SVM est une méthode de classification qui montre de bonnes performances dans la résolution des problèmes variés. Nous avons effectué une étude comparative en employant quatre types de méthode d'extraction de caractéristiques qui sont : masques de Kirsch, filtres de Gabor, moments de Zernike, transformées en ondelettes qui sont : Haar, symmlet, Coiflet et Daubechies.

Notant que les différents paramètres du classifieur SVM (tel que  $C$  qui est la fonction de coût de mauvaise classification et  $\lambda$  qui est le paramètre du noyau Gaussien) ainsi que ceux des méthodes d'extractions (fréquence  $f$ , nombre d'orientation  $N$ ,  $\sigma_x$ ,  $\sigma_y$ , ordre  $n$ ) sont fixés expérimentalement. Nous avons opté pour le noyau Gaussien pour ses bons résultats et son temps d'exécution rapide.

### 4.4 Tests et résultats

Différentes méthodes d'extractions de paramètres sont testées dans ce travail :

#### 4.4.1 Méthode basée sur les masques de Kirsch

L'application des masques de Kirsch à l'image initiale a généré quatre images de caractéristiques locales, ces dernières et les images initiales qui sont les caractéristiques globales ont été réduites à des images de taille 8x8 et 4x4.

##### 4.4.1.1 Implémentation de l'image compressée en taille 8x8

Ce mode de compression est de séparer l'image initiale en blocs de taille (2x2), le principe consiste à calculer la moyenne de chaque bloc qui sera le pixel de la l'image compressée. En conséquence le vecteur de caractéristique se compose de 320 paramètres. La recherche des paramètres optimaux du classifieur SVM à savoir  $C$  et  $\lambda$

##### La recherche du paramètre $\lambda$

En utilisant la méthode exhaustive, on fixe le paramètre  $C$  (constante de pénalité) à 10 en ajustant le paramètre  $\lambda$  d'une manière expérimentale, le tableau 4.1 montre les résultats obtenus.

$\lambda$	1	1.5	2	3	4	5	6	<b>7.5</b>	8	10
$C$	10	10	10	10	10	10	10	10	10	10
Taux(%)	57.64	89.68	91.47	89.98	91.18	92.31	92.34	<b>92.37</b>	91.97	90.93

**Tableau 4.1** Recherche du paramètre  $\lambda$

Le meilleur résultat est obtenu pour  $\lambda = 7.5$

##### La recherche du paramètre $C$

Pour le tableau 4.2 nous avons fait varier  $C$  et on a fixé  $\lambda$

$\lambda$	7.5	7.5	7.5	7.5	7.5	7.5	7.5	7.5	7.5
$C$	1	10	12	<b>15</b>	18	20	100	500	1000
Taux(%)	89.93	92.37	92.42	<b>92.72</b>	92.52	92.37	88.09	81.76	79.97

**Tableau 4.2** Recherche du paramètre  $C$

Le meilleur résultat est obtenu pour  $C = 15$

Pour la méthode de Kirsch avec implémentation de l'image compressée en taille 8x8 le meilleur taux de reconnaissance est obtenu à 92.72%.

##### 4.4.1.2 Implémentation de l'image compressée en taille 4x4

Ici la compression est de séparer l'image initiale en blocs de taille (4x4), le principe consiste à calculer la moyenne de chaque bloc qui sera le pixel de la l'image compressée. En conséquence le vecteur de caractéristique se compose de 80 paramètres.

### La recherche du paramètre $C$

Ici, on a fixé  $\lambda$  et on fait varier le paramètre  $C$ , le tableau 4.3 montre les résultats obtenus.

$C$	12	15	18	20	30	<b>40</b>	50
$\lambda$	7.5	7.5	7.5	7.5	7.5	7.5	7.5
Taux (%)	90.98	91.13	91.43	91.47	91.62	<b>91.87</b>	91.67

**Tableau 4.3** Recherche du paramètre  $C$

Le meilleur résultat est obtenu pour  $C = 40$

### La recherche du paramètre $\lambda$

On fixe  $C$  trouvé dans le tableau 4.4 et on fait varier  $\lambda$ .

$C$	40	40	40	40	40	40
$\lambda$	5	7	<b>7.5</b>	10	15	20
Taux (%)	91.47	91.57	<b>91.87</b>	91.33	89.58	89.18

**Tableau 4.4** Recherche du paramètre  $\lambda$

Le meilleur résultat est obtenu pour  $\lambda = 7.5$

Pour la méthode de Kirsch avec implémentation de l'image compressée en taille 4x4 le meilleur taux de reconnaissance est obtenu à 91.87%.

Pour la méthode de Kirsch on remarque que plus l'image est compressée moins le résultat est bon et cela revient aux pertes d'information au niveau de la compression.

## 4.4.2 Méthode basée sur les filtres de Gabor

### 4.4.2.1 Implémentation de l'image filtrée

Ici nous avons filtré l'image originale de taille 16x16 par  $N$  filtres de Gabor, en suite nous avons transformé les images filtrées en un vecteur caractéristique de  $256 \times N$  paramètres c'est-à-dire tout les pixels des images filtrées, puis on a introduit ce vecteur dans le classifieur.

Les orientations des  $N$  filtres sont obtenus par la loi suivante  $\theta_k = (k - 1)\pi/N$ , avec  $k = 1, 2, \dots, N$

### La recherche du paramètre $N$

Pour le tableau 4.5 nous avons fait varier  $N$  et on a fixé les autres paramètres.

$N$	1	<b>2</b>	3
$\sigma_x$	2.5	2.5	2.5
$\sigma_y$	1	1	1
$f$	0.1	0.1	0.1
$\lambda$	10	10	10
$C$	10	10	10
Taux (%)	91.17	<b>91.18</b>	90.83

**Tableau 4.5** Recherche du paramètre  $N$

Nous avons obtenu le meilleur résultat pour  $N = 2$  (deux orientations  $0^\circ$  et  $90^\circ$ ).

#### La recherche du paramètre $f$

Pour le tableau 4.6 nous avons fait varier  $f$  et on a fixé les autres paramètres.

$N$	2	2	2	2	2	2
$\sigma_x$	2.5	2.5	2.5	2.5	2.5	2.5
$\sigma_y$	1	1	1	1	1	1
$f$	0.02	0.09	<b>0.1</b>	0.15	1	3
$\lambda$	10	10	10	10	10	10
$C$	10	10	10	10	10	10
Taux (%)	88.09	91.08	<b>91.18</b>	90.08	87.64	87.64

**Tableau 4.6** recherche du paramètre  $f$

Nous avons obtenu le meilleur résultat pour  $f = 0.1$

#### La recherche du paramètre $\sigma_x$

Pour le tableau 4.7 nous avons fait varier  $\sigma_x$  et on a fixé les autres paramètres.

$N$	2	2	2	2	2	2	2
$\sigma_x$	0.1	0.5	1	1.5	2	<b>2.4</b>	3
$\sigma_y$	1	1	1	1	1	1	1
$f$	0.1	0.1	0.1	0.1	0.1	0.1	0.1
$\lambda$	10	10	10	10	10	10	10
$C$	10	10	10	10	10	10	10
Taux (%)	85.79	90.83	90.73	90.18	90.52	<b>91.62</b>	90.88

**Tableau 4.7** Recherche du paramètre  $\sigma_x$

Nous avons obtenu le meilleur résultat pour  $\sigma_x = 2.4$

#### La recherche du paramètre $\sigma_y$

Pour le tableau 4.8 nous avons fait varier  $\sigma_y$  et on a fixé les autres paramètres.

$N$	2	2	2	2	2	2	2
$\sigma_x$	2.4	2.4	2.4	2.4	2.4	2.4	2.4
$\sigma_y$	0.1	0.5	0.9	<b>1</b>	1.1	1.5	2.4
$f$	0.1	0.1	0.1	0.1	0.1	0.1	0.1
$\lambda$	10	10	10	10	10	10	10
$C$	10	10	10	10	10	10	10
Taux (%)	90.93	90.73	91.38	<b>91.62</b>	91.52	90.53	89.28

**Tableau 4.8** Recherche du paramètre  $\sigma_y$

Nous avons obtenu le meilleur résultat pour  $\sigma_y = 1$ .

### La recherche du paramètre $\lambda$

Pour le tableau 4.9 nous avons fait varier  $\lambda$  et on a fixé les autres paramètres.

$N$	2	2	2	2	2	2	2	2	2
$\sigma_x$	2.4	2.4	2.4	2.4	2.4	2.4	2.4	2.4	2.4
$\sigma_y$	1	1	1	1	1	1	1	1	1
$f$	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
$\lambda$	11	15	16	17	20	24	<b>25</b>	26	30
$C$	10	10	10	10	10	10	10	10	10
Taux (%)	90.58	91.82	92.07	92.22	92.62	92.62	<b>92.92</b>	92.87	92.62

**Tableau 4.9** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 25$

### La recherche du paramètre $C$

Pour le tableau 4.10 nous avons fait varier  $C$  et on a fixé les autres paramètres.

$N$	2	2	2	2	2	2
$\sigma_x$	2.4	2.4	2.4	2.4	2.4	2.4
$\sigma_y$	1	1	1	1	1	1
$f$	0.1	0.1	0.1	0.1	0.1	0.1
$\lambda$	25	25	25	25	25	25
$C$	1	9	<b>10</b>	11	20	50
Taux (%)	91.87	92.87	<b>92.92</b>	92.67	92.17	89.83

**Tableau 4.10** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 10$

Pour la méthode de Gabor avec implémentation de l'image filtrée le meilleur taux de reconnaissance est obtenu à 92.92%.

#### 4.4.2.2 Implémentation de la moyenne et la variance

Ici nous avons filtré l'image originale de taille 16x16 par  $N$  filtres de Gabor, en suite on a calculé pour chaque image filtrée la moyenne et la variance qui seront prises comme paramètres dans le vecteur caractéristique de taille  $2xN$ .

##### La recherche du paramètre $N$

Pour le tableau 4.11 nous avons fait varier  $N$  et on a fixé les autres paramètres.

$N$	2	3	4	5	7	8	<b>12</b>
$\sigma_x$	2.4	2.4	2.4	2.4	2.4	2.4	2.4
$\sigma_y$	1	1	1	1	1	1	1
$f$	0.1	0.1	0.1	0.1	0.1	0.1	0.1
$\lambda$	10	10	10	10	10	10	10
$C$	10	10	10	10	10	10	10
Taux (%)	32.53	33.23	38.41	43.19	46.58	49.47	<b>55.85</b>

**Tableau 4.11** Recherche du paramètre  $N$

Nous avons obtenu le meilleur résultat pour  $N = 12$  (douze orientations  $0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ, 105^\circ, 120^\circ, 135^\circ, 150^\circ, 165^\circ$ ).

##### La recherche du paramètre $\sigma_x$

Pour le tableau 4.12 nous avons fait varier  $\sigma_x$  et on a fixé les autres paramètres.

$N$	12	12	12	12
$\sigma_x$	<b>0.1</b>	0.2	2	2.4
$\sigma_y$	1	1	1	1
$f$	0.1	0.1	0.1	0.1
$\lambda$	10	10	10	10
$C$	10	10	10	10
Taux (%)	<b>62.58</b>	56.95	52.31	55.85

**Tableau 4.12** Recherche du paramètre  $\sigma_x$

Nous avons obtenu le meilleur résultat pour  $\sigma_x = 0.1$

##### La recherche du paramètre $\sigma_y$

Pour le tableau 4.13 nous avons fait varier  $\sigma_y$  et on a fixé les autres paramètres.

$N$	12	12	12	12	12	12	12
$\sigma_x$	0.1	0.1	0.1	0.1	0.1	0.1	0.1
$\sigma_y$	0.1	0.9	1	1.1	<b>1.2</b>	1.25	1.5
$f$	0.1	0.1	0.1	0.1	0.1	0.1	0.1
$\lambda$	10	10	10	10	10	10	10
$C$	10	10	10	10	10	10	10
Taux (%)	36.17	62.43	62.58	62.73	<b>62.83</b>	62.03	58.64

**Tableau 4.13** Recherche du paramètre  $\sigma_y$

Nous avons obtenu le meilleur résultat pour  $\sigma_y = 1.2$

### La recherche du paramètre $f$

Pour le tableau 4.14 nous avons fait varier  $f$  et on a fixé les autres paramètres.

$N$	12	12	12	12	12
$\sigma_x$	0.1	0.1	0.1	0.1	0.1
$\sigma_y$	1.2	1.2	1.2	1.2	1.2
$f$	0.05	<b>0.1</b>	0.5	1	2
$\lambda$	10	10	10	10	10
$C$	10	10	10	10	10
Taux (%)	62.83	<b>62.83</b>	62.83	62.68	62.38

**Tableau 4.14** Recherche du paramètre  $f$

Nous avons obtenu le meilleur résultat pour  $f = 0.1$

### La recherche du paramètre $\lambda$

Pour le tableau 4.15 nous avons fait varier  $\lambda$  et on a fixé les autres paramètres.

$N$	12	12	12	12	12	12	12	12
$\sigma_x$	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
$\sigma_y$	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
$f$	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
$\lambda$	9	11	11.5	12	<b>12.1</b>	12.5	13	15
$C$	10	10	10	10	10	10	10	10
Taux (%)	62.03	62.86	63.02	63.12	<b>63.17</b>	62.92	62.97	62.03

**Tableau 4.15** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 12.1$

### La recherche du paramètre $C$

Pour le tableau 4.16 nous avons fait varier  $C$  et on a fixé les autres paramètres

$N$	12	12	12	12	12
$\sigma_x$	0.1	0.1	0.1	0.1	0.1
$\sigma_y$	1.2	1.2	1.2	1.2	1.2
$f$	0.1	0.1	0.1	0.1	0.1
$\lambda$	12.1	12.1	12.1	12.1	12.1
$C$	9	11	12	<b>13</b>	20
Taux (%)	63.12	63.17	63.22	<b>63.37</b>	63.22

**Tableau 4.16** recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 13$

Pour la méthode de Gabor avec implémentation de la moyenne et la variance le meilleur taux de reconnaissance est obtenu à 63.37%.

### 4.4.2.3 Implémentation de la moyenne et l'énergie

Ici nous avons filtré l'image originale de taille 16x16 par  $N$  filtres de Gabor, en suite on a calculé pour chaque image filtrée la moyenne et l'énergie qui seront prises comme paramètres dans le vecteur caractéristique de taille  $2 \times N$ .

#### La recherche du paramètre $f$

Pour le tableau 4.17 nous avons fait varier  $f$  et on a fixé les autres paramètres.

$N$	3	3	3	3	3	3
$\sigma_x$	0.1	0.1	0.1	0.1	0.1	0.1
$\sigma_y$	1.2	1.2	1.2	1.2	1.2	1.2
$f$	0.1	0.4	<b>0.5</b>	0.6	0.7	1
$\lambda$	10	10	10	10	10	10
$C$	10	10	10	10	10	10
Taux (%)	47.38	47.63	<b>47.68</b>	47.58	47.53	47.33

**Tableau 4.17** Recherche du paramètre  $f$

Nous avons obtenu le meilleur résultat pour  $f = 0.5$

#### La recherche du paramètre $\sigma_x$

Pour le tableau 4.18 nous avons fait varier  $\sigma_x$  et on a fixé les autres paramètres.

$N$	3	3	3	3	3	3
$\sigma_x$	0.2	0.4	<b>0.45</b>	0.5	0.6	1
$\sigma_y$	1.2	1.2	1.2	1.2	1.2	1.2
$f$	0.5	0.5	0.5	0.5	0.5	0.5
$\lambda$	10	10	10	10	10	10
$C$	10	10	10	10	10	10
Taux (%)	51.76	55.25	<b>57.24</b>	56.90	53.66	44.89

**Tableau 4.18** Recherche du paramètre  $\sigma_x$  Tapez une équation ici.

Nous avons obtenu le meilleur résultat pour  $\sigma_x = 0.45$

#### La recherche du paramètre $\sigma_y$

Pour le tableau 4.19 nous avons fait varier  $\sigma_y$  et on a fixé les autres paramètres.

$N$	3	3	3	3	3	3
$\sigma_x$	0.45	0.45	0.45	0.45	0.45	0.45
$\sigma_y$	0.45	1	1.1	1.15	<b>1.2</b>	1.25
$f$	0.5	0.5	0.5	0.5	0.5	0.5
$\lambda$	10	10	10	10	10	10
$C$	10	10	10	10	10	10
Taux (%)	35.02	56.30	56.15	56.50	<b>57.24</b>	56.85

**Tableau 4.19** Recherche du paramètre  $\sigma_y$

Nous avons obtenus le meilleur résultat pour  $\sigma_y = 1.2$

### La recherche du paramètre $N$

Pour le tableau 4.20 nous avons fait varier  $N$  et on a fixé les autres paramètres.

$N$	2	3	4	6	8	<b>12</b>
$\sigma_x$	0.45	0.45	0.45	0.45	0.45	0.45
$\sigma_y$	1.2	1.2	1.2	1.2	1.2	1.2
$f$	0.5	0.5	0.5	0.5	0.5	0.5
$\lambda$	10	10	10	10	10	10
$C$	10	10	10	10	10	10
Taux (%)	47.63	57.24	58.54	60.58	60.73	<b>61.08</b>

**Tableau 4.20** Recherche du paramètre  $N$

Nous avons obtenu le meilleur résultat pour  $N = 12$  (douze orientations  $0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ, 105^\circ, 120^\circ, 135^\circ, 150^\circ, 165^\circ$ ).

### La recherche du paramètre $\lambda$

Pour le tableau 4.21 nous avons fait varier  $\lambda$  et on a fixé les autres paramètres.

$N$	12	12	12	12	12
$\sigma_x$	0.45	0.45	0.45	0.45	0.45
$\sigma_y$	1.2	1.2	1.2	1.2	1.2
$f$	0.5	0.5	0.5	0.5	0.5
$\lambda$	20	30	<b>35</b>	40	50
$C$	10	10	10	10	10
Taux (%)	69.30	70.95	<b>71.30</b>	71.30	70.15

**Tableau 4.21** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 35$

### La recherche du paramètre $C$

Pour le tableau 4.22 nous avons fait varier  $C$  et on a fixé les autres paramètres.

$N$	12	12	12	12	12	12	12	12
$\sigma_x$	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
$\sigma_y$	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
$f$	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
$\lambda$	35	35	35	35	35	35	35	35
$C$	9	11	15	20	25	<b>26</b>	27	30
Taux (%)	71.10	71.49	71.79	71.99	72.39	<b>72.24</b>	72.14	71.84

**Tableau 4.22** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 26$

Pour la méthode de Gabor avec implémentation la moyenne et l'énergie le meilleur taux de reconnaissance est obtenu à 72.24%.

Le meilleur taux de reconnaissance est obtenu avec l'image filtrée à 92.92% par contre les paramètres moyenne-variance et moyenne-énergie ne sont pas des paramètres pertinentes qui caractérisent mieux les chiffres.

#### 4.4.3 Méthode basée sur les moments de Zernike

La reconnaissance de chiffres manuscrits en utilisant les moments de Zernike est présentée comme suit : d'abord on calcule les bases orthogonales de Zernike jusqu'à l'ordre  $n$  citée dans le chapitre 2, ensuite on calcule les  $n$  moments qui sont des nombres complexes. Le vecteur de caractéristique est construit en utilisant deux méthodes : images en niveau de gris et images binarisées avec amincissement.

##### 4.4.3.1 Image en niveau de gris

###### 4.4.3.1.1 Implémentation du module

Les moments de Zernike sont calculés à partir des images en niveau de gris, le vecteur des paramètres est choisi à ce qu'il comporte les modules de ces  $n$  moments, qui est implémenté directement dans le classifieur.

##### La recherche du paramètre $n$

Pour le tableau 4.23 nous avons fait varier  $n$  et on a fixé les autres paramètres.

$\lambda$	7.5	7.5	7.5	7.5	7.5	7.5	7.5	7.5
$C$	15	15	15	15	15	15	15	15
$n$	12	10	8	6	<b>5</b>	4	3	1
Taux(%)	29.04	17.88	37.47	52.91	<b>64.27</b>	63.47	56.10	38.86

Tableau 4.23 Recherche du paramètre  $n$

Nous avons obtenu le meilleur résultat pour  $n = 5$

##### La recherche du paramètre $C$

Pour le tableau 4.24 nous avons fait varier  $C$  et on a fixé les autres paramètres.

$\lambda$	10	10	10	10	10	10	10	10	10	10
$C$	0	<b>1</b>	2	3	4	5	6	7	8	10
$n$	5	5	5	5	5	5	5	5	5	5
Taux (%)	08.81	<b>73.59</b>	73.19	72.34	71.44	70.80	70.70	70.50	70.28	69.95

Tableau 4.24 Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 1$

#### La recherche du paramètre $\lambda$

Pour le tableau 4.25 nous avons fait varier  $\lambda$  et on a fixé les autres paramètres.

$\lambda$	0.1	0.5	8	9.5	10.5	11	12	<b>15</b>	17	20
$C$	1	1	1	1	1	1	1	1	1	1
$n$	5	5	5	5	5	5	5	5	5	5
Taux (%)	17.88	17.93	67.11	72.39	74.33	75.28	75.93	<b>76.68</b>	76.67	75.98

**Tableau 4.25** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 15$

Pour l'image en niveau de gris avec le module des moments comme paramètre est obtenu avec un taux de 76.68%.

#### 4.4.3.1.2 Implémentation du module et phase

Les moments de Zernike sont calculés à partir des images en niveau de gris, le vecteur paramètre est choisi à ce qu'il comporte les modules et les phases de ces  $n$  moments, qui est implémenté directement dans le classifieur.

#### La recherche du paramètre $n$

Pour le tableau 4.26 nous avons fait varier  $n$  et on a fixé les autres paramètres.

$\lambda$	10	10	10	10	10	10
$C$	5	5	5	5	5	5
$n$	1	2	3	4	<b>5</b>	6
Taux (%)	49.27	66.51	72.34	73.74	<b>75.53</b>	71.10

**Tableau 4.26** Recherche du paramètre  $n$

Nous avons obtenu le meilleur résultat pour  $n = 5$

#### La recherche du paramètre $\lambda$

Pour le tableau 4.27 nous avons fait varier  $\lambda$  et on a fixé les autres paramètres.

$\lambda$	0.1	0.5	9	12	15	17	<b>18</b>	19	20
$C$	1	1	1	1	1	1	1	1	1
$n$	5	5	5	5	5	5	5	5	5
Taux (%)	17.88	17.93	77.29	79.47	80.86	80.91	<b>81.06</b>	80.96	80.96

**Tableau 4.27** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 18$

### La recherche du paramètre $C$

Pour le tableau 4.28 nous avons fait varier  $C$  et on a fixé les autres paramètres.

$\lambda$	18	18	18	18	18	18	18	18	18
$C$	<b>2</b>	3	4	5	6	7	8	10	12
$n$	5	5	5	5	5	5	5	5	5
Taux (%)	<b>82.01</b>	81.01	80.41	80.01	79.62	17.88	79.12	78.27	78.02

**Tableau 4.28** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 2$

Pour l'image en niveau de gris avec le module et la phase des moments comme paramètre, le meilleur taux est obtenu avec un taux de 82.01%.

Pour la méthode de Zernike le meilleur taux de reconnaissance avec l'image en niveau de gris est obtenu en implémentant le module et la phase des moments (82.01%), parce qu'on peut avoir des même modules mais de différentes phases c'est pour cela que l'ajout de la phase comme paramètre a fait augmenter le taux de reconnaissance.

#### 4.4.3.2 Image binarisée et amincie

En utilisant la méthode d'Otsu pour binariser les images initiales, puis procéder à l'utilisation de l'amincissement.

##### 4.4.3.2.1 Implémentation du module

Les moments de Zernike sont calculés à partir des images binarisées et amincies, le vecteur des paramètres est choisi à ce qu'il comporte les modules de ces  $n$  moments. Les modules des  $n$  moments qui composent le vecteur caractéristique qui est implémenté directement dans le classifieur.

### La recherche du paramètre $C$

Pour le tableau 4.29 nous avons fait varier  $C$  et on a fixé les autres paramètres.

$\lambda$	7.5	7.5	7.5	7.5	7.5	7.5
$C$	15	10	<b>5</b>	2	4	6
$n$	5	5	5	5	5	5
Taux (%)	73.54	73.84	<b>73.94</b>	73.59	73.89	73.59

**Tableau 4.29** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 5$

### La recherche du paramètre $\lambda$

Pour le tableau 4.30 nous avons fait varier  $\lambda$  et on a fixé les autres paramètres.

$\lambda$	1	5	6	7	8	<b>6.5</b>	6.6
$C$	5	5	5	5	5	5	5
$n$	5	5	5	5	5	5	5
Taux (%)	33.48	72.44	74.14	73.99	73.69	<b>74.58</b>	74.38

Tableau 4.30 Recherche du paramètre  $\lambda$ 

Nous avons obtenu le meilleur résultat pour  $\lambda = 6.5$

#### La recherche du paramètre $n$

Pour le tableau 4.31 nous avons fait varier  $n$  et on a fixé les autres paramètres.

$\lambda$	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5
$C$	5	5	5	5	5	5	5	5
$n$	0	2	<b>5</b>	8	10	12	14	4
Taux (%)	34.37	57.24	<b>74.58</b>	72.89	72.14	65.02	49.37	70.35

Tableau 4.31 Recherche du paramètre  $n$ 

Nous avons obtenu le meilleur résultat pour  $n = 6$

Pour l'image binarisée et amincie avec le module des moments comme paramètre, le meilleur taux est obtenu avec un taux de 74.58%.

#### 4.4.3.2.2 Implémentation du module et phase

Les moments de Zernike sont calculés à partir des images binarisées et amincies, le vecteur paramètre est choisi à ce qu'il comporte les modules et les phases de ces  $n$  moments. Les modules et phases des  $n$  moments qui composent le vecteur caractéristique qui est implémenté directement dans le classifieur.

#### La recherche du paramètre $\lambda$

Pour le tableau 4.32 nous avons fait varier  $\lambda$  et on a fixé les autres paramètres.

$\lambda$	7	<b>10</b>	12	15	11	9	9.5	10.5	10.1	9.9
$C$	5	5	5	5	5	5	5	5	5	5
$n$	5	5	5	5	5	5	5	5	5	5
Taux (%)	82.16	<b>83.35</b>	82.86	81.21	83.25	83.30	83.10	83.25	83.25	83.30

Tableau 4.32 Recherche du paramètre  $\lambda$ 

Nous avons obtenu le meilleur résultat pour  $\lambda = 10$

#### La recherche du paramètre $C$

Pour le tableau 4.33 nous avons fait varier  $C$  et on a fixé les autres paramètres.

$\lambda$	10	10	10	10	10	10	10	10	10	10
$C$	1	2	3	<b>6</b>	8	10	12	15	4	7
$n$	5	5	5	5	5	5	5	5	5	5
Taux (%)	81.06	81.61	83.00	<b>83.10</b>	81.41	80.96	80.41	80.01	83.02	82.90

**Tableau 4.33** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 6$

### La recherche du paramètre $n$

Pour le tableau 4.34 nous avons fait varier  $n$  et on a fixé les autres paramètres.

$\lambda$	10	10	10	10	10	10	10	10	10
$C$	6	6	6	6	6	6	6	6	6
$n$	0	1	2	3	4	<b>6</b>	7	8	
Taux (%)	31.78	45.83	68.7	76.83	81.66	<b>83.55</b>	81.81	82.46	

**Tableau 4.34** Recherche du paramètre  $n$

Nous avons obtenu le meilleur résultat pour  $n = 6$

Pour l'image binarisée et amincie avec le module et la phase des moments comme paramètre, le meilleur taux est obtenu avec un taux de 83.55%.

Pour la méthode de Zernike le meilleur taux de reconnaissance avec l'image binarisée et amincie est obtenu en implémentant le module et la phase des moments (83.55%).

Pour la méthode de Zernike le meilleur taux de reconnaissance est obtenu avec l'image binarisée et amincie, le module et phase comme paramètre. Cela est justifié par les prétraitements qu'on a appliqué à l'image initiale.

## 4.4.4 Méthode basée sur la transformée en ondelettes

La décomposition des images initiales de taille 16x16 par application de la transformée en ondelettes a donné en sortie 4 cartes (1 carte basse fréquence et 3 cartes de haute fréquence). Dans notre travail, on s'est intéressé à la carte basse fréquence (l'image approximée) de la première, deuxième et troisième résolution. Ainsi le vecteur caractéristique est composé des coefficients des approximées. Nous présentons les résultats de l'application de trois types d'ondelettes.

### 4.4.4.1 Ondelette de Daubechies

#### 4.4.4.1.1 Haar (db1)

L'ondelette de Haar (Daubechies) est appliquée sur les images initiales.

- **1<sup>ère</sup> résolution**

Ici, on s'intéresse à l'image approximée de la première résolution. Ainsi le vecteur de caractéristiques est composé de 8x8 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.35 montre les résultats obtenus.

$\lambda$	5	9.5	9.9	10	<b>10.1</b>	10.2	20	30
$C$	10	10	10	10	10	10	10	10
Taux (%)	90.53	91.62	91.82	91.97	<b>92.07</b>	91.92	91.47	89.38

**Tableau 4.35** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 10.1$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.36 montre les résultats obtenus.

$\lambda$	10.1	10.1	10.1	10.1	10.1
$C$	8	<b>9</b>	11	15	20
Taux (%)	92.02	<b>92.12</b>	91.62	90.83	90.23

**Tableau 4.36** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 9$

- **2<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la deuxième résolution. Ainsi le vecteur de caractéristiques est composé de 4x4 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.37 montre les résultats obtenus.

$\lambda$	1	5	6	6.5	<b>6.6</b>	7	8	20
$C$	8	8	8	8	8	8	8	8
Taux (%)	47.88	86.69	87.39	87.54	<b>87.64</b>	87.34	86.54	81.11

**Tableau 4.37** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 6.6$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.38 montre les résultats obtenus.

$\lambda$	6.6	6.6	6.6	6.6	6.6
$C$	7	<b>8</b>	9	10	20
Taux (%)	87.49	<b>87.64</b>	87.29	87.19	86.04

**Tableau 4.38** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 8$

- **3<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la troisième résolution. Ainsi le vecteur de caractéristiques est composé de 2x2 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.39 montre les résultats obtenus.

$\lambda$	1	<b>1.9</b>	3	4	5	10	15
$C$	10	10	10	10	10	10	10
Taux (%)	63.67	<b>68.60</b>	67.66	67.51	66.11	62.83	61.18

**Tableau 4.39** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 1.9$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.40 montre les résultats obtenus.

$\lambda$	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.9
$C$	2	3	<b>4</b>	5	6	7	9	11
Taux (%)	68.31	68.70	<b>68.80</b>	68.65	68.56	68.41	68.75	68.56

**Tableau 4.40** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 4$

On remarque qu'à chaque fois on augmente la résolution le taux de reconnaissance chute, le meilleur résultat est obtenu avec la première résolution ce qui explique la perte d'information par filtrage passe bas puisqu'on s'est intéressé à l'image approximée.

L'ondelette de bd1 (Haar) a donné un taux de 92.12%.

#### 4.4.4.1.2 Daubechies (db2)

L'ondelette de Daubechies (db2) est appliquée sur les images initiales.

- **1<sup>ère</sup> résolution**

Ici, on s'intéresse à l'image approximée de la première résolution. Ainsi le vecteur de caractéristiques est composé de 8x8 valeurs.

**La recherche du paramètre  $\lambda$** 

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.41 montre les résultats obtenus.

$\lambda$	10	13	<b>14</b>	15	16
$C$	10	10	10	10	10
Taux (%)	91.47	91.97	<b>92.02</b>	91.92	91.72

**Tableau 4.41** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 14$

**La recherche du paramètre  $C$** 

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.42 montre les résultats obtenus.

$\lambda$	14	14	14	14
$C$	<b>5</b>	9	11	15
Taux (%)	<b>92.07</b>	92.02	91.97	91.43

**Tableau 4.42** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 5$

- **2<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la deuxième résolution. Ainsi le vecteur de caractéristiques est composé de 4x4 valeurs.

**La recherche du paramètre  $\lambda$** 

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.43 montre les résultats obtenus.

$\lambda$	5	10	12.5	<b>15</b>	20
$C$	10	10	10	10	10
Taux (%)	88.68	90.28	90.53	<b>90.58</b>	90.18

**Tableau 4.43** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 15$

**La recherche du paramètre  $C$** 

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.44 montre les résultats obtenus.

$\lambda$	15	15	15	15	15
$C$	5	9	<b>10</b>	11	15
Taux (%)	90.38	90.03	<b>90.58</b>	90.53	90.13

**Tableau 4.44** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 10$

- **3<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la troisième résolution. Ainsi le vecteur de caractéristiques est composé de 2x2 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.45 montre les résultats obtenus.

$\lambda$	5	8	<b>9</b>	10	15
$C$	10	10	10	10	10
Taux (%)	81.66	82.11	<b>82.16</b>	82.11	80.46

**Tableau 4.45** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 9$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.46 montre les résultats obtenus.

$\lambda$	9	9	9	9	9
$C$	5	9	11	12	<b>15</b>
Taux (%)	81.56	82.06	82.26	82.21	<b>82.31</b>

**Tableau 4.46** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 15$

On remarque qu'à chaque fois on augmente la résolution le taux de reconnaissance chute, le meilleur résultat est obtenu avec la première résolution ce qui explique la perte d'information par filtrage passe bas puisqu'on s'est intéressé à l'image approximée.

L'ondelette de bd2 a donné un taux de 92.07%.

#### 4.4.4.1.3 Daubechie (db3)

L'ondelette de Daubechies (db3) est appliquée sur les images initiales.

- **1<sup>ère</sup> résolution**

Ici, on s'intéresse à l'image approximée de la première résolution. Ainsi le vecteur de caractéristiques est composé de 8x8 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.47 montre les résultats obtenus.

$\lambda$	5	9	10	12.5	<b>13</b>	14	15	20
$C$	10	10	10	10	10	10	10	10
Taux (%)	89.88	90.73	91.28	91.72	<b>91.82</b>	91.77	91.72	91.67

Tableau 4.47 Recherche du paramètre  $\lambda$ 

Nous avons obtenu le meilleur résultat pour  $\lambda = 13$

### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.48 montre les résultats obtenus.

$\lambda$	13	13	13	13	13	13	13
$C$	1	<b>5</b>	6	9	11	20	100
Taux (%)	91.33	<b>92.52</b>	92.42	91.92	91.77	90.08	84.25

Tableau 4.48 Recherche du paramètre  $C$ 

Nous avons obtenu le meilleur résultat pour  $C = 5$

- **2<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la deuxième résolution. Ainsi le vecteur de caractéristiques est composé de 4x4 valeurs.

### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.49 montre les résultats obtenus.

$\lambda$	5	10	<b>10.9</b>	11	12	15	20
$C$	10	10	10	10	10	10	10
Taux (%)	86.74	91.43	<b>91.87</b>	91.82	91.57	91.18	90.23

Tableau 4.49 Recherche du paramètre  $\lambda$ 

Nous avons obtenu le meilleur résultat pour  $\lambda = 10.9$

### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.50 montre les résultats obtenus.

$\lambda$	10.9	10.9	10.9	10.9	10.9	10.9	10.9	10.9
$C$	3	<b>4</b>	5	6	9	10	11	20
Taux (%)	91.82	<b>91.92</b>	91.92	91.72	91.82	91.87	91.57	89.93

Tableau 4.50 Recherche du paramètre  $C$ 

Nous avons obtenu le meilleur résultat pour  $C = 4$

- **3<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la troisième résolution. Ainsi le vecteur de caractéristiques est composé de 2x2 valeurs.

#### La recherche du paramètre $\lambda$

La valeur de  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.51 montre les résultats obtenus.

$\lambda$	5	10	11	<b>12</b>	13	15	20
$C$	10	10	10	10	10	10	10
Taux (%)	84.00	87.74	87.74	<b>88.29</b>	88.09	87.49	86.69

**Tableau 4.51** Recherche du paramètre  $\lambda$

Le Nous avons obtenu le meilleur résultat pour  $\lambda = 12$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.52 montre les résultats obtenus.

$\lambda$	12	12	12	12	12	12	12
$C$	3	5	9	<b>10</b>	11	15	20
Taux (%)	86.64	87.19	88.14	<b>88.29</b>	88.24	87.84	87.69

**Tableau 4.52** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 10$

On remarque qu'à chaque fois on augmente la résolution le taux de reconnaissance chute, le meilleur résultat est obtenu avec la première résolution ce qui explique la perte d'information par filtrage passe bas puisqu'on s'est intéressé à l'image approximée.

L'ondelette de db3 a donné un taux de 92.52%.

#### 4.4.4.1.4 Daubechie (db4)

L'ondelette de Daubechies (db4) est appliquée sur les images initiales.

- **1<sup>ère</sup> résolution**

Ici, on s'intéresse à l'image approximée de la première résolution. Ainsi le vecteur de caractéristiques est composé de 8x8 valeurs.

#### La recherche du paramètre $\lambda$

La valeur de  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.53 montre les résultats obtenus.

$\lambda$	5	10	15	<b>16</b>	17	20	23	25
$C$	10	10	10	10	10	10	10	10
Taux (%)	88.14	90.53	91.72	<b>91.77</b>	91.52	91.62	91.62	90.88

**Tableau 4.53** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 16$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.54 montre les résultats obtenus.

$\lambda$	16	16	16	16	16	16	16	16
$C$	3	4	<b>5</b>	6	9	11	15	20
Taux (%)	91.77	91.72	<b>92.32</b>	92.12	91.72	91.72	91.52	90.88

**Tableau 4.54** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 5$

- **2<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la deuxième résolution. Ainsi le vecteur de caractéristiques est composé de 4x4 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.55 montre les résultats obtenus.

$\lambda$	5	10	13	<b>15</b>	17	20
$C$	10	10	10	10	10	10
Taux (%)	84.50	90.03	90.93	<b>91.13</b>	90.78	90.68

**Tableau 4.55** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 15$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.56 montre les résultats obtenus.

$\lambda$	15	15	15	15	15	15	15	15
$C$	3	4	5	8	<b>9</b>	11	15	20
Taux (%)	91.131	91.03	90.98	91.131	<b>91.28</b>	91.131	90.43	90.43

**Tableau 4.56** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 9$

- **3<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la troisième résolution. Ainsi le vecteur de caractéristiques est composé de 2x2 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.57 montre les résultats obtenus.

$\lambda$	5	10	13	14	<b>14.5</b>	15	20
$C$	10	10	10	10	10	10	10
Taux (%)	79.72	85.7	86.59	87.39	<b>87.54</b>	87.34	86.74

**Tableau 4.57** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 14.5$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.58 montre les résultats obtenus.

$\lambda$	14.5	14.5	14.5	14.5	14.5	14.5	14.5
$C$	3	5	9	<b>10</b>	11	15	20
Taux (%)	85.13	85.7	87.29	<b>87.54</b>	87.4439	87.4439	87.1948

**Tableau 4.58** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 10$

On remarque qu'à chaque fois on augmente la résolution le taux de reconnaissance chute, le meilleur résultat est obtenu avec la première résolution ce qui explique la perte d'information par filtrage passe bas puisqu'on s'est intéressé à l'image approximée.

L'ondelette de bd4 a donné un taux de 92.32%.

Pour les ondelettes de Daubechies on a obtenu le meilleur résultat de classification 92.52% avec l'ondelette db3.

#### 4.4.4.2 Ondelettes de Symmlets

##### 4.4.4.2.1 Symmlets (sym2)

L'ondelette de Symlets (sym2) est appliquée sur les images initiales.

- **1<sup>ère</sup> résolution**

Ici, on s'intéresse à l'image approximée de la première résolution. Ainsi le vecteur de caractéristiques est composé de 8x8 valeurs.

**La recherche du paramètre  $\lambda$** 

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.59 montre les résultats obtenus.

$\lambda$	5	10	<b>14</b>	15	16	17	20
$C$	10	10	10	10	10	10	10
Taux (%)	90.68	91.47	<b>92.02</b>	91.92	91.72	91.87	91.57

**Tableau 4.59** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 14$

**La recherche du paramètre  $C$** 

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.60 montre les résultats obtenus.

$\lambda$	14	14	14	14	14	14	14
$C$	2	<b>3</b>	4	5	9	11	15
Taux (%)	91.92	<b>92.27</b>	92.17	92.07	92.02	91.97	91.43

**Tableau 4.60** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 3$

- **2<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la deuxième résolution. Ainsi le vecteur de caractéristiques est composé de 4x4 valeurs.

**La recherche du paramètre  $\lambda$** 

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.61 montre les résultats obtenus.

$\lambda$	5	10	14	14.5	<b>14.9</b>	15	16
$C$	10	10	10	10	10	10	10
Taux (%)	88.68	90.28	90.39	90.53	<b>90.63</b>	90.58	90.33

**Tableau 4.61** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 14.9$

**La recherche du paramètre  $C$** 

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.62 montre les résultats obtenus.

$\lambda$	14.9	14.9	14.9	14.9	14.9	14.9	14.9
$C$	2	3	5	8	<b>9</b>	11	20
Taux (%)	89.68	90.38	90.33	90.58	<b>90.68</b>	90.53	89.98

**Tableau 4.62** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 9$

- **3<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la troisième résolution. Ainsi le vecteur de caractéristiques est composé de 2x2 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.63 montre les résultats obtenus.

$\lambda$	5	<b>9</b>	10	11	15
$C$	10	10	10	10	10
Taux (%)	81.66	<b>82.16</b>	82.11	81.96	80.46

**Tableau 4.63** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 9$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.64 montre les résultats obtenus.

$\lambda$	9	9	9	9	9	9
$C$	5	9	<b>11</b>	12	20	100
Taux (%)	81.56	82.06	<b>82.26</b>	82.21	81.56	81.71

**Tableau 4.64** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 11$

On remarque qu'à chaque fois on augmente la résolution le taux de reconnaissance chute, le meilleur résultat est obtenu avec la première résolution ce qui explique la perte d'information par filtrage passe bas puisqu'on s'est intéressé à l'image approximée.

L'ondelette de sym2 a donné un taux de 92.27%.

#### 4.4.4.2.2 Symmlets (sym3)

L'ondelette de Symlets (sym3) est appliquée sur les images initiales.

- **1<sup>ère</sup> résolution**

Ici, on s'intéresse à l'image approximée de la première résolution. Ainsi le vecteur de caractéristiques est composé de 8x8 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.65 montre les résultats obtenus.

$\lambda$	4	5	10	12	<b>13</b>	14	15	20
$C$	10	10	10	10	10	10	10	10
Taux (%)	85.89	89.88	91.28	91.62	<b>91.82</b>	91.77	91.72	91.67

**Tableau 4.65** Recherche du paramètre  $\lambda$ 

Nous avons obtenu le meilleur résultat pour  $\lambda = 13$

**La recherche du paramètre  $C$** 

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.66 montre les résultats obtenus.

$\lambda$	13	13	13	13	13	13	13	13	13
$C$	2	3	4	<b>5</b>	8	9	11	15	20
Taux (%)	91.92	92.12	91.52	<b>92.52</b>	91.87	91.92	91.77	91.23	90.08

**Tableau 4.66** Recherche du paramètre  $C$ 

Nous avons obtenu le meilleur résultat pour  $C = 5$

- **2<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la deuxième résolution. Ainsi le vecteur de caractéristiques est composé de 4x4 valeurs.

**La recherche du paramètre  $\lambda$** 

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.67 montre les résultats obtenus.

$\lambda$	5	9	10	<b>11</b>	12	15	20
$C$	10	10	10	10	10	10	10
Taux (%)	86.74	91.08	91.43	<b>91.82</b>	91.57	91.18	90.23

**Tableau 4.67** Recherche du paramètre  $\lambda$ 

Nous avons obtenu le meilleur résultat pour  $\lambda = 11$

**La recherche du paramètre  $C$** 

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.68 montre les résultats obtenus.

$\lambda$	11	11	11	11	11	11	11	11
$C$	3	4	5	7	<b>8</b>	9	11	15
Taux (%)	91.82	91.92	91.77	91.62	<b>92.02</b>	91.82	91.57	91.13

**Tableau 4.68** Recherche du paramètre  $C$ 

Nous avons obtenu le meilleur résultat pour  $C = 8$

- **3<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la troisième résolution. Ainsi le vecteur de caractéristiques est composé de 2x2 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.69 montre les résultats obtenus.

$\lambda$	5	9	10	11	<b>12</b>	13	15
$C$	10	10	10	10	10	10	10
Taux (%)	84	87.39	87.74	87.74	<b>88.29</b>	88.09	87.49

**Tableau 4.69** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 12$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.70 montre les résultats obtenus.

$\lambda$	12	12	12	12	12	12	12
$C$	3	5	9	<b>10</b>	11	12	15
Taux (%)	86.34	87.19	88.14	<b>88.29</b>	88.24	88.19	87.84

**Tableau 4.70** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 10$

On remarque qu'à chaque fois on augmente la résolution le taux de reconnaissance chute, le meilleur résultat est obtenu avec la première résolution ce qui explique la perte d'information par filtrage passe bas puisqu'on s'est intéressé à l'image approximée.

L'ondelette de sym3 a donné un taux de 92.52%.

#### 4.4.4.2.3 Symmlets (Sym4)

L'ondelette de Symlets (sym4) est appliquée sur les images initiales.

- **1<sup>ère</sup> résolution**

Ici, on s'intéresse à l'image approximée de la première résolution. Ainsi le vecteur de caractéristiques est composé de 8x8 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.71 montre les résultats obtenus.

$\lambda$	5	10	12	<b>13</b>	14	15	16	17
$C$	10	10	10	10	10	10	10	10
Taux (%)	89.23	90.58	91.82	<b>91.87</b>	91.72	91.52	91.77	91.62

**Tableau 4.71** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 13$

### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.72 montre les résultats obtenus.

$\lambda$	13	13	13	13	13	13	13	13
$C$	2	<b>3</b>	4	5	6	9	15	20
Taux (%)	93.02	<b>93.07</b>	92.52	92.52	92.22	91.82	90.48	89.53

**Tableau 4.72** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 3$

- **2<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la deuxième résolution. Ainsi le vecteur de caractéristiques est composé de 4x4 valeurs.

### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.73 montre les résultats obtenus.

$\lambda$	5	9	10	15	19	<b>20</b>	21
$C$	10	10	10	10	10	10	10
Taux (%)	85.35	89.48	90.23	90.98	90.98	<b>91.03</b>	90.93

**Tableau 4.73** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 20$

### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.74 montre les résultats obtenus.

$\lambda$	20	20	20	20	20	20	20	20
$C$	5	8	<b>9</b>	10	11	15	20	25
Taux (%)	90.88	91.03	<b>91.08</b>	91.0314	90.9816	90.48	90.63	90.43

**Tableau 4.74** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 9$

- **3<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la troisième résolution. Ainsi le vecteur de caractéristiques est composé de 2x2 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du C est fixée et on fait varier  $\lambda$ , le tableau 4.75 montre les résultats obtenus.

$\lambda$	5	10	15	20	24	<b>25</b>	26	30
C	10	10	10	10	10	10	10	10
Taux (%)	77.97	87.34	88.58	88.68	88.69	<b>88.73</b>	88.63	88.04

**Tableau 4.75** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 25$

#### La recherche du paramètre C

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier C, le tableau 4.76 montre les résultats obtenus.

$\lambda$	25	25	25	25	25	25	25
C	3	5	9	10	11	<b>15</b>	20
Taux (%)	86.89	87.842	88.54	88.73	88.68	<b>88.93</b>	88.73

**Tableau 4.76** Recherche du paramètre C

Nous avons obtenu le meilleur résultat pour C = 15

On remarque qu'à chaque fois on augmente la résolution le taux de reconnaissance chute, le meilleur résultat est obtenu avec la première résolution ce qui explique la perte d'information par filtrage passe bas puisqu'on s'est intéressé à l'image approximée.

L'ondelette de sym4 a donné un taux de 93.07%.

Pour les ondelettes de Symmlets on a obtenu le meilleur résultat de classification 93.07% avec l'ondelette sym4.

#### 4.4.4.3 Ondelettes de Coiflets

##### 4.4.4.3.1 Coiflets (coif1)

L'ondelette de Coiflets (coif1) est appliquée sur les images initiales.

- **1<sup>ère</sup> résolution**

Ici, on s'intéresse à l'image approximée de la première résolution. Ainsi le vecteur de caractéristiques est composé de 8x8 valeurs.

**La recherche du paramètre  $\lambda$** 

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.77 montre les résultats obtenus.

$\lambda$	1	5	10	12	14	15	<b>16</b>	20
$C$	10	10	10	10	10	10	10	10
Taux (%)	28.52	90.38	91.03	91.52	91.82	91.97	<b>92.17</b>	91.52

**Tableau 4.77** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 16$

**La recherche du paramètre  $C$** 

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.78 montre les résultats obtenus.

$\lambda$	16	16	16	16	16	16	16	16	16
$C$	1	2	4	<b>5</b>	6	8	9	15	20
Taux (%)	90.08	91.38	92.12	<b>92.27</b>	92.17	92.12	92.07	91.33	91.13

**Tableau 4.78** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 5$

- **2<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la deuxième résolution. Ainsi le vecteur de caractéristiques est composé de 4x4 valeurs.

**La recherche du paramètre  $\lambda$** 

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.79 montre les résultats obtenus.

$\lambda$	0.1	1	10	<b>12</b>	13	15.5	16	17
$C$	10	10	10	10	10	10	10	10
Taux (%)	17.93	32.83	90.13	<b>90.38</b>	89.98	89.78	89.78	89.83

**Tableau 4.79** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 12$

**La recherche du paramètre  $C$** 

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.80 montre les résultats obtenus.

$\lambda$	12	12	12	12	12	12	12	12	12	12
$C$	1	2	<b>4</b>	5	6	7	8	9	15	20
Taux (%)	88.29	89.38	<b>90.23</b>	90.13	90.18	90.21	90.03	90.20	89.88	89.68

**Tableau 4.80** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 4$

- **3<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la troisième résolution. Ainsi le vecteur de caractéristiques est composé de 2x2 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du C est fixée et on fait varier  $\lambda$ , le tableau 4.81 montre les résultats obtenus.

$\lambda$	5	10	12	12.25	<b>12.5</b>	12.75	13	15	20
$C$	10	10	10	10	10	10	10	10	10
Taux (%)	83	87.19	87.29	87.29	<b>87.39</b>	87.14	87.19	86.89	86.09

**Tableau 4.81** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 12.5$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.82 montre les résultats obtenus.

$\lambda$	12	12	12	12	12	12	12	12	12
$C$	2	3	8	9	11	12	<b>15</b>	17	20
Taux (%)	84.95	85.84	86.94	87.19	87.34	87.19	<b>87.44</b>	87.14	86.99

**Tableau 4.82** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 15$

On remarque qu'à chaque fois on augmente la résolution le taux de reconnaissance chute, le meilleur résultat est obtenu avec la première résolution ce qui explique la perte d'information par filtrage passe bas puisqu'on s'est intéressé à l'image approximée.

L'ondelette de coif1 a donné un taux de 92.27%.

#### 4.4.4.3.2 Coiflets (coif2)

L'ondelette de Coiflets (coif2) est appliquée sur les images initiales.

- **1<sup>ère</sup> résolution**

Ici, on s'intéresse à l'image approximée de la troisième résolution. Ainsi le vecteur de caractéristiques est composé de 8x8 valeurs.

**La recherche du paramètre  $\lambda$** 

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.83 montre les résultats obtenus.

$\lambda$	10	15	16	17	20	<b>21</b>	25	28
$C$	10	10	10	10	10	10	10	10
Taux (%)	90.88	91.82	91.62	91.72	91.92	<b>92.12</b>	91.82	91.28

**Tableau 4.83** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 21$

**La recherche du paramètre  $C$** 

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.84 montre les résultats obtenus.

$\lambda$	21	21	21	21	21
$C$	5	<b>9</b>	11	15	20
Taux (%)	92.17	<b>92.22</b>	91.92	91.38	91.23

**Tableau 4.84** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 9$

- **2<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la deuxième résolution. Ainsi le vecteur de caractéristiques est composé de 4x4 valeurs.

**La recherche du paramètre  $\lambda$** 

La valeur du  $C$  est fixée et on fait varier le  $\lambda$ , le tableau 4.85 montre les résultats obtenus.

$\lambda$	10	15	20	25	26	27	<b>28</b>	29	30
$C$	10	10	10	10	10	10	10	10	10
Taux (%)	88.78	90.48	91.52	91.87	91.92	92.02	<b>92.20</b>	92.07	91.77

**Tableau 4.85** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 28$

**La recherche du paramètre  $C$** 

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.86 montre les résultats obtenus.

$\lambda$	28	28	28	28	28	28	28	28	28
$C$	3	4	5	6	9	<b>10</b>	11	15	20
Taux (%)	91.67	91.82	92.02	91.92	91.87	<b>92.20</b>	91.97	91.72	91.43

**Tableau 4.86** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 10$

- **3<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la troisième résolution. Ainsi le vecteur de caractéristiques est composé de 2x2 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.87 montre les résultats obtenus.

$\lambda$	5	10	15	20	21	<b>25</b>	35
$C$	10	10	10	10	10	10	10
Taux (%)	71.40	85.45	88.49	90.63	90.78	<b>91.18</b>	91.03

**Tableau 4.87** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 25$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.88 montre les résultats obtenus.

$\lambda$	25	25	25	25	25	25	25	25
$C$	1	5	9	<b>11</b>	12	15	50	100
Taux (%)	88.14	90.83	91.13	<b>91.33</b>	91.23	91.03	88.19	86.84

**Tableau 4.88** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 11$

On remarque qu'à chaque fois on augmente la résolution le taux de reconnaissance chute, le meilleur résultat est obtenu avec la première résolution ce qui explique la perte d'information par filtrage passe bas puisqu'on s'est intéressé à l'image approximée.

L'ondelette de coif2 a donné un taux de 92.22%.

#### 4.4.4.3.3 Coiflets (coif3)

L'ondelette de Coiflets (coif3) est appliquée sur les images initiales.

- **1<sup>ère</sup> résolution**

Ici, on s'intéresse à l'image approximée de la première résolution. Ainsi le vecteur de caractéristiques est composé de 8x8 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.89 montre les résultats obtenus.

$\lambda$	5	10	10.5	<b>11</b>	11.5	11.75	12	15
$C$	10	10	10	10	10	10	10	10
Taux (%)	63.37	91.33	91.28	<b>91.38</b>	91.38	91.33	91.28	90.48

**Tableau 4.89** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 11$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.90 montre les résultats obtenus.

$\lambda$	11	11	11	11	11	11	11	11
$C$	<b>1</b>	2	3	4	5	6	8	11
Taux (%)	<b>93.57</b>	93.56	93.27	92.77	92.17	91.72	91.47	91.33

**Tableau 4.90** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 1$

- **2<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la deuxième résolution. Ainsi le vecteur de caractéristiques est composé de 4x4 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.91 montre les résultats obtenus.

$\lambda$	11	13	15	16	17	<b>17.5</b>	18	20
$C$	10	10	10	10	10	10	10	10
Taux (%)	87.49	88.73	89.83	89.93	90.08	<b>90.28</b>	90.08	89.93

**Tableau 4.91** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 17.5$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.92 montre les résultats obtenus.

$\lambda$	17.5	17.5	17.5	17.5	17.5	17.5	17.5	17.5
$C$	1	<b>2</b>	3	4	5	8	12	15
Taux (%)	93.32	<b>93.37</b>	92.87	92.47	91.87	90.88	89.43	88.34

**Tableau 4.92** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 2$

- **3<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la troisième résolution. Ainsi le vecteur de caractéristiques est composé de 2x2 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.93 montre les résultats obtenus.

$\lambda$	5	10	15	20	25	30	<b>40</b>	50
$C$	10	10	10	10	10	10	10	10
Taux (%)	46.18	76.98	80.76	84	86.29	87.99	<b>88.93</b>	88.78

**Tableau 4.93** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 40$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.94 montre les résultats obtenus.

$\lambda$	40	40	40	40	40	40	40	40
$C$	3	5	9	<b>10</b>	11	12	15	20
Taux (%)	87.84	88.68	88.78	<b>88.93</b>	89.03	88.83	88.73	88.69

**Tableau 4.94** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 10$

On remarque qu'à chaque fois on augmente la résolution le taux de reconnaissance chute, le meilleur résultat est obtenu avec la première résolution ce qui explique la perte d'information par filtrage passe bas puisqu'on s'est intéressé à l'image approximée.

L'ondelette de coif3 a donné un taux de 93.57%.

#### 4.4.4.3.4 Coiflets (coif4)

L'ondelette de Coiflets (coif4) est appliquée sur les images initiales.

- **1<sup>ère</sup> résolution**

Ici, on s'intéresse à l'image approximée de la première résolution. Ainsi le vecteur de caractéristiques est composé de 8x8 valeurs.

#### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.95 montre les résultats obtenus.

$\lambda$	5	10	14	<b>15</b>	16	20
$C$	10	10	10	10	10	10
Taux (%)	46.58	89.98	90.73	<b>90.88</b>	90.58	90.48

**Tableau 4.95** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 15$

### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.96 montre les résultats obtenus.

$\lambda$	15	15	15	15	15	15	15	15	15
$C$	<b>1</b>	2	3	4	5	8	9	11	15
Taux (%)	<b>93.42</b>	93.27	93.12	92.27	92.22	91.13	90.88	90.53	89.83

**Tableau 4.96** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 1$

- **2<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la deuxième résolution. Ainsi le vecteur de caractéristiques est composé de 4x4 valeurs.

### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.97 montre les résultats obtenus.

$\lambda$	5	10	15	<b>40</b>	20	25	30
$C$	10	10	10	10	10	10	10
Taux (%)	36.62	77.27	87.69	<b>92.27</b>	89.43	90.08	90.93

**Tableau 4.97** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 40$

### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.98 montre les résultats obtenus.

$\lambda$	40	40	40	40	40	40	40	40	40
$C$	1	<b>2</b>	3	5	8	9	11	15	20
Taux (%)	91.92	<b>92.87</b>	92.82	92.52	92.57	92.52	92.17	91.38	90.68

**Tableau 4.98** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 2$

- **3<sup>ème</sup> résolution**

Ici, on s'intéresse à l'image approximée de la troisième résolution. Ainsi le vecteur de caractéristiques est composé de 2x2 valeurs.

### La recherche du paramètre $\lambda$

La valeur du  $C$  est fixée et on fait varier  $\lambda$ , le tableau 4.99 montre les résultats obtenus.

$\lambda$	10	20	25	30	40	<b>50</b>	100
$C$	10	10	10	10	10	10	10
Taux (%)	70.75	84.50	86.29	87.94	89.23	<b>90.18</b>	90.08

**Tableau 4.99** Recherche du paramètre  $\lambda$

Nous avons obtenu le meilleur résultat pour  $\lambda = 50$

#### La recherche du paramètre $C$

Ici on fixe  $\lambda$  obtenu précédemment et on fait varier  $C$ , le tableau 4.100 montre les résultats obtenus.

$\lambda$	50	50	50	50	50	50	50	50
$C$	2	3	4	5	<b>8</b>	11	15	20
Taux (%)	89.63	90.38	90.43	90.33	<b>90.53</b>	90.23	90.13	89.28

**Tableau 4.100** Recherche du paramètre  $C$

Nous avons obtenu le meilleur résultat pour  $C = 8$

On remarque qu'à chaque fois on augmente la résolution le taux de reconnaissance chute, le meilleur résultat est obtenu avec la première résolution ce qui explique la perte d'information par filtrage passe bas puisqu'on s'est intéressé à l'image approximée.

L'ondelette de coif4 a donné un taux de 92.42%.

Pour les ondelettes de Coiflets on a obtenu le meilleur résultat de classification 93.57% avec l'ondelette coif3.

#### 4.5 Interprétation des résultats

Pour la méthode de Kirsch, on remarque que plus l'image est compressée plus il y a perte d'information ce qui induit à la chute du taux de classification, cela est vérifié par les résultats obtenus précédemment l'image compressée en 8x8 a donnée un taux de 92.72%, par contre l'image compressée 4x4 a donnée un taux inférieur.

Pour la méthode de Gabor on a obtenu le meilleur résultat avec un taux de 92.92% pour l'image filtrée. Par contre pour la moyenne-énergie on a obtenu un résultat inférieur de 72.24%, et pour moyenne-variance avec un taux de 63.37%. Cela est dû au mauvais choix des paramètres.

Pour la méthode de Zernike, on remarque que pour l'image binarisée-amincie, on a obtenu un résultat meilleur de 83.55 % avec l'implémentation du module-phase. Par contre pour l'image en niveau de gris on a obtenu un résultat inférieur.

Pour la méthode d'ondelette, on remarque qu'avec la première résolution on obtient toujours le meilleur résultat car plus on augmente la résolution, on perd l'information essentielle de l'image originale.

- Pour les ondelettes de Daubechies on a obtenu le meilleur résultat de classification 92.52% avec l'ondelette db3.
- Pour les ondelettes Symmlets on a obtenu le meilleur résultat de classification 93.07% avec l'ondelette sym4.
- Pour les ondelettes Coiflets on a obtenu le meilleur résultat de classification 93.57% avec l'ondelette coif3.

Pour la méthode d'ondelette on a obtenu le meilleur taux de reconnaissance avec les ondelettes Coiflets.

#### 4.6 Conclusion

Dans ce chapitre, nous avons fait une étude comparative entre quatre méthodes d'extraction de paramètres (masques de Kirsch, filtres de Gabor, moments de Zernike et transformée en ondelettes). Plusieurs tests ont été réalisés d'une part, pour chercher le paramètre de la méthode d'extraction qui convient le mieux à notre application et d'autre part, pour régler les paramètres du classifieur SVM qui permettent d'améliorer d'avantage le taux de reconnaissance.

# Conclusion générale

### Conclusion générale

Dans le schéma de reconnaissance de chiffres manuscrits, interviennent deux étapes essentielles : l'étape d'extraction de caractéristiques et l'étape de classification. Ainsi, notre contribution dans ce mémoire, consiste à intervenir dans la première étape en choisissant les descripteurs de formes les plus discriminants possibles pour caractériser le contenu de l'image, et dans la seconde phase, nous avons implémenté le séparateur à vaste marge (SVM) avec l'approche un contre un comme classifieur et ce, grâce à ses performances.

Dans ce projet, nous avons utilisé quatre méthodes d'extraction de primitives ; la méthode de Kirsch qui consiste à extraire les composantes directionnelles (caractéristiques locales) et les caractéristiques globales à partir de l'image initiale. L'application des filtres de Gabor à l'image initiale, nous a permis d'extraire d'autres types de caractéristiques à partir des images filtrées comme l'énergie, la moyenne et la variance. Nous avons également implémenté les moments de Zernike connus comme descripteurs puissants et ce, grâce à leurs propriété d'invariance par translation, rotation et d'orthogonalité. Pour mener, une étude comparative, nous avons rajouté un autre descripteur très puissant, basé sur la transformée en ondelettes. En effet, ce dernier descripteur, surmonte tous les problèmes posés par les descripteurs précédents.

Pour la classification, on a tenté de présenter de manière simple et complète le concept de système d'apprentissage introduit par Vladimir Vapnik, les SVM. Nous avons donné une vision générale et une vision purement mathématiques des SVM qui est l'un des modèles de classification à avoir marqué la reconnaissance de formes en fournissant un cadre théorique statistique à la théorie de l'apprentissage. Cette méthode de classification est basée sur la recherche d'un hyperplan qui permet de séparer au mieux les ensembles de données. Nous avons exposé le cas linéairement séparable et le cas non linéairement séparable qui nécessite l'utilisation de fonctions ou noyaux (kernel) pour changer l'espace de représentation des données. Ces noyaux, de par leurs variétés, constituent l'une des avenues les plus prometteuses de la théorie des SVM.

Comme perspectives, nous préconisons d'utiliser d'autres méthodes d'extraction de caractéristiques afin de mieux caractériser les chiffres manuscrits. Comme il est intéressant de mener une étude comparative entre les approches des SVM un contre tous et un contre un.

# Bibliographie

- [1] Oivid Due Trier, Anil K. Jain, & Torfinn Taxt, "Feature extraction methods for character recognition a survey", *Pattern recognition*, 29(4), 641-662, 1996.
- [2] Al Falou Wassim, "Reconnaissance de caractères manuscrits par réseau de neurones", DEA Modélisation et ingénierie du logiciel scientifique, université Libanaise Saint Josef, Liban, 1998.
- [3] Richard O. Duda, Peter E. Hart , & David G. Stork, "Pattern Classification", (Second edition), New York: Wiley-Interscience, 2001.
- [4] Sébastien REBECCHI, "formalisme statistique pour ensembles de structures discrètes", support de cours, INSA de Lyon, 2009.
- [5] Nadia BENAHMED, " optimisation de réseaux de neurones pour la reconnaissance de chiffres manuscrits isolés : sélection et pondération des primitives par algorithmes génétiques ", maîtrise en génie de la production automatisée, école de technologie supérieure université du Québec, 2002.
- [6] Ousmail Faiz, Guerchouche Said, "méthode d'analyse et de la texture basée sur les parcours optimaux, matricés de concurrence et filtres de Gabor", mémoire de fin d'étude ingénieur, UMMTO, 2008.
- [7] D.S. Bolme. "ELASTIC BUNCH GRAPH MATCHING", Master of science, Colorado State University, 2003.
- [8] L.DJEMA, F.OULEBSIR-BOUMGHAR, Y. AIT BACHIR, "Parallélisation du Filtre de Gabor pour l'Extraction des Caractéristiques des Textures d'Image", SETIT, Tunisie, IEEE, 2012.
- [9] DERDOUR Khedidja, "Reconnaissance de formes du chiffre arabe imprimé : Application au code à barre d'un produit", magistère en Génie Industriel, université HADJLAKHDAR–BATNA, 2009.
- [10] S.BEDOUEHENE, "Recherche d'image par le contenu", mémoire de magister UMMTO.
- [11] Leila Chergui, Maamar Kef, Mohammed Benmohammed, "La Théorie de la Résonance Adaptative et les Moments de Zernike pour la Reconnaissance de Mots Arabes Manuscrits", cours : université Oum El Bouaghi, université Constantine.
- [12] Ouarda SOLTANI, "Restauration d'images Satellites via la transformée en Ondelette", thèse de magister en Electronique, université de Batna, 2011.
- [14] SAMIR HAKIM ABBOU, "une application de la transformée en ondelettes à la reconnaissance de commandes vocales en milieu bruités et sa mise en œuvre par processeur dédié au traitement du signal", thèse de maitrise en génie électrique, université du Québec, 2006.

- [13] N. HALLICHE, A.ALLOUANE, “application de la méthode support vector machines (SVM) pour la reconnaissance des chiffres manuscrits“, thèse de master en Electronique, UMMTO, 2011.
- [15] Guyon I, Poujaud I, Personnaz L, Dreyfus G, Denker J, & Le Cun Y, “Comparing different neural networks architectures for classifying handwritten digits2“. Int. J. Conf. on Neural Networks, 2, 127-132. Washington, USA, (1989).
- [16] N.E. ayat, “sélection de modèles automatiques des machines a vecteur de support : application à la reconnaissance de chiffres manuscrits“, thèse de doctorat, université du Québec, 2004.
- [17] V. N. Vapnik, “The Nature of Statistical Learning Theory“, Second Edition, Springer, 1999.
- [18] M. Schmidt, “Identifying speakers with support vector networks“, 1996.
- [19] J. Marques and P. Moreno, “A Study Of Musical Instrument Classification Using Gaussian Mixture Models And Support Vector Machines“, Technical Report CRL 99/4, Compaq Corporation, Cambridge Research laboratory, Cambridge, US, June 1999.
- [20] E. Osuna, R. Freund, and F. Girosi, “Training support vector machines : an application to face detection“, 1997.
- [21] S. Essid, “Classification automatique des signaux audio-fréquences : reconnaissance des instruments de musique“, Thèse de doctorat, l’Université Pierre et Marie Curie, Paris, 2005.
- [22] C. Cortes and V. N. Vapnik, “Support-vector networks“, Machine Learning, 20(3), 273-297, 1995.
- [23] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, “A Practical Guide to Support Vector Classification“, Department of computer science and information engineering, National Taiwan University, Technical report 2003.
- [24] Y. Le Cun, B. Baser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. “Backpropagation applied to handwritten zip code recognition“, Neural Computation, 1(4):541-551, Winter 1989.
- [25] J. Bromley and E. Sackinger, “Neural-network and k-nearest-neighbor classifiers“, Technical Report 11359-91 0819-16TM, ATT, 1991.