RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE ECHAHID HAMMA LAKHDAR - EL OUED
FACULTÉ DES SCIENCES EXACTES
Département D'Informatique

Mémoire de Fin D'étude
Présenté pour l'obtention du Diplôme de

MASTER ACADEMIQUE

Domaine : **Mathématique et Informatique**
Filière : **Informatique**
Spécialité : **Systèmes Distribués et Intelligence Artificielle**

Présenté par :

- **GUEDIRI Taoufik**
- **ZEKKOUR FERHAT Salim**

Thème

Reconnaissance des chiffres manuscrits à base des machines à vecteurs de supports

Soutenue le xx-xx- 2017 Devant le jury:

M.	ZAIZ Faouzi	MCA	Président
M.	GUIA Sana Alsahar	MAA	Rapporteur
M.	BERDJOUH Chafik	MAA	Encadreur

Année Universitaire: 2016-2017

Remerciements

Avant tout, nous remercions notre Dieu qui nous a donné le courage et la volonté de réaliser ce modeste travail.

Nous adressons nos vifs remerciements à notre encadreur ***Mr. BERDJOUH Chafik*** pour son effort et ses conseils.

Nous remercie vivement *Melle. SETTOU Tarablesse* qui m'a aidé et conseillé durant ce travail.

Nous remercie également *Mr. FELOUAT Hichem* qui m'a aidé dans ce travail.

Nous tenons également à remercier « ***TOUS*** » les messieurs et dames, nos professeurs qui nous ont enseigné durant deux ans de formation master en Informatique, pour leurs précieux conseils et ses orientations.

Nos remerciements vont également aux membres du jury d'avoir accepté d'évaluer nos travail. Sans oublier de remercier notre amis et notre collègues, tous d'une manière différente.

Enfin, merci à nos parents pour le soutien et l'encouragement qu'ils nous ont apporté tout au long de nos travail.

Dédicaces

Tout d'abord, je veux rendre grâce à Dieu,
le Clément et le Très Miséricordieux pour
son amour éternel.
C'est ainsi que je dédie ce mémoire à :
À ma mère pour sa tendresse et mon père
pour sa patience et encouragement,
À ma fiancée,
À mes très chers frères,
sœurs et leurs enfants,
À mes cousins et cousines, Et à mes amies
et tous ceux que j'aime.

Taoufik

Dédicaces

En témoignage d'amour et d'affection, je dédie ce travail avec une grande fierté
À mes parents qui ont été d'un dévouement exemplaire et d'un réconfort inestimable.
À mes frères, mes sœurs et toute ma famille en reconnaissance de leurs encouragements.
A tous mes amis pour leur sympathie, leur amitié et leur solidarité envers moi.
À ma fiancée.
J'espère que vous acceptez mes hautes salutations et considérations.
Que Dieu puisse vous protéger.

Salim

Résumé

La reconnaissance de chiffres manuscrits présente un défi très grand et joue un rôle très important dans le monde actuel pour rendre les machines capable de connaitre comme un homme et capable de résoudre des problèmes complexes tel que la reconnaissance de chiffres du (chèque bancaire, relevé des compteurs d'électricité, gaz et eau, etc.....). Malgré les tentatives de rendre la machine apprendre comme les humains, Mais jusqu'aujourd'hui, aucune machine capable de comprendre 100% de reconnaissance de chiffres manuscrits.

Les travaux de ce mémoire s'inscrivent dans le cadre de la reconnaissance automatique des chiffres manuscrits. L'approche proposée se compose essentiellement de deux étapes : extraction de caractéristiques et classification des pixels de l'image en utilisant une nouvelle méthode d'apprentissage: les machines à vecteurs de support (SVM). Une étude comparative a été menée avec l'utilisation de la méthode SVM et essai de différents noyaux (03) : Gaussien, polynomial et linéaire. L'approche proposée a été testée sur des images de chiffres manuscrits et a montré pour l'ensemble des tests réalisés, un taux moyen de bonne reconnaissance des chiffres.

Mots clés: reconnaissance de chiffres manuscrits, classification, extraction de caractéristiques, noyaux, SVM.

ملخص

ان التعرف على الارقام المكتوبة بخط اليد أصبح يلعب دورا هاما في العالم وذلك بجعل الآلة ذكية قادرة على حل المشاكل المعقدة مثل التعرف على الارقام الموجودة في (وصل عدادات الكهرباء والغاز، الصكوك البريدية والبنكيية... الخ) ، فبالرغم من المحاولات الكثيرة لتحقيق هذا لاتوجد لحد الآن آلة تستطيع التعرف 100 % على الأرفام المكتوبة بخط اليد.

يرتكز عملنا في هذه المذكرة على التعرف التلقائي على الأرقام المكتوبة بخط اليد، النهج المقترح من طرفنا يتكون أساسا من خطوتين مهمتين: استخراج الخصائص وتصنيف وحدات الصورة باستخدام طريقة تعلم جديدة :(SVM). وقد أجرينا دراسة مقارنة باستخدام SVM واختبار مختلف الأنوية (03): قوس، متعدد الحدود والخطية.

النهج المقترح تم تجربته على صور الأرقام المكتوبة بخط اليد واظهرت جميع التجارب التي اجريت معدل جيد للتعرف على الارقام.

كلمات البحث: التعرف التلقائي على الارقام المكتوبة بخط اليد ،التصنيف، استخراج الخصائص،الأنوية ، SVM .

Abstract

The knowledge of the handwritten digits recognition represents a great defy and play an important role in our world to enable the machine of knowing and distinguishing as human being. It resolves many complicated problems making the human life more easier, In spite of repetitive essays to replace the human by the machine but until now there is no machine that could recognition 100 % the handwritten digits.

This work based in automatic recognition of handwritten digits. The proposed approach consists essentially of two steps: extraction of characteristics and classifying pixels of the image using a new learning method: support vector machines (SVM). A comparative study was carried out using the SVM method and testing different kernels (03): Gaussian, polynomial and linear. The proposed approach was tested on images of handwritten digits and showed for all the tests performed, an average rate of good recognition of the digits.

Keywords: the handwritten digits recognition, classification, extraction of characteristics, kernels, SVM.

Table de matières

Liste des figures

1. Reconnaissance de l'écriture

2. Machines à Vecteur de Support

3. Conception & Mise en œuvre

Liste des Tableaux

Introduction générale

Depuis la fin des années soixante, des travaux intensifs ont été accomplis dans le domaine de la reconnaissance de l'écriture. En effet, des systèmes commerciaux sont aujourd'hui disponibles, particulièrement pour la poste (lecture les chiffres des chèques, tri postale) et dans les banques (traitement des chèques, des factures). À l'heure actuelle, les recherches s'orientent vers l'interprétation de l'écriture manuscrite et libre sans contraintes de très grande dimension.

Dans ce mémoire, nous nous intéressons à la reconnaissance des chiffres manuscrits hors - ligne qui reste aujourd'hui un thème de recherche ouvert. En effet, bien que le nombre de classes naturelles soit très réduit (chiffres '0' à '9'), on trouve à l'intérieur de chacune d'entre elles, une très grande variabilité de l'écriture, de plus, les conditions souvent relativement précaires dans les quelles sont écrits les chiffres (chèques écrits rapidement sur un coin de table) et la variabilité du matériel utilisé (utilisation de divers stylos, de différentes qualités de papier) tendent à compliquer la reconnaissance.

Plusieurs générations de machines d'apprentissage ont vu le jour dans le but de classifier, de catégoriser ou de prédire des structures particulières dans les données. Mais la plupart de ces techniques éprouvent de grandes difficultés à traiter les données de très haute dimension. Pour surmonter ce problème, on procède souvent par la sélection d'une partie des attributs des données pour réduire la dimension de l'espace d'entrée. Mais dans ce cas, on aura besoin d'utiliser des hypothèses simplificatrices qui ne se vérifient pas toujours en pratique. Par ailleurs, une méthode issue récemment d'une formulation de la théorie de l'apprentissage statistique due en grande partie à l'ouvrage de Vapnik en 1995 intitulé « *the nature of Learning statistical theory* » surmonte ce problème en imposant que le nombre de paramètres soit linéairement lié au nombre des données d'apprentissage. Cette technique est appelée Support Vector Machines (SVM) ou machines à vecteurs de support.

L'objet de ce mémoire est l'implémentation des SVMs pour la reconnaissance des chiffres manuscrits.

Ce mémoire est organisé en quatre chapitres :

- ✓ Le premier chapitre présenter le schéma standard de reconnaissance des chiffres manuscrits en mettant l'accent plus précisément sur le bloc classifieur.
- ✓ Le deuxième chapitre décrit en détail le concept ainsi que la formulation mathématique des SVMs.
- ✓ Le dernier chapitre, va mettre l'accent sur la conception & mise en œuvre de notre système proposée. Et enfin la conclusion générale.

Chapitre 1

Reconnaissance de l’écriture

1. introduction

La reconnaissance automatique de l'écriture s'intéresse à la conception et à la réalisation de systèmes capables d'interpréter et de transformer un texte sous forme d'image en un fichier texte. Grâce aux progrès récents de la puissance informatique, de nombreuses techniques de reconnaissance de l'écriture ont été également développées et perfectionnées, notamment pour les écritures des chiffres. Cependant, la grande variabilité inhérente à la nature de l'écriture manuscrite a rendu ce domaine de recherche très actif. Nous présentons dans ce chapitre, un aperçu de l'état de l'art des techniques dans les principaux modules de reconnaissance automatique de l'écriture manuscrits.

2. Production et reconnaissance de documents

La reconnaissance de documents est le processus inverse de la production. De la forme papier, elle essaie de remonter à la forme logique. La figure (1.1) illustre les deux processus.

Le document papier est saisi à l'aide d'un scanner de manière à obtenir une image sous la forme électronique (couleur, résolution). Une image bruitée et biaisée appelée image brute. Cette image est ensuite prétraitée pour fournir une image épurée avec une représentation plus claire.

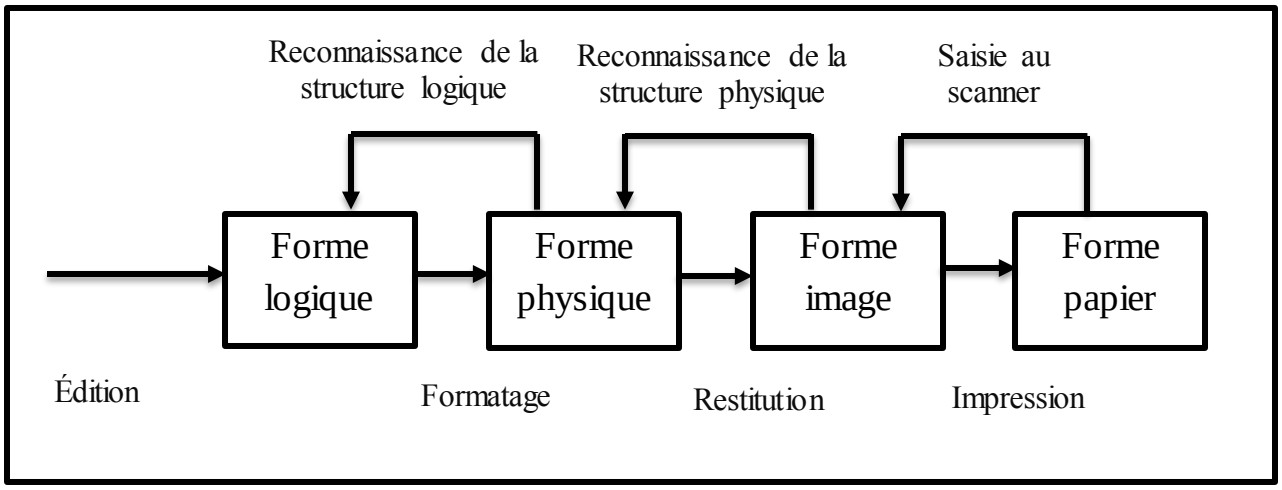


Figure 1.1 : Processus de production et de reconnaissance de documents.

La reconnaissance de la structure physique (ou forme physique) consiste d'une part à la détection et la classification des différentes zones de l'image en texte, graphique, table, formule, dessin ou photo et d'autre part à la découpe du texte en colonnes, paragraphes, lignes, mots et signes. Finalement, la reconnaissance de la structure logique (ou forme logique) consiste à faire un étiquetage logique aux différents objets de la structure physique et à réorganiser ces objets conformément au flux de lecture [LR, 2001].

3. Schéma générale d'un système de reconnaissance

Dans cette partie nous présenterons les grandes étapes qui composent une chaine de reconnaissance de l'écriture manuscrite. Nous aborderons les phases suivantes : acquisition (scanning, numérisation), prétraitement, segmentation à des chiffres séparés ou segments reliés à un chiffre, extraction des caractéristiques et classification.

La figure (2.1) illustre le schéma général d'un système de reconnaissance des chiffres.

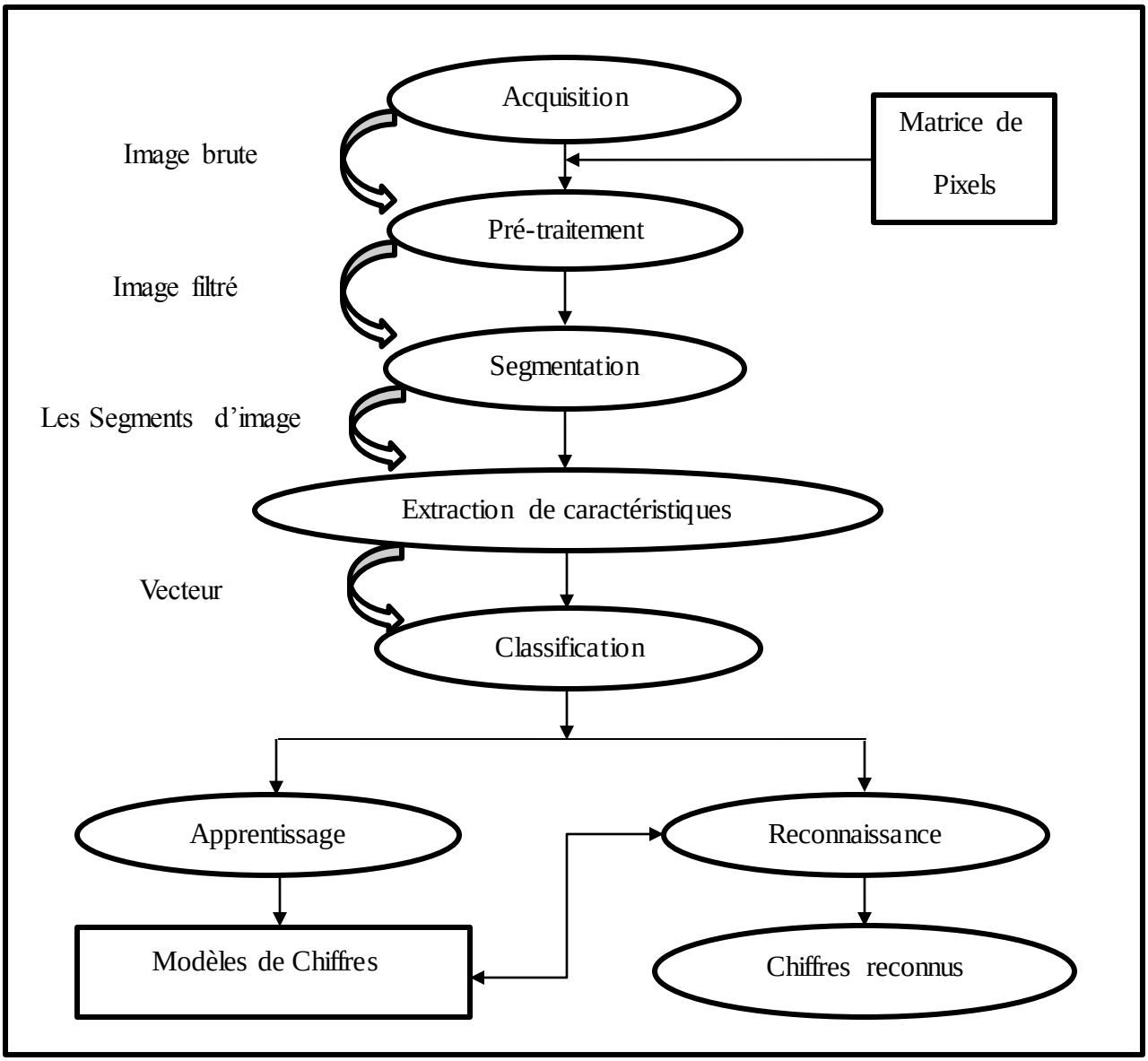


Figure 1.2 : Schéma général d'un système de reconnaissance des chiffres.

3.1. Acquisition de l'image

Les méthodes de reconnaissance se distinguent tout d'abord par le type d'acquisition des données. Cette phase consiste à capter l'image d'un texte au moyen des capteurs physiques (scanner, caméra,…) et de la convertir en grandeurs numériques adaptés au système de traitement, avec un minimum de dégradation possible [SH, 2007]. Il existe deux types :

3.1.1. Acquisition de l'écriture hors ligne

Dans le contexte de l'écriture hors ligne, les systèmes d'acquisition les plus courants sont essentiellement des scanners ou des caméras linéaires .La résolution de l'image numérisée influence les étapes ultérieures d'un système de lecture automatique. Il n'est communément admis que la résolution optimale d'une image en fonction de l'épaisseur du trait d'écriture (voir la figure 1.3).

3.1.2. Acquisition de l'écriture en ligne

Dans le cas de l'acquisition « en ligne », les dispositifs d'acquisition les plus répandus sont des tablettes à numériser ou des « papiers électroniques ». L'échantillonnage du tracé délivre une série de coordonnées décrivant la trajectoire du stylet au cours du temps. Les tablettes plus perfectionnées permettent également d'avoir accès aux informations de pression et d'inclinaison du stylo (voir la figure 1.3).

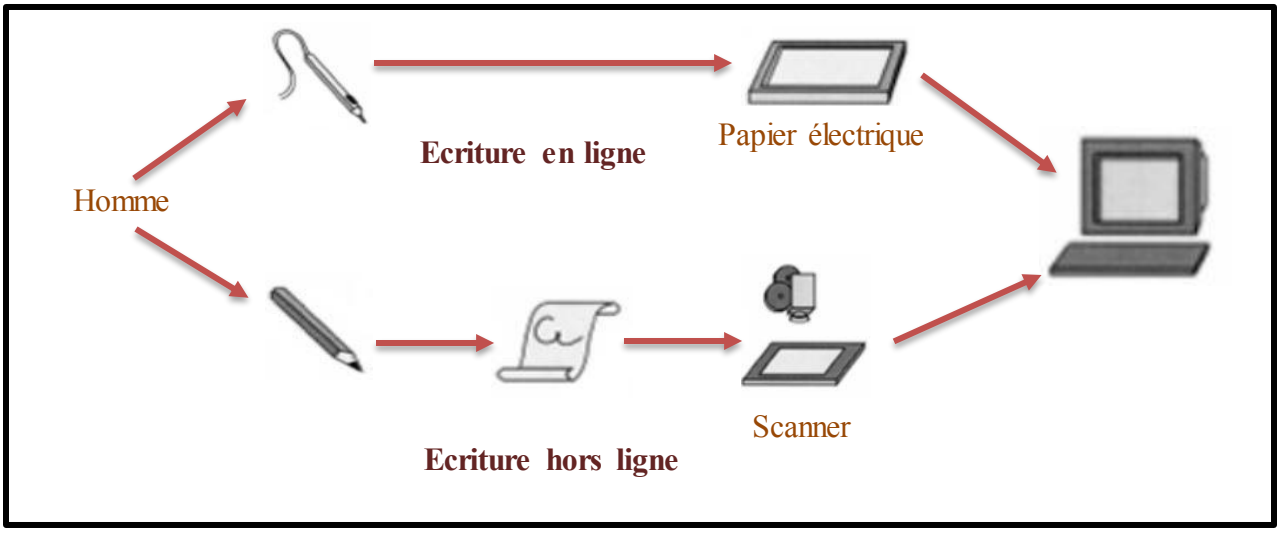


Figure 1.3 : Communication écriture Homme-machine.

3.2. Prétraitement

Le but des prétraitements est la réduction du bruit, de la distorsion, et de la variation des styles pour faciliter les traitements ultérieurs tels que la segmentation et l'extraction de caractéristiques. Ils comprennent principalement : la binarisation, la suppression du bruit, la normalisation et la squelettisation.

3.2.1. Binarisation de l'image

La représentation binaire d'une image pour faciliter les traitements ultérieurs, il permet de séparer l'information utile (l'écriture) du fond par l'utilisation d'un seuil de binarisation approprié, ce seuil doit traduire la limite des contrastes forts et faibles dans l'image. Selon la méthode de calcul du seuil de binarisation, on distingue deux types de binarisation : le seuillage globale et le seuillage adaptatif. Pour plus des techniques de binarisation voir [TT, 1995].

3.2.1.1. Seuillage global

Le seuillage global consiste à prendre un seuil ajustable, mais identique pour toute l'image. Chaque pixel de l'image est comparé à ce seuil et prend la valeur blanc ou noir selon qu'il est supérieur ou inférieur. Cette classification ne dépend alors que du niveau de gris du pixel considère.

Cette méthode convient pour les documents simples et de bonne qualité (visuel). Néanmoins, elle n'est plus applicable lorsque la qualité d'impression du texte n'est pas constante dans toute la page, des caractères peuvent être partiellement perdus. Pour l'évaluation du seuil optimal, on se réfèrera aux travaux de N. Otsu [NO, 1979].

3.2.1.2. Seuillage adaptatif

Dans les documents pour lesquels l'intensité du fond et l'intensité de la forme peuvent varier au sein du document, un seuillage global est inadapté. Il devient nécessaire de choisir le seuil de binarisation de manière locale. On calcule un seuil de binarisation pour chaque pixel de l'image, en fonction de son voisinage.

Par exemple J. Bernsen dans [BD, 1986] propose le calcul suivant pour déterminer le seuil de binarisation de chaque pixel :

$$Z_{min}(x,y) = min_{(x,y)\in v(x,y)}\ I(x,y) \qquad Eq: (1.1)$$

$$Z_{max}(x,y) = max_{(x,y)\in v(x,y)}\ I(x,y) \qquad Eq: (1.2)$$

Où :

v(x, y) : est le voisinage du pixel (x, y)

I(x, y) : est la valeur de l'intensité du pixel (x, y) dans l'image

$Z_{min}(x,y)$: Est la valeur minimale de l'intensité dans le voisinage du pixel considère

$Z_{max}(x,y)$: Est la valeur maximale de l'intensité dans le voisinage du pixel considère

La moyenne des valeurs $Z_{min}(x,y)$ et $Z_{max}(x,y)$ est utilisée pour déterminer le seuil de binarisation du pixel I(x, y) :

$$T(x,y) = \frac{Z_{min}(x,y) + Z_{max}(x,y)}{2} \qquad Eq:(1.3)$$

Donc :

$$I(x,y) = \begin{cases} 1 \; si \; I(x,y) \leq T(x,y) \\ 0 \; sinon \end{cases} \qquad Eq:(1.4)$$

La figure suivant illustré l'étape de la binarisation :

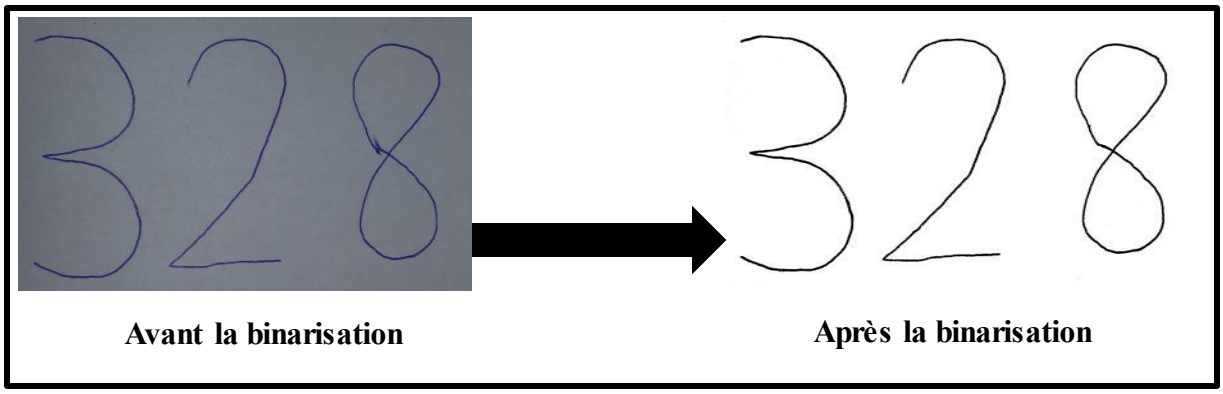


Figure 1.4 : Exemple de la binarisation.

3.2.2. Elimination du bruit

Le problème du bruit est très important mais très difficile particulièrement dans le cas où l'utilisateur écrit sur une feuille ayant le fond complexe comme un papier couleur. Une technique simple est de faire la soustraction entre l'image entrée et l'image d'un papier couleur (Il n'y a pas d'écriture). Cependant, les deux images doivent être alignées et les parties d'écriture qui traversent les éléments du fond vont être enlevées (voir la figure 1.5).

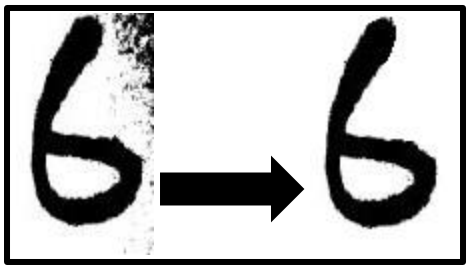


Figure 1.5 : Exemple d'Elimination du bruit.

3.2.3. Normalisation de l'image

Pour faciliter l'étape de reconnaissance, les chiffres segmentés doivent être normalisées 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.

3.2.4. Squelettisation

Le processus de squelettisation permet de réduire et de compacter la taille de l'image, et trouver un axe médian, définit comme l'ensemble de pixels S qui possèdent une égalité de distance par rapport aux pixels de frontière qui les entourent. La sortie de ce processus est un squelette d'un chiffre manuscrit [SA, 2008]. Le processus s'effectue par passes successives pour déterminer si un tel ou tel pixel est essentiel pour le garder ou non dans le tracé.

La représentation squelettique possède les avantages suivants:

- une bonne manière pour représenter les relations structurelles entre les composants de modèle (échantillon).
- Très utilisée dans les systèmes de reconnaissance de chiffres et caractères ... etc

Remarque :

Selon la qualité du document à traiter, le type de l'écriture et la méthode d'analyse adoptée, une ou plusieurs techniques de prétraitement sont utilisées. Mais pas forcément toutes.

3.3. Segmentation des chiffres

Pour simplifier la tâche de reconnaissance, la segmentation permet de diviser la chaîne de chiffres manuscrits en plusieurs entités. Pour faire la segmentation, à diviser la chaîne des chiffres en plusieurs images comportant chacune un chiffre isolé.

En pratique, deux étapes de segmentation sont nécessaires. La première étape est la segmentation primaire qui sépare les caractères selon les passages par zéro (ou par l'utilisation d'un seuil prédéfini) de l'histogramme de projection verticale. A l'issu de cette étape, tous les chiffres liés, chevauchés ou partiellement superposés ne sont pas segmentés. Aussi, les chiffres fragmentés risquent d'être coupés en deux. Pour cela, la deuxième étape dite segmentation secondaire opte à définir des règles de segmentation à partir d'une analyse des types de connexion [GGSG, 1995].

3.3.1. La segmentation primaire

Elle s'exécute lorsque les chiffres ne présentent pas des défauts de chevauchement ou de connection. La séparation des chiffres s'effectue par l'utilisation de l'histogramme de projection verticale. Elle s'applique sur l'image binaire qui consiste à localiser les chiffres en recherchant les plages correspondant à un nombre de pixels noirs non nuls. La séparation des chiffres contigüs consiste à détecter les passages par zéro de l'histogramme de projection [LRFCT, 2002] qui permet alors la détermination de l'emplacement de chaque chiffre de l'image. L'avantage de cette méthode est qu'elle permet de réaliser une segmentation sans connaissance à priori des images des chiffres. En revanche, son inconvénient majeur est qu'elle ne permet pas d'isoler les chiffres lorsqu'ils sont naturellement connectés ou en chevauchement. De plus, cette méthode est très sensible à la rotation et à la variabilité du style d'écriture (écriture détachée). La figure (1.6) illustre un exemple de mauvaise segmentation lorsque les chiffres sont en chevauchement dû à une écriture oblique.

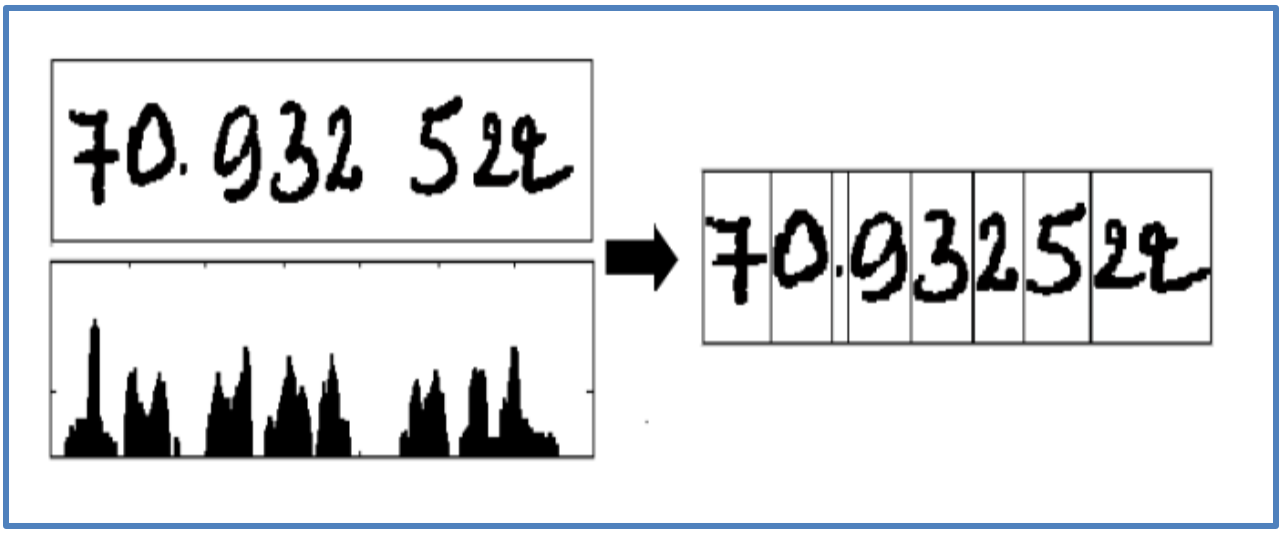


Figure 1.6 : Segmentation des chiffres par l'histogramme de projection verticale

La segmentation primaire exige une bonne qualité d'acquisition de l'image du chèque avec une phase de prétraitement efficace.

3.3.2. La segmentation secondaire

La procédure de segmentation secondaire fait une analyse fine de l'image déterminée à l'issue de la segmentation primaire, afin d'y détecter la présence éventuelle de plusieurs chiffres se chevauchant ou connectés entre eux. Aussi, il est nécessaire de développer des méthodes adaptées aux chiffres chevauchés ou connectés. Pour plus de détails voir [LRFCT, 2002].

3.4. Extraction de caractéristiques

L'extraction de caractéristiques consiste à transformer une image (chiffre...) en un vecteur de caractéristiques de taille fixe.

Cette transformation revient à changer l'espace de représentation des données, du plan de l'image vers un espace à ***n*** dimensions (R^n).

Le choix des caractéristiques est critique, et influence nettement le résultat. Ces caractéristiques doivent avoir les propriétés suivantes :

- La discrimination : permettre une bonne différentiation entre les classes de symboles à reconnaitre.
- Nombre de dimensions limite : le phénomène de malédiction de la dimensionnalité doit être évité afin de maintenir un nombre de dimension limité.
- L'invariante : les caractéristiques doivent être invariantes par translation, rotation, et changement d'échelle, elles doivent être également indépendantes de la distorsion et du bruit.

De nombreuses catégories des méthodes d'extraction de caractéristiques existant, nous avons distingué cinq catégories principales de caractéristiques [MNCC, 2007]:

- caractéristiques topologiques,
- caractéristiques statistiques,
- caractéristiques structurelles,
- globales ou locales, et
- superposition des modèles et corrélation.

3.4.1. Caractéristiques topologiques ou métrique

Ce type de caractéristiques est basé sur des densités de pixels. Dans ce type de caractéristiques, on compte également les profils et histogrammes. Pour maintenir un vecteur de taille fixe, on divise l'image en un nombre fixe de bandes horizontales et verticales. Les caractéristiques sont les moyennes des valeurs sur ces bandes [FM, 2008].

Elles consistent à compter dans une forme le nombre de trous, évaluer les concavités, mesures des pentes et autres paramètres de courbures et évaluer des orientations, mesurer la longueur et l'épaisseur des traits, détecter les croisements et les jonctions des traits, mesures les surfaces et les périmètres,… [SC, 2005].

3.4.2. Caractéristiques structurelles

Elles ressemblent beaucoup aux caractéristiques topologiques. La différence est qu'elles sont généralement extraites non pas de l'image brute mais à partir du squelette ou du contour de la forme en donnant ses propriétés globales et locales. Ainsi, on ne parle plus de trous, mais de boucles ou de cycles dans une représentation filiforme de l'objet. Parmi ces caractéristiques on peut citer [SC, 2005] :

- Les traits et les anses dans les différentes directions ainsi que leurs tailles.
- Les points terminaux.
- Les points d'intersections, Les boucles…….etc.

3.4.3. Caractéristiques statistiques

Les caractéristiques statistiques décrivent une forme en termes d'un ensemble de mesures extraites à partir de cette forme.

3.4.4. Caractéristiques globales ou locales

Les caractéristiques globales cherchent à représenter au mieux la forme générale d'un objet et sont donc calculées sur des images relativement grandes (ex : Couleur "Histogrammes , les moment statistique, cohérence spatiale (ie: Histogramme de connexité) , ..." texture et forme). Les caractéristiques locales sont calculées lors d'un parcours des pixels de l'image avec un pas d'analyse qui dépend de la modélisation, du type de caractéristique et de la taille de l'image [SC, 2005].

3.4.5. Superposition des modèles et corrélation

La méthode de « template matching » appliquée à une image binaire (en niveaux de gris ou squelettes), consiste à utiliser l'image de la forme comme vecteur de caractéristiques pour être comparé à un modèle (template) pixel par pixel dans la phase de reconnaissance, et une mesure de similarité est calculée [SH, 2007].

3.5. Classification

La classification consiste à affecter une forme donnée à une classe prédéfinie. Cette phase regroupe les deux tâches d'apprentissage et de décision. En effet, à partir de la même description de la forme en paramètres, elles tentent, toutes les deux d'attribuer cette forme à un modèle de référence. Le résultat de l'apprentissage est soit la réorganisation ou le renforcement des modèles existants en tenant compte de l'apport de la nouvelle forme, soit la création d'un nouveau modèle de l'apprentissage.

Le résultat de la décision est un « avis » sur l'appartenance ou non de la forme aux modèles de l'apprentissage. [AY, 1992].

4. Méthodes de classification

Généralement, un système de reconnaissance est construit à partir des connaissances à priori sur le problème. Le fait de classer un objet correspond donc à prendre une décision sur base d'une ou plusieurs règles. Dès lors, une des premières approches pour automatiser le traitement, fut d'extraire la connaissance experte de spécialistes sous forme de règles. Ainsi, pour chaque catégorie, on disposait d'un ensemble de règles permettant de déterminer l'appartenance d'un objet à la-dite catégorie (ou classe). Bien que cette façon ait été largement utilisée jusqu'à la fin des années 80, on en conçoit aisément les limites. En effet, quand le nombre d'exemples et de classes est très important, il devient difficile de rassembler toutes les connaissances expertes nécessaires aux prises de décision. Dans les années 90, l'approche Machine Learning (ML) ou apprentissage machine devient très populaire. Il s'agit d'apprendre automatiquement les règles de décision sur base d'un ensemble d'exemples déjà classés. L'apprentissage machine inclut entre autres les méthodes supervisées et non supervisées.

Dans les techniques non-supervisées, les objets sont présentés sans leur catégorie. Un exemple type de méthodes non-supervisées est le clustering dont le but est de créer des groupes (clusters) d'objets présentant des caractéristiques semblables [JC, 2002].

4.1. Réseaux de neurones

Un réseau de neurones est un assemblage de neurones connectés entre eux. Un réseau réalise une ou plusieurs fonctions algébriques de ses entrées, par composition des fonctions réalisées par chacun des neurones. La capacité de traitement de ce réseau est stockée sous forme de poids d'interconnexions obtenus par un processus d'apprentissage à partir d'un ensemble d'exemples d'apprentissage.

Il arrive souvent que les exemples de la base d'apprentissage comportent des valeurs approximatives ou bruitées. Si on oblige le réseau à répondre de façon quasi parfaite relativement à ces exemples, on peut obtenir un réseau qui est biaisé par des valeurs erronées. Par exemple, imaginons qu'on présente au réseau des couples situés sur une droite d'équation, mais bruités de sorte que les points ne soient pas exactement sur la droite. S'il y a un bon apprentissage, le réseau répond $ax + b$ pour toute valeur de x présentée. S'il y a sur apprentissage, le réseau répond un peu plus que ou un peu moins, car $(xi, (f(xi))$ chaque couple positionné en dehors de la droite va influencer la décision. [Ha, 2005]

4.2. Support Vecteur Machine (Machines à Vecteur de Support)

Les réseaux de neurones éprouvent de grandes difficultés à traiter les données de très haute dimension, de plus, ils possèdent un grand nombre de paramètres d'apprentissage à fixer par l'utilisateur. Pour surmonter ce problème, on procède souvent par la sélection d'une partie des attributs des données pour réduire la dimension de l'espace d'entrée. Mais dans ce cas, on aura besoin d'utiliser des hypothèses simplificatrices qui ne se vérifient pas toujours en pratique. Par ailleurs, une méthode issue récemment d'une formulation de la théorie de l'apprentissage statistique due en grande partie à l'ouvrage de Vapnik en 1995 intitulé « the nature of learning theory » [VV, 1995] surmonte ce problème en imposant que le nombre de paramètres soit linéairement lié au nombre des données d'apprentissage. Cette technique est appelée Support Vector Machines(SVM) ou machines à vecteurs de support. L'avantage principal provient du fait qu'il y a peu de paramètres à régler comparativement aux réseaux de neurones. Elle laisse très peu de place aux paramètres utilisateurs, bien que récemment proposée elle a fait l'objet d'un nombre important de publications.

5. Conclusion

Dans ce chapitre, nous avons présenté les différentes étapes de reconnaissance des chiffres manuscrits et les méthodes de classification fondés sur les machines d'apprentissage.

Dans le système de reconnaissance, nous nous intéressons plus précisément au classifieur fondé sur les SVMs .Aussi, nous abordons dans le prochain chapitre, la description détaillée de cette méthode de classification appliquées à la reconnaissance des chiffres manuscrits.

Chapitre 2

Machines à Vecteur de Support

1. Introduction

L'apprentissage machine basé sur la notion de généralisation à partir d'un grand nombre de données couvre des domaines tels que la reconnaissance de forme et la régression ne cesse d'avoir un développement dans ses méthodes et techniques, ceux-ci ne les a pas empêché de dévoiler des limites qui réduisent leur efficacité face à la complexité des problèmes du domaine, en même temps d'autre méthodes ont étaient misent en œuvre et dès leur première apparition elles ont surpassé les méthodes existantes auparavant, les SVMs sont une de ces nouvelles méthodes largement utilisés récemment. Dans ce chapitre qui suit on va présenter cette méthode.

2. Définition

Les "Support Vector Machins" appelés aussi « maximum margin classifier» (séparateur à vaste marge) sont des techniques d'apprentissage supervisé basés sur la théorie de l'apprentissage statistique (généralement considérés comme la 1ére réalisation pratique de cette théorie [BSAS, 2002]) et respectant les principes du (SRM) « structural risk minimization » (trouver un séparateur qui minimise la somme de l'erreur de l'apprentissage [VK, 2001]), un SVM repose sur les deux notions de vaste marge et fonction Kernel nous allons l'expliquer dans ce chapitre plus tard.

Les SVMs sont considéré comme l'un des modèles les plus importants parmi la famille des méthodes à Kernel, Ils ont gagné une forte popularité grâce à leur succès dans la reconnaissance des chiffres manuscrits avec un taux d'erreur de 1.1% en phase de test.

3. Apprentissage statistique et SVM

La notion d'apprentissage étant importante, nous allons commencer par effectuer un rappel. L'apprentissage par induction permet d'arriver à des conclusions par l'examen d'exemples particuliers. Il se divise en apprentissage supervisé et non supervisé. Le cas qui concerne les SVM est l'apprentissage supervisé. Les exemples particuliers sont représentés par un ensemble de couples d'entrée/sortie. Le but est d'apprendre une fonction qui correspond aux exemples vus et qui prédit les sorties pour les entrées qui n'ont pas encore été vues. Les entrées peuvent être des descriptions d'objets et les sorties la classe des objets donnés en entrée [HB, 2006].

Donc on peut schématise par :

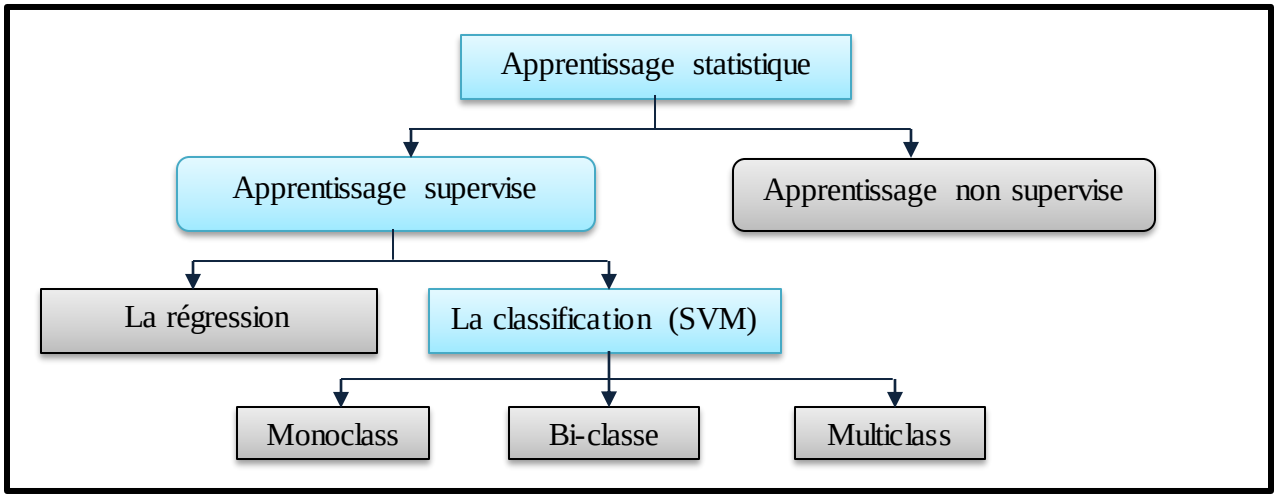


Figure 2. 1: Schéma des différents types d'apprentissage.

3.1 Objectif de l'apprentissage statistique

Effectuer une classification consiste à déterminer une règle de décision capable, à partir d'observations externes, d'assigner un objet à une classe parmi plusieurs. Le cas le plus simple consiste à discriminer deux classes. D'une manière plus formelle, la classification bi-classe revient à estimer une fonction $f : x \rightarrow \{+1, -1\}$ à partir d'un ensemble d'apprentissage constitué de couples (x_i, y_i). suivant une distribution de probabilité $P(x, y)$ inconnue, tels que :
$(x_i, y_i) \in X \times Y \quad o\grave{u} \quad i = 1, \ldots, N_x \; et \; y = \{+1, -1\}$,

De sorte à ce que f classifie correctement des exemples inconnus (x_i, y_i). Par exemple, on peut assigner x_t à la classe $(+1)$ si $(x_t) \geq 0$, et à la classe (-1) sinon. Les exemples inconnus sont supposés suivre la même distribution de probabilité $P(x, y)$ que ceux de l'ensemble d'apprentissage. La meilleure fonction f est celle obtenue en minimisant le risque :

$$R[f] = \int L\,[f(x), y]\, d\,P(x, y). \qquad (2.1)$$

Où L désigne une fonction de coût, comme par exemple : $L[f(x), y] = (f(x) - y)^2$.

3.2 Sur-apprentissage et sous-apprentissage

Quand on parle de la phase d'apprentissage deux notions de base très importante doivent être discutées :

- Le sur-apprentissage, et
- Le sous-apprentissage.

A. Sur-apprentissage :

Le sur –apprentissage (appelé overfitting en anglais) survient lorsqu‘ on cherche à trop «coller » aux données d‘entrainement (comme dans le cas d‘un apprentissage par cœur) donc c‘est le cas où la classificateur est classifier les exemples d‘autre classe à la classe, vue (figure suivant) Le modelé de haut degré est en situation de sur-apprentissage.

B. Sous-apprentissage :

C‘est la situation au contraire si la classe de fonctions considérée par l‘algorithme d‘apprentissage n‘est pas assez « riche » pour pouvoir décrire la diversité présente dans les données alors que c‘est le cas où la classificateur est classifier les exemples d‘un classe à l‘autre classe, vue (figure suivant) Le modèle linéaire est en situation de sous-apprentissage.

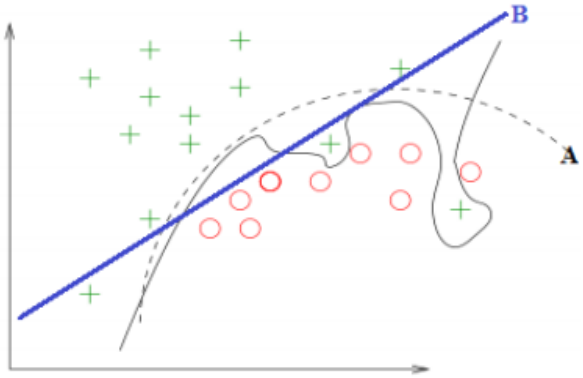


Figure 2.2: **A**: Cas Sur-apprentissage, **B** : Cas sous-apprentissage.

4. Notions de base

D‘abord nous allons voire quelques notions de base de SVM comme hyperplan, marge, support vecteur et séparateur Optimale :

4.1. Hyperplan

Un hyperplan qui sépare les deux ensembles de points [HB, 2006] (les deux classes) qu‘est montre dans figure (2.2).

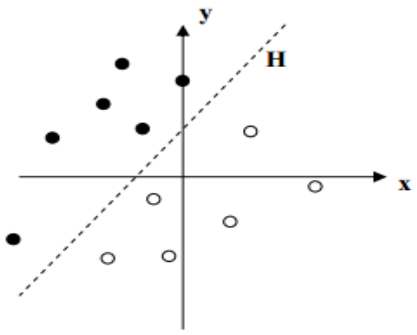


Figure 2.3: Exemple d‘un hyperplan.

Supposons pour l'instant que les échantillons de l'ensemble d'apprentissage sont séparables par un hyperplan (Figure 2.4), i.e. on choisit des fonctions de décision de la forme :

$$f(x) = \langle x, w \rangle + b. \quad (2.2)$$

4.2. Marge

La marge est la distance minimale entre les échantillons de l'ensemble d'apprentissage et la frontière de décision. La marge peut à son tour être mesurée grâce au vecteur poids w: puisque nous supposons que les échantillons sont séparables, on peut redéfinir w et b de sorte à ce que les échantillons x les plus proches de l'hyperplan satisfassent $|\langle x, w \rangle + b| = 1$.

4.3. Support vecteur

Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support.

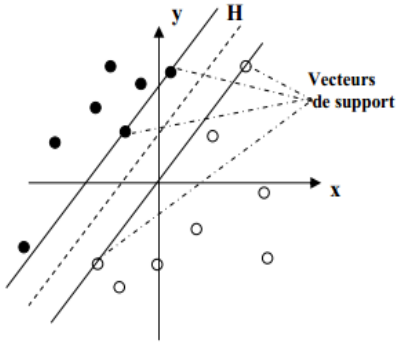


Figure 2.4: Exemple de vecteurs de support.

4.4. Séparateur Optimale « vaste marge »

Il est évident qu'il existe une multitude d'hyperplan valide mais la propriété remarquable des SVM est que cet hyperplan doit être optimal. Nous allons donc en plus chercher parmi les hyperplans valides, celui qui passe « au milieu » des points des deux classes d'exemples «voire figure (2.4)». Intuitivement, cela revient à chercher l'hyperplan le « plus sûr » [PMLA, 2003].

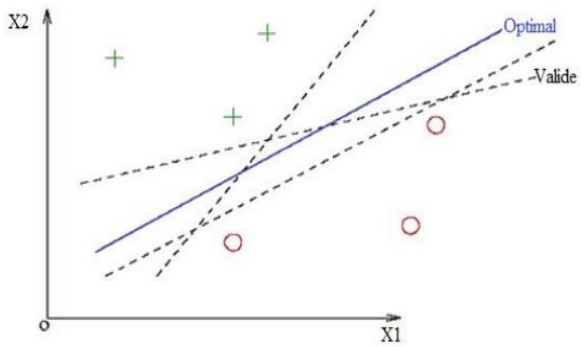


Figure 2.5: Exemple d'hyperplan séparateur optimal.

En effet, supposons qu'un exemple n'ait pas été décrit parfaitement, une petite variation ne modifiera pas sa classification si sa distance à l'hyperplan est grande. Formellement, cela revient à chercher un hyperplan dont la distance minimale aux exemples d'apprentissage est maximale.

On appelle cette distance « marge » entre l'hyperplan et les exemples. L'hyperplan séparateur optimal est celui qui maximise la marge. Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge [PMLA, 2003].

5. Pourquoi maximiser la marge ?

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. Dans le schéma qui suit, la partie droite nous montre qu'avec un hyperplan optimal, un nouvel exemple reste bien classé alors qu'il tombe dans la marge. On constate sur la partie gauche qu'avec une plus petite marge, l'exemple se voit mal classé [HB, 2006].

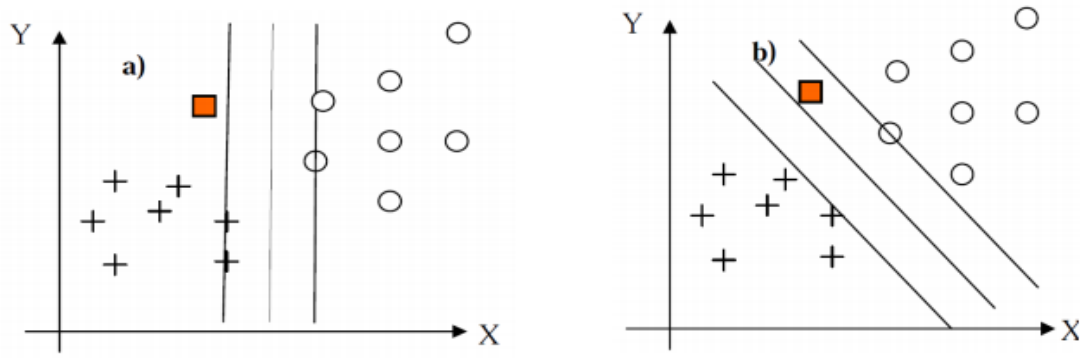


Figure 2. 6: a) Hyperplan avec faible marge, b) Meilleur hyperplan séparateur [HB, 2006]

En général, la classification d'un nouvel exemple inconnu est donnée par sa position par rapport à l'hyperplan optimal. Dans le schéma suivant, le nouvel élément sera classé dans la catégorie des «+ ».

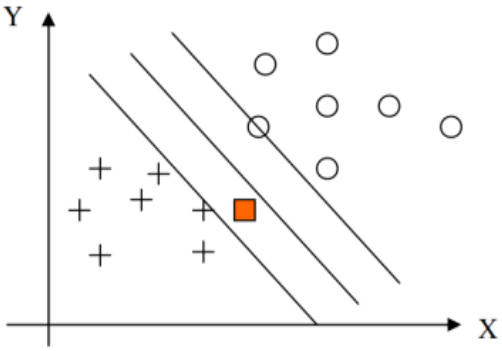


Figure 2.7: Exemple de classification d'un nouvel élément.

6. Linéarité et non-linéarité

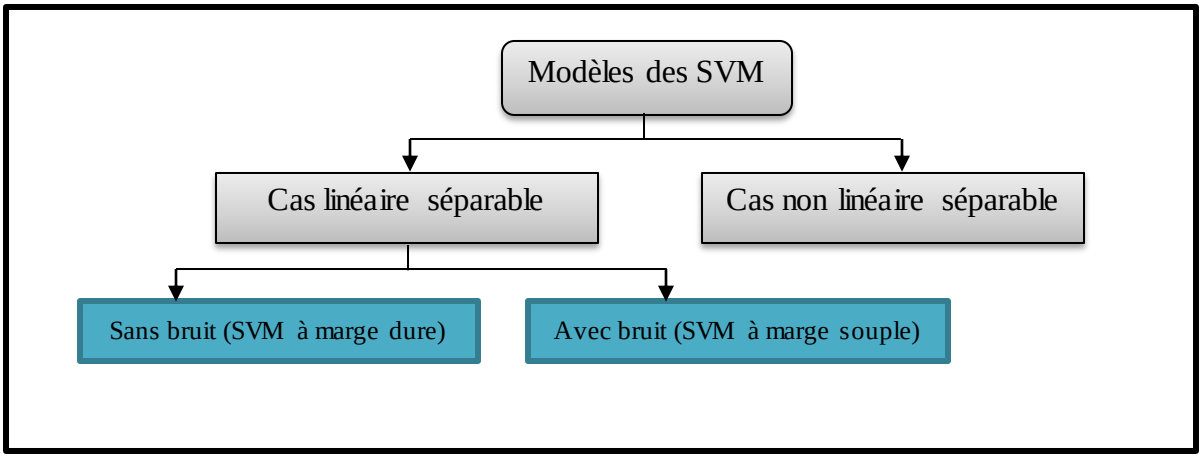


Figure 2.8: Schéma spectacles les différents modèle des SVM.

Parmi les modèles des SVM, on constate les cas linéairement séparable et les cas non linéairement séparable. Les premiers sont les plus simples de SVM car ils permettent de trouver facilement le classificateur linéaire. Dans la plupart des problèmes réels il n'y a pas de séparation linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables [HB, 2006].

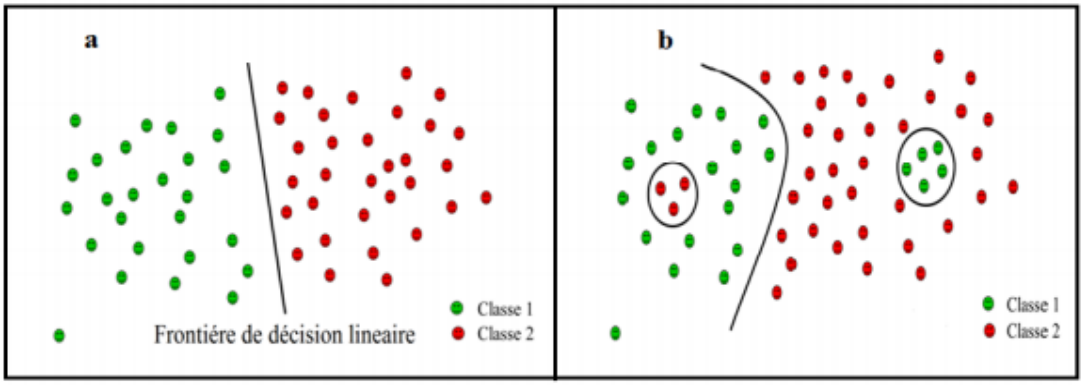


Figure 2.9: a) Cas linéairement séparable, b) Cas non linéairement séparable [HB, 2006]

6.1. Cas linéairement séparable

Ces cas sont les plus simples de SVM car ils permettent de trouver facilement le classificateur linéaire et dans un modèle linéaire, on a $f(x) = w.x + b$.

- L'hyperplan séparateur (frontière de décision) a donc pour équation $w.x + b = 0$
- La distance d'un point au plan est donnée par $d(x) = \frac{|w.x+b|}{\|w\|}$

- L'hyperplan optimal est celui pour lequel la distance aux points les plus proches (marge) est maximale.

6.1.1. SVM à marge dure

C'est le cas où on ne trouve pas des exemples dans l'espace de généralisation.

Maximisation de la marge

La marge est la distance du point le plus proche à l'hyperplan. Soient x_1 et x_2 eux points de classes différentes : $(f(x_1) = +1$ et $f(x_2) = -1)$, $w.x_1 + b = +1$ et $w.x_2 + b = -1$

Donc $(w.(x_1 - x_2)) = 2$, d'où : $\left(\frac{w}{\|w\|}.(x_1 - x_2)\right) = \frac{2}{\|w\|}$ [OB, 2001].

Considérons maintenant deux échantillons x_1 et x_2 de classes différentes telles qu'on ait $w.x_1 + b = +1$ et $w.x_2 + b = -1$.

- La distance entre xi et l'hyperplan est donnée comme ça :

$$di(w, b, x_i) = \frac{|w.x_i + b|}{\|w\|} \tag{2.3}$$

- La marge γ correspond alors à la distance entre x1 et x2 mesurée perpendiculairement à l'hyperplan :

$$\delta(w, b) = min_{x_i, y_i = -1}\, d(w, b, x_i) + min_{x_i, y_i = 1}\, d(w, b, x_i)$$

$$= min_{x_i, y_i = -1} \frac{|w.x_i + b|}{\|w\|} + min_{x_i, y_i = 1} \frac{|w.x_i + b|}{\|w\|}$$

$$= \frac{1}{\|w\|}\left[min_{x_i, y_i = -1} |w.x_i + b| + min_{x_i, y_i = 1} |w.x_i + b|\right]$$

$$= \frac{1}{\|w\|}[1 + 1]$$

$$= \frac{2}{\|w\|}$$ [TW, 2012].

On peut donc en déduire que maximiser la marge (maximiser $\frac{2}{\|w\|}$) revient à minimiser$||w||$ alors que minimiser $(\frac{1}{2}\|w\|^2)$ sous certaines contraintes que nous verrons dans les paragraphes suivants.

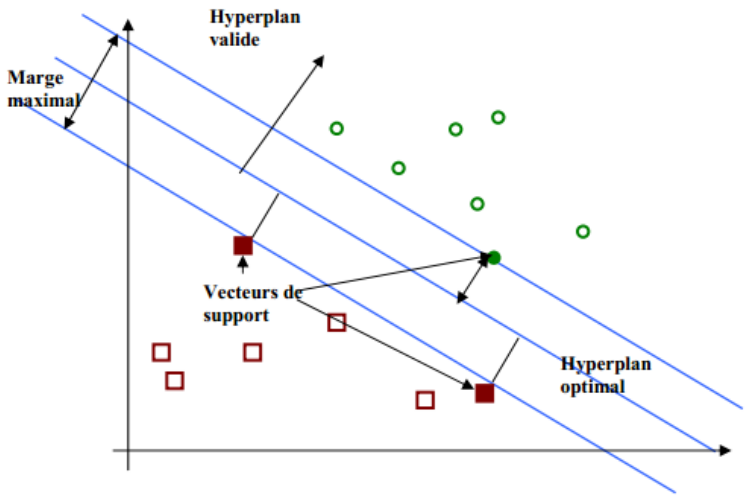


Figure 2.10: Exemple de marge maximal (SVM binaire à marge dure).

6.1.2. Cas non séparable :- SVM à marge molle «souple » (Avec bruit)

En réalité, un hyperplan séparateur n'existe pas toujours, et même s'il existe, il ne représente pas généralement la meilleure solution pour la classification. En plus une erreur d'étiquetage dans les données d'entrainement (un exemple étiqueté $+1$ au lieu de -1 par exemple) affectera crucialement l'hyperplan. Dans le cas où les données ne sont pas linéairement séparables, ou contiennent du bruit (outliers : données mal étiquetées) les contraintes de l'équation (2.3) ne peuvent être vérifiées, et il y a nécessité de les relaxer un peu. Ceci peut être fait en admettant une certaine erreur de classification des données (figure 2.11) ce qui est appelé « SVM à marge souple (Soft Margin) ».

On part du problème primal linéaire et on introduit des variables « ressort » pour assouplir les contraintes :

$$\begin{cases} min\ \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{n}\varepsilon_i \\ \forall i, y_i(w.x_i + b) \geq 1 - \varepsilon_i \end{cases}$$

On pénalise par le dépassement de la contrainte. La seule différence est la borne supérieure C sur les α. Où C est un paramètre positif libre (mais fixe) qui représente une balance entre les deux termes de la fonction objective (la marge et les erreurs permises) c.-à-d. entre la maximisation de la marge et la minimisation de l'erreur de classification.

On en déduit le problème dual qui a la même forme que dans le cas séparable [PMLA, 2003]

$$: \quad \begin{cases} max\ \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i.x_j \\ \forall i, 0 \leq \alpha_i \leq c \\ \sum_{i=1}^{n}\alpha_i y_i = 0 \end{cases}$$

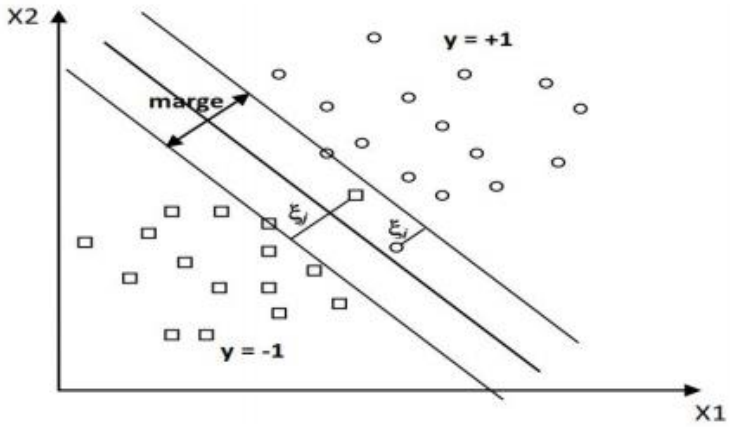


Figure 2.11: SVM binaire à marge souple

6.2 Cas non linéairement séparable

Il existe néanmoins des cas où on ne peut pas classer les entrées de façon linéaire [PMLA, 2003]. Pour surmonter les inconvénients des cas non linéairement séparable, l'idée des SVM est de changer l'espace des données. La transformation non linéaire des données peut permettre une séparation linéaire des exemples dans un nouvel espace. On va donc avoir un changement de dimension. Cette nouvelle dimension est appelé « espace de ré-description ».

En effet, intuitivement, plus la dimension de l'espace de ré-description est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples est élevée. Ceci est illustré par le schéma suivant [HB, 2006].

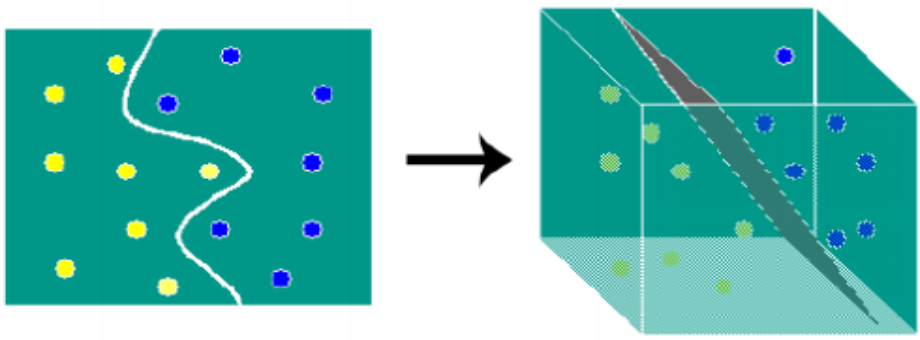


Figure 2.12: Exemple de changement de l'espace de données.

On a donc une transformation d'un problème de séparation non linéaire dans l'espace de représentation en un problème de séparation linéaire dans un espace de ré-description de plus grande dimension. Cette transformation non linéaire est réalisée via une fonction noyau.

En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. On peut citer les exemples de noyaux suivants : polynomiale, gaussien, sigmoïde et laplacien [HB, 2006].

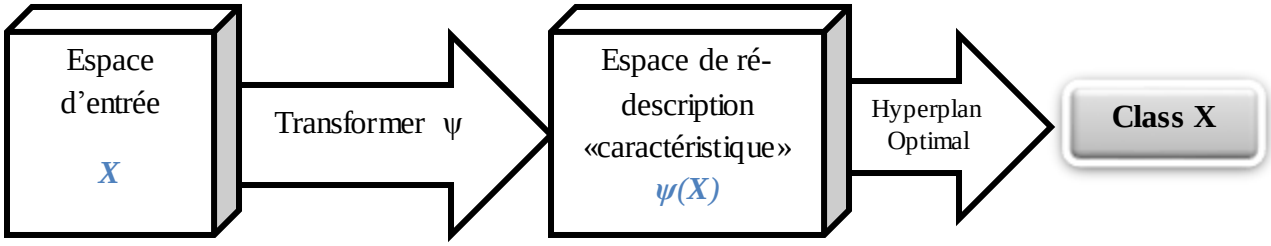


Figure 2.13: le principe général de technique SVM dans ce cas.

6.2.1. Fonction noyau (kernel)

Dans les cas linéaire, on pouvait transformer les données dans un espace où la classification serait plus aisée. Dans ce cas, l'espace de re-description utilisé le plus souvent est R (ensemble des nombres réels). Il se trouve que pour des cas non linéaires, cet espace ne suffit pas pour classer les entrées. On passe donc dans un espace de grande dimension [OB, 2001].

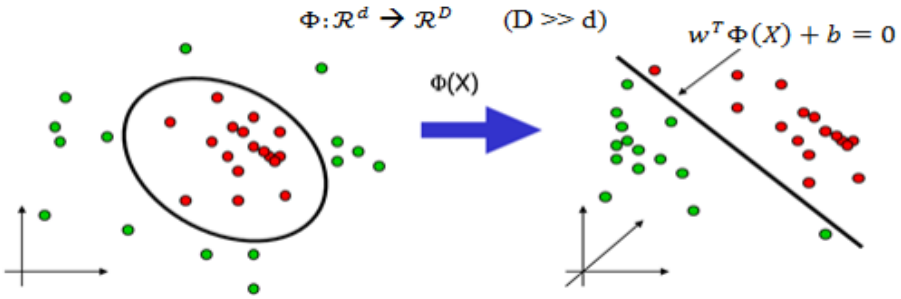


Figure 2.14: Illustration de passage à R3 [OB, 2001]

Le passage dans F = R3 (i.e. D=3). Rend possible la séparation linéaire des données. On doit donc résoudre :

$$\begin{cases} max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \varphi(x_i) \varphi(x_j) \\ \forall i, 0 \leq \alpha_i \leq c \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

Et la solution à la forme: $f(x) = \sum_{i=1}^{n} \alpha_i^* y_i \varphi(x_i).\varphi(x) + b$

Le problème et sa solution ne dépendent que du produit scalaire $\varphi(x_i)\varphi(x')$. Plutôt que de choisir la transformation non-linéaire $:X \to F$, on choisit une fonction K:(nombres réels) appelés fonction noyau.

Elle représente un produit scalaire dans l'espace de représentation intermédiaire. Du Coup k est linéaire (ce qui nous permet de faire le rapprochement avec le cas linéaire des paragraphes précédents). Cette fonction traduit donc la répartition des exemples dans cet espace $k(x,x') = \varphi(x).\varphi(x')$. Lorsque k est bien choisie, on n'a pas besoin de calculer la représentation des exemples dans cet espace pour calculer φ.

Exemple :

soit $x = (x_1, x_2)$ et $\varphi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$. Dans l'espace intermédiaire, le produit scalaire donne

$$\varphi(x)\varphi(x') = x_1^2 x_1'^2 + 2x_1x_2x_1'x_2' + x_2^2 x_2'^2$$

$$= (x_1x_1' + x_2x_2')^2$$

$$= (x.x')^2$$

On peut donc calculer $\varphi(x) \times \varphi(x')$ sans calculer φ: $k(x,x') = (x.x')^2$. K représentera donc le noyau pour les entrées correspondantes mais devra néanmoins remplir certaines conditions [PV, 2003].

Exemples de noyaux :

- Linéaire $K(x,x') = x.x'$.
- Polynomial $K(x,x') = (x.x')^d \; ou \; (c + x.x')^d$.
- Gaussien $K(x,x') = e^{-\left||x-x'|\right|.\frac{2}{\sigma}}$.
- Laplacien $K(x,x') = e^{-||x-x'||.\frac{1}{\sigma}}$.

On remarque en général que le noyau gaussien donne de meilleurs résultats et groupe les données dans des paquets nets. En pratique, on combine des noyaux simples pour en obtenir de plus complexe [OB, 2001].

6.2.2 Condition de Mercer

Une fonction k symétrique est un noyau si $(k(xi,xj))\, i,j$ est une matrice définie positive. Dans ce cas, il existe un espace F et une fonction φ tels que $k(x,x') = \varphi(x) \times \varphi(x')$. [OB, 2001]

Problèmes :

- ✓ Cette condition est très difficile à vérifier.
- ✓ Elle ne donne pas d'indication pour la construction de noyaux.
- ✓ Elle ne permet pas de savoir comment estφ.

7. SVMs multiclasse

La plupart des problèmes de classification sont a multi classe, les SVMs ont étaient conçu initialement pour résoudre des problèmes de classification a deux classes, en revanche d'autres méthodes ont permis l'extension des approches SVM pour traiter ce type de problème.

7.1. Un contre tous (One Versus the Rest)

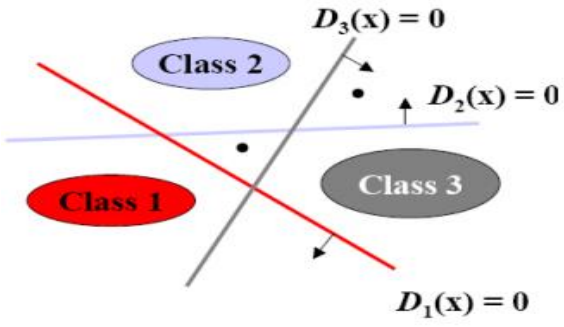


Figure 2.15 : Trois classes séparées par la méthode d'un contre tous avec des séparateur linéaire

Pour obtenir des classifieurs pour les M classes, on définit un ensemble de classifieur binaire (à deux classes) $f_1 \dots f_m$, chaque classifieur sépare la classe i de toute les autres classes produisant ainsi M fonction de décision de type sgn, la valeur $+1$ est attribué aux donnée appartenant à la classe i et la valeur -1 et attribué aux données appartenant aux classes restantes [BSAS, 2002]:

$$f^j(x) = sgn\left(g^j(x)\right)$$

$$\text{Avec} \quad g^j(x) = \sum_{i=1}^{m} y_i \alpha_i^j k(x, x_i) + b^j$$

La valeur maximale de $f^j(x)$ désigne l'argument j de f comme classe d'un exemple x_i, si la différence entre deux grand $g^j(x)$ est inferieur a un seuil θ lors de la classification d'un exemple x, cet exemple sera rejeté est ne sera affecté a aucune classe, cet écartement de l'exemple est appelé *Reject decision*.(figure 2.16)

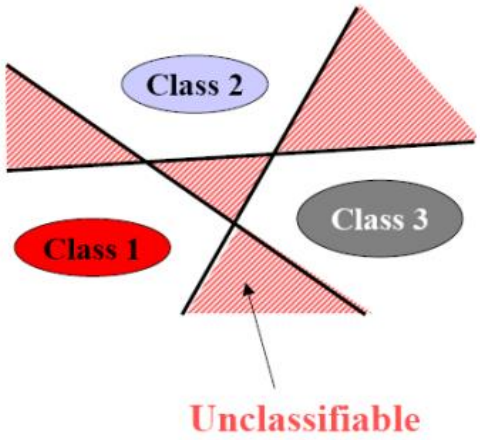


Figure 2.16 : Les surfaces rouges représente le « Reject decision »

7.2. Classification par pair (*Pairwise classification*)

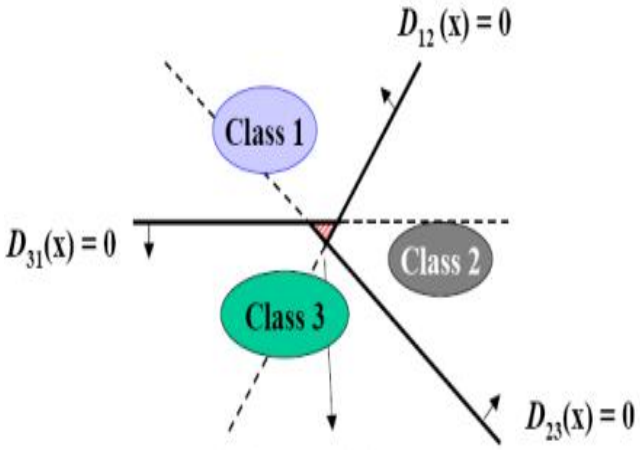


Figure 2.17 : Classification multiclasse par paire

Connu sous le nom de classification un contre un (one versus one), dans cette méthode on détermine un classifieur pour chaque pair de classe, pour M classe on aura $\frac{M.(M-1)}{M-2}$ classifieurs binaire (le nombre de fonctions de décision est égale aux nombre de classifieurs), le nombre des séparateurs est habituellement plus grand a celui des séparateurs sur la méthode un contre tous, pour un $M = 10$ on a besoin de 45 classifieurs binaires, chaque classe effectue un vote pour affecter un point x, la classe majoritaire après vote sera celle à qui le point est affecté.

D'après (figure 2.17) on remarque que la région des points non classifiable existe toujours.

8. Complexité

Nous allons évaluer la complexité (temps de calcul) de l'algorithme SVM. Elle ne dépend que du nombre des entrées à classer (d) et du nombre de données d'apprentissage (n).

On montre que cette complexité est polynomiale en n taille de la matrice hessienne = n²

$$dn^2 \leq Complexité \leq dn^3$$

En effet, on doit au moins parcourir tous les éléments de la matrice ainsi que toutes les entrées. Pour un très grand nombre de données d'apprentissage, le temps de calcul explose. C'est Pourquoi les SVMs sont pratiques pour des « petits » problèmes de classification [HB, 2006].

9. Architecture générale d'une machine à vecteur support

Une machine à vecteur support, recherche à l'aide d'une méthode d'optimisation, dans un ensemble d'exemples d'entrainement, des exemples, appelés vecteurs support, qui caractérisent la fonction de séparation. La machine calcule également des multiplicateurs associés à ces vecteurs.

Les vecteurs supports et leurs multiplicateurs sont utilisés pour calculer la fonction de décision pour un nouvel exemple. Le schéma de la relation (2.4) résume l'architecture générale d'une SVM dans le cas de la reconnaissance des chiffres manuscrits [BR, 2012].

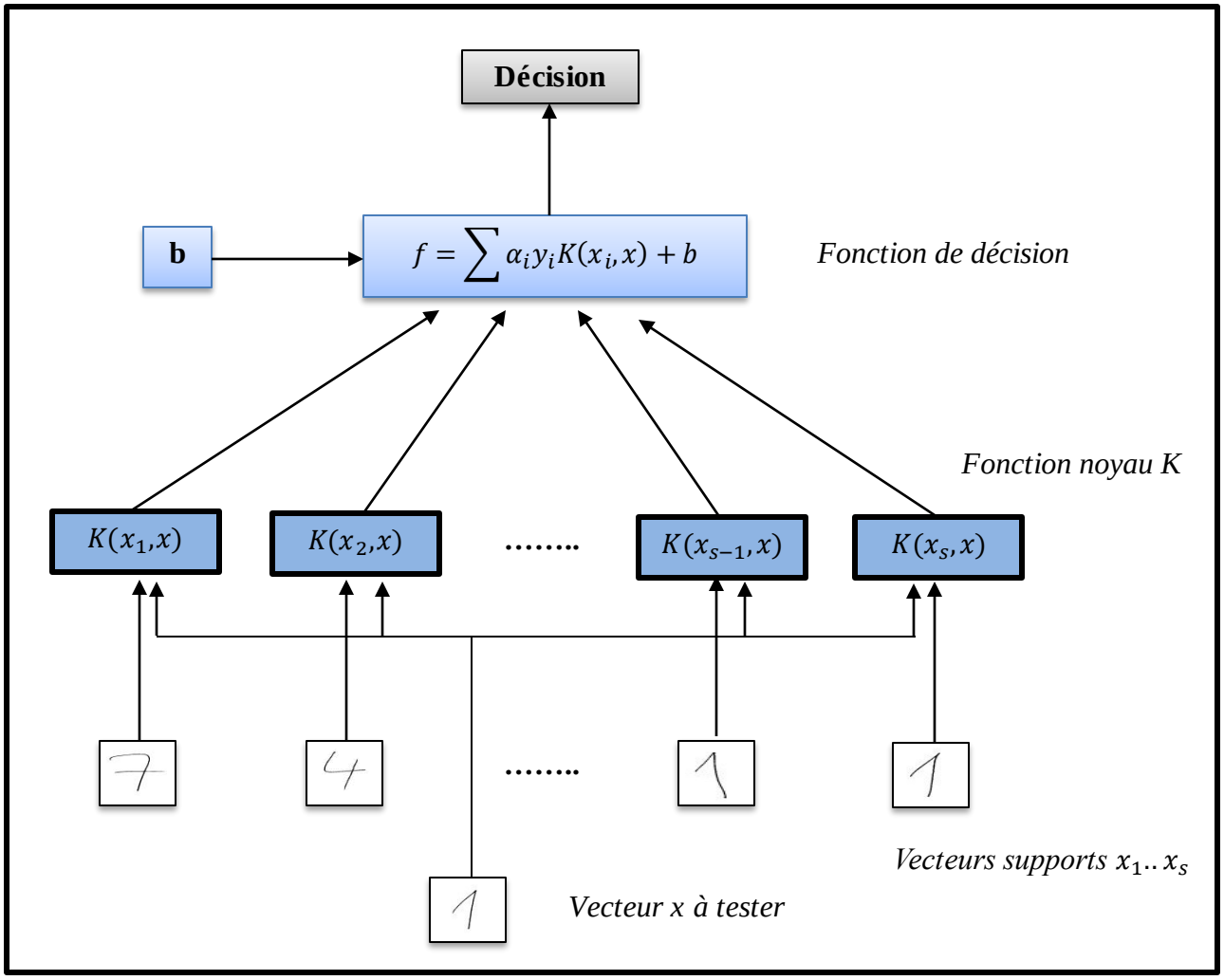


Figure 2.18: Architecture d'une machine à vecteur support

10. Les domaines d'applications

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 diagnostiques médicales et ce même sur des ensembles de données de très grandes dimensions.

11. Exemple d'application

Notre exemple d'application des SVMs présente une comparaison entre un Perceptron et un SVM en phase d'apprentissage pour un problème de classification linéaire.

Exemple d'apprentissage : points $a_i(x_1, x_2)$.

Nombre d'exemples : 454

Classes : 2 désigné par ± 1

Perceptron

-Perceptron à une couche d'entrée à 2 neurones, couche de sortie un seul neurone.

-Fonction de décision : tangente hyperbolique.

-Algorithme d'apprentissage: Widrow-Hoff

Modèle SVM

SVM linéaire a marge dure.

Implémentation sous MATLAB

Résultat

Temps de calcul pour le Perceptron : 46.86 sec

Tempe de calcul pour le SVM : 3.93 sec

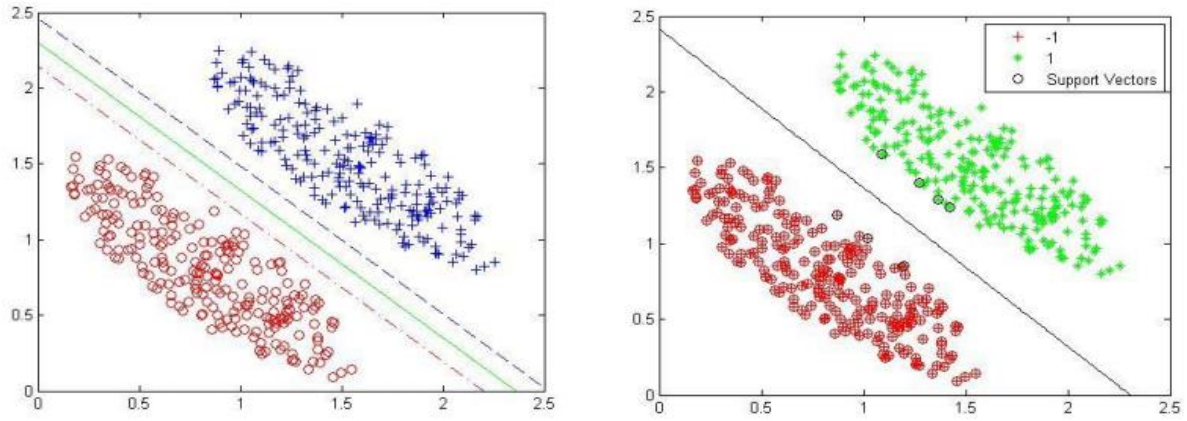


Figure 2.19 : A) Apprentissage par Perceptron, B) Apprentissage par SVM.

12. Avantages et inconvénients

Avantages

- Absence d'optimum local.
- contrôle explicite du compromis entre la complexité du classifieur et l'erreur.
- Possibilité d'utilisation de structure de données comme les chaines de caractères et arbres comme des entrées.
- traitement des données a grandes dimensions.

Inconvénients

- Demande des données négatives & positives en même temps.
- Besoin d'une bonne fonction Kernel.
- Problèmes de stabilité des calculs dans la résolution de certains programme quadratique a contraintes.

13. Conclusion

Dans ce chapitre, nous avons tenté de présenter de manière simple et complète le concept de système d'apprentissage introduit par Vladimir Vapnik, les Machines à Vecteurs de Support. Nous avons donné une vision générale et une vision purement mathématiques des SVM.

Cette méthode de classification est basée sur la recherche d'un hyperplan qui permet de séparer au mieux des ensembles de données. Nous avons exposé les cas linéairement séparable et les cas non linéairement séparables qui nécessitent l'utilisation de fonction noyau (kernel) pour changer d'espace. Cette méthode est applicable pour des tâches de classification à deux classes, mais il existe des extensions pour la classification multi classe. Nous nous sommes ensuite intéressés aux différents domaines d'application. Il existe des extensions que nous n'avons pas présentées, parmi lesquelles l'utilisation des SVM pour des tâches de régression, c'est-à-dire de prédiction d'une variable continue en fonction d'autre variable.

Chapitre 3

Conception & Mise en œuvre

1. Introduction

Dans le chapitre précédent, nous avons présenté la méthode de classification SVM, inspirée de la théorie statistique de l'apprentissage de *Vladimir Vapnik* introduite en 1995. Dans ce chapitre nous allons présenter notre contribution dans le système de reconnaissance de chiffres manuscrits, puis nous allons présenter la mise en œuvre de notre application en donnant les outils nécessaires, tel que le langage de programmation. Ensuite, nous présenterons quelques résultats concernant l'apprentissage et la reconnaissance de chiffres manuscrits.

2. Mise en œuvre du système

Dans cette section, nous allons présenter une conception du système en donnant une vue globale du système, ensuite nous allons détaillé chaque module composant le système séparément. Le système peut être vu comme étant 05 modules complétant chacun une tâche du processus de reconnaissance :

- Acquisition
- Pré-traitement
- Segmentation
- Extraction de caractéristiques
- Classification

(Voir la figure 3.1) :

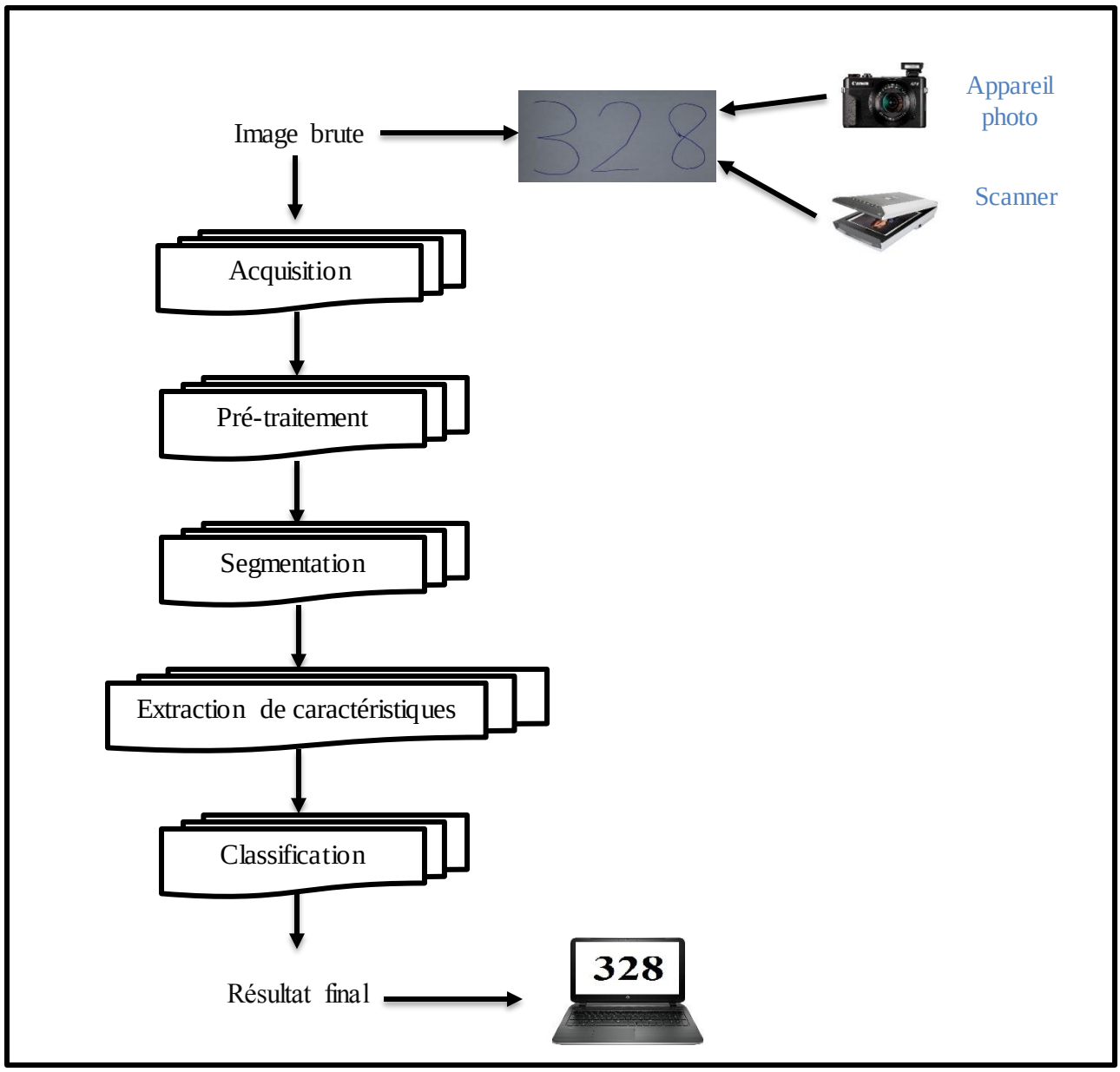


Figure 3. 1: Illustration des modules du système.

2.1. Acquisition

L'acquisition est la première étape du processus de reconnaissance de chiffres manuscrits. Ce module consiste tout simplement à acquérir ou charger l'image et la sauvegarder en un format d'image connu (par exemple : jpeg, gif, bmp…etc.) et /ou la transformer en un matrice de pixels dans le but de l'utiliser directement.

2.2. Pré-traitement

L'image obtenue dans la phase d'acquisition n'est qu'une image brute (bruitée). Durant la phase de prétraitement nous allons utiliser notre proposition dans cette phase comme la normalisation et la squelettisation afin de rendre cette image prête pour des traitements futurs.
Les étapes principales de prétraitement sont :

- Binarisation
- Filtrage
- Normalisation
- Squelettisation

2.2.1. Binarisation

La première étape de prétraitement pour l'image en entrée est l'étape de binarisation. Ce traitement nécessite pour seul paramètre un seuil. Tout pixel au-dessus de ce seuil sera considéré «blanc», «noir» (0,1) (La figure (3.2) illustre l'étape de binarisation).

$$img(i,g) = \begin{cases} 0, & img(i,j) < 128 \\ 1, & img(i,j) \geq 128 \end{cases}$$

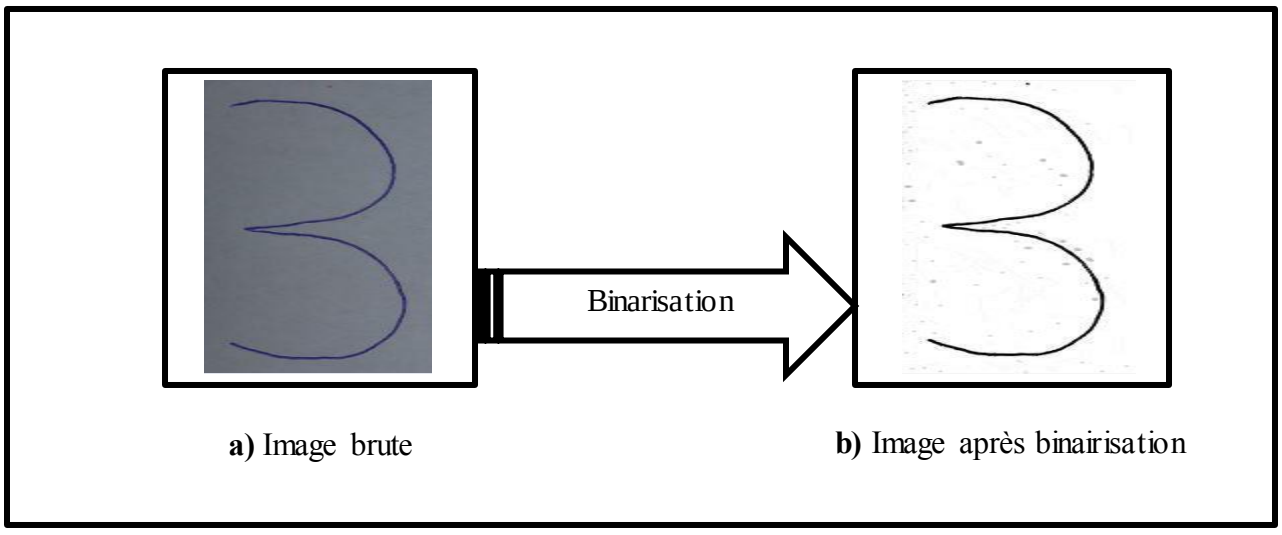


Figure 3.2 : Application de binairisation.

2.2.2 Filtrage

L'image représente maintenant après la binarisation contient des imperfections (image bruitée), alors le filtrage consiste à suivre certains processus afin d'éliminer ces tâches et de représenter en mieux les points d'intérêt**.**

Pour réaliser l'étape de filtrage, nous proposons une élimination de tous les pixels qui respectent la condition suivante : un pixel noir (**1**) entouré par des pixels blanc (**0**) ou un pixel blanc (**0**) entre deux pixels noir (**1**). (Voir le code de fonctions ci-après) :

```
Fonction entoure_blanc (mat [,], i, j : entier) : booléen ;
Debut
  Si (mat [i-1,j]=0 et mat[i,j+1]=0 et
       mat[i+1,j]=0 et mat [i,j-1]=0) alors
     entoure_blanc ← vrai;
  Sinon
     entoure_blanc ← faux ;
  Finsi
Fin.
```

```
Fonction entoure_noir (mat [,], i, j : entier) : booléen ;
Debut
  Si (mat [i-1,j]=1 et mat[i,j+1]=1 et
       mat[i+1,j]=1 et mat [i,j-1]=1) alors
     entoure_noir ← vrai;
  Sinon
     entoure_noir ← faux ;
  Finsi
Fin.
```

```
Fonction Filtrage (matrice [,] : entier) : matrice [,] ;
Var i, j : entier;
Debut
  Pour (i=1 à matrice.longueur) faire
      Pour (j=1 à matrice. largeur) faire
        Si (entoure_blanc [i, j]) alors
          matrice [i, j] = 0;
        Finsi
        Si (entoure_noir [i, j]) alors
          matrice [i, j] = 1;
        Finsi
      FinPour
  FinPour
  Filtrage ← matrice;
Fin.
```

L'étape de filtrage est illustrée par la figure suivante:

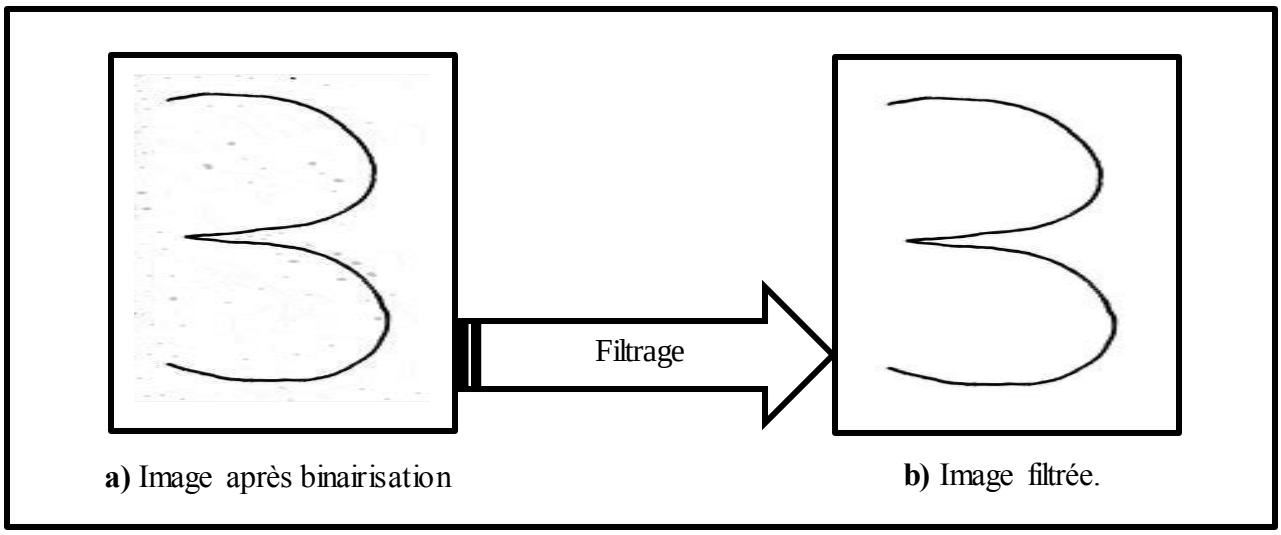
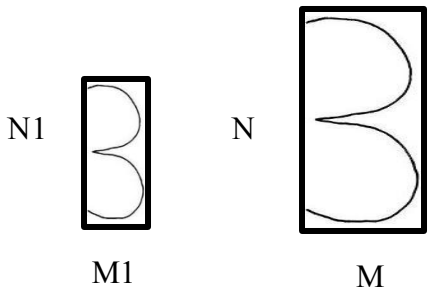


Figure 3.3 : Illustration d'un exemple de filtrage.

2.2.3. Normalisation

La normalisation peut être indispensable pour certains types de systèmes (comme les SVM), La taille d'un chiffre peut varier d'une écriture à l'autre, ce qui peut causer une instabilité des paramètres (descripteurs de ce chiffre). Une technique naturelle de prétraitement consiste à ramener les chiffres à la même taille (figure 3.4), et dans ce système nous proposons cette technique pour ramener tous les images à la taille (32*32) pixels.

On commence par la présentation d'une technique de variation de la taille d'une image. Soit un objet (chiffre) de taille N1×M1 qu'on souhaite ramener à la taille N×M (32*32).



La technique suivie est :

Si (i, j) est un pixel dans l'image finale (de taille N*M), le pixel correspondant dans l'image source (de taille (N1*M1)) est ([(i*N1)/N], [(j*M1)/M].

L'algorithme de variation de taille est donc :

1- Soit (**a**) l'image source et (**b**) l'image finale

2-

Pour i = 0 **à** N-1

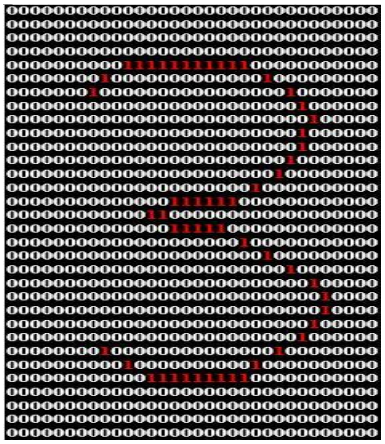
  Pour j = 0 **à** M-1

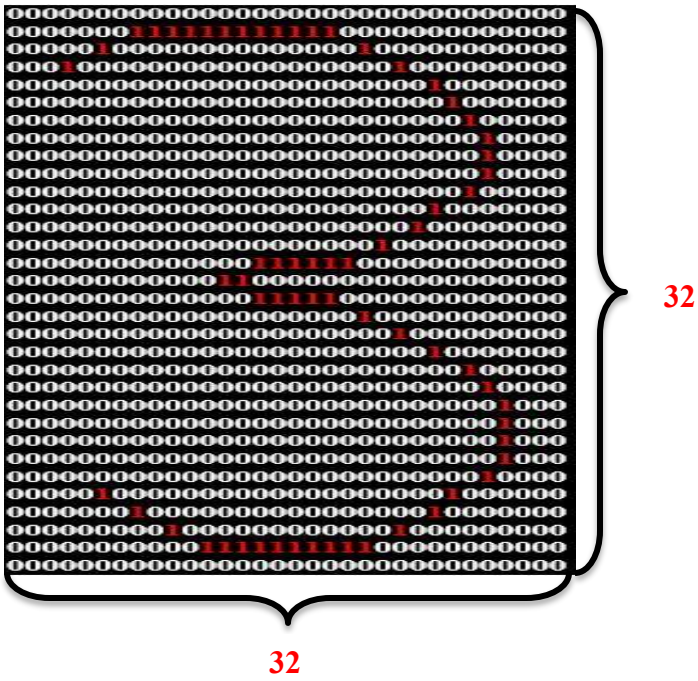
    b (i, j) = a([(i*N1)/N],[(j*M1)/M])

  Finpour

Finpour

Voici un exemple de résultat d'application de cette technique :





a) L'image source. **b)** L'image finale.

Figure 3.4 : Normalisation de la taille.

Notons que l'image résultante de cette technique de normalisation amène à une image trop épaisse. Il faut donc une phase de squelettisation pour obtenir une image d'épaisseur uniforme entre toutes les images de la base d'apprentissage et pour celles des nouvelles observations.

2.2.4. Squelettisation

La squelettisation est une opération qui consiste à transformer l'image en sa représentation en "fil de fer" (squelette a un pixel d'épaisseur 1).C'est une manière de représenter l'information indépendamment de l'épaisseur initiale de l'écriture. Cette technique constitue l'étape de squelettisation du chiffre.

Les deux critères retenus pour les méthodes de squelettisation sont les suivants :

1. l'épaisseur du chiffre aminci doit être de 1.
2. le chiffre 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 de squelettisation 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.

1ère 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 ci-après présente les 8-voisins d'un point p_1.

p_9	p_2	p_3
p_8	$\mathbf{p_1}$	p_4
p_7	p_6	p_5

Si p_1 est d'indice (i, j), alors p_9 est d'indice (i-1, j-1), p_2 d'indice (i-1, j), p_3 d'indice (i-1, j+1), p_8 d'indice (i, j-1), p_4 d'indice (i, j+1), p_7 d'indice (i+1, j-1), p_6 d'indice (i+1, j),et p_5 d'indice (i+1,j+1).

$S(p_1)$ = nombre de transitions $0-1$ dans la séquence $p_2, p_3, \ldots, p_9$.

Par exemple, pour le point p1 ci-après on a : $N(p_1)=4$ et $S(p_1)=3$ dans la matrice suivante.

0	**0**	**1**
1	**p_1**	**0**
1	**0**	**1**

Après l'application de l'étape 1 pour chaque point dans la région binaire, on obtient une image dans laquelle on a effacé tous les points qui vérifient les conditions **a**, **b**, **c**, **d**. Il faut noter que dans cette étape on n'efface un point qu'après le balayage de toute l'image.

2ème 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 4 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 :

1- appliquer l'étape 1 à chaque point de l'image.

2- effacer tout point vérifiant les 4 conditions **a**, **b**, **c**, **d**.

3- appliquer l'étape 2 à l'image obtenue suite à l'étape 1.

4- effacer tout point vérifiant les 4 conditions **a1**, **b1**, **c1**, **d1**.

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

Voici la représentation algorithmique de cette méthode de squelettisation :

1- Soit x une matrice représentant l'image originale de dimension (32×32).

a : sera une matrice de dimension (32×32).

b : sera une matrice de dimension (32×32).

NN = 0 (NN représente $N(p_1)$)

S = 0 (S représente $S(p_1)$)

cond = false (cond est une variable qui sera vraie si l'algorithme est fini)

2-

Tantque (cond = false) **faire**

a = x ;

(étape 1)

Pour i = 1 à 32 **faire**

Pour j = 1 à 32 **faire**

NN=0 ; S=0 ;

Si x(i, j) =1 **alors**

NN = x(i-1,j) + x(i-1,j+1) + x(i,j+1) + x(i+1,j+1) + x(i+1,j) + x(i+1,j-1) + x(i,j-1) + x(i-1,j-1);

Si x(i-1,j)=0 et x(i-1,j+1)=1 **alors** S=S+1 ; **finsi**

Si x(i-1,j+1)=0 et x(i,j+1)=1 **alors** S=S+1 ; **finsi**

Si x(i,j+1)=0 et x(i+1,j+1)=1 **alors** S=S+1 ; finsi

Si x(i+1,j+1)=0 et x(i+1,j)=1 **alors** S=S+1 ; **finsi**

Si x(i+1,j)=0 et x(i+1,j-1)=1 **alors** S=S+1 ; **finsi**

Si x(i+1,j-1)=0 et x(i,j-1)=1 **alors** S=S+1 ; **finsi**

Si x(i,j-1)=0 et x(i-1,j-1)=1 **alors** S=S+1 ; **finsi**

Si x(i-1,j-1)=0 et x(i-1,j)=1 **alors** S=S+1 ; **finsi**

finsi

Si NN ≥ 2 **et** NN ≤ 6 **et** S = 1 **et** x(i-1,j)*x(i,j+1)*x(i+1,j)=0 **et** x(i,j+1)*x(i+1,j)*x(i,j-1)=0 **alors**

a(i,j) = 0 ;

finsi

finpour

fnipour

b = a ;

(étape 2)

Pour i = 1 à 32 **faire**

Pour j = 1 à 32 **faire**

NN=0 ; S=0 ;

Si a(i,j) =1 **alors**

NN= a(i-1,j) + a(i-1,j+1) + a(i,j+1) + a(i+1,j+1) + a(i+1,j) + a(i+1,j-1) + a(i,j-1) + a(i-1,j-1) ;

Si a(i-1,j)=0 et a(i-1,j+1)=1 **alors** S=S+1 **finsi**

Si a(i-1,j+1)=0 et a(i,j+1)=1 **alors** S=S+1 **finsi**

Si a(i,j+1)=0 et a(i+1,j+1)=1 **alors** S=S+1 **finsi**

Si a(i+1,j+1)=0 et a(i+1,j)=1 **alors** S=S+1 **finsi**

Si a(i+1,j)=0 et a(i+1,j-1)=1 **alors** S=S+1 **finsi**

Si a(i+1,j-1)=0 et a(i,j-1)=1 **alors** S=S+1 **finsi**

Si a(i,j-1)=0 et a(i-1,j-1)=1 **alors** S=S+1 **finsi**

Si a(i-1,j-1)=0 et a(i-1,j)=1 **alors** S=S+1 **finsi**

finsi

Si NN ≥ 2 **et** NN ≤ 6 **et** S = 1 et a(i-1,j)*a(i,j+1)*a(i+1,j)=0 **et** a(i,j+1)*a(i+1,j)*a(i,j-1)=0 **alors**

b(i,j) = 0;

finsi

finpour

finpour

(test de changement de l'image)

cond = true ;

Pour i = 1 à 32 **faire**

Pour j = 1à **faire**

Si x(i,j) ≠ b(i,j) **alors** cond = false **finsi**

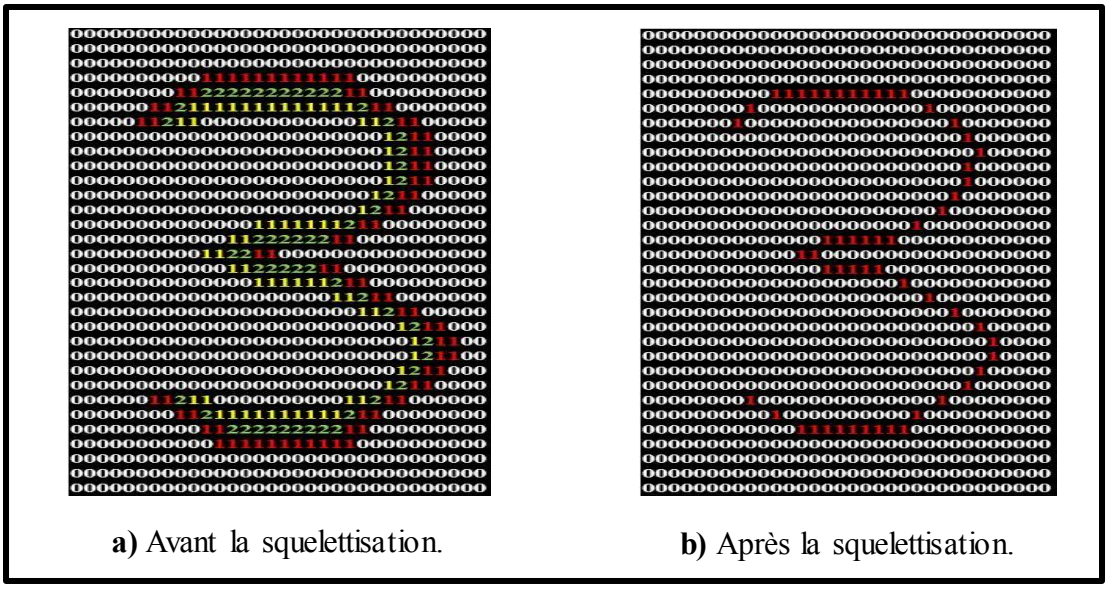
finpour

finpour

fintantque

La matrice b contient alors l'image après squelettisation.

Voici d'exemple illustrant l'application de cet algorithme.



a) Avant la squelettisation. **b)** Après la squelettisation.

Figure 3.5 : Le résultat de squelettisation

2.3. Segmentation

C'est la troisième étape du processus de reconnaissance de chiffres manuscrits. Dans cette phase on utilise la nuance en couleur - noir/blanc - de l'image épurée pour obtenir les différents segments composants le chiffre.

La figure 3.6 illustre cette tâche.

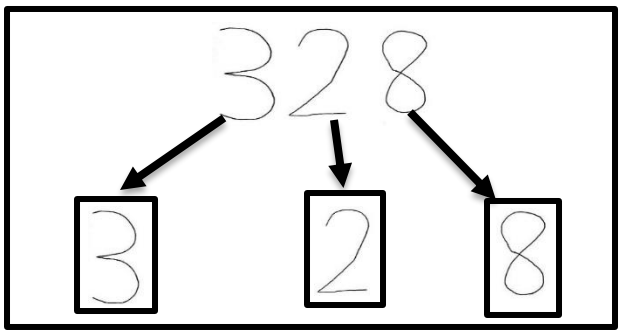


Figure 3.6 : résultat de segmentation de l'image du chiffre « 328 ».

Cette tâche est réalisée à l'aide des fonctions suivantes :

```
Fonction selection_petit_matrice (mat : vecteur[1..n,1..n]du entier, largeur, longueur :entier,) : vecteur[1..n,1..n]entier;
Var
    cord : vecteur[1..4]du entier ;
    er :entier ;
Debut
 Tantque (Line < NbLine) faire
        cord [0] = select_clonne1(mat, largeur, longueur, cord [1]);
        cord [1] = select_clonne2(mat, largeur, longueur, cord [0]);
        cord [2] = select_line1(mat, largeur, longueur, cord [0], cord [1]);
        cord [3] = select_line2(mat, largeur, longueur, cord [0], cord [1]);
        er = cord [1];
  FinTantque
 selection_petit_matrice ← cord ;
Fin.
```

```
Fonction select_colone1 (matrice [,][,] :entier ; NbLine, NbColonne :entier):entier;
Var
    Line, Colonne, Valeur_retour : entier;
    S: booleén;
Debut
 S ← vrai;
 Line ← 0 ;
 Colonne ← 0 ;
 Tantque (Line < NbLine) faire
  Tantque (Colonne <NbColonne) faire
  Si (matrice [Line, Colonne]=1)
   Valeur_retour ← Line;
   S ← false;
   Break ; // stop la boucle
  Finsi
  Colonne ← Colonne+1 ;
  FinTantque
  Colonne ← 0 ;
  Si (S=false) alors
   Break ;
  Finsi
  Line ← Line+1 ;
 FinTantque
 select_colone1 ← Valeur_retour ;
Fin.
```

```
Fonction select_colone2 (matrice [,][,] :entier ; NbLine, NbColonne, debuut :entier):entier;
Var
    Line, Colonne, Valeur_retour : entier;
Debut
 Line ← 0 ;
 Colonne ← debuut;
  Tantque (Colonne < NbLine) faire
   Si (matrice [Colonne, Line]=1)
     Colonne ← Colonne+1 ;
     Line ← 0;
   Sinon
     Line ← Line +1 ;
     Si (Line= NbColonne) alors
       Valeur_retour ← Colonne;
       Break ;
     Finsi
   Finsi
  FinTantque
 select_colone2 ← Valeur_retour ;
Fin.
```

```
Fonction select_line1 (matrice [,][,] :entier ; NbLine, interval1, interval2 :entier):entier;
Var
    Line, Colonne, Valeur_retour : entier;
    S: booleén;
Debut
 S ← vrai;
 Line ← 0 ;
 Colonne ← interval1 ;
 Tantque (Line < NbLine) faire
  Tantque (Colonne <interval2) faire
   Si (matrice [Colonne, Line]=1)
    Valeur_retour ← Line;
    S ← false;
    Break; // stop la boucle
   Finsi
   Colonne ← Colonne+1 ;
  FinTantque
  Colonne ← interval1 ;
  Si (S=false) alors
   Break ;
  Finsi
  Line ← Line+1 ;
 FinTantque
 select_line1 ← Valeur_retour ;
Fin.
```

```
Fonction select_line2 (matrice [,][,] :entier ; NbColonne, interval1, interval2 :entier):entier;
Var
    Line, Colonne, Valeur_retour : entier;
    S: booleén;
Debut
 S ← vrai;
 Line ← NbColonne - 1;
 Colonne ← interval1 ;
 Tantque (Line >= 0) faire
  Tantque (Colonne <interval2) faire
   Si (matrice [Colonne, Line]=1)
    Valeur_retour ← Line;
    S ← false;
    Break; // stop la boucle
   Finsi
   Colonne ← Colonne+1 ;
  FinTantque
  Colonne ← interval1 ;
  Si (S=false) alors
   Break ;
  Finsi
  Line ← Line - 1 ;
 FinTantque
 select_line2 ← Valeur_retour ;
Fin.
```

Exemple

La figure (3.7) illustre l'étape de segmentation

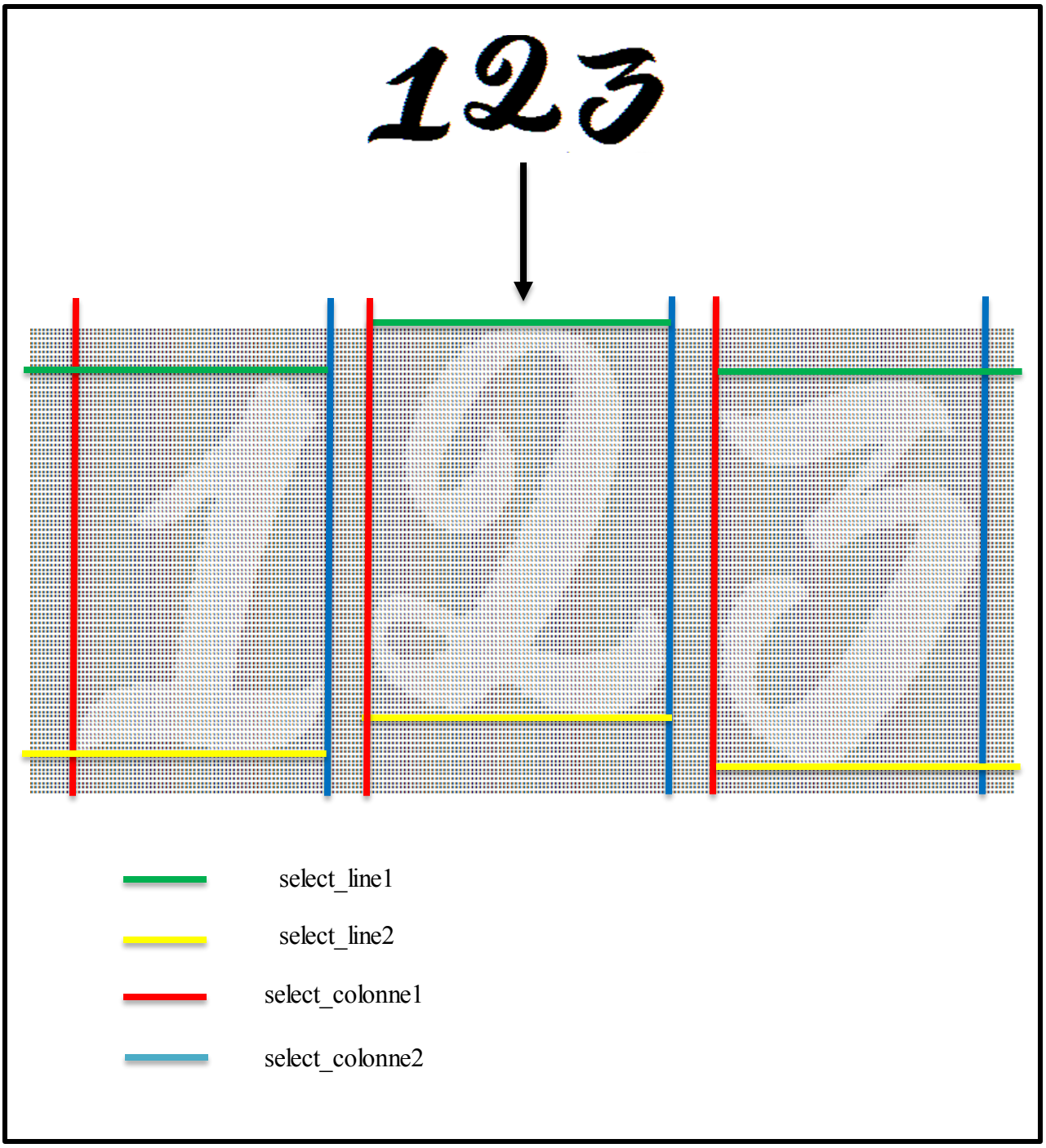


Figure 3.7 : Etape de segmentation de du chiffre « 123 ».

2.4. Extraction des caractéristiques

L'extraction des caractéristiques consiste à utiliser une technique d'analyse (statistique, structurelle et globale ou local,…etc.) pour obtenir les caractéristiques qui donnent une bonne description de l'image de chiffres. Pour ce faire, Il existe une diversité des méthodes citer dans chapitre 1 mais dans ce système nous avons utilisé une technique d'extraction globale "Descripteur couleur exactement Cohérence spatiale" qui est très efficace et flexible et donne bon représentation de notre type d'image, notre descripteur est considère comme concaténation de deux histogramme (histogramme vertical et horizontale).

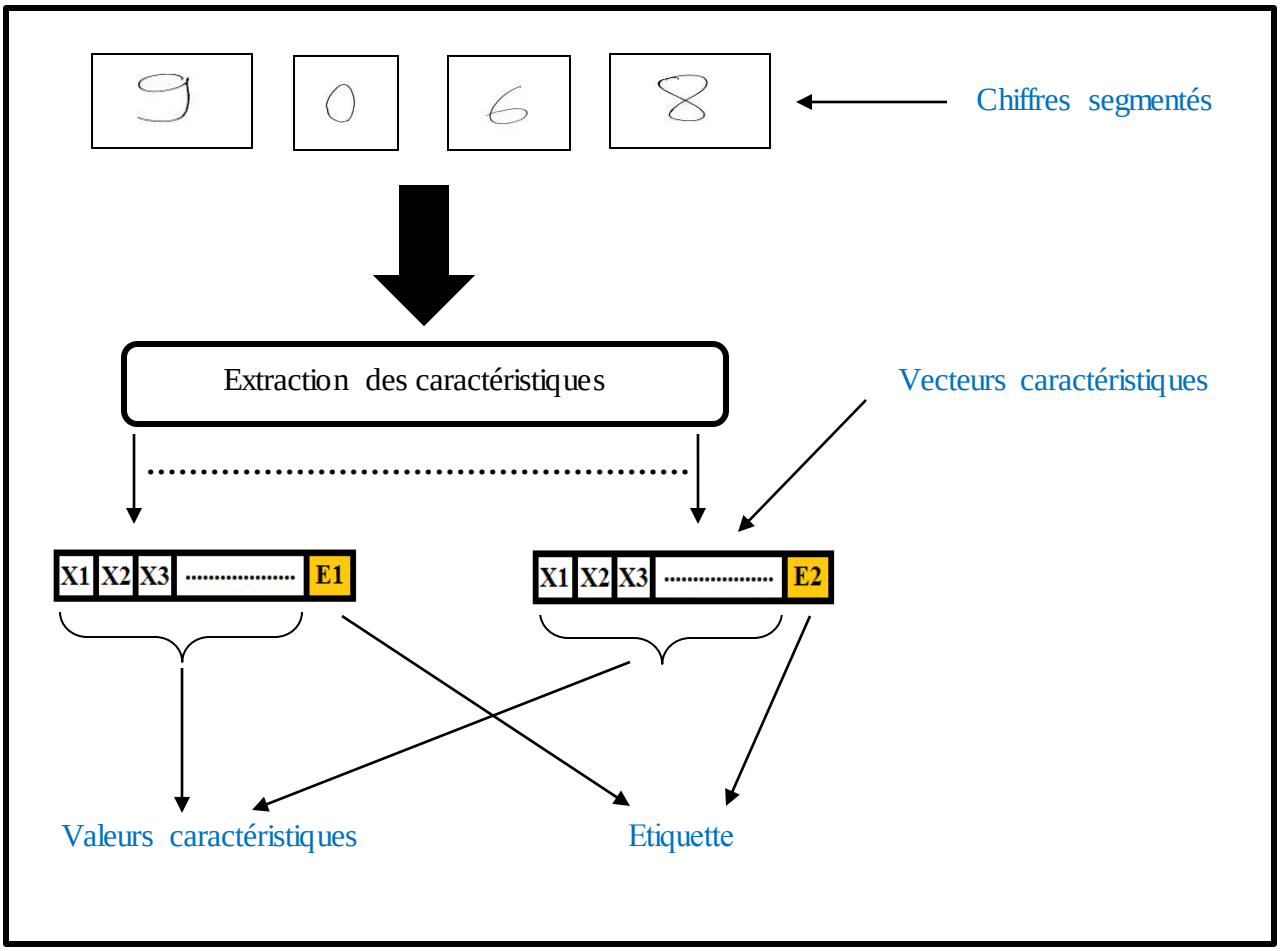


Figure 3.8 : Illustration d'une extraction des caractéristiques.

La technique utilisée pour extraire les vecteurs de caractéristiques sont :

Histogramme latéral : est une représentation du nombre de valeurs égales à (1) dans chaque ligne des lignes de la matrice d'image.

Algorithme d'extraction de caractéristiques

```
Fonction extract_vecteur (matrice [,] : entier) : matrice [,] ;
Var inpu : vecteur [1..32] entier ;
Debut
  Pour (i=1 à 32) faire
     Pour (j=1 à 32) faire
       Si (matrice [i,j]==1) alors
            inpu [i] ← inpu [i] +1 ;
       Finsi
     FinPour
     Pour (k=1 à 32) faire
       Si (matrice [k, i]==1) alors
           inpu [i+32] ← inpu [i+32] +1;
       Finsi
     FinPour
  FinPour
  extract_vecteur ← inpu;
Fin.
```

L'exemple suivant illustrée l'extraction des caractéristiques.

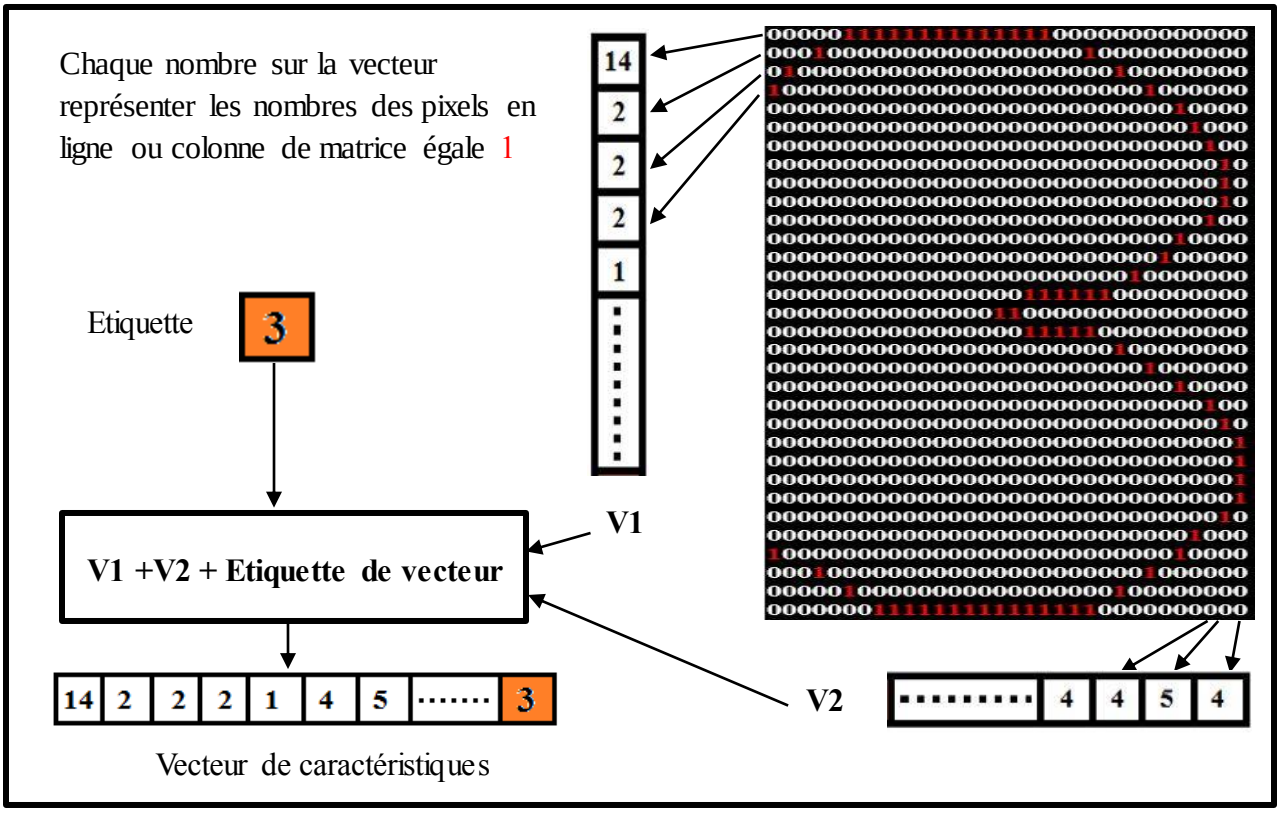


Figure 3.9 : Illustration plus détaille d'une extraction des caractéristiques.

2.5. Classification

Cette phase consiste à utiliser une méthode de classification (dans notre cas SVM). Elle se divise en deux sous phases: apprentissage et test ou décision. La première consiste à initialiser la base des modèles, autant que la deuxième consiste à assigner une classe pour chaque nouveau exemple donnée (vecteur caractéristique).

Classifieur SVM

Un classifieur SVM est chargé de prendre le vecteur de caractéristiques (Taille, nb_Pixels, Etiquette …etc) généré par le module précédent comme une donnée d'entrée, rechercher un hyperplan séparateur qui sépare les exemples dans la phase d'apprentissage et faire une décision de classification dans la phase d'identification (voir la figure 3.10).

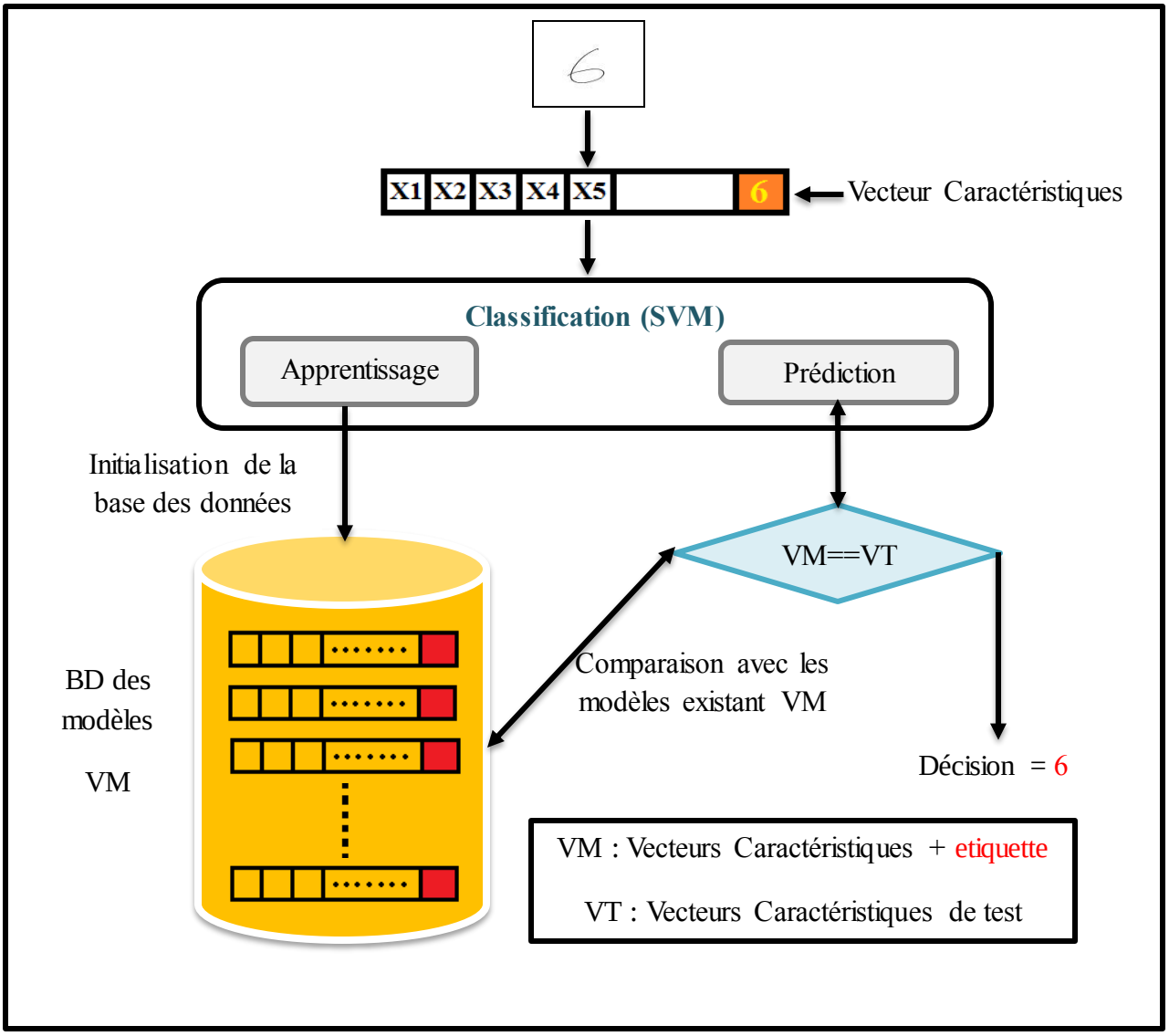


Figure 3.10 : Illustration d'une Phase de classification.

Dans le module SVM, il y a deux phases : une pour l'apprentissage et l'autre pour la classification.

2.5.1. Apprentissage

Cette phase consiste à initialiser ou créer la base des modèles en sauvegardant les caractéristiques des différents chiffres.

2.5.2. Prédiction

Elle consiste à utiliser les caractéristiques extraites dans la phase précédente pour attribuer une classe en se basant sur les données de la base des modèles.

La figure (3.11) représente la relation entre ces deux phases.

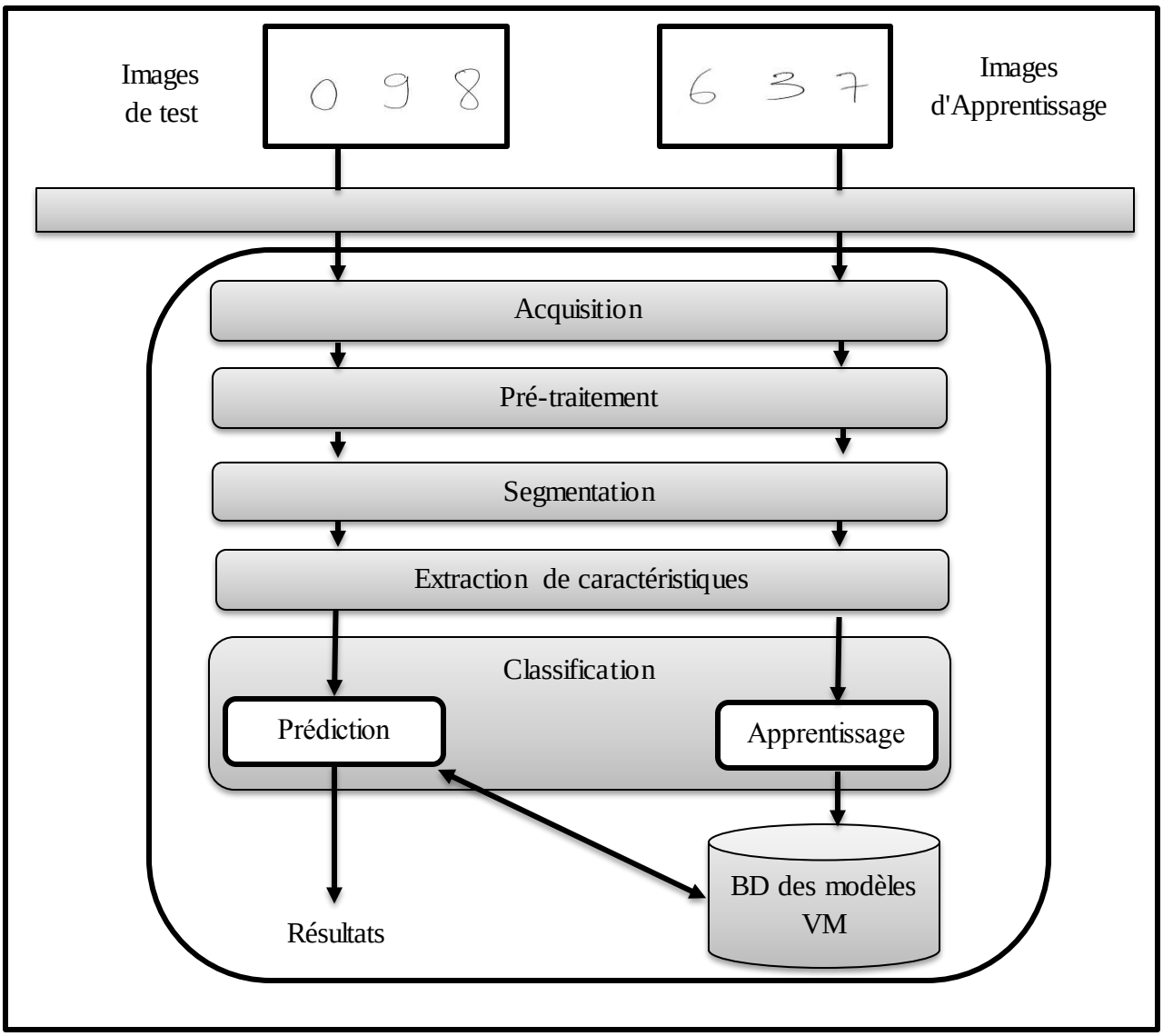


Figure 3.11 : Illustration des deux phases utilisées d'un classifieur SVM.

3. l'implémentation

3.1. Environnement de travail

3.1.1. Langage de programmation

Visual Studio 2015 est un environnement de programmation riche en outils comportant toutes les fonctionnalités nécessaires pour créer des projets C# de toute taille.

Le langage star de la nouvelle version de Visual Studio et de l'architecture .NET est C#, un langage dérivé du C++. Il reprend certaines caractéristiques des langages apparus ces dernières années et en particulier de Java. C# peut être utilisé pour créer, avec une facilité incomparable, des applications Windows et Web. C# devient le langage de prédilection d'ASP.NET qui permet de créer des pages Web dynamiques avec programmation côté serveur. [GL, 2006]

Les points forts de C#

- libération automatique des objets.
- Le compilateur C#, comme Java, produit un p-code (bytecode) MSIL, Microsoft Intermediate Language, destiné à fonctionner dans une machine virtuelle (managed execution environment).
- C# a une syntaxe plus familière pour les programmeurs C, C++, Java et Perl.
- La syntaxe élégante et économique est empruntée à C et C++. - Grandes capacités d'introspection de classe (réfection).

La plateforme WPF

Windows Presentation Foundation (WPF) (nom de code Avalon) est la spécification graphique de Microsoft .NET. Il intègre le langage descriptif XAML qui permet de l'utiliser d'une manière proche d'une page HTML pour les développeurs.

3.1.2 NET Framework

Les programmes en C# s'exécutent sur le .NET Framework, composant intégral de Windows qui inclut un système d'exécution virtuel appelé Common Language Runtime (CLR) et un jeu unifié de bibliothèques de classes. Le CLR est l'implémentation commerciale de l'infrastructure du langage commun (CLI) de Microsoft, norme internationale constituant la base de toute création d'environnements d'exécution et de développement et assurant le fonctionnement homogène des langages et des bibliothèques. [Msd, 2015]

3.1.3 Microsoft Visual Studio.Net

Visual Studio .NET est un jeu complet d'outils de développement permettant de générer des applications Web ASP, des services Web XML, des applications bureautiques et des applications mobiles. Visual Basic .NET, Visual C++ .NET, Visual C# .NET et Visual J# .NET utilisent tous le même environnement de développement intégré (IDE, integrated development environment), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du .NET Framework, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de services Web XML. [Ms, 2015]

3.2. Présentation de l'application

Dans cette partie, nous allons présenter l'interface de notre application et quelques résultats de l'étape de reconnaissance de chiffres manuscrits.

3.2.1. Interface d'apprentissage

Tout d'abord, l'utilisateur de notre application sur l'interface d'apprentissage doit trouverez deux fenêtres :

- *Education form the database* : Pour afficher la fenêtre d'apprentissage à partir de notre base de données.
- *Manual education* : Pour afficher la fenêtre d'éducation manuellement par l'écriture.

Apprentissage à partir de la base de données

L'utilisateur de notre application doit cliquer sur le bouton *Start training* pour faire apprendre la machine à partir de la base de données, la figure (3.12) illustre l'interface d'apprentissage à partir de BD (modelé VM) de notre application.

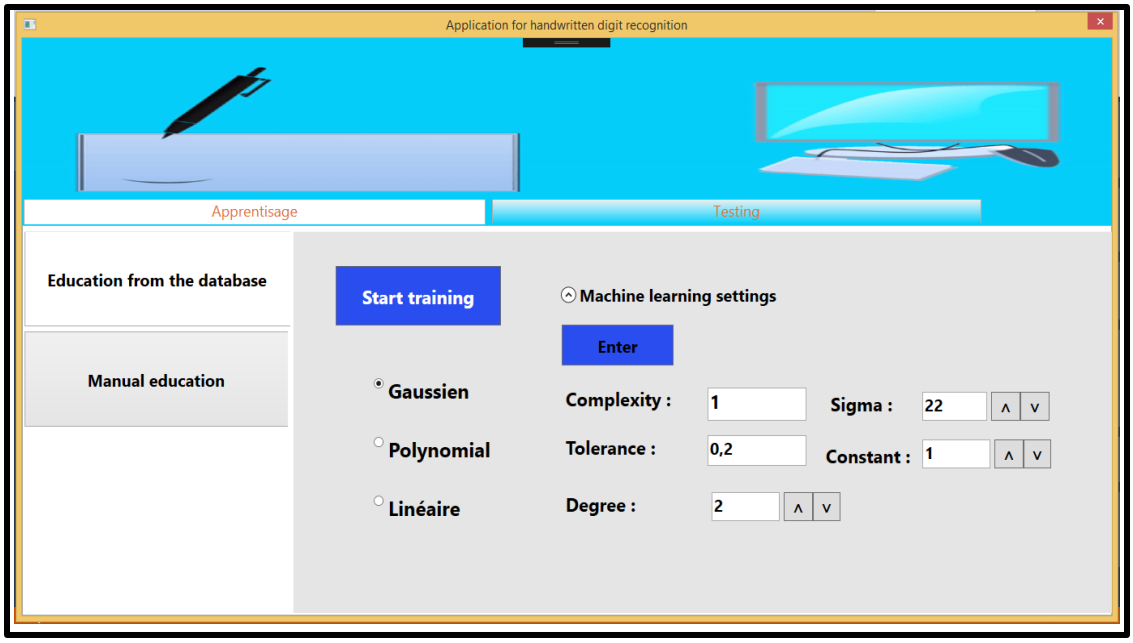


Figure 3.12 : L'interface d'apprentissage de notre application.

La fenêtre contient les boutons suivants :

- *Start training* : pour le commencer d'apprentissage à partir de notre base de données.
- *Les boutons radio (Gaussien, Polynomial, Linéaire) :* pour choisir le type de SVM noyaux.
- *Enter* : Pour entrer les valeurs de machine SVM (bouton pour les développeurs).

Écriture manuelle pour l'apprentissage

La figure (3.13) illustre l'opération :

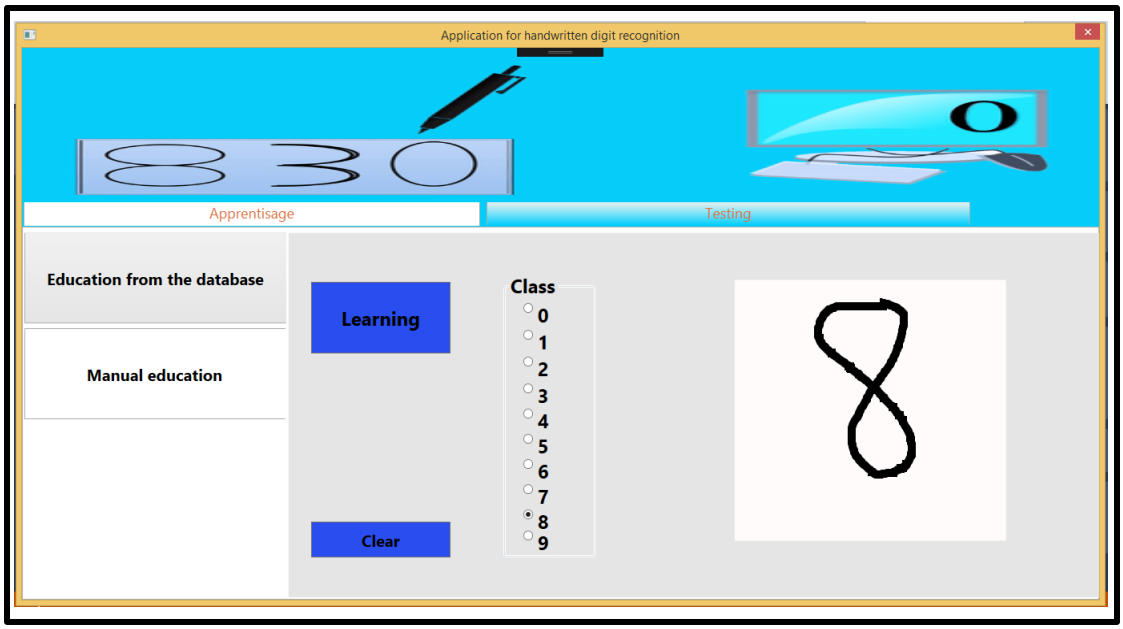


Figure 3.13 : L'interface d'apprentissage manuelle.

Cette fenêtre se compose de deux boutons :

- *Learning* : Pour éducation la machine à partir le panneau d'écriture.
- *Clear* : Pour effacer le panneau d'écriture.

3.2.2. Interface reconnaissance

L'utilisateur peut choisir :

- *Image recognition* : Pour afficher la fenêtre de reconnaissance par charger une image.
- *Writing recognition* : Pour afficher la fenêtre de reconnaissance pour l'écriture manuelle.

La reconnaissance en chargeant une image

La figure (3.14) illustre La reconnaissance avec chargement d'une image.

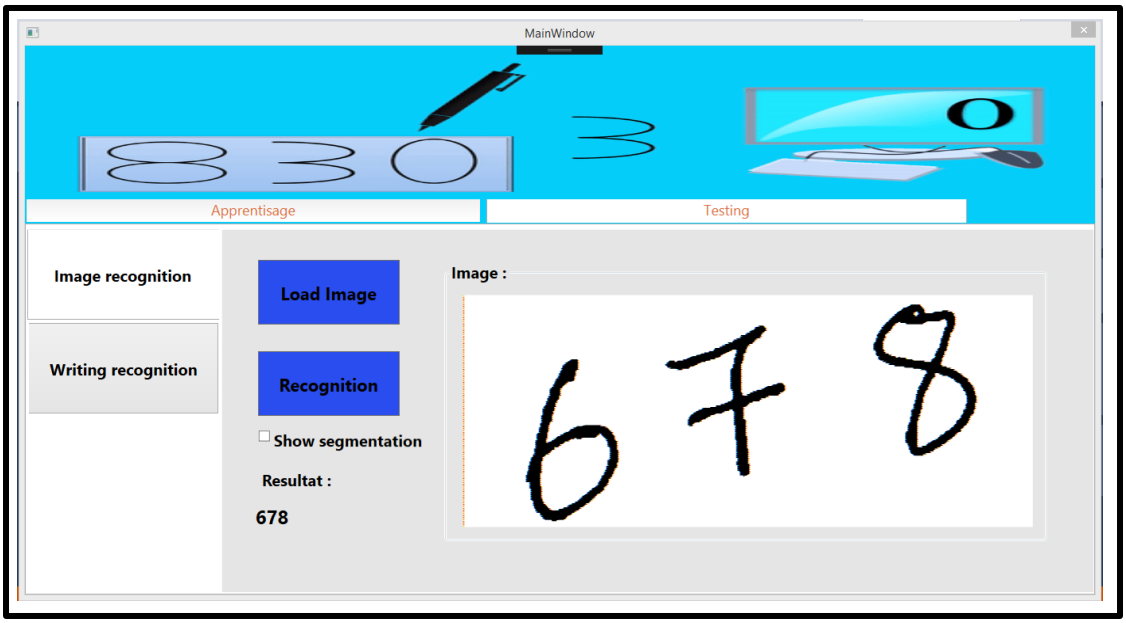


Figure 3.14 : L'interface de reconnaissance par charger une image.

Dans cette fenêtre, on trouve deux boutons :

- *Load image* : Pour charger une image à partir l'ordinateur.
- *Recognition* : Pour commencer le processus de reconnaissance.

La reconnaissance par l'écriture sur le panneau

La figure (3.15) illustre La reconnaissance par l'écriture directe sur le panneau.

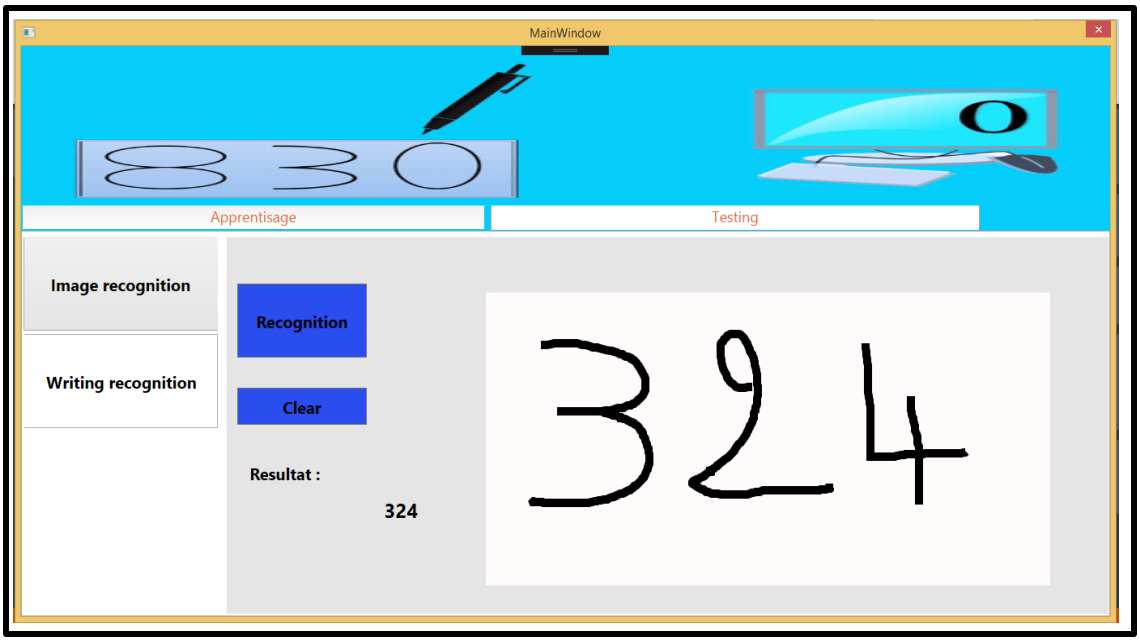


Figure 3.15 : L'interface de reconnaissance par l'écriture sur le panneau.

Dans cette fenêtre, on trouve deux boutons :

- *Recognition* : Pour commencer le processus de reconnaissance.
- *Clear* : Pour effacer le panneau d'écriture.

3.3. Test et Résultats

Nous avons choisis quelques exemples de chiffres manuscrits. Ces exemples sont scannés à l'aide d'une imprimante-scanner CANON MF4730 et avec une résolution de 32*32 pixels dans notre application.

En notre application nous avons utilisé trois types de noyaux : linéaire, gaussien et polynomial. En comparant leurs résultats les uns aux autres en reconnaissance de chiffres manuscrits. Les trios noyaux implémenté ont été entraînés sur une base d'apprentissage (1000 échantillons) et testées sur une base de tests (50 échantillons) en utilisant les mêmes vecteurs de caractéristiques.

Le tableau suivant illustre le test et le résultat de chaque chiffre par le noyau linéaire.

Chiffre	Nombre exemples de test	Taux de reconnaissance
0	50	81 %
1	50	85 %
2	50	84 %
3	50	82 %
4	50	82 %
5	50	74 %
6	50	78 %
7	50	80 %
8	50	76 %
9	50	84 %

Tableau 3.1 : Résultats de test de noyau linéaire.

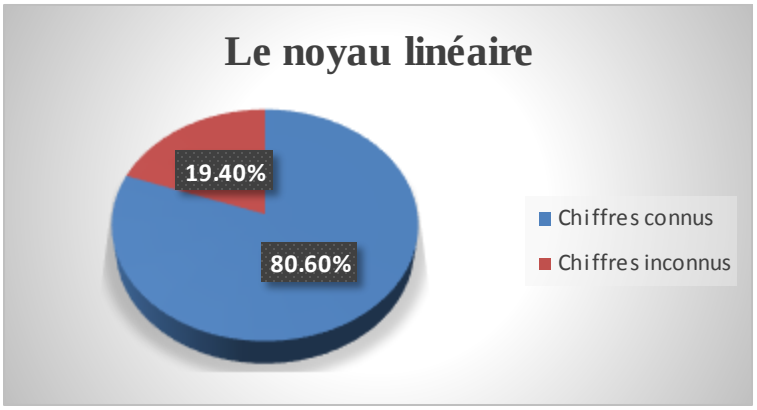


Figure 3.16 : Taux de reconnaissance de noyau linéaire.

Le tableau suivant illustre le test et le résultat de chaque chiffre par le noyau gaussien.

Chiffres	Nombre Exemples de test	Taux de reconnaissance
0	50	95 %
1	50	93 %
2	50	96 %
3	50	90 %
4	50	99 %
5	50	90 %
6	50	88 %
7	50	94 %
8	50	90 %
9	50	98 %

Tableau 3.2 : Résultats de test de noyau gaussien.

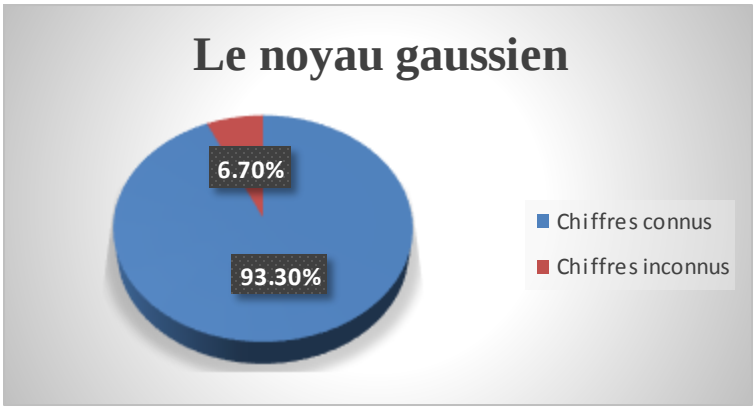


Figure 3.17 : Taux de reconnaissance de noyau gaussien.

Le tableau suivant illustre le test et le résultat de chaque chiffre par le noyau polynomial.

Chiffres	Nombre Exemples de test	Taux de reconnaissance
0	50	94 %
1	50	91 %
2	50	95 %
3	50	90 %
4	50	95 %
5	50	90 %
6	50	81 %
7	50	90 %
8	50	88 %
9	50	94 %

Tableau 3.3 : Résultats de test de noyau polynomial.

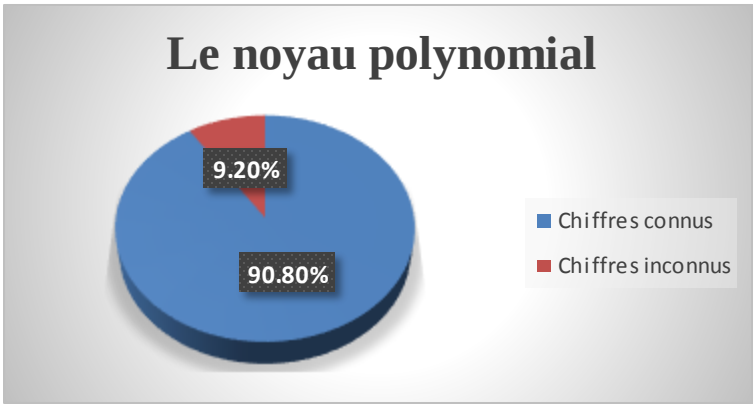


Figure 3.18 : Taux de reconnaissance de noyau polynomial

Comparaison des résultats

Nous présentons (tableau 3.4) les résultats obtenues avec les meilleures configurations trouvées à la fois pour les trois de types de noyaux : linéaire, gaussien et polynomial. Où : les paramètres C,D et T désignent respectivement la complexité , dégré et la tolérance , pour les trois noyaux de classifieur SVM.

Réseau	Paramètres	Taux de reconnaissance
Le noyau linéaire	C=1, D=0.2 et T=500	80.60 %
Le noyau gaussien	C=1, D=0.2 et T=500	93.70 %
Le noyau polynomial	C=1, D=0.2 et T=500	90.80 %

Tableau 3.4 : Performances des trois noyaux.

Les meilleurs résultats obtenus sont de 80.60 % pour le noyau linéaire, de 93.70 % pour le noyau gaussien et de 90.80 % pour le noyau polynomial.

On remarque en général que le noyau gaussien donne de meilleurs résultats dans le taux de reconnaissance.

L'influence de la taille de la base d'apprentissage sur le classifieur permet de formuler deux idées:

- L'augmentation de la taille de la base d'apprentissage permet de diminuer l'erreur tout en augmentant le taux de reconnaissance.
- Le processus d'apprentissage est donc efficace.

4. Conclusion

Nous avons présenté dans ce chapitre les différentes étapes qui peuvent conduire à une conception convenable d'un système de reconnaissance de chiffres manuscrits, nous avons choisi pour l'extraction des caractéristiques une méthode hybride dont on combine les vecteurs de descripteur statistique et la méthode SVM pour la reconnaissance des différentes classes.

Ensuite nous avons cité les outils qui nous avons utilisé pour développer notre application ensuite, nous avons présenté leurs interfaces et quelques résultats expérimentaux concernant les techniques l'apprentissage et la reconnaissance de chiffres manuscrits.

Conclusion générale

Jusqu'aujourd'hui, La reconnaissance de l'écriture et la reconnaissance de chiffres manuscrits en particulier présente un défi très grand, malgré les efforts et les travaux intensifs réalisés dans ce domaine, aucun système de reconnaissance de chiffres manuscrits n'est jugé fiable à 100%, Mais au fur et à mesure les autre essayent d'améliorer les scores pour de meilleurs résultats. Et elle joue un rôle très important dans le monde actuel. Elle est capable de résoudre des problèmes complexes et rendre les activités de l'homme plus simple.

L'objectif de ce travail est l'étude et l'implémentation des machines à vecteurs de support ou SVMs (Support Vector Machines) pour la reconnaissance des chiffres manuscrits.

En général, la plupart des méthodes d'apprentissage machine comme réseau de neurones possèdent un grand nombre de paramètres d'apprentissage à fixer par l'utilisateur. En contrepartie, la formulation élégante de *SVM* laisse très peu de place aux paramètres utilisateurs.

La méthode des SVM que nous avons implémentée a été testée sur des chiffres manuscrits d'une base avec 1000 chiffres pour l'apprentissage et 50 chiffres pour le test en utilisant trois noyaux de cette méthode de classification : le noyau linéaire, le noyau gaussien et le noyau polynomial.

Les résultats obtenus montrent que le noyau gaussien permet de fournir des meilleurs résultats comparativement aux autres noyaux.

Nous pouvons envisager plusieurs perspectives possibles pour étendre ce travail :

- L'évaluation sur d'autres bases de données ainsi que sur d'autres types de noyaux.
- Les méthodes d'extraction des caractéristiques peuvent être envisagées (par exemple en utilisant les caractéristiques qui contiennent les paramètres structurels extraits du squelette de la forme).
- Ajout d'une deuxième segmentation pour faire une analyse fine de l'image déterminée à l'issue de la segmentation primaire, afin d'y détecter la présence éventuelle de plusieurs chiffres se chevauchant ou connectés entre eux.

Bibliographie

[AY, 1992] A. Belaid, et Y. Brlaid, 1992. << *Reconnaissance des formes. Méthodes et application* >>; P 14 - 17.

[BD, 1986] Bernsen. Dynamic thresholding of gray-level images. Proc. 8th Int. Conf. Pattern Recognition Paris, pages 1251–1255, 1986.

[BR, 2012] BEY. R : « *Une segmentation hybride d'images IRM cérébrales par combinaison de l'algorithme K-Means et Quad-Tree Génétique* », Mémoire pour l'obtention du MASTER EN INFORMATIQU, 2012.

[BSAS, 2002] Bernhard Scholkopf, Alexander J. Smola "*Learning with Kernels, Support Vector Machines, Regularization, Optimization, and Beyond*", the MIT Press 2002.

[FM, 2008] F. Menasri : «*Segmentation d'image Application aux documents anciens* », Thèse Docteur de l'Université Paris Descartes en Informatique, France, Juin 2008.

[GGSG, 1995] Congedo, G., Dimauro, G., Impedovo, S. and Pirlo, G., 1995. Segmentation of Numeric Strings, Proc. of Third Int. Conf. on Document Analysis and Recognition, 14-15 Aout ,Montrea l, pp.1038-1041.

[GL, 2006] : Gérard Leblanc, Livre C# et .NET version 2, Edition EYROLLES, 2006.

[Ha, 2005] Hanoi, 15 juillet 2005. *Réseaux de neurones pour la reconnaissance des formes*.

[HB, 2006] Mohamadally .H, Fomani.B : « *SVM machine à vecteurs de support ou séparateur à vaste marge* », BD Web, ISTY3, Versailles St Quentin, France, janvier 2006.

[JC, 2002] J. Callut, 2002 <<*Implémentation efficace des Support Victor Machines pour la classification*». Mémoire présenté en vue de l'obtention du grade de Maître en Informatique.

[LR, 2001] L. Robadey :" 2 (CREM):Une méthode de reconnaissance structurelle de documents complexes basée sur des patterns bidimensionnels ". Thèse de doctorat soumise à la Faculté des Sciences de l'Université de Fribourg, Suisse, 2001.

[LRFCT, 2002] Oliveira, L. S., Sabourin, R., Bortolozzi, F., Suen, C. Y., 2002. Automatic Recognition of Handwritten Numerical Strings: A Recognition and Verification Strategy, IEEE Transactions on Pattern Analysis and Machine Intelligence – PAMI. Vol. 24, 1438–1454.

[MNCC, 2007] Mohammed Cheriet, Nawwaf Kharma, Cheng Lin Liu, and Ching Suen. Character Recognition Systems: A Guide for Students and Practitioners. Wiley-Interscience, 2007.

[Ms, 2015] https://msdn.microsoft.com/fr-fr/library/aa291755(v=vs.71).aspx

[Msd, 2015] https://msdn.microsoft.com/fr-fr/library/z1zx9t92.aspx

[NO, 1979] N. Otsu. A threshold selection method from gray-level histograms. In IEEE Trans. System, Man Cybernetics9, pages 62–66, 1979.

[OB, 2001] O. Bousquet : « *Introduction aux ' Support Vector Machines '(SVM)* », Centre de Mathématiques Appliquées Ecole Polytechnique, Palaiseau, Orsay, Novembre 2001.

[PM, 2003] P. Mahé : «*Noyaux pour graphes et Support Vector Machines pour le criblage virtuel de molécules* », Rapport de stage, DEA MVA 2002/2003, Septembre 2003.

[PMLA, 2003] P. Mahé, L. Ait-Ali : « *Projet d'apprentissage statistique SVM pour l'apprentissage non supervisé*», DEA MVA, Février 2003.

[PV, 2003] P. Vincent : «*Modèles à noyaux à structure locale* », Thèse présentée à la Faculté des études supérieures en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.) en informatique, Université de Montréal, Octobre 2003.

[SA, 2008] Shubair A et al.:" *Off-line Arabic handwritten word segmentation using rotational invariant segments features* ". The international Arab journal of information technology, Vol. 5, No. 2, April 2008.

[SC, 2005] S. Chevalier et al : « *Étude de primitives spectrales pour la reconnaissance de caractères manuscrits dans le cadre d'une approche markovienne 2D* », DGA/Centre d'Expertise Parisien, France, Novembre 2005.

[SH, 2007] S. Haitaamar :" *segmentation de texte en caractère pour la reconnaissance optique de l'écriture arabe*". Université EL-HADJ LAKHDHAR Batna, Juillet 2007.

[TT, 1995] Oivind Due Trier and Torfinn Taxt. Evaluation of binarization methods for document images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(3) :312–315, 1995.

[TW, 2012] TOURQUI.W : « *Recherche d'images par le contenu symbolique (approche statistique)* », Mémoire pour l'obtention du MASTER EN INFORMATIQUE, 2012.

[VK, 2001] Vojislav Kecman, "*Learning and Soft Computing Support Vector Machines, Neural Networks, and Fuzzy Logic Models*", the MIT Press 2001.

[VV, 1995] V.Vapnik, 1995 .*the nature of statistical learning throry*. springer_verlag, New_York, USA. V. K. Govindan, 1990. << *Character recognition - A review* >>; Department of Electrical communication engineering, India; CNED'94; P 671-672.