

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et de la
Recherche Scientifique
Université Ahmed Draia – Adrar
Faculté des Sciences et de la Technologie
Département des Mathématiques et Informatique



Mémoire de fin d'étude, en vue de l'obtention du diplôme de Master en informatique

Option : Système d'Intelligents.

Thème

Lecture Automatique de Plaques d'Immatriculation par les Machines à Vecteurs de Support

Préparé par

GADOUI Siham et KEBIR Fatima

Devant la commission d'examen composée de

- | | | |
|--------------------------|-----|------------|
| • Mr. Abdelouahab OUAHAB | MAA | Président |
| • Dr. El Mamoun MAMOUNI | MCA | Rapporteur |
| • Mr. MEDIANI | MAA | Examineur |

Année universitaire 2019-2020

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
الْحَمْدُ لِلَّهِ الَّذِي
خَلَقَ الْمَوَدَّعَةَ
الْحَمْدُ لِلَّهِ الَّذِي
خَلَقَ الْمَوَدَّعَةَ

Résumé

Ce projet consiste à réaliser un système de détection et de reconnaissance des plaques d'immatriculation de véhicules algériens. Pour cela nous avons un système complet de localisation, de segmentation et de reconnaissance des chiffres que nous avons appelé LAPI.

Le système LAPI a été développé en passant par trois phases. Tout d'abord un prétraitement a été effectué pour filtrer et réduire le bruit de l'image originale. Ensuite une phase de détection de contour sur l'image obtenue en utilisant le filtre Prewitt, et enfin nous appliquons une lecture OCR pour reconnaître les chiffres de la plaque d'immatriculation en utilisant les Machines à Vecteurs de Support (SVM) comme une méthode de classification, des résultats encourageants ont été obtenus.

Mots clés

Détection de contour, Machines à Vecteurs de Support (SVM), OCR, Segmentation, Lecture Automatique de Plaques d'Immatriculation (LAPI).

Abstract

Algerian vehicle license plates. For that we realized a state of the art and a preliminary study in order to gather the necessary information to make it possible to define the different treatments to be implemented in this system.

The system LAPI has been developed through three phases. Firstly a pretreatment was to filter and reduce the noise of the original image. Then a contour detection phase on the image obtained using segmentation, and finally we apply an OCR reading to recognize the character forms on the license plate using Support Vector Machines (SVM) as a classification method.

Key words

Edge detection, Support Vector Machines, OCR, Segmentation, Lecture Automatique de Plaques d'Immatriculation (LAPI), mathematic morphologies.

ملخص

يتضمن هذا المشروع إنشاء نظام للكشف عن لوحات ترخيص المركبات الجزائرية والتعرف عليها. من أجل ذلك قمنا بدراسة أولية من أجل جمع المعلومات اللازمة لجعل من الممكن تحديد التقنيات اللازمة التي سيتم تطبيقها في هذا النظام. تم تطوير نظام LAPI من خلال ثلاث مراحل. أولاً، كانت المعالجة المسبقة هي تصفية وتقليل ضوضاء الصورة الأصلية. ثم مرحلة الكشف عن الأشكال الهندسية في الصورة التي تم الحصول عليها باستخدام التجزئة، وأخيراً قمنا بتطبيق قراءة OCR للتعرف على أحرف لوحة الترخيص باستخدام Support Vector Machines (SVM) كطريقة تصنيف.

الكلمات المفتاحية

Détection de contour, Machines à Vecteurs de Support (SVM), OCR, Segmentation, Lecture Automatique de Plaques d'Immatriculation (LAPI), Morphologie mathématique.

Remerciements

Tous nos remerciements s'adressent tous d'abord à tout puissant ALLAH, d'avoir guidé nos pas vers le chemin du savoir.

La première personne que nous tient à remercier sont notre encadrant [Mr.MAMOUNI El Mamoun](#), pour l'orientation, la confiance, la patience qui m'a constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité.

Nous joignons à ces remerciements le président du jury Mr.OUAHAB et examinateur Mr.MEDIANI pour avoir accepté de faire partie du jury et pour leur lecture minutieuse et avisé des nombreuses pages qui suivent.

Nous tient à exprimer mes sincères remerciements à tous les professeurs qui m'ont enseignés et qui par leurs compétences nous avons soutenu dans la poursuite de mes études.

Enfin, nous remercions tous ceux qui, de près ou de loin, sont contribué à la réalisation de ce travail.

Dédicace

Le plus grand merci s'adresse au bon Dieu

Je dédie ce modeste travail à :

Ma mère, qui a œuvré pour ma réussite,

Mon père, qui peut être fier de trouver ici le résultat de longues années
de sacrifices,

Mes très chers frères et mes sœurs,

Toute la famille sans exception et tous mes amis

Mon binôme Siham,

Tous ceux qui ont contribué de près ou de loin pour que ce projet soit
possible,

Je vous dis merci.

Fatima

Dédicace

Je dédie ce modeste travail,

A mon père qui s'est tant donné et sacrifié,

A ma très chère mère qui m'a toujours soutenue,

A mes très chers frères,

A mes sœurs,

A mon binôme Fatima,

Je remercie tous ceux qui, de près ou de loin, sont contribué à la
réalisation de ce travail.

Je vous dis merci.

Siham

Table des matières

Résumé	III
Remerciements	V
Dédicace.....	VI
Liste des figures	XII
Liste des tableaux	XIV
Introduction générale	1

Chapitre 1: Reconnaissance de formes

1.Introduction.....	5
2.Reconnaissance des formes.....	5
2.1Intelligence Artificielle.....	5
2.2Définitions.....	6
3.Historique.....	7
4.Méthodes de reconnaissance de formes.....	8
5.Schéma général d'un système de Reconnaissance des formes	9
5.1Préparation des données	9
5.1.1Numérisation... ..	9
5.1.2Prétraitement.....	9
5.1.3Calcul des représentations.....	10
5.2Apprentissage.....	10
5.3Classification.....	10
5.4Post traitement.....	10
6.Applications typiques de la reconnaissance des formes	10
6.1Marketing.....	10
6.2Finances.....	11
6.3Usinage.....	11
6.4Energie.....	11
6.5Lecture automatisée.....	11
6.6Sécurité.....	11

7.Conclusion.....	11
-------------------	----

Chapitre 2: traitement d'images et segmentation

1.Introduction.....	14
2.Définition du traitement d'images.....	14
2.1Image.....	15
2.2Représentation d'image	15
2.2.1Image matricielle.....	16
2.2.2Image vectorielle	16
2.3Principe d'acquisition d'image.....	16
2.4Caractéristiques d'une image... ..	17
2.4.1Taille.....	17
2.4.2Résolution.....	17
2.4.3Bruit.....	17
2.4.4Contraste.....	17
2.4.5Histogramme.....	17
2.4.6Luminance.....	18
2.4.7Images à niveaux de gris.....	18
2.4.8Image en couleur.....	18
2.5Les principales étapes de traitement d'image.....	18
2.6Type de données manipulées... ..	19
2.7Les opérateurs de traitement d'image.....	19
2.7.1Filtres linéaires... ..	20
2.7.2Lisage.....	20
2.7.3Détection de contours.....	21
2.8Opérateurs morphologiques.....	23
2.8.1Conversion de l'image en niveau de gris	23
2.8.2L'érosion.....	23
2.8.3La dilatation.....	23
2.8.4L'ouverture.....	23
2.8.5La fermeture.....	24
3.Segmentation d'image	24

3.1	Définition de la segmentation.....	24
3.2	Le choix d'une technique de segmentation.....	26
3.3	Objectifs de la segmentation.....	26
3.4	Les stratégies de segmentation.....	26
3.4.1	Les stratégies scalaires.....	26
3.4.2	Les stratégies marginales.....	27
3.4.3	Les stratégies vectorielles.....	27
3.4.4	Les Stratégie hybride.....	28
3.5	Méthodes de classification.....	28
3.5.1	Réseaux de neurones.....	29
3.5.2	Machines à Vecteur de Support (Support Vector Machine).....	29
3.5.3	K plus proche voisins (K-NN).....	29
4.	Conclusion.....	30

Chapitre 3 : Machines à vecteurs de support (SVM)

1.	Introduction.....	33
2.	Pourquoi les Machine à Vecteurs de Support (SVM) ?.....	33
3.	Définition de SVM.....	33
4.	Historique.....	34
5.	Apprentissage statistique et SVM.....	35
6.	Principe de fonctionnement général de SVM.....	35
6.1	Notions de base.....	35
6.1.1	Hyperplan.....	35
6.1.2	Vecteurs de support.....	36
6.1.3	Marge.....	36
6.2	Pourquoi maximiser la marge ?.....	37
6.3	Linéarité et non-linéarité.....	38
6.3.1	Cas linéairement séparable.....	38
6.3.2	Cas non linéairement séparable.....	38
6.4	Fondements mathématiques.....	39
6.4.1	Cas des données linéairement séparables.....	39
6.4.2	Cas des données non-linéairement séparables.....	40
6.4.3	Fonction noyau.....	41
6.5	SVM multi-classes.....	42

6.5.1	Approche Un contre un (One versus One)	42
6.5.2	Approche Un contre tous (One versus All)	44
1.	Graphe de décision.	46
8.	SVMs basées arbres de décision	48
9.	Architecture générale d'une machine à vecteur support.	48
10.	Utilisation des SVM	50
11.	Avantages et inconvénients.	50
11.1	Avantages.	50
11.2	Inconvénients.	50
12.	Conclusion.	51

Chapitre 4 : Résultats expérimentaux et discussions

1	Introduction.	53
2	Ressources matérielles et logicielles	53
2.1	Ressources matérielles	53
2.2	Ressources logicielles.	53
2.2.1	Environnement informatique.	53
2.2.2	SVM multiclass	54
3.	Caractéristiques des plaques d'immatriculation Algériennes	54
4.	Architecture générale de système LAPI	56
5.	Les interfaces de l'application	57
5.1	Système de détection de plaque	57
5.1.1	Charger l'image	58
5.1.2	Prétraitement.	59
5.1.3	Localisation de la plaque.	62
5.2	Système de segmentation.	63
5.2.1	Segmentation basée sur la projection et les composants connexes	63
5.3	Système de reconnaissance de chiffre.	64
5.3.1	Description de notre base de données	64
5.3.2	L'extraction des primitives.	65
6	Expérimentations et Résultats.	68
7	Conclusion.	70
	Conclusion générale.	72
	Référence.	74

Liste des figures

Figure 1.1: Intelligence Artificielle :de la fiction à la réalité . Erreur ! Signet non défini.	
Figure1.2 : Schéma général d'un système de reconnaissance des formes	15
Figure 2.1: Différents types d'images.....	15
Figure 2.2: Filtre de Prewitt	21
Figure 2.3 : Filtre de Sobel	22
Figure 2.4: Filtre de Canny	22
Figure 2.5 : Exemple de dilation et érosion.	23
Figure 2.6: Stratégie scalaire	33
Figure 2.7 : Les stratégies marginales	34
Figure 2.8: Les stratégies vectorielles.....	34
Figure3.1: Exemple d'un hyperplan sépare deux ensembles de points.....	41
Figure 3.2 : Exemple de vecteurs de support	41
Figure 3.3 : Exemple de marge maximal (hyperplan valide)	42
Figure 3.4 : a) Hyperplan avec faible marge, b) Meilleur hyperplan séparateur.....	42
Figure 3.5 : Exemple de classification d'un nouvel élément.....	43
Figure 3.6 : a) Cas linéairement séparable, b) Cas non linéairement séparable	43
Figure 3.7 : Transformation de l'espace de données.....	44
Figure 3.8 : Le passage par une description des données peut permettre une séparation linéaire des exemples.....	48
Figure 3.9 : Approche une-contre-une.....	50
Figure 3.10 : Approche une-contre-reste avec des zones d'indécision.....	51

Figure 3.11 : Résolution des cas d'indécision dans la méthode 1vsR.....	52
Figure 3.12 : Graphe de décision acyclique orienté à quatre classes.....	54
Figure 3.13 : SVM multi classe par arbre de décision.....	55
Figure 3.14 : Architecture d'une machine à vecteur support.....	56
Figure 4.1 : Composantes de base de l'immatriculation des véhicules Algérienne.....	55
Figure 4.2 : Quelques exemples de plaques d'immatriculation Algérienne.....	55
Figure 4.3 : Architecture générale de notre système.....	56
Figure 4.4 : Interface Système de reconnaissance des plaques d'immatriculation Algérienne	57
Figure 4.5 : Interface Système de détection de plaque.....	57
Figure 4.6 : Interface de charger l'image.....	58
Figure 4.7 : Image originale transformée en images en niveau de gris.....	59
Figure 4.8 : Application la binarisation sur une image en niveau de gris.....	60
Figure 4.9 : Application de l'algorithme de Prewitt sur une image en niveau de gris.....	61
Figure 4.10 : Application les opérations l'érosion et fermeture.....	61
Figure 4.11 : Projection verticale et horizontal de l'image d'opération morphologie.....	62
Figure 4.12 : Localisation de plaque de matricule.....	63
Figure 4.13 : la plaque d'immatriculation en caractères avec et sans recadrage.....	63
Figure 4.14 : Processus de segmentation des caractères de plaques d'immatriculation...	64
Figure 4.15 : Exemple de matrice de distribution (4*4) de chiffre "9"	66
Figure 4.16 : Interface de création des corpus.....	68
Figure 4.17 : Interface de démarrer l'apprentissage.....	69
Figure 4.18 : Interface de la classification.....	69

Liste des tableaux

Tableau 4.1: Représentation de caractéristique technique de l'ordinateur de développement	53
Tableau 4.2: Description de notre base de données.....	65
Tableau 4.3: Des échantillons de notre base de données.....	65
Tableau 4.4: Vecteur des caractéristiques de chiffre "0" Zéro.....	67



Introduction générale

Introduction générale

La reconnaissance automatique de plaques d'immatriculation des véhicules est le plus intéressant et stimulant sujet de recherche à partir de ces dernières années. Il est montré que les plaques d'immatriculation sont forme et de taille différentes et ont également couleur différente dans différents pays.

Le numéro d'immatriculation représente un moyen efficace pour identifier les véhicules. Il s'agit d'une information unique pour chaque voiture.

L'identification des plaques d'immatriculation utilisée surtout dans un contexte de sécurité. Les informations extraites de la plaque peuvent être utilisées pour diverses finalités, comme la recherche de véhicules suspects ou encore la lutte contre la criminalité, etc. Ceci rend leur lecture cruciale et inévitable dans tous ces domaines.

En Algérie, les plaques d'immatriculation ont la même forme (fond blanc ou jaune avec la couleur noire pour les caractères). Pour ces conditions, on va proposer un système qui se concentre sur deux phases principales : La localisation (prétraitement et la détection des plaques) et la reconnaissance des plaques d'immatriculation (la segmentation et la reconnaissance des chiffres).

Ce mémoire est composé de quatre chapitres :

❖ **Le 1er chapitre** présent quelques généralités sur la vision artificielle et nous avons définies quelque notion de reconnaissance des formes, ainsi que les différentes approches existantes pour la reconnaissance de formes.

❖ **Le 2èm chapitre** nous allons présenter des études sur le traitement d'images et segmentation et les notions de base nécessaires à la compréhension de ces techniques.

Introduction générale

❖ **Le 3^{em} chapitre** présente une description générale de la méthode de classification consiste au SVM pour la reconnaissance des chiffres des plaques d'immatriculation.

❖ **Le 4^{em} chapitre** présente l'environnement de travail et la réalisation de chaque phase de notre système ainsi que les résultats obtenus avec l'évaluation des performances des différentes phases.



Chapitre 1

Reconnaissance

de formes

1. Introduction

L'un des principaux objectifs de l'ordinateur est d'automatiser les tâches habituellement effectuées par l'homme. La plupart des activités humaines reposent sur une faculté importante de notre cerveau : pour voir distinguer entre les formes. Par exemple, reconnaître une pomme d'une orange ; un bon diagnostic d'un mauvais, ou encore un sous-marin d'une baleine. Pour pouvoir automatiser ce genre de tâches, l'ordinateur doit obligatoirement acquérir la faculté de reconnaître les formes.

Dans ce chapitre, nous allons parler d'une manière générale sur le domaine de reconnaissance de formes et l'intelligence artificielle en présentant les différentes étapes constituant un système de reconnaissance de formes et son domaine d'application.

2. Reconnaissance des formes

2.1 Intelligence Artificielle

Le terme d'Intelligence Artificielle (IA) désigne toutes les techniques et méthodes qui permettent de doter des systèmes informatiques de capacités proches de celles de l'être humain. La science-fiction s'est approprié ce domaine en imaginant des robots, dotés d'une intelligence qui serait semblable à celle de l'homme. Toutefois, de nos jours, notre maîtrise de l'IA est bien plus modeste ; les systèmes développés ne sont capables que d'imiter certaines capacités de l'homme. Prenons par exemple des robots alliant des capacités de perception de l'environnement et de commande qui sont capables de se déplacer de façon autonome en évitant les obstacles (voir figure 1.1).

Le but de ma thèse est de réaliser un système permettant de détecter des piétons devant un véhicule à partir d'une caméra embarquée. Mon travail se situe au niveau de la perception ; effectivement, il s'agit de recréer artificiellement les capacités de perception d'un conducteur qui chercherait de façon active les piétons situés devant son véhicule avec lesquels il pourrait entrer en collision. Le système désiré est donc apparenté à un système d'Intelligence Artificielle [3].

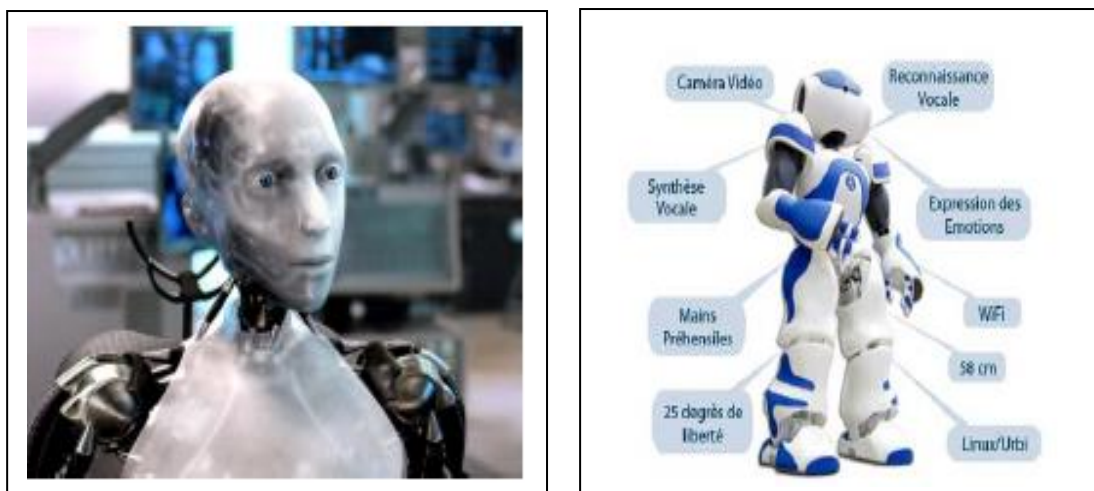


Figure 1.1: Intelligence Artificielle :de la fiction à la réalité [3].

- a) exemple d'un robot imaginé par la science-fiction (extrait du film I-Robots) ayant les mêmes capacités que l'être humain,
- b) exemple des capacités d'un robot actuel (NAO de la société AldebaranRobotics).

2.2 Définitions

On désigne par reconnaissance de formes (ou parfois reconnaissance de motifs) un ensemble de techniques et méthodes visant à identifier des motifs à partir de données brutes afin de prendre une décision dépendant de la catégorie attribuée à ce motif. On considère que c'est une branche de l'intelligence artificielle qui fait largement appel aux techniques d'apprentissage automatique et aux statistiques. Les formes ou motifs à reconnaître peuvent être de nature très variée. Il peut s'agir de contenu visuel (code barre, visage, empreinte digitale...) ou sonore (reconnaissance de parole), d'images médicales (rayon X, EEG, IRM...) ou multi spectrales (images satellitaires) et bien d'autres. Watanabe a défini une forme comme : « l'opposé du chaos ; c'est une entité vaguement définie, à laquelle on peut associer un nom ». En des termes informatiques, une forme est un ensemble de valeurs, appelés attributs, auxquels est associé un nom (ou étiquette), qui est leur classe. Plusieurs formes peuvent avoir la même classe, on dit alors que ce sont les exemples ou réalisations de la classe. Le problème que cherche à résoudre la reconnaissance des formes est d'associer une classe à une forme inconnue (qui n'a pas encore de classe associée). On considère souvent la Reconnaissance des formes comme un problème de classification : trouver la fonction qui affecte à toute forme inconnue sa classe la plus pertinente. Elle est partie intégrante de tout système intelligent destiné à la prise de décision [4].

3. Historique

Or que ce soit pour déchiffrer un texte dactylographié ou manuscrit, pour compter des chromosomes, reconnaître une tumeur, un char ou un avion de guerre, la compréhension de l'image, sa classification passe toujours par la reconnaissance d'une forme. « *Plusieurs approches théoriques ont été développées* », explique *Olivier Faugeras*.

« Les premières consistaient à faire des calculs à partir de l'image et construire des représentations symboliques de plus en plus complexes, d'abord en deux dimensions tel que sur l'image, puis tridimensionnelles, pour tenter de restituer une description proche de notre propre vision. » Un peu partout dans le monde, les chercheurs ont mis au point des méthodes mathématiques permettant de détecter les contours des objets à partir des changements rapides de contraste dans l'image, des ombres et des lumières, des régions homogènes en couleur, en intensité, en texture.

« Dès 1964, des chercheurs français, Georges Matheron (1930-2000) et Jean Serra, ont développé une autre approche théorique (baptisée morphologie mathématique) et un outil spécifique (l'analyseur de texture breveté en 1965, ndlr) d'abord pour analyser des microphotographies de terrain et évaluer des teneurs en minerai, puis pour d'autres applications comme la cytologie (caractérisation et comptage de cellules) » rappelle *Olivier Faugeras*. En 1968, ils créent le Centre de morphologie mathématique de l'Ecole des Mines de Fontainebleau. Leurs outils d'analyse et d'interprétation d'images sont longtemps restés franco-français, jusqu'à ce qu'un américain, Robert Haralick (Université du Kansas à cette époque, de Seattle actuellement), en fasse une large publicité dans les années 1980, en les adaptant à de nombreuses applications : industrielles comme l'inspection radiographique des ailes d'avions de Boeing, aériennes ou médicales.

D'autres chercheurs, comme les américains Marvin Minsky et Seymour Papert du MIT (Massachusetts Institute of Technology) ont considéré le problème dans l'autre sens, en cherchant à formaliser et à faire reproduire par l'ordinateur notre propre processus de reconnaissance d'images, donc notre propre vision. Cette démarche était dans l'air du temps, au coeur des promesses de « l'intelligence artificielle » qui devait permettre de mettre l'intelligence en équations et doter les ordinateurs de toutes les capacités humaines de raisonnement, mémoire, perception. Or la vision s'est révélée être un domaine particulièrement complexe à modéliser tant elle est basée sur une quantité phénoménale de connaissances à priori fondée sur notre intelligence et notre expérience [2].

4. Méthodes de reconnaissance de formes

- Mise en correspondance de graphes.
- Méthode Bayésienne.
- Estimation Paramétrique.
- Classifieur linéaire.
- Réseau de neurones.
- Local feature focus.
- Support Vector Machine (SVM).
- Polyèdres de contrainte.
- Méthode des hypercubes.

Un algorithme bien connu pour la détection de formes, la transformée de Hough, est une méthode d'estimation paramétrique [4].

La reconnaissance de motifs peut être effectuée au moyen de divers algorithmes d'apprentissage automatique tels [4] :

- un réseau de neurones.
- une analyse statistique.
- l'utilisation de modèles de Markov cachés.
- une recherche d'isomorphisme de graphes ou sous-graphes.

Les formes recherchées peuvent être des formes géométriques, descriptibles par une formule mathématique, telles que :

- Cercle ou ellipse.
- Courbes de Bézier, splines.
- Droite.

Elles peuvent aussi être de nature plus complexe :

- Lettre.
- Chiffre.
- Empreinte digitale.

Les algorithmes de reconnaissance peuvent travailler sur des images en noir et blanc, avec en blanc les contours des objets se trouvant dans l'image. Ces images sont le fruit d'algorithmes de détection de contours. Ils peuvent aussi travailler sur des zones de l'image prédéfinies issues de la segmentation de l'image.

5. Schéma général d'un système de Reconnaissance des formes

La majorité des systèmes de Reconnaissance des formes ont le schéma de fonctionnement suivant (voir figure 1.2).

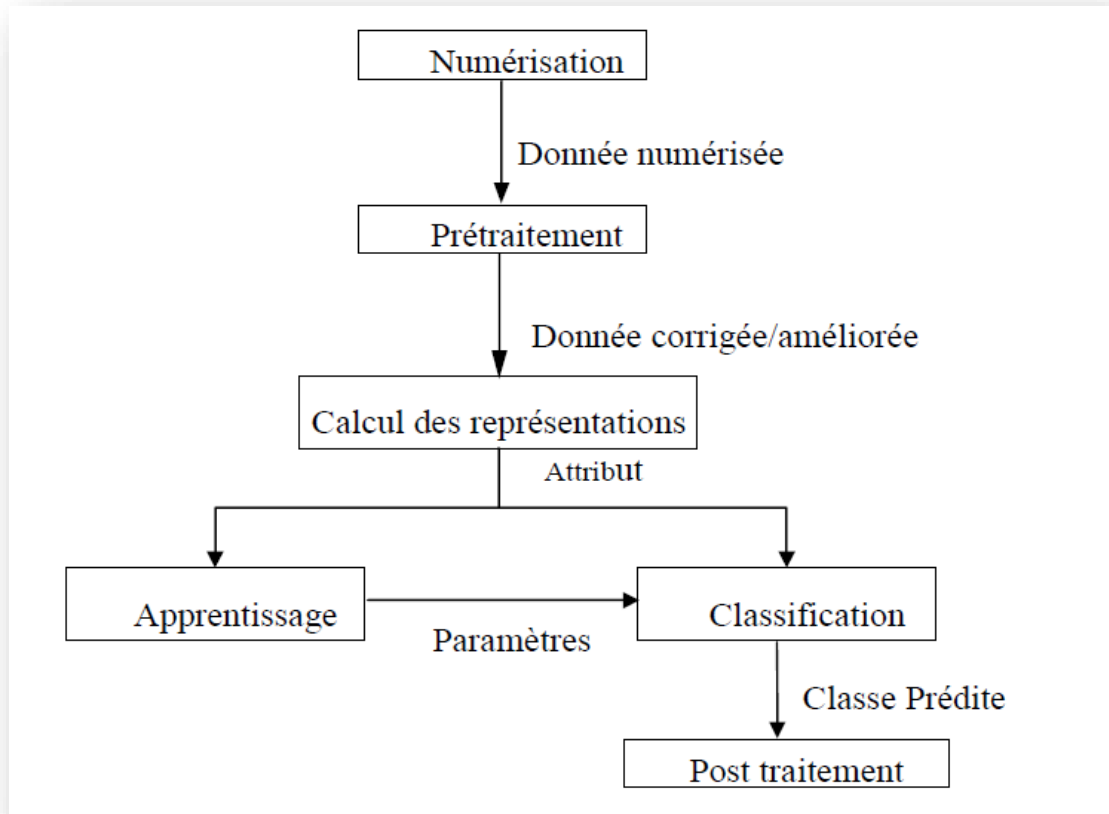


Figure1.2 : Schéma général d'un système de reconnaissance des formes [1].

5.1 Préparation des données

5.1.1 Numérisation

A partir des informations du monde physique, construire une représentation des données directement manipulable par la machine. Des capteurs (microphone, caméra, instruments de mesure) convertissent les signaux reçus du monde réel en une représentation numérique discrète. L'espace résultant, appelé "espace de représentation", a une dimension r très grande lui permettant de disposer du maximum d'informations sur les formes numérisées [1].

5.1.2 Prétraitement

Consiste à sélectionner dans l'espace de représentation l'information nécessaire au domaine d'application. Cette sélection passe souvent par l'élimination du bruit, la normalisation des données, ainsi que par la suppression de la redondance. Le nouvel espace de représentation a

une dimension r' très inférieure à r mais demeure un espace de grande dimension et contient des informations encore assez primitives [1].

5.1.3 Calcul des représentations

Il s'agit de la phase finale de la préparation des données. Elle fournit un certain nombre de "caractéristiques" ou "paramètres" (fameux attributs) en utilisant des algorithmes de sélection et/ou d'extraction d'attributs. Les attributs étant limités en nombre, "l'espace des paramètres" ainsi obtenu est de dimension p très petite par rapport à r' [4].

5.2 Apprentissage

L'apprentissage ou entraînement, est une partie importante du système de reconnaissance. Le classificateur étant généralement une fonction paramétrique, l'apprentissage va permettre d'optimiser les paramètres du classificateur pour le problème résoudre, en utilisant des données d'entraînement. Lorsque les données d'entraînements ont préalablement classées, l'apprentissage est dit "supervisé", sinon il est "non supervisé"[4].

5.3 Classification

Cette phase est le noyau de la Reconnaissance des formes. En utilisant les modèles (paramètres) obtenus lors de l'apprentissage, le classificateur assigne à chaque forme inconnue sa ou ses formes les plus probables [4].

5.4 Post traitement

Cette phase a pour but de corriger les résultats de la classification en utilisant des outils spécifiques au domaine d'application. Par exemple pour un système de reconnaissance de textes manuscrits, le classificateur se charge de classer chaque caractère séparément, alors que le post traitement applique un correcteur orthographique sur tout le texte pour valider et éventuellement corriger le résultat de la classification. Bien que facultative, cette phase permet d'améliorer considérablement la qualité de la reconnaissance [4].

6. Applications typiques de la reconnaissance des formes

6.1 Marketing

La reconnaissance des formes est souvent utilisée pour classer les consommateurs selon les produits qu'ils sont susceptible d'acheter. Elle est aussi utilisée par les sociétés de vente pour classer les clients selon qu'ils soient de bons ou mauvais payeurs, ou encore selon qu'ils vont oui ou non passer à la concurrence [1].

6.2 Finances

Les systèmes de reconnaissance des formes sont utilisés pour la détection de transactions bancaires frauduleuses ainsi que la prédiction des banqueroutes [1].

6.3 Usinage

La qualité des produits dépend souvent de la paramétrisation correcte, et les relations exactes entre la qualité et les valeurs des paramètres n'est pas claire. Les systèmes de reconnaissance des formes sont utilisés pour classer les paramètres selon la qualité des produits qu'ils sont susceptibles de générer. Ils permettent ainsi de réduire le nombre d'essais ce qui fait gagner du temps et de l'argent [1].

6.4 Energie

Les systèmes de reconnaissance des formes sont utilisés pour prévoir la consommation électrique (réduite, normale, élevée), permettant ainsi aux clients de réduire si nécessaire leur consommation, et aux producteurs de mieux gérer leurs unités de production [1].

6.5 Lecture automatisée

Les systèmes de reconnaissance des formes permettent de numériser les anciens documents ainsi que les archives, non pas sous la forme d'images, mais plutôt sous une forme textuelle [1].

6.6 Sécurité

La reconnaissance vocale et rétinienne est un exemple d'applications typiques de la reconnaissance des formes pour l'authentification. La vérification des signatures est aussi très populaire [1].

7. Conclusion

De manière générale, on peut dire que les systèmes de reconnaissance des formes sont des fonctions de classification qui associent à toute forme inconnue sa classe la plus probable. Malgré la difficulté de la tâche, les systèmes de reconnaissance des formes ont connus un grand succès dans des domaines variés.

Un système de reconnaissance des formes contient principalement quatre modules. Un module de préparation de données qui se charge de faciliter la tâche de reconnaissance en représentant les données en entrée sous une forme adaptée à la classification. Un module d'apprentissage qui va optimiser la fonction de classification pour le domaine d'application ciblé. Un module de classification qui va classer les formes inconnues et un module de post

Chapitre 1 : Reconnaissance de formes

traitement qui va corriger les résultats du classificateur en utilisant des mécanismes spécifiques au domaine d'application.



Chapitre 2

Traitement d'images et segmentation

1. Introduction

Le traitement d'images est un domaine très vaste qui a connu, et qui connaît encore, un développement important depuis quelques dizaines d'années.

Nous avons désignée par traitement d'images numériques l'ensemble des techniques permettant de modifier une image numérique afin d'améliorer ou d'en extraire des informations.

Dans ce chapitre, nous abordons les notions de base nécessaires à la compréhension des techniques de traitement d'images. Ensuite, nous allons donner un aperçu sur les différentes techniques connues dans ce domaine.

2. Définition du traitement d'images

Le **traitement d'images** est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information [5].

Il s'agit d'un sous-ensemble du traitement du signal dédié aux images et aux données dérivées comme la vidéo par opposition au sous-ensemble consacrées aux sons et autres signaux [6].

La compréhension du traitement d'images commence de ce qu'est une image. Le mode et les conditions d'acquisition et de numérisation des images traitées conditionnent largement les opérations qu'il faudra réaliser pour extraire de l'information. En effet de nombreux paramètres entrent en compte, les principaux étant:

- **La résolution d'acquisition et le mode de codage utilisé lors de la numérisation**, qui déterminent le degré de précision des éventuelles mesures de dimensions.
- **Les réglages optiques utilisés**, (dont la mise au point) qui détermine par exemple la netteté de l'image.
- **Les conditions d'éclairage**, qui déterminent une partie de la variabilité des images traitées.
- **Le bruit de la chaîne de transmission d'image**.

Quelques exemples types d'informations qu'il est possible d'obtenir d'une image numérique :

- La luminance moyenne ;
- Le contraste moyen ;
- La couleur prédominante ;

- Le taux d'acuité moyen (précis ou flou) ;
- Le taux d'uniformité des couleurs ;
- La présence ou l'absence de certains objets.

2.1 Image

Une image est avant tout un signal 2D (x, y) Souvent, cette image représente une réalité 3D (x, y, z) , il existe deux points de vue sur la notions de l'image [6] :

- ❖ D'un point de vue humain : Une image contient plusieurs informations sémantiques il faut interpréter le contenu au-delà de la valeur des nombres.
- ❖ D'un point de vue mathématique : Une image est une matrice de nombres représentant un signal, plusieurs outils permettent de manipuler ce signal.

Il existe trois principaux types d'images :

- ✓ Images en niveaux de gris, dont la valeur appartient à l'ensemble suivant $\text{Image}(x,y) \in [0..255]$.
- ✓ Images binaires (uniquement en noire et blanc), dont la valeur ne peut accéder 0 ou 1ce qui nous donne la formule suivante : $\text{Image}(x, y) \in \{0, 1\}$.
- ✓ Images couleurs qui fait un mélange des trois couleurs, $\text{ImageRed}(x,y)\text{ImageGreen}(x,y)\text{ImageBlue}(x, y)$.

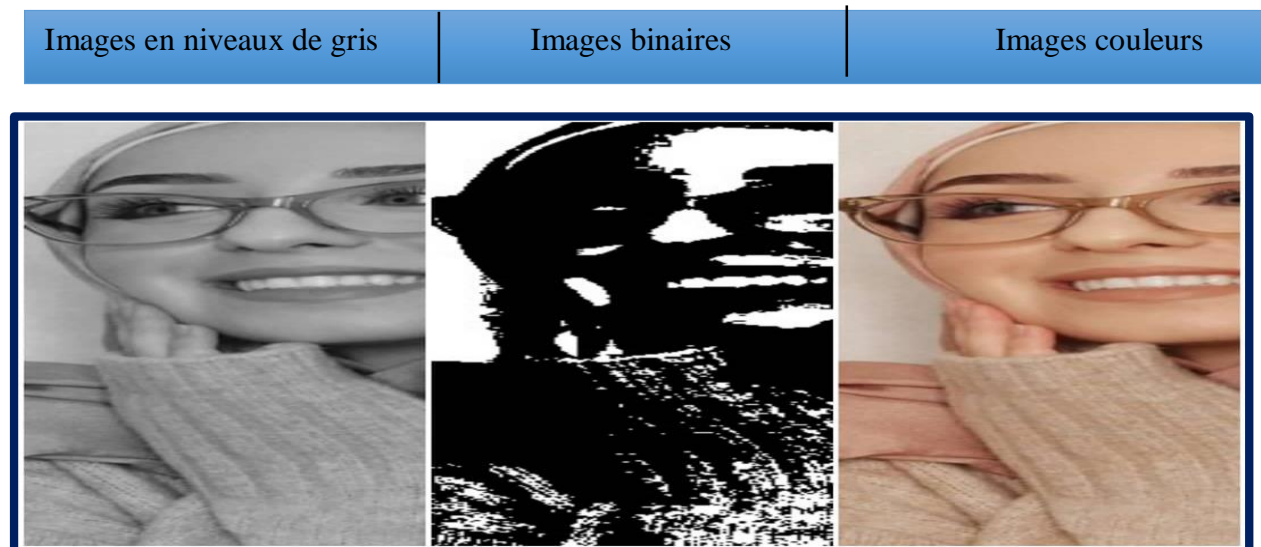


Figure 2.1: Différents types d'images.

2.2 Représentation d'image

D'un point de vue informatique, l'image est avant tout un fichier numérique qui va reconstituer un rendu visuel au travers des calculs. A la base, il existe deux technologies bien distinctes pour produire ce rendu [5].

- **Images matricielles**

- **Images vectorielles**

2.2.1 Image matricielle

Encore appelée image Bitmap, l'image matricielle est certainement celle la plus répandue et la plus couramment utilisée. Elle est tout simplement constituée d'une grille de milliers de Pixels représentant les points de couleur successifs de l'image.

Nous parlons de points lorsque ces images sont imprimées ou destinées à l'impression (Photographies, publicités, cartes etc.) Et nous parlons de pixels pour les images stockées sous forme « binaire » ou numérique.

Chaque Pixel est alors un tout petit carré ayant une place bien définie avec sa propre couleur et l'image devient alors un quadrillage de Pixels qui, placés les uns à côté des autres, restituent le rendu visuel [6].

2.2.2 Image vectorielle

L'image Vectorielle utilise également la technique du Pixel, mais cette fois, leur position et leur couleur ne sont pas figées puisqu'elles sont calculées dynamiquement par le logiciel [6].

Une image vectorielle est une image numérique composée d'objets géométriques individuels (segments de droite, polygones, arcs de cercle, etc.) définis chacun par divers attributs de forme, de position, de couleur, etc.

Par exemple pour afficher une ligne, le logiciel détermine le point de départ, le point d'arrivée puis la trajectoire à suivre. Ensuite, il calcule et positionne l'ensemble des pixels nécessaires pour afficher cette ligne. Il en va de même pour des formes et des couleurs plus complexes [6].

Par exemple, une image vectorielle d'un cercle est définie par des attributs de types : position du centre, rayon, etc.... [5].

2.3 Principe d'acquisition d'image

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images [6].

Des sources lumineuses éclairent une scène composée d'objets ; Chaque objet absorbe et renvoie cette énergie lumineuse ; enfin le capteur d'images transforme l'énergie lumineuse renvoyée dans sa direction en un signal électrique.

Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation.

Ces systèmes de saisie, dénommés optiques, peuvent être classés en deux catégories principal :

- **Acquisition à partir d'une caméra.**
- **Acquisition à partir d'un scanner.**

2.4 Caractéristiques d'une image

L'image est un ensemble structure d'informations caractérisée par les paramètres suivants :

2.4.1 Taille

On appelle taille le nombre de pixels constituant l'image, c'est-à-dire sa « dimension informatique » (le nombre de colonnes de l'image que multiplie son nombre de lignes).

Une image numérique est composée d'une grille de pixels. Ces pixels sont autant de petits carrés porteurs d'une information de couleur élémentaire [6].

2.4.2 Résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre).

Nous utilisons aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur ; plus grand est ce nombre, meilleure est la résolution [5].

2.4.3 Bruit

Le bruit dans une image est l'ensemble des pixels qui parasitent la photo, c'est un phénomène qui consiste en la dégradation progressive de la qualité de l'image au fur et à mesure que l'on augmente la sensibilité du capteur, il est dû aux limites en faible lumière du capteur et de l'électronique de l'appareil photo et par conséquent entraîne une perte de la netteté dans les détails de l'image [5].

2.4.4 Contraste

Le contraste est une propriété intrinsèque d'une image qui quantifie la différence de luminosité entre les parties claires et sombres d'une image, le contraste d'une photo est lié à la notion de luminosité, instinctivement une image paraît contrastée lorsqu'elle mêle des zones très lumineuses et des zones très sombres [6].

2.4.5 Histogramme

L'histogramme de niveaux des gris d'une image est une fonction qui permet de représenter la distribution des couleurs de l'image, en indiquant pour chaque valeur entre le noir (0) et le blanc (255), combien il y a de pixels de cette valeur dans l'image ; en abscisse (axe x) : le niveau de gris (de 0 à 255) ; en ordonnée (axe y) : le nombre de pixels.

Les pixels sombres apparaissent à gauche de l'histogramme, les pixels clairs à droite de l'histogramme et les pixels gris au centre de l'histogramme [6].

2.4.6 Luminance

La luminance encore appelée brillance est le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface pour un observateur lointain.

Une surface très lumineuse présente une forte luminance, tandis qu'une surface parfaitement noire aurait une luminance nulle [6].

2.4.7 Images à niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, nous pouvons attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255.

Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant.

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la "couleur" de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux [6].

2.4.8 Image en couleur

Même s'il est parfois utile de pouvoir représenter des images en noir et blanc, les applications multimédias utilisent le plus souvent des images en couleurs. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation.

Nous pouvons représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B) [6].

2.5 Les principales étapes de traitement d'image

En réalité, il n'existe pas de méthode de traitement d'images générale pour tous les domaines d'application possible. Il faut appliquer à chaque domaine des algorithmes spécifiques. Ces derniers utilisent le plus souvent des techniques classiques à savoir : la segmentation, la classification, la reconnaissance des frontières, la reconnaissance des caractères, etc. [6].

De manière schématique toute méthode de traitement d'images comprend ces 3 étapes majeures :

- **Prétraitements des images.**

- **Amélioration des images.**
- **Analyse des images.**

2.6 Type de données manipulées

Le traicteur d'image dispose principalement d'images numériques, donc échantillonnées. Il dispose également de données intermédiaires de diverses natures : cartes de régions, listes de points connexes, tableaux de valeurs mesurées, etc.

En ce qui concerne les images proprement dites, la représentation la plus utilisée est celle d'un tableau à 2 dimensions composé d'un ensemble de lignes et de colonnes.

Chaque cellule du tableau, appelée pixel, contient une valeur quantifiée. Cette valeur est une sémantique dépendant du type de signal qu'elle code (intensité lumineuse du point, distance à un point de référence, ou numéro de la région d'appartenance par exemple). Dans le cas des images 3D d'IRM, la représentation n'est plus un tableau à 2 dimensions mais un tableau à 3 dimensions.³ [6].

2.7 Les opérateurs de traitement d'image

Le Par analogie avec les opérateurs mathématiques, on appelle opérateurs de traitement d'images des traitements plus ou moins complexes prenant en entrée une image ou un ensemble d'informations relatif à une image, et produisant une image ou un ensemble d'informations relatif aux données initiales[5].

On classe généralement les opérateurs en différentes familles, en fonction des informations qu'ils acceptent en entrée et qu'ils fournissent en sortie, et en fonction des transformations qu'ils font subir aux données. Ainsi, par exemple, on distingue (cette liste est loin d'être exhaustive) :

- **Opérateurs image → image:**
 - **Opérateurs de modifications pixel à pixel** (aussi appelés opérateurs point à point) : changement de la dynamique de l'image, opérateurs binaires pixel à pixel (et, ou, xor, etc.);
 - **Opérateurs locaux** (traitent les pixels en fonction de leur voisinage) : opérateurs de flou, opérateurs morphologiques (érosion, dilatation, squelette, etc), opérateurs de détection de contours ;
 - **Opérateurs dans l'espace fréquentiel** : opérateurs de réduction du bruit, filtres passe-bande (souvent utilisés en première approche pour améliorer l'image, on les appelle alors des opérateurs de prétraitement) ;
 - **Opérateurs globaux** : calcul des distances ;
 - **Opérateurs de croissance de régions** : ligne de partage des eaux.
- **Opérateurs image → ensemble d'informations:**

- Opérateurs de segmentation en frontières, en régions ;
- Opérateurs de classification de pixels ;
- Opérateurs de calcul de paramètres.

➤ Opérateurs ensemble d'informations → image :

- Constructeurs d'image à partir d'une carte de régions ou d'une liste de frontières.

Les parties suivantes s'attachent à détailler les différents opérateurs et leurs Applications habituelles, puis à présenter la manière dont ils sont combinés pour construire une application de traitement d'images.

2.7.1 Filtres linéaires

Un filtre linéaire transforme un ensemble de données d'entrée en un ensemble de données de sortie selon une opération mathématique appelée convolution. Lorsqu'il s'agit de, données numérisées comme dans le cas du traitement d'image, la relation entre les valeurs des pixels de sortie et celle des pixels d'entrée est décrite par un tableau de nombres, généralement carré, appelé matrice de convolution. Le temps de calcul est souvent réduit lorsqu'on veut séparer un filtre en deux filtres dont la convolution mutuelle permet de le reconstituer. Cette remarque est utilisée en particulier pour créer un filtre à deux dimensions à partir de deux filtres à une seule dimension (vecteurs) dans le sens horizontal et le sens vertical [6].

2.7.2 Lisage

Ceux-ci sont des filtres passe-bas qui coupent plus ou moins les plus hautes fréquences. Ils sont utilisés pour atténuer les bruits d'origines les plus diverses qui polluent l'information, en particulier dans la détection de contours considérée ci-après.

Techniquement, il s'agit de traductions discrètes de filtres continus qui, comme ceux-ci, ne modifient pas le niveau global du signal. Les termes de la matrice de convolution sont donc généralement des entiers à diviser par leur somme [6].

- **Filtre uniforme:** il est obtenu par convolution de deux filtres unidimensionnels rectangulaires. Toutes les composantes de la matrice ont la même valeur. L'imperfection de ce filtre réside dans le fait qu'il introduit des déphasages.
- **Filtre pyramidal :** la convolution d'un filtre rectangulaire avec lui-même conduit à un filtre triangulaire grâce auquel les phases ne sont plus modifiées. Le filtre pyramidal obtenu à partir de filtres triangulaires dans les deux directions.
- **Filtre gaussien :** ce filtre très populaire utilise la loi de probabilité de Gauss (voir Loi normale multidimensionnelle). Des approximations de plus en plus précises peuvent être obtenues, selon le théorème central limite par itération de l'un des filtres précédents.

2.7.3 Détection de contours

Ces filtres transforment l'image d'entrée en une image noire sauf aux points où un contour est détecté qui est marqué en blanc. Les valeurs absolues importent peu, il est sans intérêt d'échanger d'échelle comme pour un lissage [6].

La détection est basée sur la dérivation selon les deux coordonnées. Si on considère classiquement les signaux comme des sommes de sinusoides, la dérivation apparaît comme un filtre passe-haut qui introduit donc du bruit à l'origine de faux contours. Pour l'amateur il est recommandé, avant d'utiliser un filtre simple, d'atténuer ce bruit par passage dans un filtre flou. Des méthodes plus élaborées ont été systématisées pour les professionnels.

- **Filtre de Prewitt**

le filtre de Prewitt introduit un flou, chacune des deux matrices étant le produit du filtre dérivation dans la direction considérée par un filtre de flou rectangulaire selon l'autre direction [6].

Les différents masques de convolution de Prewitt sont :

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

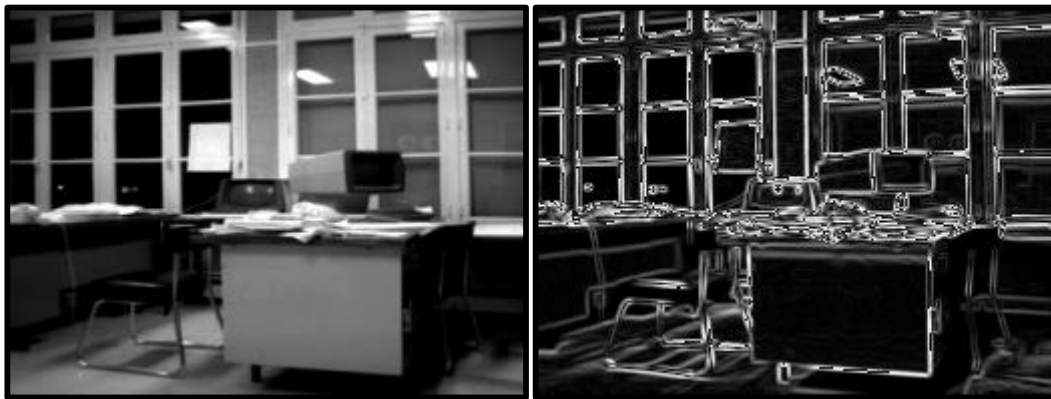


Figure 2.2: Filtre de Prewitt [16].

- **Filtre de Sobel**

la technique précédente est améliorée en remplaçant le filtre rectangulaire par un filtre triangulaire [6].

Les différents masques de convolution de Sobel sont :

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



Figure 2.3 : Filtre de Sobel [16].

- **Filtre de Canny**

C'est un filtre de Sobel précédé par un lissage gaussien et suivi par un seuillage. Ce filtre est conçu pour être optimal, au sens de trois critères [6].

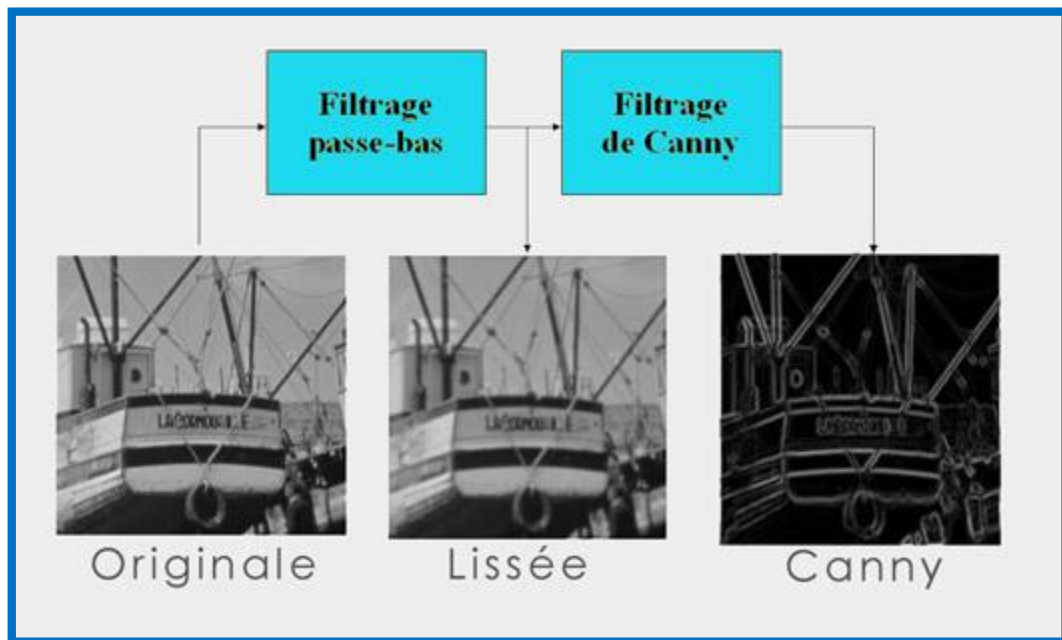


Figure 2.4: Filtre de Canny [7].

2.8 Opérateurs morphologiques

Les opérateurs morphe-mathématiques se sont initialement appliqués sur des images en noir et blanc (Matheron et Serra, 1965). Ils ont ensuite été étendus à des images en niveaux de gris par Dougherty en 1978. Pour les appliquer séparément à chaque composante couleur [7]

2.8.1 Conversion de l'image en niveau de gris

L'image qui est en origine en couleur sera transformée en une image en niveau de gris, un pixel est représenté sur 8 bits et le nombre de composant couleur pour chaque pixel sera donc égal à 1.

2.8.2 L'érosion

Elle permet d'éliminer les pixels noirs isolés au milieu des parties blanches de l'image, la séparation des objets à l'endroit des étranglements, le rétrécissement des objets de grande taille et la disparition des petites composantes [7]

2.8.3 La dilatation

Elle élimine les tâches blanches dans des zones noires mais ajoute des pixels noirs au contour des objets présents dans l'image [7].

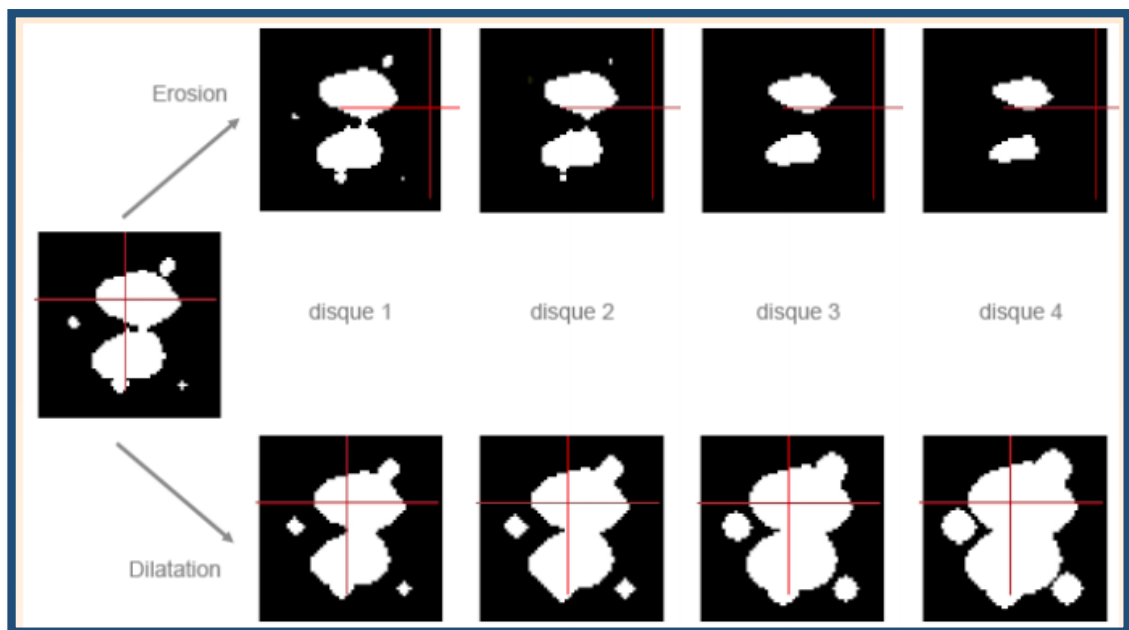


Figure 2.5 : Exemple de dilution et érosion.

2.8.4 L'ouverture

Une érosion suivie d'une dilatation s'appelle une ouverture. L'ouverture a pour propriété d'éliminer toutes les parties des objets qui ne peuvent pas contenir l'élément structurant.

2.8.5 La fermeture

Une dilatation suivie d'une érosion s'appelle une fermeture. Il a pour propriété de combler tout ce qui est de taille inférieur à l'élément structurant.

3. Segmentation d'image

3.1 Définition de la segmentation

La segmentation est un traitement de bas niveau qui consiste à créer une partition de l'image A en sous-ensembles R_i , appelés régions tels qu'aucune région ne soit vide, l'intersection entre deux régions soit vide et l'ensemble des régions recouvre toute l'image [8].

Une région est un ensemble de pixels connexes ayant des propriétés communes qui les différencient des pixels des régions voisines [8].

On peut dire que :

La segmentation est la partition d'une image en un ensemble de régions qui ne se chevauchent pas et dont l'union est l'image entière. Quelques règles à suivre pour obtenir une segmentation sont :

1. Les régions doivent être uniformes et homogènes par rapport à certaines caractéristiques (niveau de gris, écart type, gradient).
2. Leurs intérieurs doivent être simple et sans beaucoup de petits trous (des parties de région non segmentés).
3. Les régions adjacentes doivent avoir des valeurs très différentes par rapport à la caractéristique prise en compte dans la segmentation.
4. Les limites de chaque région doivent être simples et spatialement précises.

En termes mathématiques:

$$\left\{ \begin{array}{l} \bigcup_{i=1}^n R_i = I \\ R_i \neq \Phi \quad \forall i = 1 \dots n \\ R_i \cap R_j = \Phi \quad \forall i, j \text{ avec } i \neq j \end{array} \right.$$

Où \mathbf{R}_i est l'ensemble des régions formant l'image \mathbf{I} .

- Est une opération appliquée à l'image qui consiste à subdiviser une scène réelle, en parties constituantes ou objets, en projetant une scène réelle sur un plan.. Il faut donc disposer d'un certain nombre d'attributs, représentatifs des régions que l'on cherche à extraire pour procéder [35].

- Il existe deux techniques permettant la mise en oeuvre de la segmentation ; la première connue sous le nom de segmentation implicite et la deuxième c'est la segmentation explicite [4].

✓ La segmentation implicite

Les méthodes de segmentation implicite s'inspirent des approches utilisées dans le domaine de la parole, où le signal est divisé en intervalles de temps réguliers et procèdent à une sur-segmentation importante de l'image du mot à pas fixe (un ou quelques pixels). Cela permet d'assurer un taux de présence important des points de liaison entre lettres considérées.

La segmentation s'effectue pendant la reconnaissance qui assure son guide. Le système recherche dans l'image, des composantes ou des groupements de graphèmes (La plus petite unité distinctive de l'écriture) qui correspondent à ses classes de lettres. Classiquement, il peut le faire de deux manières :

- Soit par fenêtrage : le principe est d'utiliser une fenêtre mobile de largeur variable (qui n'est pas facile à déterminer) pour trouver des séquences de points de segmentation potentiels qui seront confirmés ou non par la reconnaissance de caractères. Elle nécessite deux étapes : La génération d'hypothèses de segmentation (séquences de points obtenus par le fenêtrage), le deuxième est le choix de la meilleure hypothèse de la reconnaissance (validation).
- soit par recherche de primitives : il s'agit de détecter les combinaisons de primitives qui donneront la meilleure reconnaissance [35].

✓ La segmentation explicite

Cette approche souvent appelée dissection, elle est antérieure à la reconnaissance et n'est pas remise en cause pendant la phase de reconnaissance. Les hypothèses des caractères sont déterminées à partir des informations de bas niveau présentes sur l'image.

Ces hypothèses sont définitives et doivent être d'une grande fiabilité car la moindre erreur de segmentation remet en cause la totalité des traitements ultérieurs.

Les approches de segmentation explicite s'appuient sur une analyse morphologique du mot manuscrit pour localiser des points de segmentation potentiels. Elles sont particulièrement adaptées à l'analyse de la représentation bidimensionnelle et donc plus souvent utilisées dans les systèmes de reconnaissance hors-ligne de mots.

Certaines méthodes de segmentation explicite sont basées sur une analyse par morphologiques mathématiques exploitent les concepts de régularité et singularité du tracé, analyse des contours supérieurs ou inférieurs du mot.

Les points de segmentation potentiels détectés sont confirmés à l'aide de diverses heuristiques [4].

3.2 Le choix d'une technique de segmentation

Le choix est lié à [8] :

- La nature de l'image (éclairage, contours, texture, etc.).
- Aux opérations en aval de la segmentation (compression, reconnaissance des formes, Mesures, etc.).
- Aux primitives à extraire (droites, régions, textures, etc.).
- Aux contraintes d'exploitation (temps réel, espace mémoire, etc.).

3.3 Objectifs de la segmentation

- Fournir des régions homogènes (selon un critère donné) .
- Localiser de manière précise les contours des régions .
- L'étude et l'interprétation des structures anatomiques .
- Réduction de bruit [8].

3.4 Les stratégies de segmentation

En traitement d'image, il existe quatre types différents stratégies de la segmentation des images couleur, à savoir les stratégies scalaires, marginales, vectorielles et hybrides [8].

3.4.1 Les stratégies scalaires

L'approche scalaire commence par une fusion des n composantes de l'image multi composante. L'image sous forme mono-composante ainsi obtenue est traitée par la suite (**Figure 2.6**). Par exemple, une image vectorielle couleur codée dans l'espace RVB peut être transformée, avant traitement, en une image scalaire. Cela peut être réalisé au moyen d'une analyse en composantes principales dont on ne retient que la première composante.

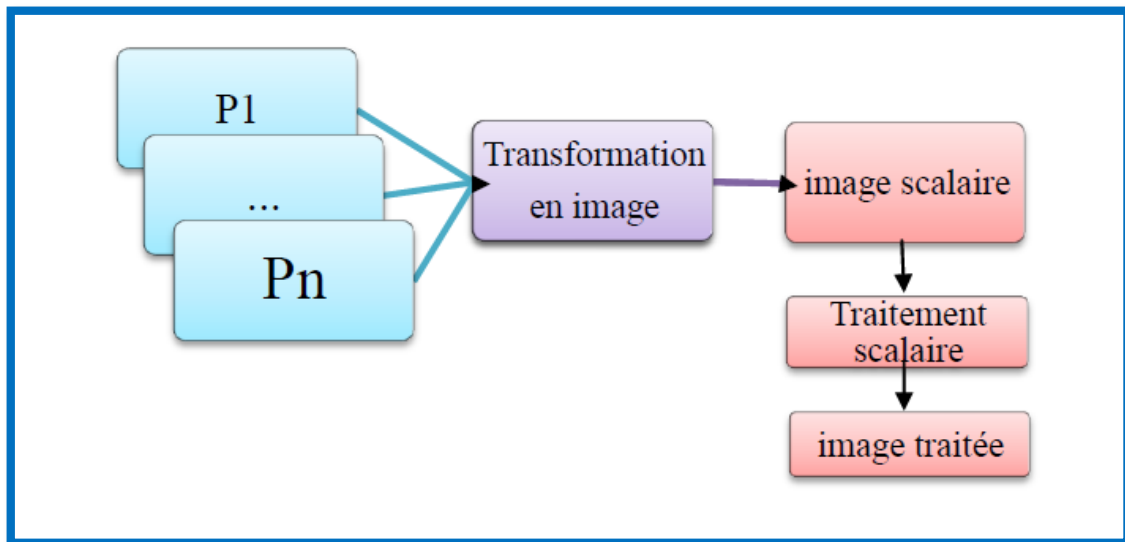


Figure 2.6: Stratégie scalaire [8].

3.4.2 Les stratégies marginales

L'approche marginale opère un traitement scalaire sur chaque composante de l'image. Les résultats obtenus sont ensuite fusionnés en une seule image mono-composante (Figure 2.7). Cette stratégie demande autant de traitements qu'il y a de composantes.

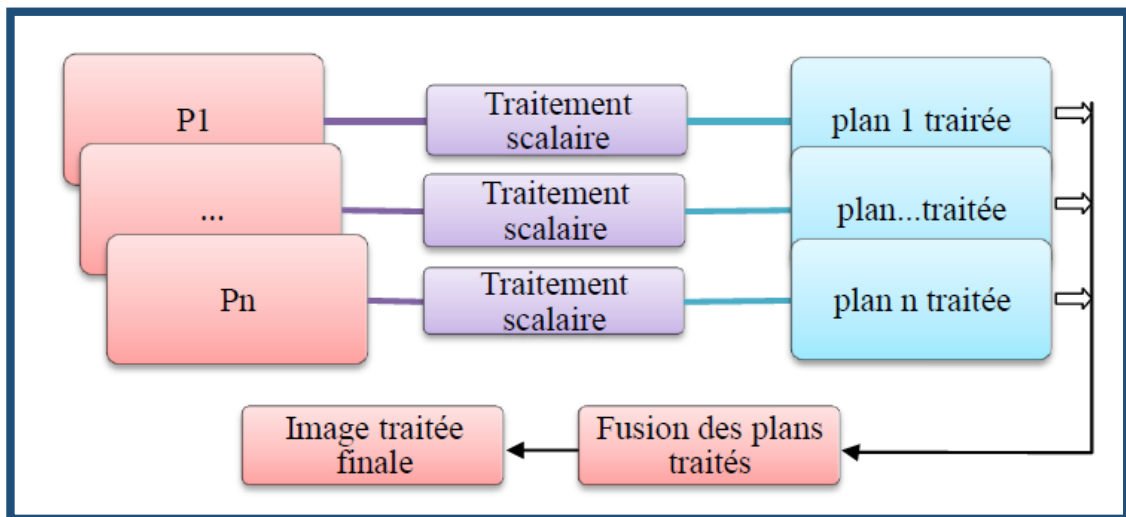


Figure 2.7 : Les stratégies marginales [8].

3.4.3 Les stratégies vectorielles

L'approche vectorielle utilise directement, et de façon globale, l'information vectorielle présente dans l'image (Figure 2.8). La corrélation entre composantes est ainsi prise en compte.

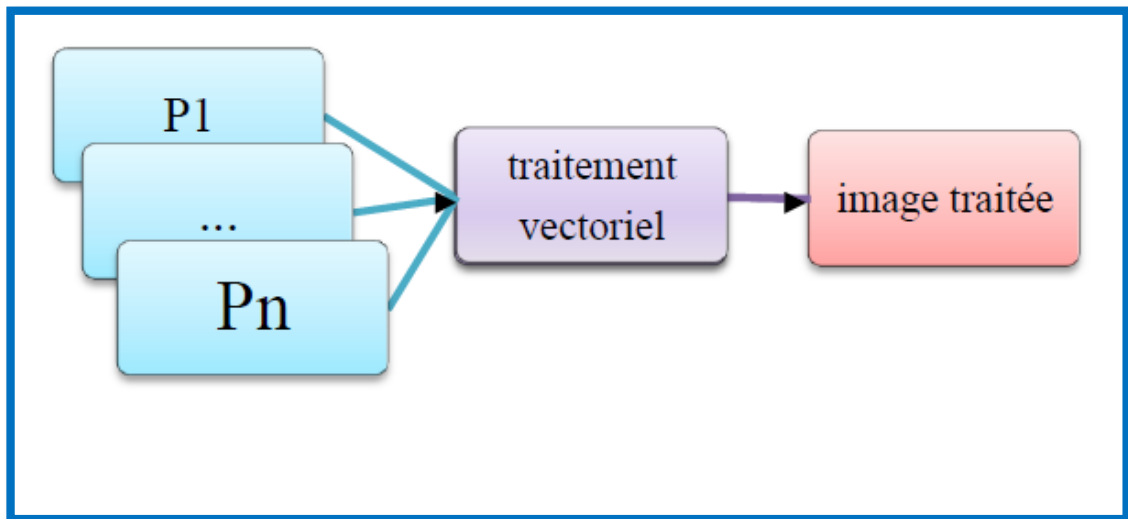


Figure 2.8: Les stratégies vectorielles [8].

3.4.4 Les Stratégie hybride

Le mélange des deux stratégies marginales et vectorielles résulte la stratégie hybride. Elle est couramment utilisée dans la segmentation par analyse d'histogrammes des images couleur.

3.5 Méthodes de classification

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

Elles s'appliquent à un grand nombre d'activités humaines et conviennent en particulier au problème de la prise de décision automatisée. La procédure de classification sera extraite automatiquement à partir d'un ensemble d'exemples.

Un exemple consiste en la description d'un cas avec la classification correspondante.

Un système d'apprentissage doit alors, à partir de cet ensemble d'exemples, extraire une procédure de classification, il s'agit en effet d'extraire une règle générale à partir des données observées. La procédure générée devra classer correctement les exemples de l'échantillon et avoir un bon pouvoir prédictif pour classer correctement de nouvelles descriptions.

Parmi les méthodes qui ont été développées dans ce contexte (machine d'apprentissage), nous distinguons par exemple les réseaux de neurones et les machines à vecteurs du support (SVM : Support Vector Machines, en anglais) ainsi que K plus proches voisins (K Nearest Neighbors) [9].

3.5.1 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 [13].

3.5.2 Machines à Vecteur de Support (Support Vector Machine)

L'algorithme des machines à vecteurs de support a été développé dans les années 90 par Vapnik. Il a initialement été développé comme un algorithme de classification binaire supervisée. Il s'avère particulièrement efficace de par le fait qu'il peut traiter des problèmes mettant en jeu de grands nombres de descripteurs, qu'il assure une solution unique (pas de problèmes de minimum local comme pour les réseaux de neurones) et il a fourni de bons résultats sur des problèmes réels[10][11] .

L'algorithme sous sa forme initiale revient à chercher une frontière de décision linéaire entre deux classes, mais ce modèle peut considérablement être enrichi en se projetant dans un autre espace permettant d'augmenter la séparabilité des données [12].

3.5.3 K plus proche voisins (K-NN)

L'algorithme KPPV affecte une forme inconnue à la classe de son plus proche voisin en le comparant aux formes stockées dans une classe de références nommée prototypes. Il renvoie les K formes les plus proches de la forme à reconnaître suivant un critère de similarité. Une stratégie de décision permet d'affecter des valeurs de confiance à chacune des classes en compétition et d'attribuer la classe la plus vraisemblable (selon la métrique choisie) à la forme inconnue [15].

Cette méthode présente l'avantage d'être facile à mettre en œuvre et fournit de bons Résultats. Son principal inconvénient est lié à la faible vitesse de classification due au nombre important de distances à calculer [14].

4. Conclusion

Dans ce chapitre nous avons vu la notion de l'image ainsi que ses différentes caractéristiques, puis les opérateurs de traitement d'image concrets du domaine de traitement d'image.

Chapitre 3

Machines à vecteurs de support (SVM)

1. Introduction

Parmi les méthodes à noyaux, inspirées de la théorie statistique de l'apprentissage de Vladimir Vapnik, les Machines à Vecteurs de Support (SVM) constituent la forme la plus connue. SVM est une méthode de classification binaire par apprentissage supervisé, elle fut introduite par Vapnik en 1995. Cette méthode est donc une alternative récente pour la classification.

Elle repose sur l'existence d'un classificateur linéaire dans un espace approprié. Puisque c'est un problème de classification à deux classes, cette méthode fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonctions dites noyau (kernel) qui permettent une séparation optimale des données [19].

Les SVM ont été utilisés dans différents domaines d'expertise, tels que les diagnostics médicaux, le traitement d'image, le traitement du signal, etc. Dans ce chapitre, nous présentons les aspects théoriques de la méthode SVM.

2. Pourquoi les Machine à Vecteurs de Support (SVM) ?

L'algorithme des machines à vecteurs de support a été développé dans les années 90 par le russe Vladimir Vapnik. Initialement, les SVM ont été développé comme un algorithme de classification binaire supervisée. Il s'avère particulièrement efficace de par le fait qu'il peut traiter des problèmes mettant en jeu de grands nombres de descripteurs, qu'il assure une solution unique (pas de problèmes de minimum local comme pour les réseaux de neurones) et il a fourni de bons résultats sur des problèmes réels [19].

L'algorithme sous sa forme initiale revient à chercher une frontière de décision linéaire entre deux classes, mais ce modèle peut considérablement être enrichi en se projetant dans un autre espace permettant d'augmenter la séparabilité des données. On peut alors appliquer le même algorithme dans ce nouvel espace, ce qui se traduit par une frontière de décision non linéaire dans l'espace initial [19].

3. Définition de SVM

Les SVM sont un ensemble de techniques d'apprentissage supervisé de nouvelle génération basés sur les progrès récents de la théorie de **l'apprentissage statistique**, ils

sont basés sur le principe de la maximisation de la marge (séparation des classes [18]). Principalement conçues pour résoudre ces problèmes.

- Des problèmes de discrimination, permettant de décider à quelle classe appartient un échantillon.
- Cependant, ils peuvent aussi résoudre des problèmes de régression visant à prédire la valeur numérique d'une variable (Scholkopf et Smola, 2001). C'est-à-dire de prédiction d'une variable continue en fonction d'autres variables, Comme c'est le cas par exemple dans de la prédiction de consommation électrique en fonction de la période de l'année, de la température, etc.

La résolution de ces deux problèmes se base sur la construction d'une fonction h qui, à un vecteur d'entrée x , fait correspondre une sortie y ($y = h(x)$) [18].

Les SVM constituent la forme la plus connue parmi les méthodes à noyaux. Ce sont des classificateurs qui reposent sur deux idées clés :

- ✓ La notion de marge maximale.
- ✓ La notion de Fonction noyau.

4. Historique

Les séparateurs à vastes marges reposent sur deux idées clés : la notion de marge maximale et la notion de fonction noyau. Ces deux notions existaient depuis plusieurs années avant qu'elles ne soient mises en commun pour construire les SVM.

L'idée des hyperplans à marge maximale a été explorée dès 1963 par Vladimir Vapnik et A. Lerner, et en 1973 par Richard Duda et Peter Hart dans leur livre *Pattern Classification*. Les fondations théoriques des SVM ont été explorées par Vapnik et ses collègues dans les années 70 avec le développement de la Théorie de Vapnik-Chervonenkis, et par Valiant.

L'idée des fonctions noyaux n'est pas non plus nouvelle : le théorème de Mercer date de 1909, et l'utilité des fonctions noyaux dans le contexte de l'apprentissage artificiel a été montrée dès 1964 par Aizermann, Bravermann et Rozoener.

Ce n'est toutefois qu'en 1992 que ces idées seront bien comprises et rassemblées par Boser, Guyon et Vapnik dans un article, qui est l'article fondateur des séparateurs à vaste marge. L'idée des variables ressorts, qui permet de résoudre certaines limitations pratiques importantes, ne

sera introduite qu'en 1995. À partir de cette date, qui correspond à la publication du livre de Vapnik, les SVM gagnent en popularité et sont utilisés dans de nombreuses applications. Un brevet américain sur les SVM est déposé en 1997 par les inventeurs originaux.

5. 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 [19].

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 [19].

6. Principe de fonctionnement général de SVM

6.1 Notions de base

6.1.1 Hyperplan

On appelle hyperplan séparateur un hyperplan qui sépare les deux classes, en particulier il sépare leurs points d'apprentissage. Comme il n'est en générale pas possible d'en trouver un, on se contentera donc de chercher un hyperplan discriminant qui est une approximation au sens d'un critère à fixer (maximiser la distance entre ces deux classes) [18].

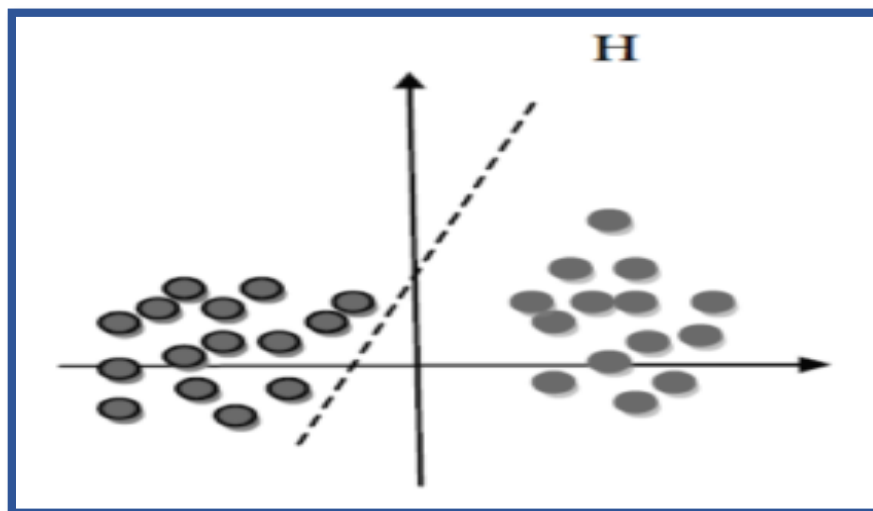


Figure 3.1: Exemple d'un hyperplan sépare deux ensembles de points.

6.1.2 Vecteurs de support

Ce sont les points de la frontière entre les deux classes des données parmi l'ensemble total d'apprentissage, ces points sont appelés vecteurs de support [18].

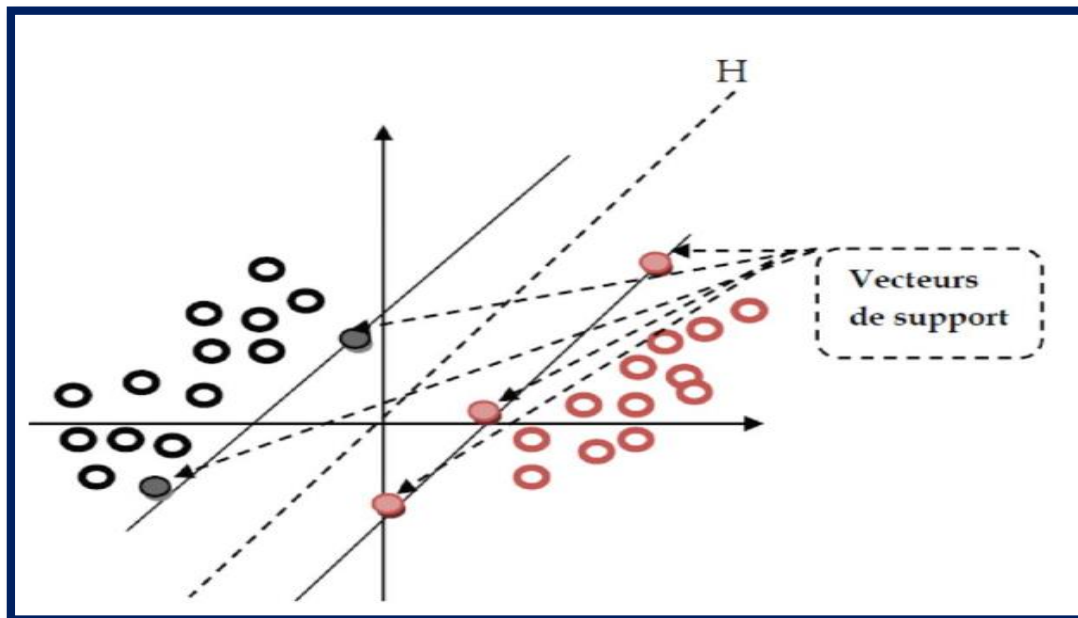


Figure 3.2 : Exemple de vecteurs de support [18].

6.1.3 Marge

Il existe une infinité d'hyperplans capable de séparer parfaitement les deux classes d'exemples. Le principe des SVM est de choisir celui qui va maximiser la distance minimale entre l'hyperplan et les exemples d'apprentissage (i.e. la distance entre l'hyperplan et les vecteurs de support), cette distance est appelée la marge [18].

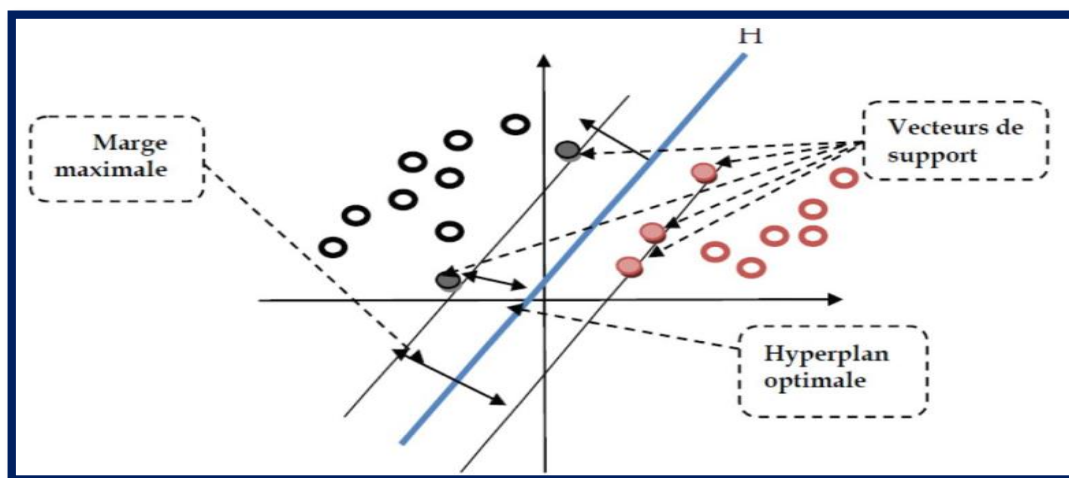


Figure 3.3 : Exemple de marge maximal (hyperplan valide) [18].

6.2 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 [19].

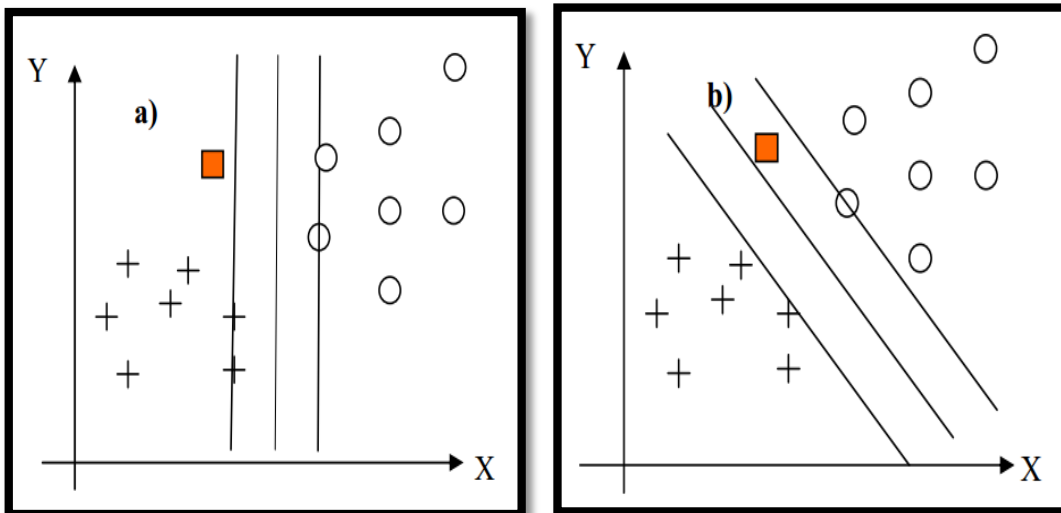


Figure 3.4 : a) Hyperplan avec faible marge, b) Meilleur hyperplan séparateur [19]

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 « + ».

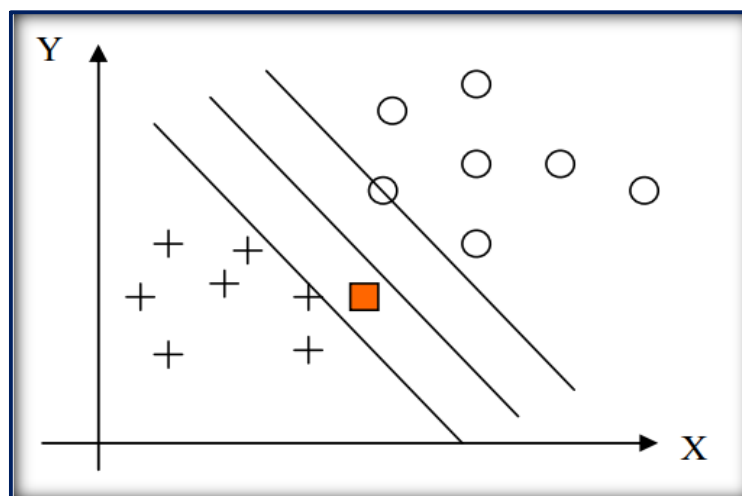


Figure 3.5 : Exemple de classification d'un nouvel élément.

6.3 Linéarité et non-linéarité

Parmi les modèles des SVM, on constate les cas linéairement séparable et les cas non linéairement séparable.

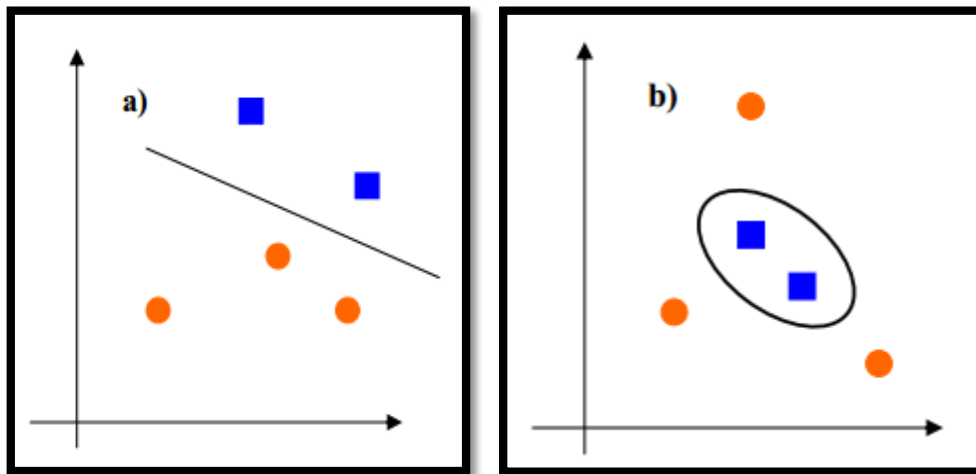


Figure 3.6 : a) Cas linéairement séparable, b) Cas non linéairement séparable [19].

6.3.1 Cas linéairement séparable

Les cas linéairement séparable sont les plus simples de SVM car ils permettent de trouver facilement le classificateur linéaire.

6.3.2 Cas non linéairement séparable

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 [19].

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 rescription ». En effet, intuitivement, plus la dimension de l'espace de rescription est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples est élevée [19]. Ceci est illustré par le schéma (**Figure 3.7**)

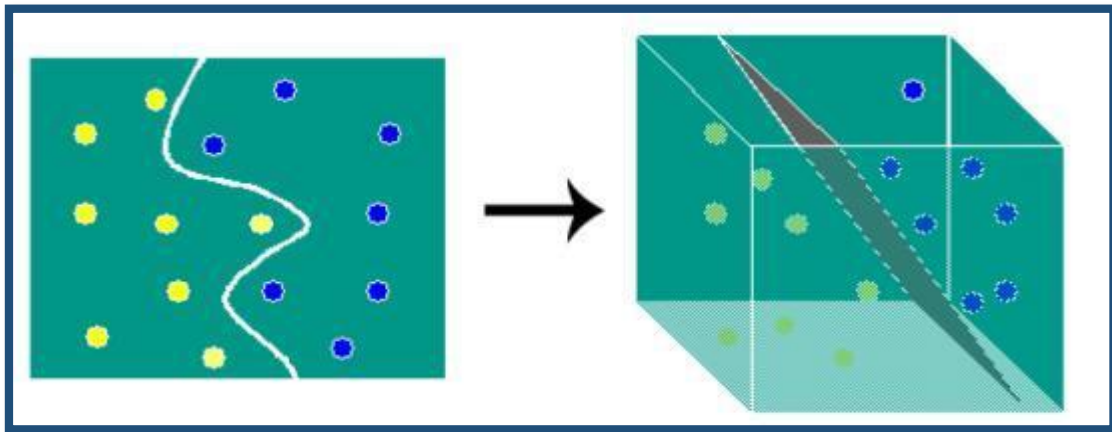


Figure 3.7 : Transformation de l'espace de données.

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 représentations de plus grande dimension. Cette transformation non linéaire est réalisée via une fonction noyau [19].

6.4 Fondements mathématiques

6.4.1 Cas des données linéairement séparables

Considérons « l » points: $\{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$, $x_i \in \mathbb{R}^N$. Avec $i=1 \dots L$ et $y_i \in \{\pm 1\}$. Ces points sont classés en utilisant une famille de fonctions linéaires définies par :

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \quad (3.1)$$

Avec $\mathbf{w} \in \mathbb{R}^N$ et $b \in \mathbb{R}$ de telle sorte que la fonction de décision concernant l'appartenance d'un point à l'une des deux classes soit donnée par :

$$F(\mathbf{x}) = \text{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b) \quad (3.2)$$

La fonction (3.1) représente l'équation de l'hyperplan H. La fonction de décision (3.2) va donc observer de quel côté de H se trouve l'élément de x [20].

On appelle la marge d'un élément la distance euclidienne prise perpendiculairement entre H et x . Si on prend un point quelconque t sur H, cette marge peut s'exprimer en :

$$\mathbf{M}_x = \frac{\mathbf{w}}{\|\mathbf{w}'\|} (\mathbf{x} - \mathbf{t}) \quad (3.3)$$

La marge de toutes les données est définie comme étant :

$$M = \min_{x \in E} M_x \quad (3.4)$$

- L'approche de classification par SVM tend à maximiser cette marge pour séparer le plus clairement possible deux classes. Intuitivement, avoir une marge la plus large possible sécurise mieux le processus d'affectation d'un nouvel élément à l'une des classes [7].
- Un SVM fait donc partie des classificateurs à marge maximale.
- Un classificateur à marge maximale est un classificateur dont l'hyperplan optimal séparant deux classes est une solution du problème d'optimisation mathématique suivant (**forme primale**) :

$$\text{Minimiser } \frac{1}{2} \|\mathbf{w}\|^2 (\langle \mathbf{w}, \mathbf{x} \rangle + \mathbf{b}) \quad \mathbf{z} \geq \mathbf{x} \in \mathbf{E} \quad (3.5)$$

- La fonction objective de ce problème est le carré de l'inverse de la double marge qu'on veut maximiser. La contrainte unique correspond au fait que les éléments \mathbf{x} doivent être bien placés.
- La résolution de ce problème nécessite de fixer les paramètres \mathbf{w} et \mathbf{b} qui constituent les variables α_i de la machine d'apprentissage.
- Les classificateurs à marge maximale donnent de bons résultats lorsque les données sont linéairement séparables.
- La tâche de discrimination est de trouver un hyperplan qui sépare deux (ou plus) ensembles de vecteurs.
- Pour la détection et la reconnaissance des vertèbres, ces deux classes peuvent être région de vertèbre ou non [7].

6.4.1.1 Forme duale

La formulation primale peut être transformée en formulation duale en utilisant les multiplicateurs de Lagrange. L'équation (3.5) s'écrit alors sous la forme suivante [7] :

$$L(\mathbf{w}, \mathbf{b}, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^l \alpha_i (y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + \mathbf{b}) - 1) \quad (3.6)$$

6.4.2 Cas des données non-linéairement séparables

On part du problème primal linéaire et on introduit des variables « ressort » pour assouplir les contraintes [19] :

$$\begin{cases} \min \frac{1}{2} \|\mathbf{w}\|^2 + c \sum_{i=1}^n \varepsilon_i \\ \forall i, y_i (\mathbf{w} \cdot \mathbf{x}_i + \mathbf{b}) \geq 1 - \varepsilon_i \end{cases} \quad (3.9)$$

On pénalise par le dépassement de la contrainte.

On en déduit le problème dual qui a la même forme que dans le cas séparable [19] :

$$\begin{cases} \max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\ \forall i, 0 \leq \alpha_i \leq c \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases} \quad (3.8)$$

La seule différence est la borne supérieure C sur les α .

6.4.3 Fonction noyau

La transformation de l'espace de représentation des données d'entrées en un espace de plus grande dimension F, dans lequel il est probable qu'il existe une séparatrice linéaire, afin de pouvoir traiter des cas où les données ne sont pas linéairement séparables. Les fonctions du noyau sont utilisées pour cartographier efficacement les données d'entrée qui peuvent ne pas être séparées linéairement en un espace de grande dimension, où des méthodes linéaires peuvent alors être appliquées [18].

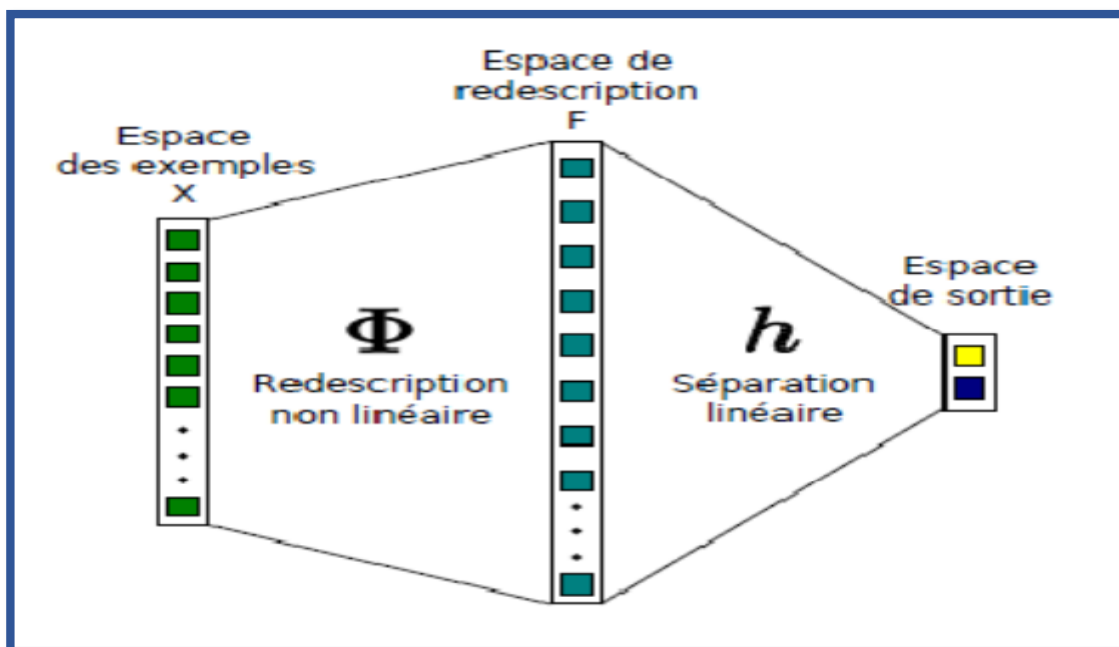


Figure 3.8 : Le passage par une description des données peut permettre une séparation linéaire des exemples.

6.4.3.1 Exemples de noyaux

- Linéaire $k(x, x') = x \cdot x'$
- Polynomial $k(x, x') = (x \cdot x')^d$ ou $(c + x \cdot x')^d$
- Gaussien $k(x, x') = e^{-\|x-x'\|^2/\sigma}$
- Laplacien $k(x, x') = e^{-\|x-x'\|/\sigma}$ [19].

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 [19].

6.5 SVM multi-classes

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

6.5.1 Approche Un contre un (One versus One)

Cette méthode, appelée aussi "pairwise". Elle consiste à utiliser un classifieur pour chaque paire de classes. Au lieu d'apprendre K fonctions de décisions, la méthode 1vs1 discrimine chaque classe de chaque autre classe, ainsi $K(K-1)/2$ fonctions de décisions sont apprises. Pour chaque paire de classes $(k; s)$, la méthode 1vs1 définit une fonction de décision binaire : $\mathfrak{R} \rightarrow \{-1, +1\}$.

L'affectation d'un nouvel exemple se fait par liste de vote. On teste un exemple par le calcul de sa fonction de décision pour chaque hyperplan. Pour chaque test, on vote pour la classe à laquelle appartient l'exemple (classe gagnante). On définit pour le faire la fonction de décision binaire $H_{KS}(x)$ de l'équation (3.9).

$$H_{KS}(x) = \text{signe}(F_{KS}(x))$$

$$= \begin{cases} +1 & \text{si } f_{ks}(x) > 0 ; \\ 0 & \text{sinon} \end{cases} \quad (3.9)$$

Sur la base des $K(K-1)/2$ fonctions de décision binaires, on définit K autres fonctions de décision l'équation (3.10) :

$$H_K(x) = \sum_{s=1}^m H_{KS}(x) \quad (3.10)$$

Un nouvel exemple est affecté à la classe la plus votée. La règle de classification d'un nouvel exemple x est donnée par l'équation (3.11) :

$$K^* = \underset{(1 \leq k \leq K)}{\text{Arg}} (\text{Max} H_k(x)) \quad (3.11)$$

Malheureusement, la fonction 3.11 peut être vérifiée pour plusieurs classes, ce qui produit des zones d'indécisions. La méthode de vote affecte dans ce cas, un exemple aléatoirement à l'une des classes les plus votées.

(Figure 3.9) représente un exemple de classification de trois classes avec la zone d'indécision.

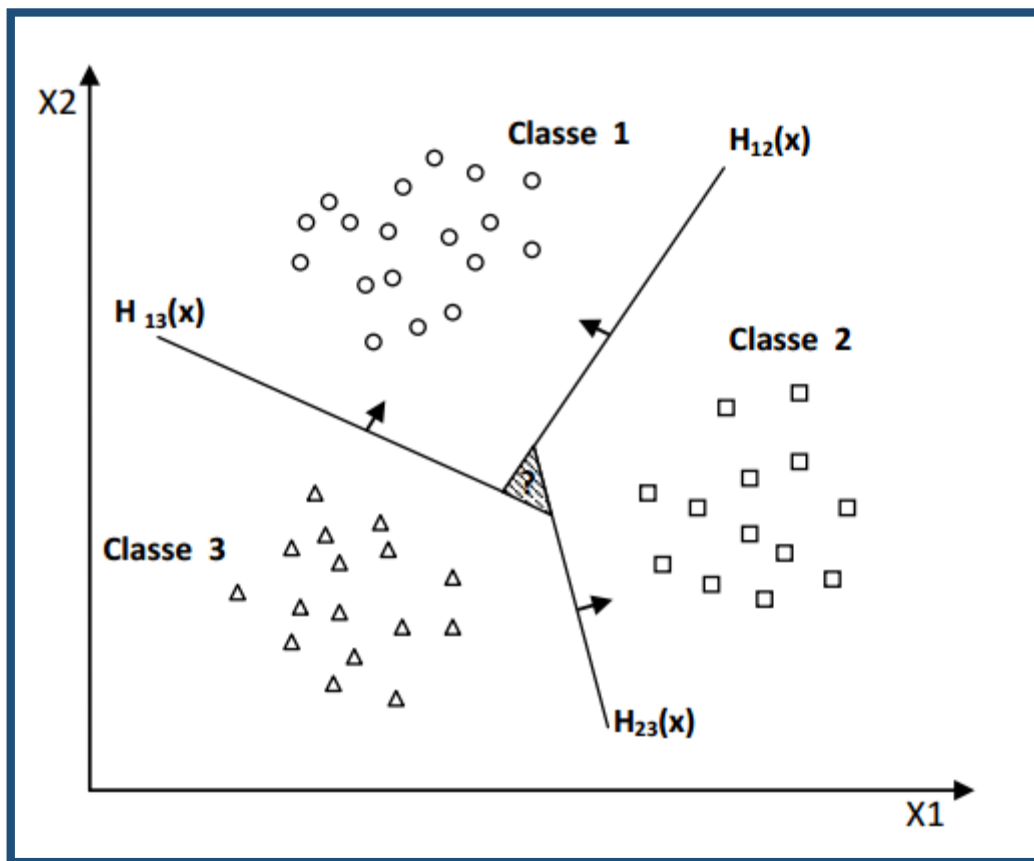


Figure 3.9 : Approche une-contre-une.

Bien que La méthode 1vs1 utilise, pour l'entraînement, un nombre plus important d'hyperplans que la méthode 1vsR, elle est souvent plus rapide. Cela est dû, d'une part, au

nombre limité d'exemples utilisés pour entraîner chaque hyperplan, et d'autre part, à la simplicité des problèmes à résoudre. En effet, chaque deux classes prises à part sont moins chevauchées que toutes les classes [20].

6.5.2 Approche Un contre tous (One versus All)

C'est la méthode la plus simple et la plus ancienne. Selon la formulation de Vapnik, elle consiste à déterminer pour chaque classe k un hyperplan $H_k(w_k; b_k)$ la séparant de toutes les autres classes. Cette classe k est considéré comme étant la classe positive (+1) et les autres classes comme étant la classe négative (-1), ce qui résulte, pour un problème à K classes, en K SVM binaires. Un hyperplan H_k est défini pour chaque classe k par la fonction de décision suivante :

$$H_k(x) = \text{signe} (\langle w_k, x \rangle + b_k)$$
$$= \begin{cases} +1 & \text{si } f_k(x) > 0 ; \\ 0 & \text{sinon} \end{cases} \quad (3.12)$$

La valeur retournée de l'hyperplan permet de savoir si x appartient à la classe k ou non. Dans le cas où il n'appartient pas à k ($H_k(x) = 0$), nous n'avons aucune information sur l'appartenance de x aux autres classes. Pour le savoir, on présente x à tous les hyperplans, ce qui donne la fonction de décision de l'équation (3.13) suivante :

$$K^* = \underset{(1 \leq k \leq K)}{\text{Arg Max}} (H_k(x)) \quad (3.13)$$

Si une seule valeur $H_k(x)$ est égale à 1 et toutes les autres sont égales à 0, on conclut que x appartient à la classe k . Le problème est que l'équation (3.13) peut être vérifiée pour plus d'une classe, ce qui produit des régions d'ambiguïté, et l'exemple x est dit non classifiable. La (Figure 3.10) représente un cas de séparation de 3 classes.

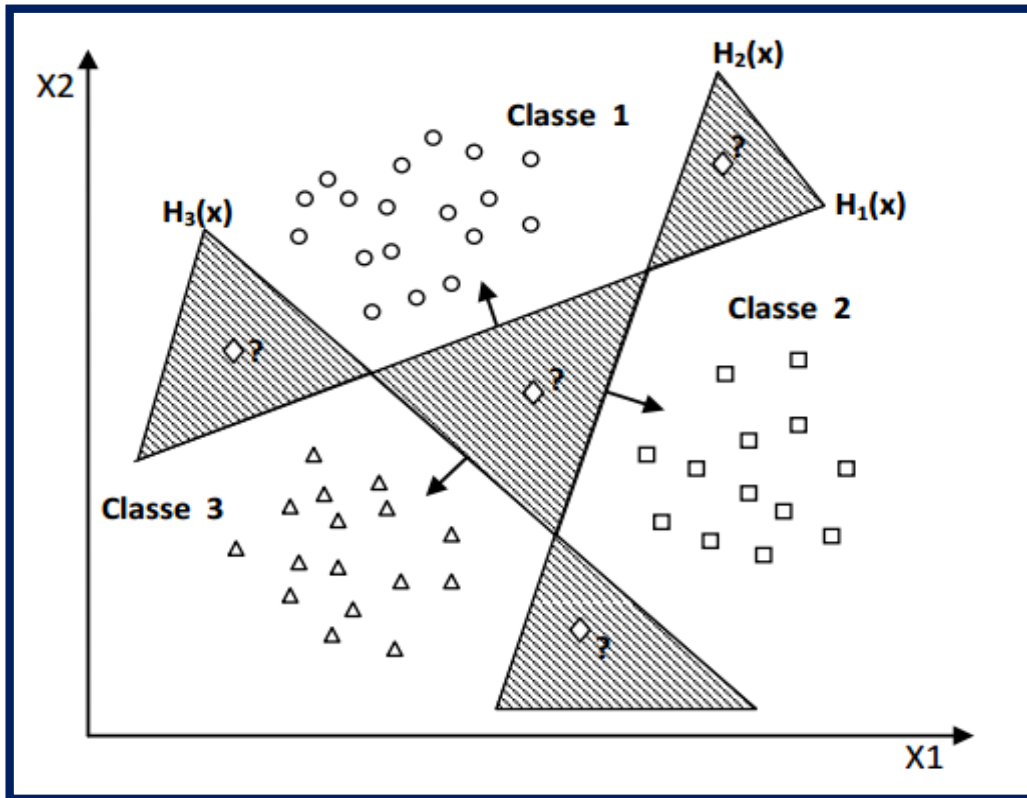


Figure 3.10 : Approche une-contre-reste avec des zones d'indécision

Pour surmonter cette situation, la méthode 1vsR utilise le principe de "le gagnant prend tout" ("winner-takes-all") : la classe k retenue est celle qui maximise

$f_k(\mathbf{x}) = \langle \mathbf{w}_k, \mathbf{x} \rangle + b_k$ de l'équation (3.14).

$$K^* = \underset{(1 \leq k \leq K)}{\text{Arg Max}} (\langle \mathbf{w}_k, \mathbf{x} \rangle + b_k) \quad (3.14)$$

Géométriquement interprétée, tout nouvel exemple \mathbf{x} est affecté à la classe dont l'hyperplan est le plus loin de \mathbf{x} , parmi les classes ayant $H(\mathbf{x}) = 1$ (Figure 3.11).

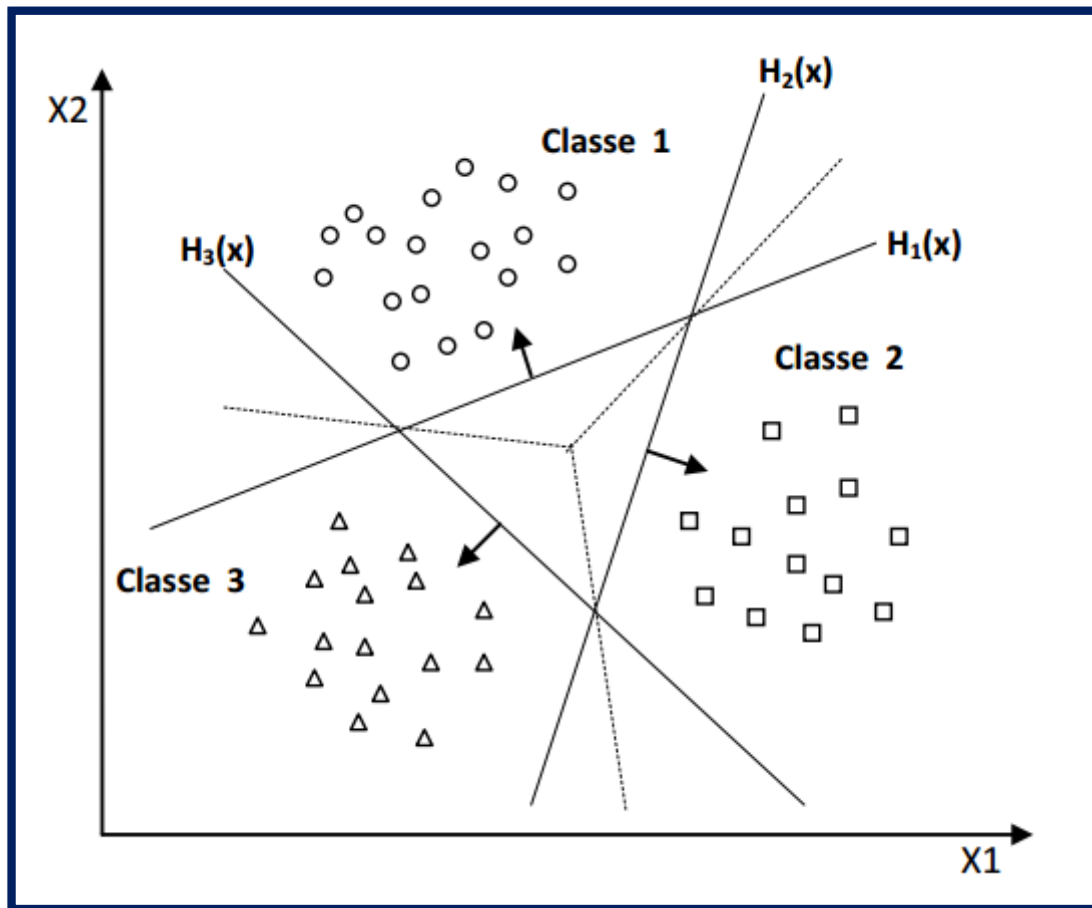


Figure 3.11 : Résolution des cas d'indécision dans la méthode 1vsR.

Souvent, la méthode 1vsR est critiquée à cause de son asymétrie puisque chaque hyperplan est entraîné sur un nombre d'exemples négatifs beaucoup plus important que le nombre d'exemples positifs. Par exemple dans le cas de l'OCR, le classifieur du caractère 'A' est entraîné sur des exemples positifs représentant 'A' et des exemples négatifs représentant tous les autres caractères. La méthode une contre une suivante est une méthode symétrique qui corrige ce problème [20].

1. Graphe de décision

C'est une méthode qui a été développée pour résoudre le problème des zones d'indécision dans la méthode 1vs1. Premièrement, l'entraînement est effectué par la même méthode 1vs1 de la section précédente pour obtenir $K(K-1)/2$ hyperplans. Puis, au lieu d'utiliser le vote pour l'affectation des nouveaux exemples, on construit un graphe de décision. Pour cela, on définit une mesure E_{ks} de la capacité de généralisation sur les différents hyperplans obtenus c.-à-d. pour chaque paire de classes. Cette mesure représente le rapport entre le nombre de vecteurs supports de l'hyperplan et le nombre d'exemples des deux classes correspondantes l'équation (3.15) :

$$E_{ks} = \frac{N_{vs}}{N_{exemples}} \quad (3.15)$$

Après la phase d'apprentissage on construit un graphe de décision qui sera utilisé pour la classification selon les étapes suivantes :

1. Créer une liste L contenant toutes les classes.
2. Si L contient une seule classe, créer un nœud étiqueté de cette classe et arrêter.
3. Calculer pour chaque paire de classes (i, j) la capacité de généralisation E_{ij} de l'hyperplan obtenu dans la phase d'entraînement 1vs1.
4. Rechercher les deux classes k et s dont E_{ks} est maximum.
5. Créer un nœud N du graphe étiqueté de $(k ; s)$.
6. Créer un graphe de décision à partir de la liste $L - \{ k \}$, de la même manière, et l'attacher au fils gauche de N .
7. Créer un graphe de décision à partir de la liste $L - \{ s \}$, de la même manière, et l'attacher au fils droit de N .

On obtient ainsi un graphe de décision similaire à l'exemple de (Figure 3.12) dont les feuilles sont les classes et les nœuds internes sont les hyperplans :

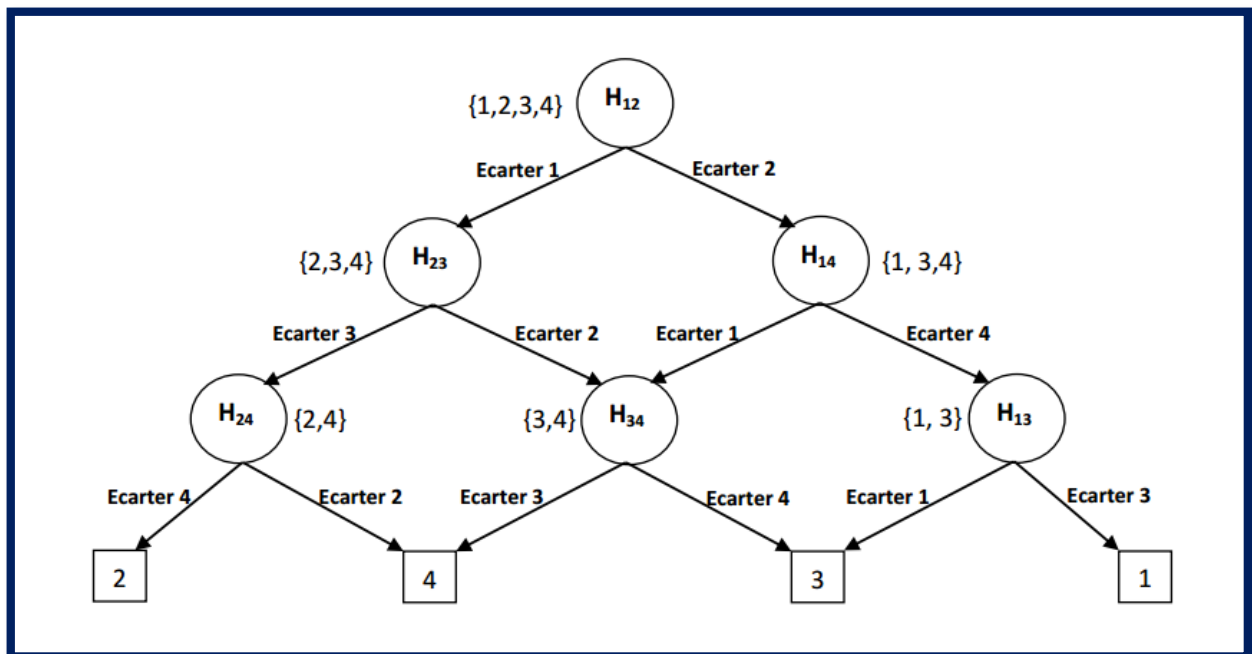


Figure 3.12: Graphe de décision acyclique orienté à quatre classes.

Un nouvel exemple x est exposé premièrement à l'hyperplan de la racine. Si la décision est positive, on continue avec le fils gauche sinon avec le fils droit jusqu'à atteindre une feuille. La

feuille atteinte représente la classe de l'exemple x . Contrairement à la méthode de vote, qui teste pour classifier un exemple $K(K-1)/2$ hyperplans, la méthode DAG en teste uniquement $K-1$, ce qui la rend très rapide en classification par rapport aux méthodes 1vs1 et 1vsR [20].

8. SVMs basées arbres de décision

Dans cette méthode, on apprend pour K classes, $(K-1)$ hyperplans. Chaque hyperplan sépare une ou plusieurs classes du reste, selon un découpage choisi. On peut choisir, par exemple un découpage semblable à la méthode 1vsR où l'hyperplan H_i sépare la classe i des classes $i+1, i+2, \dots, K$ (Figure 3.13)

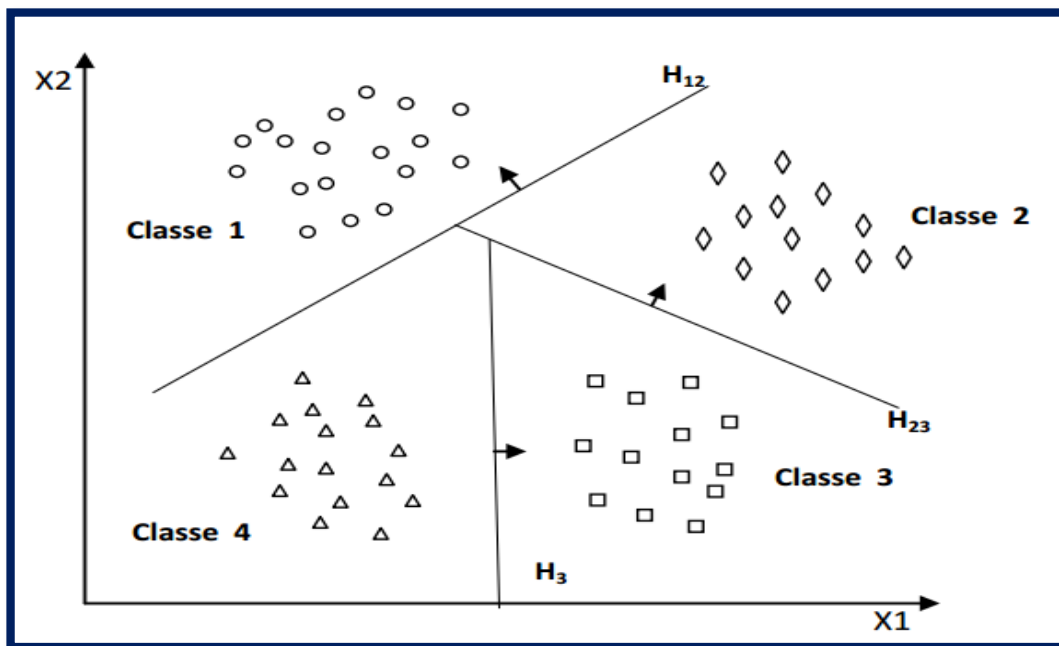


Figure 3.13 : SVM multi classe par arbre de décision.

Dans la phase de classification, pour classer un nouvel exemple x , on teste les hyperplans dans l'ordre croissant et on s'arrête sur le premier hyperplan qui retourne une valeur de décision positive. L'exemple x appartient alors à la classe positive de cet hyperplan.

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'entraînement, des exemples, appelés vecteurs support, qui caractérisent la fonction de séparation. La machine calcule également des multiplicateurs associés à ces vecteurs. 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 [20].

Le schéma de (**Figure 3.14**) résume l'architecture générale d'une SVM dans le cas de la reconnaissance des chiffres manuscrits.

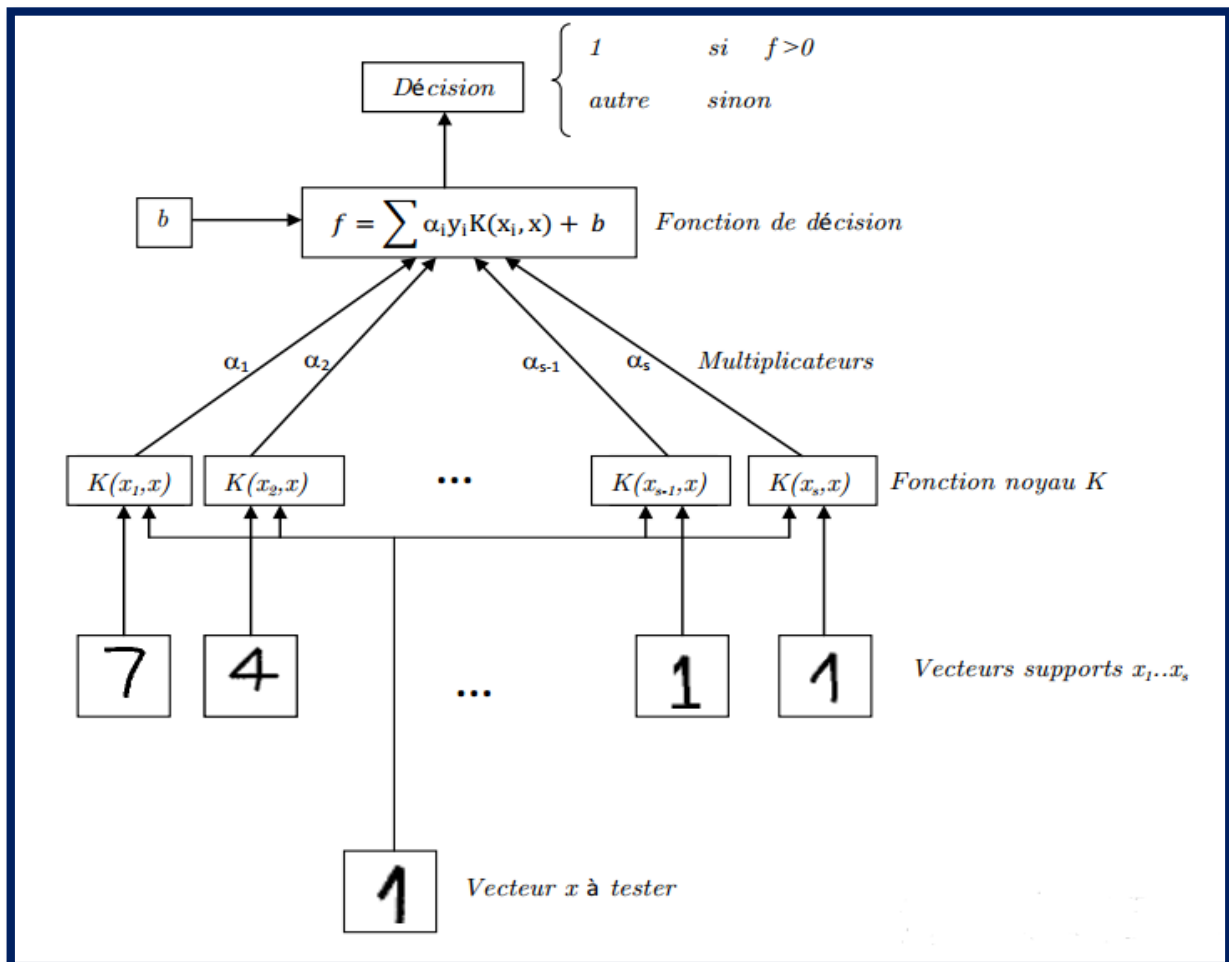


Figure 3.14 : Architecture d'une machine à vecteur support.

La fonction noyau K est utilisée pour calculer la distance entre le vecteur à tester x et chaque vecteur support dans l'espace de caractéristique. Les résultats sont ensuite linéairement combinés en utilisant les multiplicateurs de Lagrange et ajoutés au biais b . Le résultat final f permet de décider à propos du nouveau vecteur : si $f(x)$ est positive, il s'agit du chiffre "1", sinon, il s'agit d'un autre chiffre [20].

10. Utilisation des SVM

Les SVM présentent une bonne précision de classification, même si seulement une quantité modeste de données de formation est disponible, ce qui les rend particulièrement adaptées à une approche dynamique et interactive de la reconnaissance d'expression [18].

Les SVM fournissent des performances de pointe dans les applications du monde réel telles que la catégorisation des textes, la reconnaissance des caractères manuscrits, la classification des images, l'analyse des bio séquences, etc., et sont maintenant définis comme l'un des outils standard pour l'apprentissage et la transmission des machines exploitation minière [18].

Les machines à vecteurs de support ont déjà été utilisées avec succès dans une variété d'applications de classification, y compris la reconnaissance d'identité et de texte ainsi que l'analyse de données sur les microarrays d'ADN [18].

11. Avantages et inconvénients

11.1 Avantages

- Très efficace en dimension élevée.
- Un nombre de paramètres faible à régler.
- Une grande vitesse d'apprentissage.
- Ils sont aussi efficaces dans le cas où la dimension de l'espace est plus grande que le nombre d'échantillon d'apprentissage.
- Traitement des problèmes linéaires ou non linéaires selon la fonction du noyau.
- Un nombre restreint d'échantillons suffit à la détermination des vecteurs supports permettant la discrimination entre les classes. En conséquence, ces algorithmes demandent moins de mémoire [18].

11.2 Inconvénients


- Si le nombre d'attributs est beaucoup plus grand que le nombre d'échantillons, les performances seront moins bonnes.
- Comme il s'agit de méthodes de discrimination entre les classes, elles ne fournissent pas des estimations de probabilités [18].

12. Conclusion

Dans ce chapitre, on a donné une vision générale et une vision purement mathématiques des SVM, on a tenté de présenter d'une manière simple et complète le concept de système d'apprentissage introduit par Vladimir Vapnik : les « Support Vecteur Machine ». Cette méthode de classification est basée sur la recherche d'un hyperplan qui permet de séparer au mieux des ensembles de données.

Les machines à support vecteur, sont un outil récent en pleine expansion. Leurs atouts principaux, résident dans leur capacité de généralisation et l'optimisation qui se fait sans minimum local, et sans phénomène de sur-apprentissage contrairement à d'autres outils plus classiquement utilisés (réseaux de neurones, ...) ce qui en fait un outil intéressant à développer.

Nous présentons dans le chapitre suivant l'implémentation et résultats de notre système de reconnaissance automatique de plaques d'immatriculation.



Chapitre 4

Application et résultats

1 Introduction

Dans ce dernier chapitre, on va citer les phases principales sur lesquelles notre système de reconnaissance de plaque d'immatriculation Algérienne LAPI se repose, tels : Le chargement d'images, le prétraitement, la localisation des plaques, la segmentation des chiffres et enfin leur reconnaissance en utilisant la méthode de classification SVM (Machines à Vecteurs de Support) ainsi que la description de l'environnement matériel et logiciel pour l'implémentation de notre application LAPI.

2 Ressources matérielles et logicielles

2.1 Ressources matérielles

Notre système est développé dans un ordinateur dont les caractéristiques techniques, sont les suivantes (voir Tableau 3.1):

Composante	Description
Processeur	Intel Core i3 CPU 2.00GHz
RAM	4.00 Go

Tableau 4.1: Représentation de caractéristique technique de l'ordinateur de développement.

- ✓ Utilisé pour réaliser notre système de reconnaissance et aussi la création de notre base de données (les corpus).

D'autre part nous avons utilisé une autre PC portable pour la création de base de données

PC portable (Intel Celeron CPU 2.16 GHz, avec 2Go de RAM)

2.2 Ressources logicielles

2.2.1 Environnement informatique

- Système d'exploitation : Windows 10 professionnel 64 bit.
- Langage de programmation utilisé : on a choisi langage Embarcadero XE3 (C++Builder.).

Nous avons développé notre système à l'aide du langage C++ Builder. Qui est un environnement de programmation visuel orienté objet pour le développement rapide d'applications (RAD). En utilisant Embarcadero XE3 C++Builder, on peut, avec un minimum de codage manuel, créer de performantes applications.

Embarcadero XE3 C++Builder est éditeur de logiciel qui met à la disposition des développeurs des outils unifiés pour le développement multiplateformes, la modélisation, l'administration et les données fournit tous les outils nécessaires pour développer.

2.2.2 SVM multiclass

Nous avons utilisé l'implémentation *SVMmulticlass* qui est un moteur SVM développé par Thorsten Joachims en 2008.

SVMmulticlass est une implémentation open-source de référence pour les machines à support vectoriel.

Elle est disponible sur [21] avec documentation, exemples, et références bibliographiques. Le code est implémenté en C et devenu un des moteurs les plus utilisés actuellement. Il possède de nombreuses fonctionnalités et des caractéristiques très attractives pour l'utilisateur :

- Il permet d'utiliser des nombreux kernels prédéfinis, et même d'utiliser des kernels définis par l'utilisateur.
- Il implémente des algorithmes rapides pour l'optimisation en termes de temps de calcul et gestion de mémoire.
- Il fournit tout un ensemble de paramètres qui permettent d'évaluer les performances des SVM après chaque processus d'apprentissage, et plus concrètement :
 - ✓ Il montre une estimation de la dimension de Vapnik-Chervonenkis.
 - ✓ Il calcule la précision sur la base d'apprentissage selon une procédure « leaveone out ».
- Il permet d'aller au-delà de la classification pour aborder des problèmes de régression, et il admet une variante appelée SVM-Struct pour les problèmes multi variés et structurés [14].

3. Caractéristiques des plaques d'immatriculation Algériennes

Comme on s'intéresse aux plaques d'immatriculation algériennes, on commence par donner quelques règles de base pour les plaques d'immatriculation de véhicules en Algérie:

- La plaque est composée de dix chiffres (de 00000 100 01 à 99999 399 48) désignant le numéro d'enregistrement du véhicule.
- Commençant par la droite:
 - Les deux premiers chiffres entre 01 et 48 correspondent à la wilaya.
 - Les deux premiers chiffres (entre 00 et 99) parmi les trois chiffres qui sont au milieu de la plaque correspondent à l'année de fabrication de véhicule. Puis le 3^{ème} chiffre restant (il peut être 1 ou 2 ou bien 3) signifie le type de l'automobile: 1 pour les voitures, 2 pour

Chapitre 4 : Résultats expérimentaux et discussions

Les camions et 3 spécifié pour certains marques de véhicules (par exemple : les marques expressetJ9). [7] - Les cinq derniers chiffres signifient que le véhicule est le (ième) véhicule qui est entré dans la wilaya destinée. [7] (voir Figure4.1)

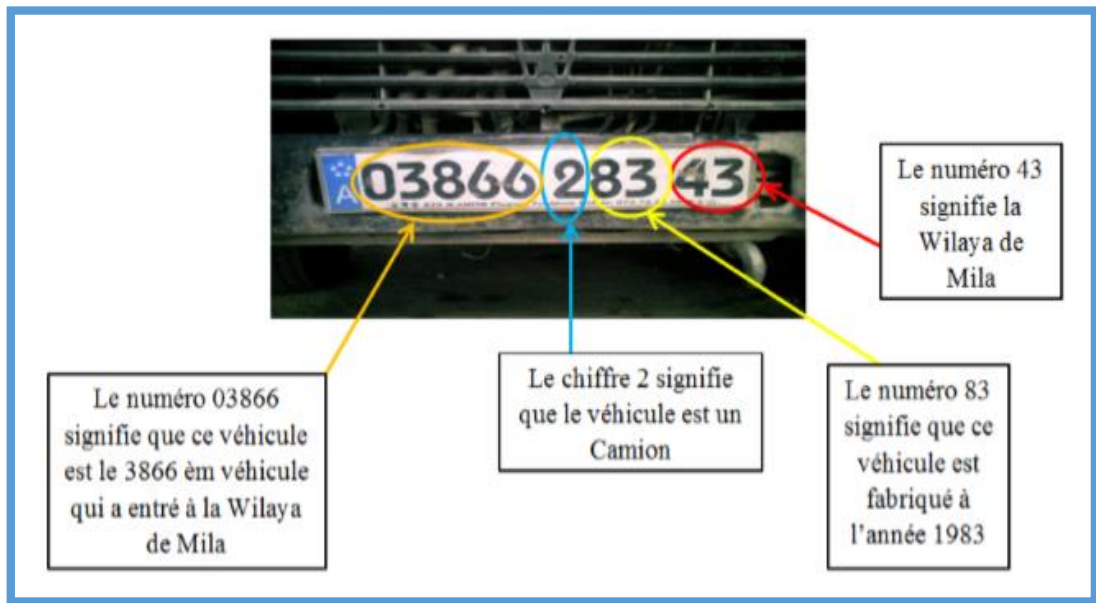


Figure4.1: Composantes de base de l'immatriculation des véhicules Algérienne. [7]

- Récemment les gens aiment ajouter quelques illustrations sur la plaque d'immatriculation de leurs voitures, comme par exemple le drapeau Algérien, le symbole DZ,etc.



Figure4.2: Quelques exemples de plaques d'immatriculation Algérienne

4. Architecture générale de système LAPI

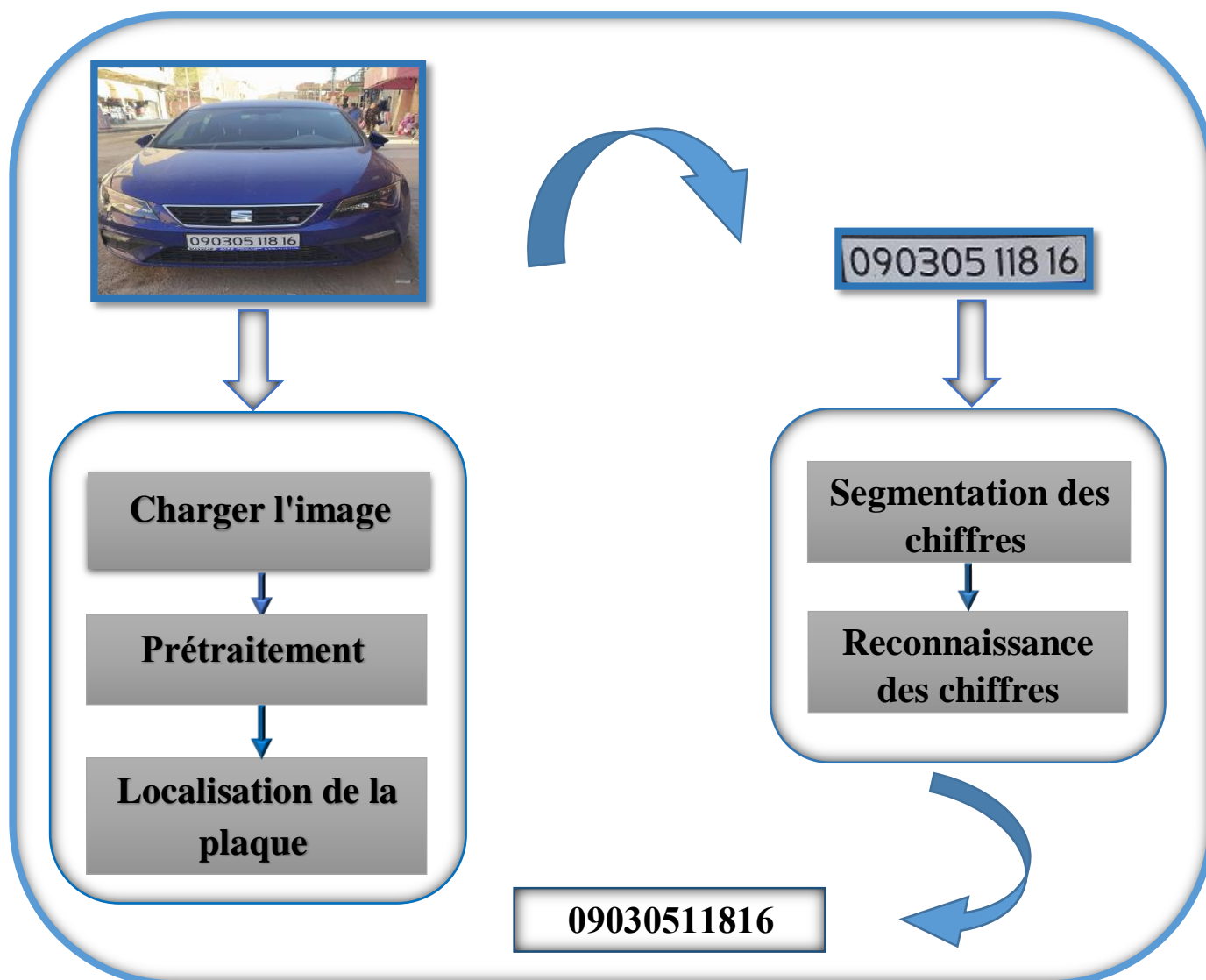


Figure 4.3: Architecture générale du notre système.

5. Les interfaces de l'application

Pour mieux comprendre le fonctionnement de l'application, les figures qui suivent décrivent la chronologie des principales fonctionnalités de l'application.



Figure 4.4 : Interface Système de reconnaissance des plaques d'immatriculation Algérienne.

5.1 Système de détection de plaque

- Description de notre base de données

Lors de l'évaluation des performances d'un système de reconnaissance il est nécessaire de disposer d'une base de données.

Nous avons photographié des voitures à Adrar, certaines dans le centre-ville, d'autres à l'université et certaines dans les palais voisins.

Nous avons construit une base de données contenant 44 images

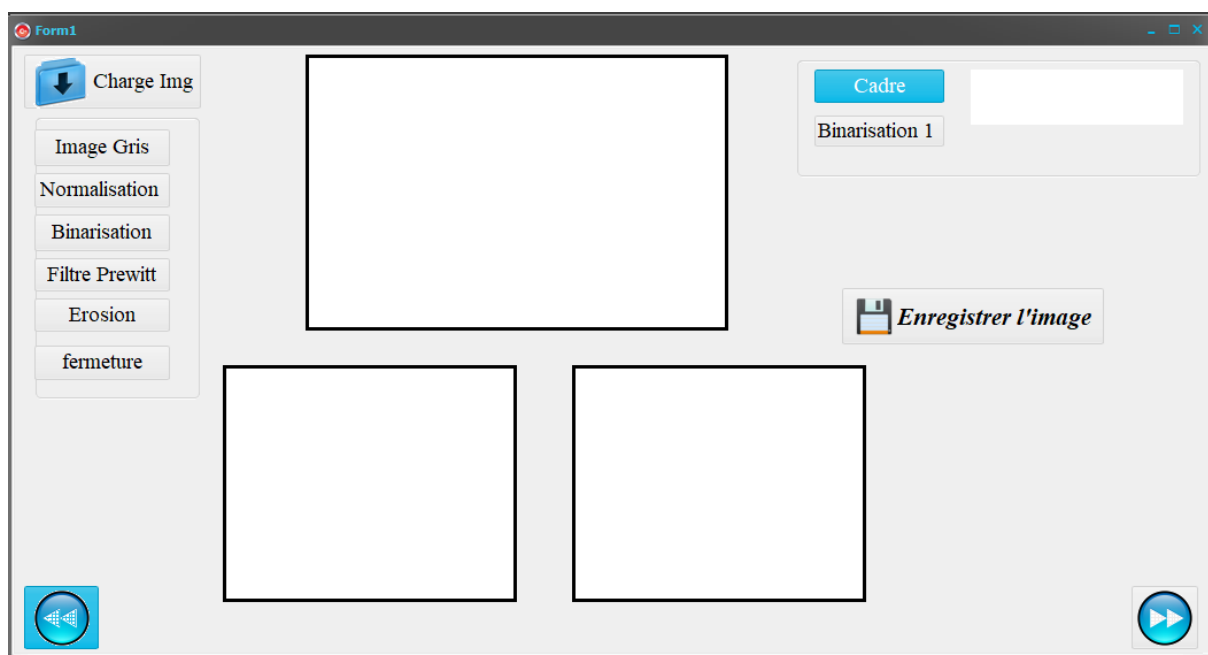


Figure 4.5: Interface Système de détection de plaque.

5.1.1 Charger l'image

Le système LAPI va charger l'image d'un véhicule à partir d'une base de données contenant d'un ensemble de 44 images.

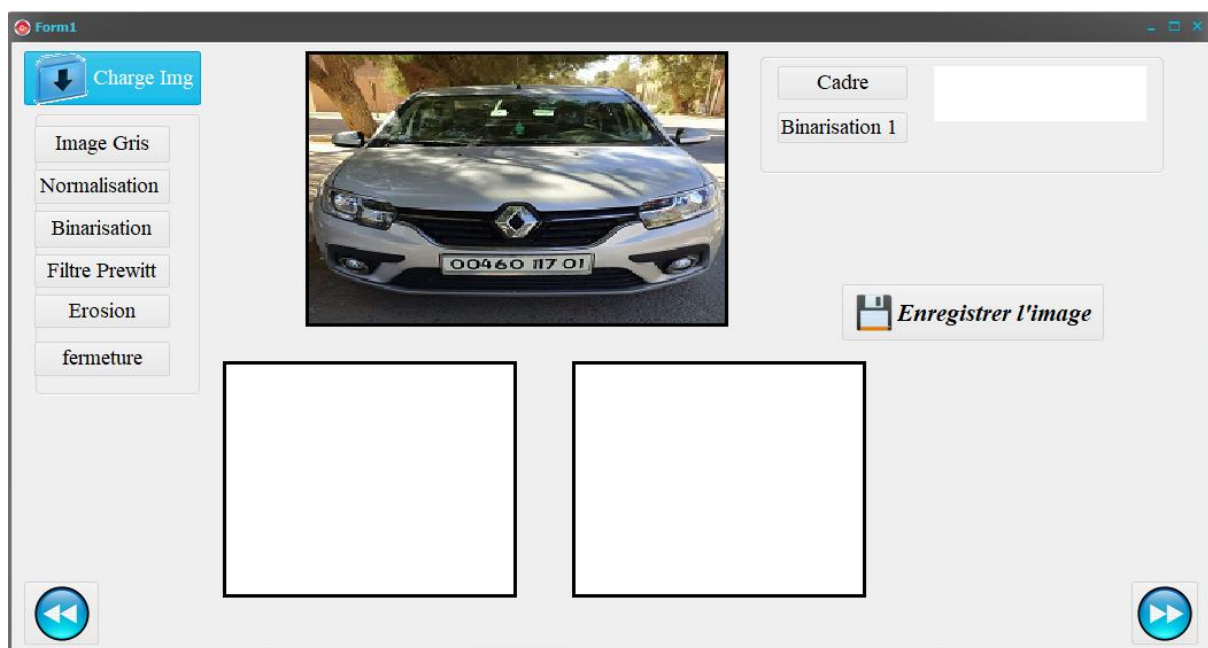


Figure 4.6: Interface de charger l'image.

5.1.2 Prétraitement

Lorsque le chargement de l'image de véhicule est réalisé, le système LAPI va subir une séquence de prétraitements. Généralement, ces prétraitements sont assez classiques en traitement d'image et ont pour but de préparer l'image tracée à la phase suivante d'analyse.

Dans cette étape on procède à une série d'opérations : la normalisation, la binarisation, la conversion de l'image en niveau de gris, et détection de contours.

5.1.2.1 Normalisation

Nous avons effectué cette opération pour éliminer les conditions qui peuvent fausser les résultats, comme la différence de taille. Pour cela nous avons proposé une procédure qui permet de normaliser l'image encadrée dans une dimension de 200*250 pixels. Après la normalisation de la taille, toutes les images se retrouvent définies dans une matrice de même taille.

5.1.2.2 Conversion de l'image en niveau de gris

Après avoir chargé l'image nous passons à un prétraitement de l'image qui consiste à convertir l'image couleur d'un niveau RGB en image avec des niveaux de gris. Cela nous permettra de réduire les nuances et préparer un passage plus sûr vers une image composée que du blanc et du noir.

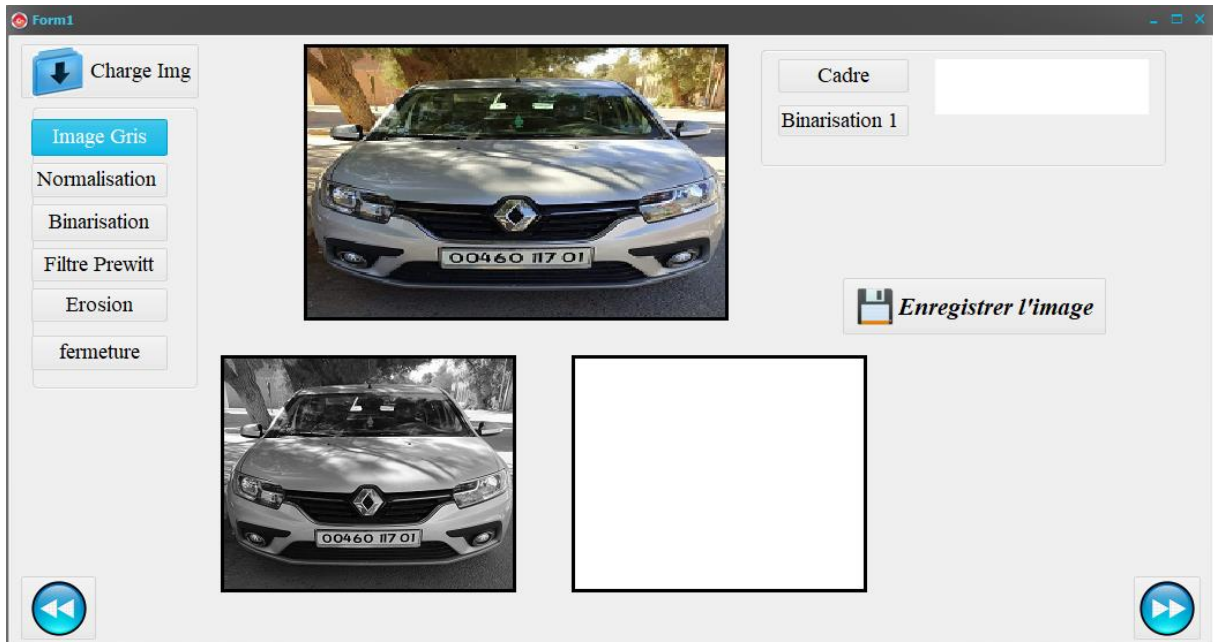


Figure 4.7 : Image originale transformée en images en niveau de gris.

5.1.2.3 Conversion de l'image en binaire

Dans cette partie, l'image sera transformée en noire et blanc. Pour le faire, nous avons fixé un seuil $S1$ ($S1=10777215$). Si on considère $f(i)$ comme la valeur en niveau de gris du pixel i , alors on pose la condition suivante :

$$\left\{ \begin{array}{l} \text{si } f(i) > s1 \text{ alors } f(i) = 0 \\ \text{sinon} \\ f(i) = 1 \end{array} \right.$$

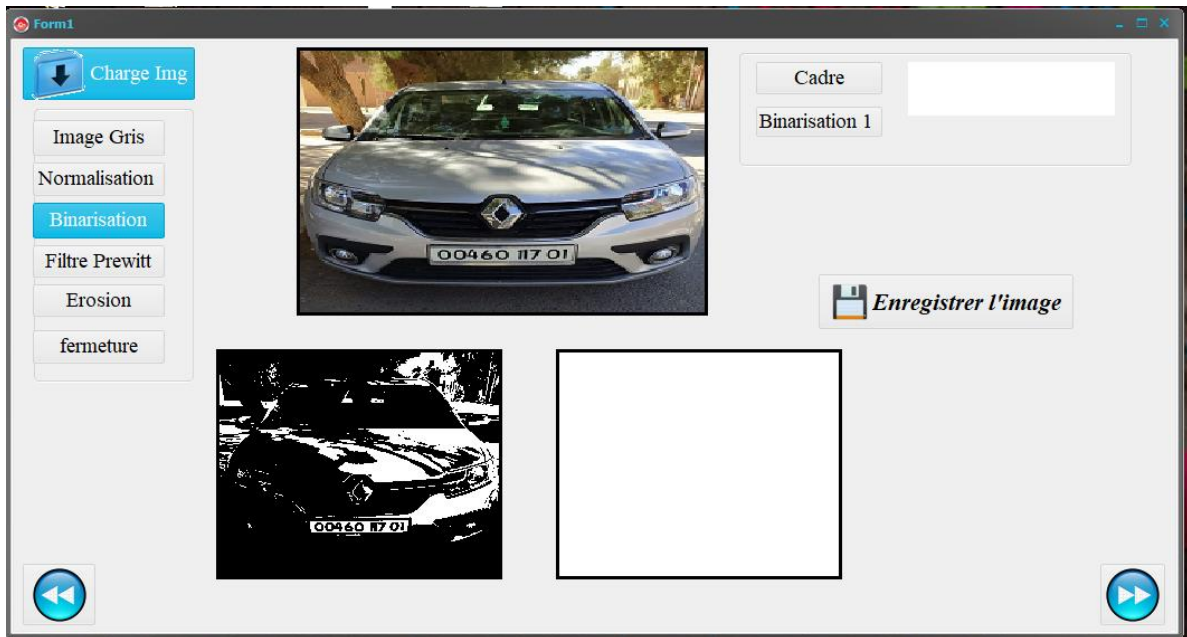


Figure 4.8: Application la binarisation sur une image en niveau de gris.

5.1.2.4 Détection de contour

La détection de contours en traitement d'image est une étape essentielle et incontournable car elle permet une réduction importante de la quantité d'information relative à une image, tout en préservant des informations structurelles comme les contours et les frontières des images. Pour faire cela nous allons appliquer un filtre «*Prewitt*», ainsi on obtiendra le résultat suivant comme le montre la figure suivante.

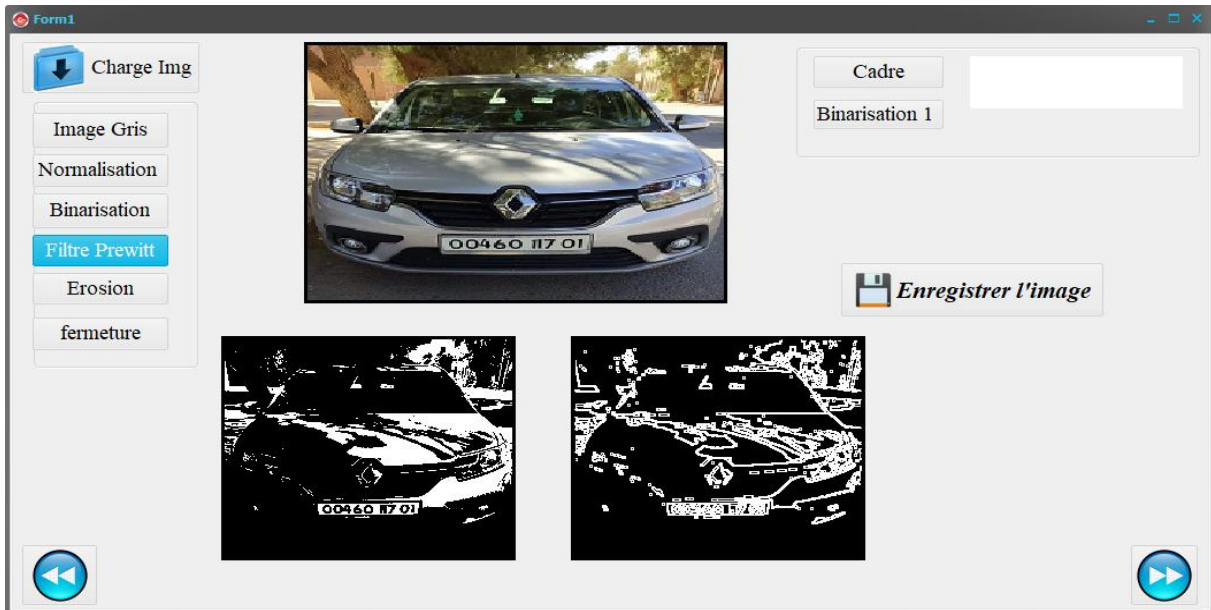


Figure 4.9: Application de l'algorithme de Prewitt sur une image en niveau de gris.

5.1.2.5 Opération morphologique

En appliquant une opération morphologique proche, on peut supprimer des espaces vides et connectez toutes les régions qui ont un nombre élevé d'arêtes. Dans cette étape, nous avons des régions pouvant contenir des plaques.

Dans cette étape on a utilisé l'érosion et fermeture, Le traitement est achevée avec une érosion en motif rectangulaire, ce qui va réparer les coins des rectangles et supprimer Les point apparus dans l'image du a quelques restes de pixels noires des étapes précédentes .Fermeture est Remplissage des trous.

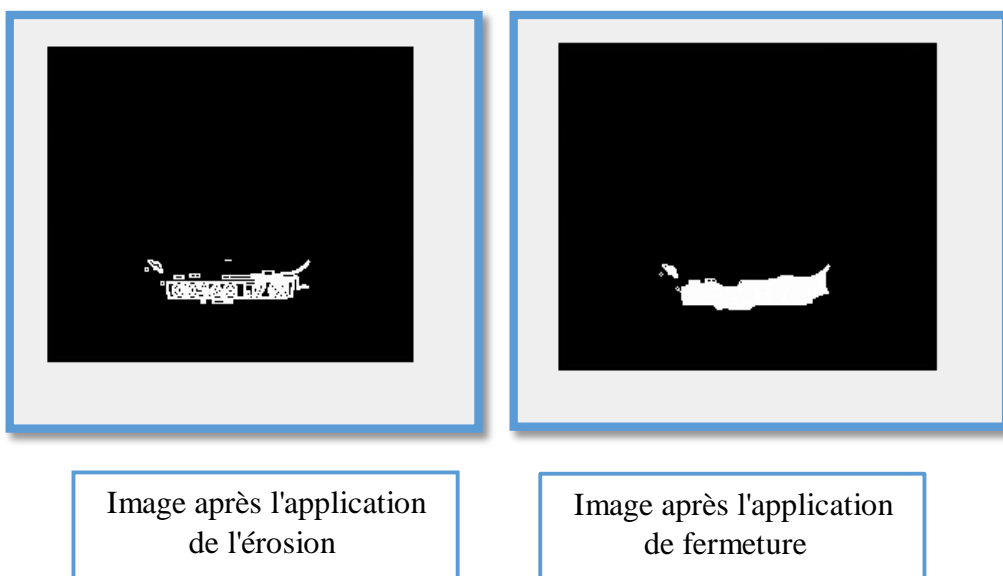


Figure 4.10: Application les opérations de l'érosion et fermeture.

5.1.3 Localisation de la plaque

La localisation et l'isolement de la plaque est la phase la plus importante et la plus complexe, elle détermine la rapidité et la robustesse du système. Après les opérations que nous avons fait on obtient la plaque.

Après l'opération morphologique, on peut détecter la région plaque par une analyse statistique. On va utiliser la projection verticale de l'image pour faire ressortir des bandes verticales qui contiennent le plus probablement la plaque.

La projection verticale de l'image, est un graphe qui représente la somme des contours verticaux de l'image suivant l'axe y. La projection horizontale est définie comme la somme des contours verticaux de l'image suivant l'axe x.

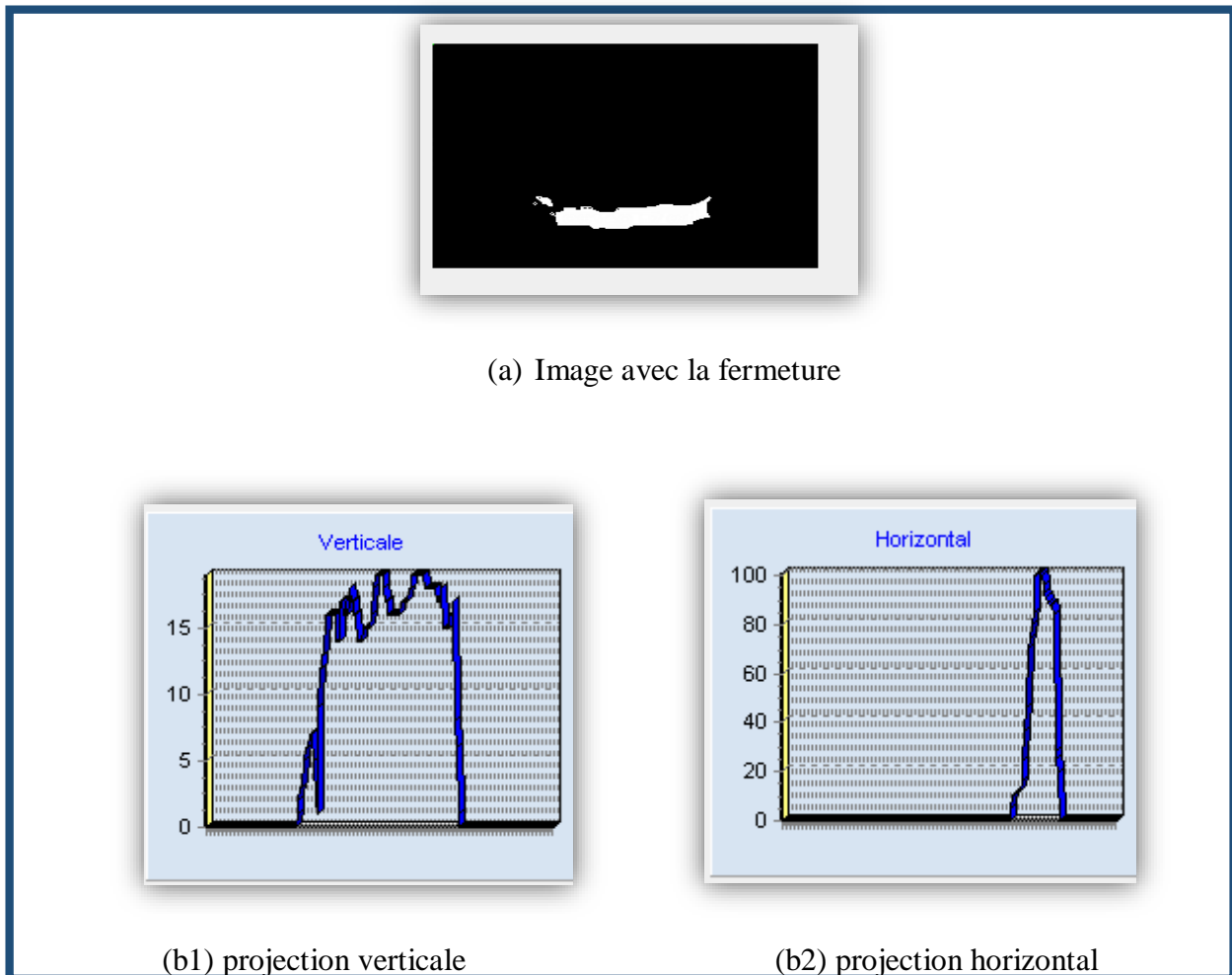


Figure 4.11: Projection verticale et horizontale de l'image d'opération morphologique.

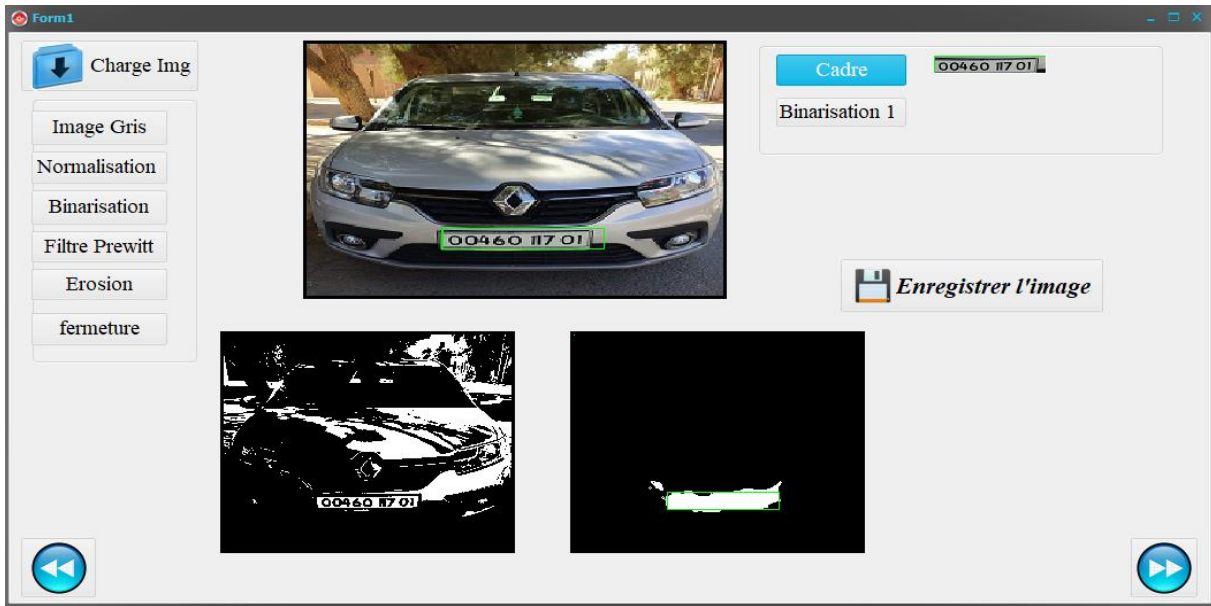


Figure 4.12: Localisation de plaque de matricule.

5.2 Système de segmentation

La plaque une fois extraite va être segmenté en séparant les chiffres. La segmentation traditionnelle basée sur les caractéristiques de la projection verticale des caractères en binaire et sur l'extraction de composants connexes est faisable et valable pour l'image de bonne qualité, mais cette méthode repose trop lourdement sur les résultats précédents de localisation de la plaque [7].

- Recadrage de la plaque :

Pour une meilleure segmentation, nous avons pensé à un recadrage en blanche de la plaque pour fusionner le contour avec le reste.



(a) Plaque avec recadrage

(b) Plaque sans recadrage

Figure 4.13: la plaque d'immatriculation en caractères avec et sans recadrage.

5.2.1 Segmentation basée sur la projection et les composants connexes

Il existe deux types de segmentation des caractères : la segmentation horizontale et la segmentation verticale.

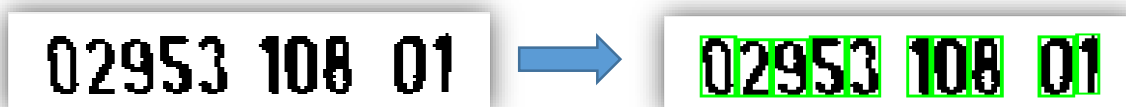
5.2.1.1 Segmentation horizontale

Elle est principalement utilisée pour filtrer les bruits des frontières supérieures et inférieures, supprimer certains rivets et localiser les limites supérieures et inférieures de la région des caractères de la plaque d'immatriculation.

5.2.1.2 Segmentation verticale

Elle est principalement utilisée pour filtrer les bruits des frontières gauche et droite, supprimer certains rivets entre les caractères et segmenter tous les caractères de la plaque d'immatriculation sans erreur.

- La méthode de segmentation que nous avons la supposé est assez simple. En effet, le système LAPI va séparer les caractères selon les pixels des objets les plus connectés puis il va tracer ces derniers par des rectangles.
- Le système LAPI a pu segmenter 34 plaques parmi 36 avec un taux de segmentation 98.74% ce que signifie que la méthode de segmentation utilisée est considérablement fiable.
- Le processus de segmentation est montré dans la figure ci-dessous



(a) Plaque d'immatriculation détectée

(b) Caractères segmentés.

Figure 4.14: Processus de segmentation des caractères de plaques d'immatriculation.

5.3 Système de reconnaissance de chiffre

5.3.1 Description de notre base de données

Lors de l'évaluation des performances d'un système de reconnaissance il est nécessaire de disposer d'une base de données.

Après la détection de la plaque d'immatriculation, Nous avons capté chaque chiffre dans la plaque. Nous avons construit une base de données contenant 400 images.

Les systèmes	Images
Nombre des plaques	36
Nombre des classes	10
Nombre de chiffre pour chaque classe	40

Nombre totale des images	400
Nombre des images pour la Base d'apprentissage	300
Nombre des images pour le test	100

Tableau 4.2: Description de notre base de données.

N° Classe	Des échantillons de la base de données
1	0
2	1
3	2
4	4
5	3
6	5
7	6
8	7
9	8
10	9

Tableau 4.3: Des échantillons de notre base de données.

5.3.2 L'extraction des primitives

C'est l'une des étapes les plus délicates et les plus importantes en OCR. La reconnaissance d'un caractère passe d'abord par l'analyse de sa forme et l'extraction de ses traits caractéristiques (primitives) qui seront exploités pour son identification.

Nous avons utilisé la technique de matrice de distribution pour que nous présentions dans la section suivante.

5.3.2.1 Construction de la matrice de distribution

La construction de la matrice de distribution est une phase très importante dans notre système. Pour une matrice de distribution de taille $N \times N$, le principe consiste à superposer une grille ($N \times N$) sur l'image du caractère et calculer pour chacune des cellules $[i,j]$ résultantes le nombre de pixels noir, puis attribuer ce nombre à la case $[i,j]$ de la matrice de distribution .

Nous considérons par exemple la représentation de chiffre « 9 » dans sa forme isolée, sur une matrice de distribution de 4×4 .

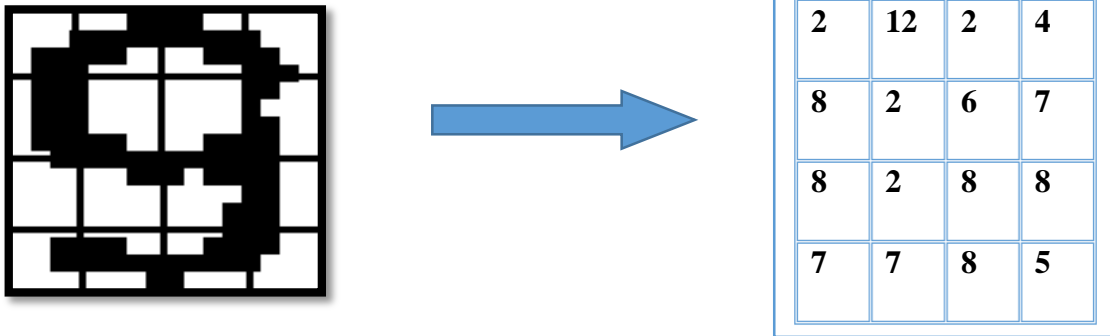


Figure 4.15 : Exemple de matrice de distribution (4×4) de chiffre "9".

Pour le système proposé nous avons utilisé une matrice de distribution de taille 4×4 et l'image de chiffre est de taille 16×16 après la normalisation. Nous présentons ci-après l'algorithme qui permet de construire la matrice de distribution

Algorithme de construction de la matrice de distribution

```

M : Image de chiffre 16 x 16;
NB : compteur;
D : Matrice de distribution 4 x 4 ;
I, J, L, H: Entier;
L = 0 ; H = 0;
Tant que (L < 16 ) faire
    Tant que (H < 16 ) faire
        NB = 0;
        Pour (I = L jusqu'à L + 4 )
            Pour (J = H jusqu'à H + 4 )
                Si pixel M [I, J] est noire alors NB = NB + 1;
            D [L div 4 , H div 4 ] = NB;
            H = H + 4 ;
        L = L + 4 ;
    
```

5.3.2.2 Structure de vecteur de caractéristique

Dans le système de reconnaissance le vecteur de caractéristique est de taille 16, il représente les valeurs de matrice de distribution (4x4).

- **Exemple :**

Les cases	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
vecteur	8	15	15	8	13	0	1	13	14	0	0	13	8	15	15	7

Tableau 4.4: Vecteur des caractéristiques de chiffre "0" Zéro.

5.3.2.3 Corpus

Le corpus est un fichier texte possède une structure particulière. Chaque ligne du corpus représente le vecteur de caractéristique d'une image, la première valeur de la ligne est le numéro de la classe, ensuite chaque valeur de vecteur de caractéristique est précédé par un index (Index: Valeur).

Nous avons présenté ci-dessous une ligne de corpus pour l'image "3" qui a le numéro de la classe 3.

```
3 1:11 2:0 3:4 4:6 5:9 6:4 7:1 8:11 9:11 10:16 11:12 12:11  
13:2 14:4 15:8 16:3
```

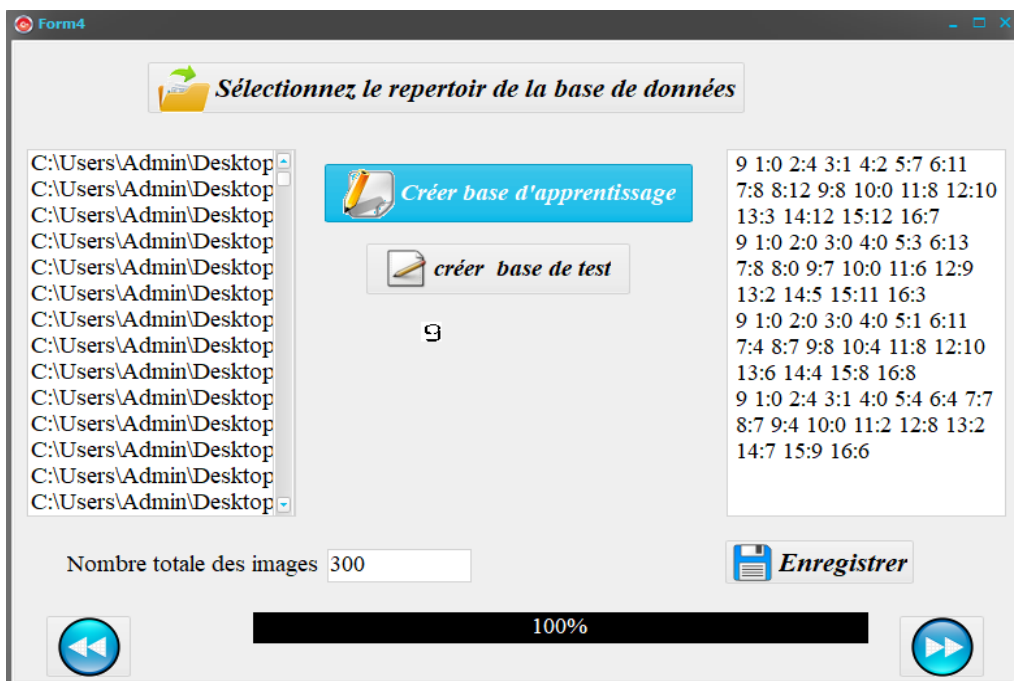


Figure 4.16 : Interface de création des corpus.

6 Expérimentations et Résultats

Dans cette phase, on a utilisé la méthode de classification : Machines à Vecteurs de Support(SVM).

- **Approche:** on a utilisé l'approche un contre une.
- **Type de noyau:** RBF

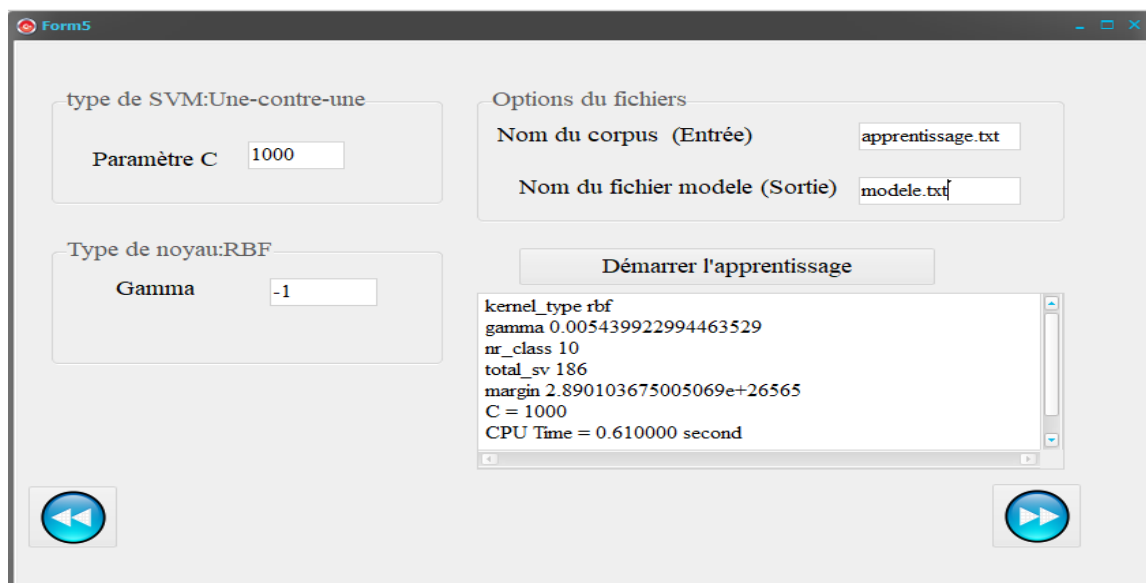


Figure 4.17: Interface de démarrer l'apprentissage.

- Le taux de reconnaissance de système de reconnaissance des chiffres, nous obtenons 98%. (voir Figure 4.18).

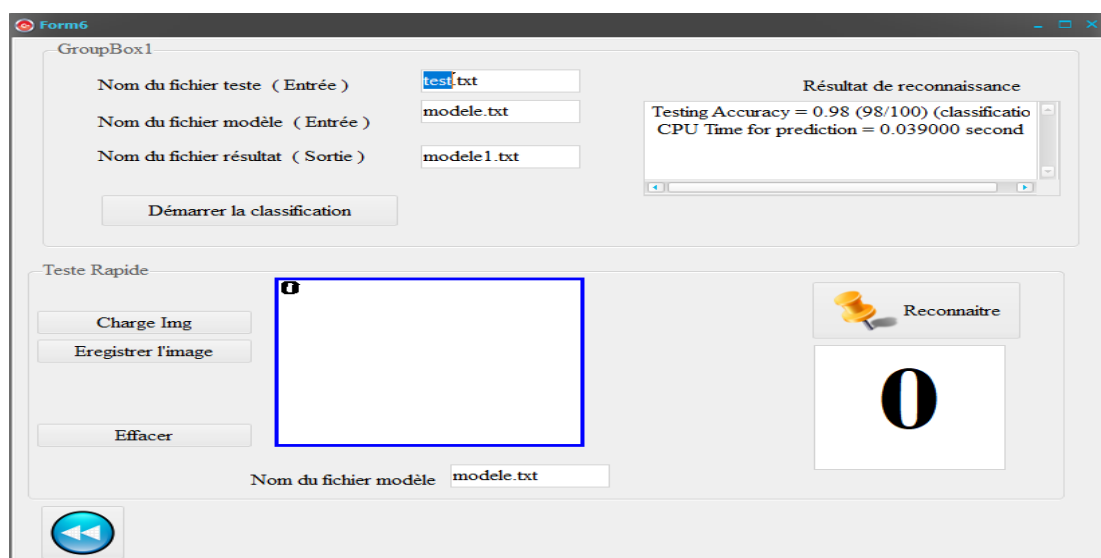


Figure 4.18 : Interface de la classification.

7 Conclusion

Dans ce chapitre, nous avons présenté un système à base de SVM pour la reconnaissance des plaques d'immatriculation Algérienne, en se basant sur trois étapes principales: localisation de la plaque, séparation des chiffres et finalement la décision.

Ce système a été réalisé en grande partie grâce aux différentes phases de traitement que nous avons utilisé.



Conclusion

Générale

Conclusion générale

Le travail s'inscrit dans le cadre générale de la détection et reconnaissance automatique des plaques d'immatriculation Algérienne plus précisément, il exploite une approche basée sur les machines à vecteurs de support appliquée à la reconnaissance des chiffres imprimés.

Malgré que la localisation et l'isolement de la plaque est la phase la plus complexe .Dans notre système, pour localiser et détecter la plaque d'immatriculation dans l'image de la voiture, nous avons appliqué une série d'opérations telles que : niveau de gris, Binairisation, filtre Prewitt,....

Dans notre système LAPI, nous avons opté la méthode de machine à vecteurs de support pour décider la classe d'appartenance de chaque chiffre.

Après quelques tests effectués sur des plaques d'immatriculation, nous remarquons que la méthode de SVM est efficace pour la reconnaissance. Nous estimons que cette méthode a une bonne performance car notre base d'apprentissage est vraiment trop petite par rapport à ce qu'elle devrait être.

Comme perspectives, nous souhaitons améliorer les résultats obtenus en analysant les cas de défaillances et traiter les



Référence

Bibliographique



Références

- [1] Pr. MESHOUL .S «Approches bio-inspirées pour la reconnaissance de formes», université mentouri de constantine faculte des sciences de l'ingenieur departement d'informatique.
- [2] Mr. Alain LIORET« La reconnaissance de Forme Comment améliorer les techniques de reconnaissance de forme 3D ? », Mémoire de Recherche Le 06/06/2010 à Paris.
- [3] LAETITIA LEYRIT «Reconnaissance d'objets en vision artificielle : application à la reconnaissance de piétons», Université Blaise Pascal - Clermont-Ferrand II, 2010. Français.
- [4] M.DJABEUR DJEZZAR,F.BENKADA «Mise au Point d'une Application de Reconnaissance de Formes»,Université Abou Bakr Belkaid– Tlemcen Faculté des Sciences Département d'Informatique2016-2017.
- [5] I.NOR, S.SIDHOUM «Développement d'une application de détection et de reconnaissance de plaques d'immatriculation(LAPIA) », Université Abou Bakr Belkaid– Tlemcen Faculté des Sciences Département d'Informatique 2016-2017.
- [6] Aboubacar Beidari Abdoul Nasser;Bourezg Mohamed «Développement d'une application de fiable détection et de reconnaissance de plaques d'immatriculation», Université Abou Bakr Belkaid– Tlemcen Faculté des Sciences Département d'Informatique 2017-2018.
- [7] AKACEM, RAHMANI «Système de reconnaissance des plaques d'immatriculation Algérienne», Université d'Adrar Faculté des Sciences et de la Technologie Département des Mathématiques et Informatique 2014-2015.

- [8] Belmerabet Sarra, Bardjak Nawal Devant «Segmentation d'image», Université Larbi Ben M'hidi Oum El Bouaghi Faculté des Sciences Exactes et Sciences de la Nature et de la Vie Département des Mathématiques et Informatique 2016-2017.
- [9] S.Fayssl et B.Abdrrahmane « Reconnaissance des chiffres manuscrits par les SVM », Thèse de licence. Université Ahmed Draïa D'Adrar, 2014-2015.
- [10] C. Bahlmann, B. Haasdonk, H. Burkhardt « On-line Handwriting Recognition with Support Vector Machines A Kernel Approach », Publ. in Proc. of the 8th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR), pp. 49–54, Germany, 2002.
- [11] M. Li « Sequence and Text Classification: Features and Classifiers », Thèse de phd, Ecole de sciences d'informatique, Université d'est d'Anglia, Norwich NR4 7TJ, Juillet 2006.
- [12] P. Mahé « Noyaux pour graphes et Support Vector Machines pour le criblage virtuel de molécules », Rapport de stage, DEA MVA 2002/2003, Septembre 2003.
- [13] Hanoi « Réseaux de neurones pour la reconnaissance des formes », 15 juillet 2005.
- [14] Mr.Mamouni, «Sélection expérimentale de modèles SVM multi-classes application à la reconnaissance des caractères arabes», Thèse de magister. Université des Sciences et des Technologies d'Oran Mohamed BOUDIAF ,2012.
- [15] S. Snoussi maddouri, « Modèle perceptif neuronal à vision globale-locale pour la reconnaissance de mots manuscrits arabes », 2002.
- [16] http://www.tsi.enst.fr/pages/enseignement/ressources/mti/localisation_et_epaisseur3/index.html 07/05/2020
- [17] BAHMED FARAH «reconnaissance automatique de logo de voitures», université des sciences et de la technologie d'oran mohamed boudiaf faculte de genie electrique departement d'electronique... 05/10/2010.

[18] h.ouakkaf belgasmi «reconnaissance automatique des expressions faciales par support vector machine», université larbi ben m'hidi oum el bouaghi faculté des sciences exactes et des sciences de la nature et de la vie...06 juin 2017.

[19] M. ZAIZ Faouzi «Les Supports Vecteurs Machines (SVM) pour la reconnaissance des caractères manuscrits arabes» Université Mohamed Khider – BISKRA...15/07/2010.

[20] A. DJEFFAL «Utilisation des méthodes Support Vector Machine (SVM) dans l'analyse des bases de données» Université Mohamed Khider - Biskra ... 2011/2012.

[21] http://www.cs.cornell.edu/People/tj/svm%5Flight/svm_multiclass.html 13/03/2020

