

République Démocratique et Populaires Algérienne
Ministère de l'enseignement supérieur et la recherche scientifique
Centre universitaire de NAAMA
Spécialité informatique

Classification des chiffres manuscrits utilisant SVM

Présenté par :
Fasla Fatima Zahra
Nair Narimane

Encadré par:
Dr.SIDAOUI Boutkhil

Année Universitaire: 2021-2022

Remerciement :

Nous remercions ALLAH de nous avoir donné toutes les capacités et qualifications pour continuer notre chemin d'étude et accomplir ce travail.

Nous tenons à exprimer notre gratitude aux personnes qui nous ont aidé de près ou de loin dans la réalisation de ce mémoire.

En premier lieu, Dr.SIDAOUI BOUTKHIL qui nous a proposé ce sujet. Nous le remercions pour nous avoir encadré, guidé, aidé, conseillé et surtout pour sa patience et sa compréhension. Nous apprécions son esprit critique et ses encouragements qui nous ont permis de compléter ce travail

Nous remercions également les membres du jury qui ont bien voulu examiner et évaluer ce travail .

Un merci spécial à tous les enseignants qui nous ont enseigné tout au long de notre parcours éducatif , sans oublier nos collègues qui nous avons appris et atteint ce point ensemble.

Enfin, nous voudrions remercier nos parents pour leur soutien constant et leurs encouragements

Table des matières

Table des figures	4
Liste des tableaux	5
Introduction générale	6
1 Reconnaissance des chiffres manuscrits	8
1.1 Introduction :	8
1.2 Problèmes de reconnaissance des manuscrits	8
1.3 Schéma général de reconnaissance :	8
1.3.1 Acquisition :	9
1.3.2 Prétraitement d'image :	10
1.3.3 Segmentation	12
1.3.4 Extraction des caractéristiques	12
1.3.5 Classification	13
1.3.6 Reconnaissance	13
1.4 Conclusion	13
2 Machine vecteurs de support(SVM)	14
2.1 Introduction :	14
2.2 Apprentissage automatique	14
2.3 Types d'apprentissage automatique	15
2.3.1 Apprentissage Supervisé	16
2.3.2 Apprentissage Non Supervisé	17
2.4 Techniques d'apprentissage automatique	17
2.5 Machines à vecteurs de support	18
2.5.1 Concepts du base :	19
2.6 Types de classification SVM	19
2.6.1 Linéairement séparable.	20
2.6.2 Non linéairement séparable.	22
2.7 Fonctionnement de Kernel	22
2.7.1 Différentes fonctions du kernel	23
2.8 SVM multiclass	23
2.8.1 One Vs One	23
2.8.2 One Vs All	24
2.9 Conclusion :	24

3	Implementation	25
3.1	Introduction :	25
3.2	Environnement de travail	25
3.2.1	MATLAB	25
3.2.2	LIBSVM	25
3.2.3	MNIST	26
3.3	Architecture de système de classification des chiffres manuscrit	26
3.3.1	Présentation de l'application	27
3.3.2	Expérimentation et discussion	32
3.4	Résultat du classification	34
3.5	Conclusion :	36
	Conclusion générale	37

Table des figures

1.1	schéma général de reconnaissance	9
1.2	les deux types de reconnaissance des manuscrits [10]	10
1.3	binarisation d'une image brute avec un seuil $S= 125$	10
1.4	exemple de reduction du bruit	11
1.5	squelettisation de chiffre 8	12
1.6	Exemple chaîne de chiffre cursif 29	12
1.7	Extraction des caractéristiques d'image de "0"	13
2.1	Relation IA, Apprentissage Automatique et science des données [32]	15
2.2	Schéma de décomposition des approches par apprentissage automatique [7]	16
2.3	Structure de modèle de réseau de neurones artificiels	18
2.4	principe de séparateur SVM	19
2.5	les différents séparateurs que les données peut avoir	20
2.6	les deux cas de séparation : (a)linéaire et (b) non linéaire	20
2.7	L'idée de base du fonctionnement du kernel [38]	22
2.8	Exemple de l'approche OVO	24
2.9	Exemple de l'approche OVA	24
3.1	Certains chiffres de la base de données MNIST	26
3.2	Architecture de notre système de classification des chiffres manuscrits	27
3.3	Interface principale	28
3.4	Interface pour charger les données	28
3.5	Interface pour créer des modèles	29
3.6	Interface pour tester les modèles	29
3.7	l'utilisation de la fenêtre de création des fichiers SVM	30
3.8	La fenêtre pour créer les modèles	31
3.9	La fenêtre pour tester les modèles	32
3.10	graphe du précision basé sur les résultats des modèles qu'on a testés	34
3.11	Moyenne des résultats des tests de deux approches utilisées	35
3.12	Résultats d'accuracy du test de 9 différents modèles	35
3.13	Moyennes des résultats des tests de kernels utilisés	36

Liste des tableaux

3.1	Exemple d'apprentissage de 4 différents modèles du kernel polynomiale	33
3.2	Exemple de test du 4 modèles de kernel polynomiale	33
3.3	Résultats d'apprentissage utilisant le kernel RBF	33
3.4	Resultats de le test des modeles de kernel RBF	33
3.5	Résultats d'apprentissage de 4 modèles de kernel linéaire	34
3.6	Test des modèles linéaire	34
3.7	Resultats d'apprentissage et test de modèles choisis au hasard	36

Introduction générale

Les humains peuvent facilement résoudre et reconnaître n'importe quel problème tel que la distinction des visages, l'identification des objets, la reconnaissance des sons de la langue et l'apprentissage par l'expérience, mais ce n'est pas la même chose dans le cas d'une machine. De nombreuses techniques et méthodes devraient être mises en œuvre pour qu'elle fonctionne presque comme l'humain.

La reconnaissance automatique de l'écriture manuscrite est la transcription de données manuscrites en texte au format numérique, destiné à être utilisé par l'ordinateur. Ce domaine fait l'objet de recherches depuis les années 1950.

D'autre part, la reconnaissance des chiffres manuscrits a une grande importance et un grand usage, tel que la reconnaissance des chiffres de l'écriture manuscrite en ligne sur les tablettes informatiques, la reconnaissance des codes postaux sur le courrier pour le tri du courrier postal, le traitement des montants des chèques bancaires, les saisies numériques dans les formulaires remplis par main, etc...

Puisque les ordinateurs deviennent omniprésents et que de plus en plus de gens l'utilisent, il serait plus bénéfique si les machines peuvent reconnaître l'écriture manuscrite, mais les chiffres manuscrits ne sont pas toujours de la même taille, épaisseur ou orientation et position par rapport aux marges. donc en raison de la variation de forme et d'orientation des chiffres manuscrits, il est difficile pour une machine de les interpréter.

Même ainsi, la recherche dans ce domaine est toujours ouverte. En fait, divers systèmes de reconnaissance ont été proposés dans de nombreuses applications ces dernières années où une précision de reconnaissance élevée est toujours souhaitée [4].

L'apprentissage automatique (Machine Learning en anglais) est un champ qui forme la machine à faire le travail par elle-même sans aucune interaction humaine. C'est un processus d'enseignement du système informatique afin de réaliser des prédictions en lui fournissant des données .

Les algorithmes d'apprentissage automatique utilisent des ensembles de données du monde réel pour former et dériver des modèles de prédictions plus précis et sophistiqués

L'une des principales phases de la reconnaissance est la classification, donc en utilisant l'apprentissage automatique, nous utilisons les techniques de classification.

Il existe de nombreux classificateurs qui peuvent être utilisés comme , Naive bayes , support vector machine (SVM), le voisin le plus proche et les algorithmes d'apprentissage profond comme les réseaux de neurones convolutifs (CNN).

Dans ce travail, nous utilisons l'une des techniques d'apprentissage automatique populaires, d'apprentissage automatique, nommée la machine à vecteurs de support (SVM) pour classer les chiffres manuscrits en 10 classes prédéfinies (0-9)

Ce mémoire est organisé comme suit :

- Dans le premier chapitre nous présentant l'idée générale de la reconnaissance des chiffres manuscrits on indiquant les étapes de reconnaissance ainsi que la classification de l'image numérique.
- Le deuxième chapitre est consacré aux notions d'apprentissage et à la méthode SVM.
- Le troisième chapitre donne des informations générales sur la mise en œuvre de notre travail, l'environnement de travail et la base de données utilisée.

Chapitre 1

Reconnaissance des chiffres manuscrits

1.1 Introduction :

Certains problèmes de reconnaissance reposent sur un grand ensemble de données à traiter. le traitement d'un grand volume de ces données dans une meilleure forme sera très utile pour une reconnaissance plus précise.

Une image en deux dimensions est un très bon exemple de problème avec un grand ensemble de données[6].

Reconnaissance de chiffres manuscrits représente la capacité d'une machine à reconnaître des chiffres manuscrits humains à partir de différentes sources , telles que des images, des papiers, des écrans tactiles, etc. quelle que soit la source, au final, nous aurons des images comme entrée . ces images doivent d'abord être traitées dans le but d'améliorer leurs qualités afin d'en extraire des informations .

1.2 Problèmes de reconnaissance des manuscrits

Lorsqu'il s'agit du contenu manuscrit, la machine peut rencontrer plusieurs obstacles dans l'interprétation des caractères manuscrits, notamment des sources de mauvaise qualité, l'ambiguïté et variations de style d'écriture d'une personne à une autre .

Deux humains n'ont pas exactement la même écriture, et même deux ensembles d'écritures d'une même personne pour un même mot ne sont pas complètement identiques.

D'une personne à l'autre, la variance peut inclure l'inclinaison, la taille, la forme, l'orientation et la cursivité ou disjonction des caractères manuscrits[1].

En ce qui concerne les chiffres manuscrits, cette variété peut également causer des problèmes de similitude lorsque de nombreuses personnes écrivent différents chiffres d'une manière qu'il serait difficile à la machine de les classer en raison de la similitude entre eux, des chiffres tels que 1 et 7, 5 et 6, 3 et 8, 2 et 5, 2 et 9, etc...

1.3 Schéma général de reconnaissance :

la reconnaissance est un processus réalisé par plusieurs étapes qui commence en premier lieu par l'acquisition d'image suivi par un pré-traitement de cette dernière afin d'extraire les informations et qui seront classées .

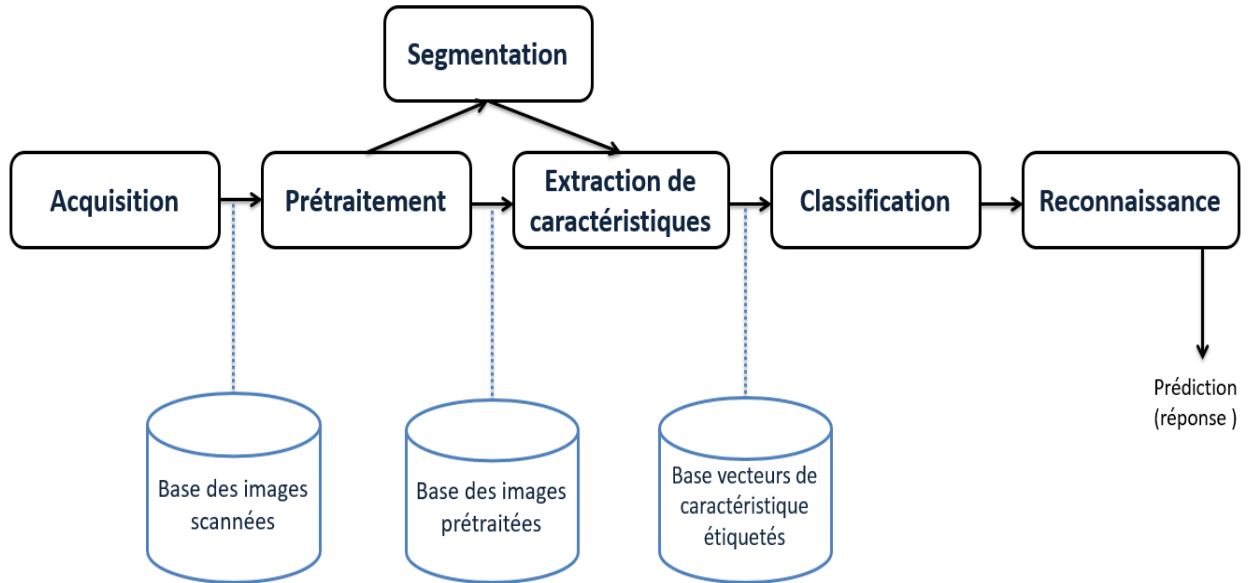


FIGURE 1.1 – schéma général de reconnaissance

1.3.1 Acquisition :

La phase d'acquisition des données consiste à numériser le document où se trouve l'écriture manuscrite pour le mettre sous forme d'image.

Les systèmes de reconnaissance ont deux approches qui sont fondées sur des données d'entrée en ligne et hors ligne.[30] (Figure.1.2).

- Pour la reconnaissance en ligne, les données sont capturées à l'aide d'une tablette électronique et sont enregistrées durant le processus d'écriture [22] donc elle est basée sur les données de trajectoire du stylo dans le temps réel.
- L'écriture hors-ligne, se traduit par une bitmap¹, produite soit à partir d'une caméra soit d'un scanner[22], de sorte que la reconnaissance hors ligne ne repose que sur les données de pixels.

1. une image représentée par un tableau bidimensionnel de valeurs pour les pixels

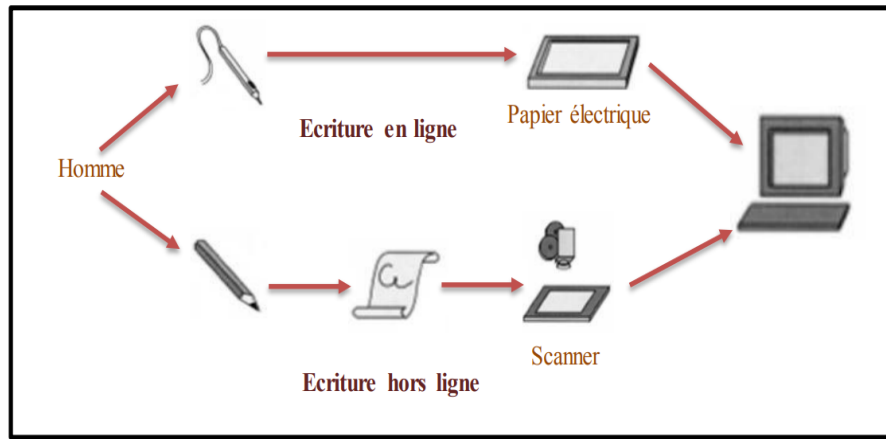


FIGURE 1.2 – les deux types de reconnaissance des manuscrits [10]

1.3.2 Prétraitement d'image :

L'image est soumise à diverses opérations pour l'ajuster afin que les résultats soient plus prêts pour l'extraction d'informations ou pour un traitement ultérieur.

il existe de nombreuses techniques de prétraitement de l'image numérique qui peuvent être utilisées selon la nécessité de l'image telles que La binarisation d'image, la réduction du bruit, la correction de l'inclinaison, la normalisation de la taille d'image, le lissage et la squelettisation.

- **Binarisation :**

La binarisation (le seuillage) consiste à convertir une image en couleur ou en niveau de gris en une image binaire en changeant les valeurs d'intensité d'une image à 0 ou 1, réduisant essentiellement l'information dans l'image de 256 nuances de gris à 2 : noir et blanc, où les pixels dont la valeur est supérieure à un seuil S prennent 1 et le niveau des autres est 0.

Par exemple, dans la Figure.1.3, nous choisissons seuil $S=125$, donc dans cette image chaque pixel qui a une valeur de 125 ou plus prend la valeur de 1 (noir) et chaque pixel qui a une valeur inférieure à 125 prend la valeur de 0 (blanc).



FIGURE 1.3 – binarisation d'une image brute avec un seuil $S= 125$

- **Normalisation :**

La normalisation d'image est un processus de traitement d'image typique qui modifie la plage de valeurs d'intensité de pixel.

Normaliser veut dire transformer d'une certaine façon l'image pour que la comparaison avec d'autres images ait un sens (d'où le terme normalisation) et qu'elles soient toutes structurés de la même façon et la même taille. Notons que certaines formes (caractères, chiffres, mots) sont inclinées ou penchées donc il est nécessaire de normaliser en pente cette forme afin de segmenter la forme (par exemple segmentation d'un mot en lettres). alors la normalisation consiste à corriger la pente d'un mot ou à redresser l'inclinaison des lettres dans un mot afin de faciliter la segmentation. [29]

- **Réduction du bruit :**

Le bruit est causé par un éclairage insuffisant lors de la prise de vue qui se caractérise par l'apparition de grains colorés particulièrement visibles dans les zones sombres.

En traitement d'image, il est souvent indispensable de débruiter l'image car ce bruit est une anomalie qui dégrade sa qualité [28], afin de restituer la vraie image comme dans la Figure 1.4

les images débruitées pourraient inévitablement perdre certains détails. étant donné que le bruit, les contours et la texture sont des composants à haute fréquence, il est difficile de les distinguer dans le processus de débruitage [12].

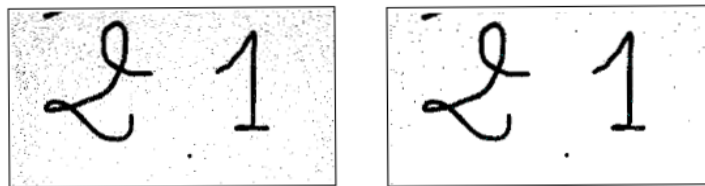


FIGURE 1.4 – exemple de reduction du bruit

- **Squelettisation :**

La squelettisation est une étape de prétraitement fondamentale dans de nombreux algorithmes de traitement d'image et de reconnaissance de formes car les images squelettisées sont plus faciles à traiter et elles réduisent le temps de traitement pour les opérations ultérieures.

Il s'agit d'une opération morphologique utilisée pour supprimer les pixels de premier plan sélectionnés des images binaires.[35] (figure 1.5)



FIGURE 1.5 – squelettisation de chiffre 8

1.3.3 Segmentation

La segmentation d'une image est effectuée pour isoler les caractères de l'image en différentes sous-images. Chaque sous-image est considérée comme un caractère individuel

La segmentation occupe une grande partie du processus de prétraitement et d'extraction des informations mais; elle pose un problème difficile et inévitable dans le domaine de reconnaissance d'écriture manuscrite surtout lorsqu'on a une écriture cursive (Figure 1.6), donc les résultats de la reconnaissance dépendent de la sortie de la segmentation

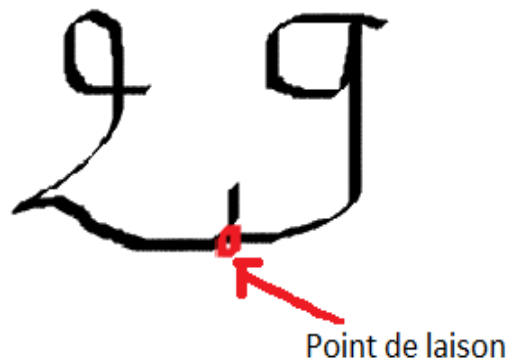


FIGURE 1.6 – Exemple chaîne de chiffre cursif 29

1.3.4 Extraction des caractéristiques

L'extraction des caractéristiques est au cœur de la reconnaissance de caractères [26], elle consiste à transformer une image en un vecteur de caractéristiques (x_1, x_2, \dots, x_n) de taille fixe [15]. (figure 1.7)

Pour la prise de décision, un système de reconnaissance n'a besoin que de l'information pertinente pour différencier un objet d'un autre.[14], donc un bon ensemble de caractéristiques

extraites doit représenter des caractéristiques particulières à une classe[23] et être aussi invariant que possible aux distorsions et variations attendues que les caractères peuvent avoir dans une application spécifique.[37] autrement dit,le type ou le format des caractéristiques extraites doit correspondre aux exigences du classificateur choisi.

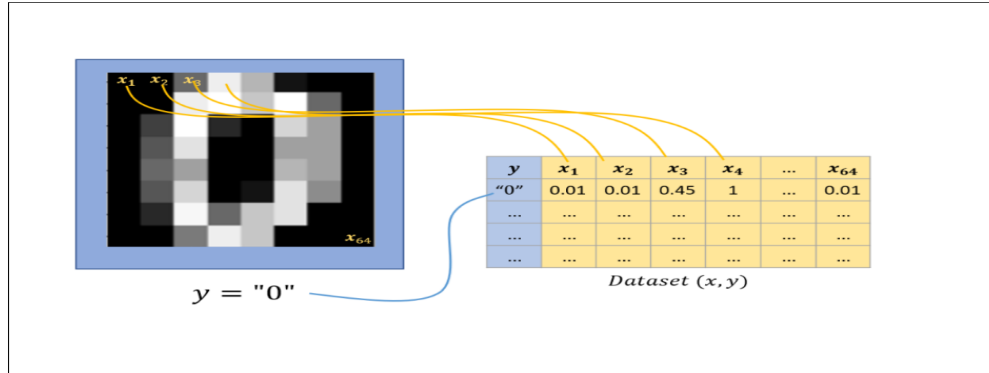


FIGURE 1.7 – Extraction des caractéristiques d'image de "0"

1.3.5 Classification

La classification est une étape très importante dans tout système de reconnaissance car il faut déterminer quelle est la stratégie approprié à adapté pour avoir un bon résultat de classification déterminant la classe d'appartenance de l'objet en entrée, donc le choix du classifieur est très important,il constitue l'élément de décision dans un système de reconnaissance.

1.3.6 Reconnaissance

système de reconnaissance implique l'utilisation d'un modèle entraîné qui reconnaît les unités individuelles de base mentionnées précédemment. Il inclut une comparaison de ces unités individuelles avec chaque classe de référence et la mesure d'un score de similarité (sous forme de distance ou de score de probabilité), Le score de similarité du modèle est utilisé pour décider quelle classe correspond le mieux à l'individu inconnu.

1.4 Conclusion

Dans ce premier chapitre nous avons mis l'accent sur différents problèmes de reconnaissance des chiffres manuscrits. par la suite nous avons présenté le schéma général de reconnaissance que présente le cheminement de plusieurs de plusieurs étapes qui commence par l'acquisition suivi du Pré-traitement d'image et la segmentation, ainsi que d'extraction des caractéristiques , la classification et enfin la reconnaissance qui met en exergue les techniques de l'apprentissage automatique , plus spécifiquement SVM qui sera abordée dans le deuxième chapitre.

Chapitre 2

Machine vecteurs de support(SVM)

2.1 Introduction :

Dans le chapitre précédent, nous avons donné un aperçu des systèmes de reconnaissance des chiffres manuscrits, dans lequel nous avons expliqué les différentes phases de système de reconnaissance manuscrite, y compris la phase de classification. Dans ce chapitre, nous présentons les différentes techniques d'apprentissage automatique en particulier SVM pour les problèmes de classification.

2.2 Apprentissage automatique

L'apprentissage automatique est un champ d'étude fait partie de l'Intelligence Artificielle (IA) est utilisé en science et analyse des données (Analytics and Data Science). (Figure 2.1) qui consiste à laisser l'ordinateur apprendre quel calcul effectuer, plutôt que de lui donner ce calcul , on entend par calcul une programmation explicite, comme l'a dit Arthur Samuel¹, en 1959 : «L'apprentissage automatique est la science qui permet aux ordinateurs d'apprendre sans être explicitement programmé ». c'est-a-dire,les technologies d'apprentissage automatique permettent aux ordinateurs d'apprendre à effectuer une tâche, sans avoir reçu d'instructions explicites.

1. un mathématicien américain et un pionnier du domaine de l'apprentissage automatique qui a développé un programme pouvant apprendre tout seul comment jouer aux Dames en 1959.

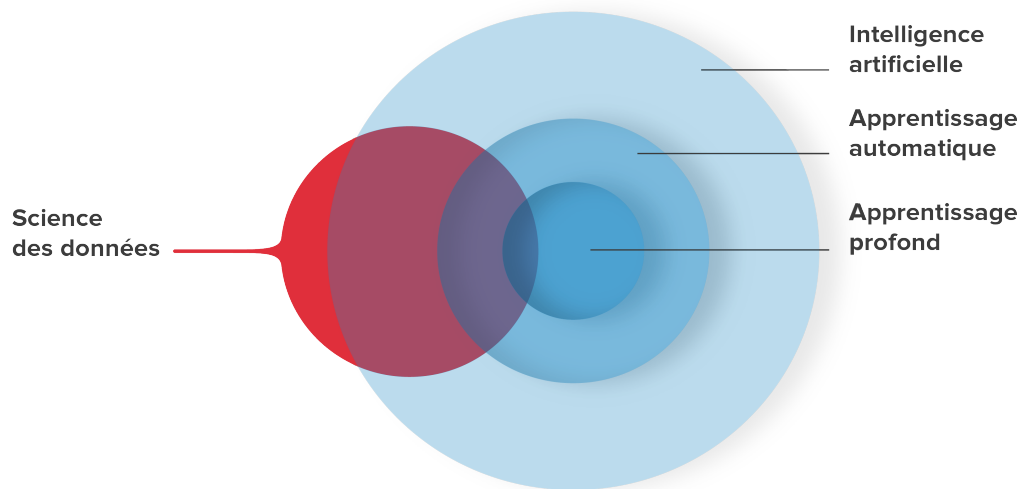


FIGURE 2.1 – Relation IA, Apprentissage Automatique et science des données [32]

L'apprentissage automatique permet d'apprendre à partir de données. L'information est la base de l'apprentissage, les capteurs de l'environnement, données textuelles, les images, données de séries chronologiques dans l'économie ..etc. il existe de nombreux exemples d'informations qui peuvent être utilisées pour l'apprentissage [18].

En 1998, Tom Mitchell² a donné une définition plus moderne de l'apprentissage automatique en disant : «L'apprentissage automatique est l'étude des algorithmes informatiques qui permettent aux programmes informatiques de s'améliorer automatiquement à travers l'expérience ».

À l'instar des humains, les algorithmes d'apprentissage automatique apprennent par l'expérience. Par exemple, après avoir vu de multiples exemples d'un objet, un algorithme d'apprentissage automatique peut devenir capable de reconnaître cet objet dans de nouveaux scénarios inédits.

2.3 Types d'apprentissage automatique

Les algorithmes d'apprentissage peuvent se catégoriser selon le mode d'apprentissage qu'ils emploient. Une machine est capable d'apprendre en utilisant des méthodes d'apprentissage qui s'inspirent fortement de la façon dont les humains apprennent à faire des choses. Parmi ces méthodes, on compte :[34]

- L'apprentissage supervisé (Supervised Learning)
- L'apprentissage non supervisé (Unsupervised Learning)

2. un informaticien américain et professeur de l'Université E. Fredkin à l'Université Carnegie Mellon (CMU)

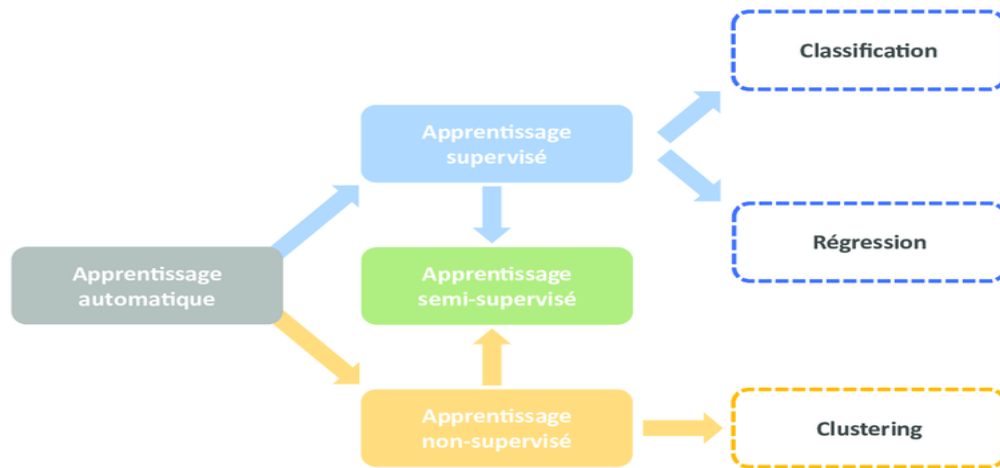


FIGURE 2.2 – Schéma de décomposition des approches par apprentissage automatique [7]

2.3.1 Apprentissage Supervisé

On parle d'apprentissage supervisé lorsque l'on fournit à une machine beaucoup d'exemples qu'elle doit étudier, il nous permet de faire des prédictions basées sur un modèle qui est obtenu à partir d'un algorithme choisi et entraîné sur des données

Pour maîtriser l'apprentissage supervisé, il faut absolument comprendre et connaître les 4 notions suivantes : [34]

- **Dataset :**

Apprendre à partir d'exemples (Dataset) lorsque l'on fournit à une machine beaucoup d'exemples (\mathbf{x}, \mathbf{y}) dans le but de lui faire apprendre la relation qui relie \mathbf{x} à \mathbf{y} .

on compile ces exemples (\mathbf{x}, \mathbf{y}) dans un tableau que l'on appelle Dataset :

- La variable \mathbf{y} porte le nom de **target** (la cible). C'est la valeur que l'on cherche à prédire.
- La variable \mathbf{x} porte le nom de **feature** (facteur). Un facteur influence la valeur de \mathbf{y} et on a en général beaucoup de features $(\mathbf{x}_1, \mathbf{x}_2, \dots)$ dans notre Dataset que l'on regroupe dans une matrice \mathbf{X}

- **Modèle et ses paramètres :**

à partir de ce Dataset, on développe un modèle. Il peut s'agir d'un modèle linéaire, ou bien un modèle non-linéaire.

les paramètres de la fonction (linéaire ou non linéaire) sont les paramètres d'un modèle

- **Fonction Coût :**

On appelle fonction de coût l'ensemble des erreurs renvoyées par un modèle par rapport à l'ensemble de données (Dataset).

Le but est avoir un bon modèle, c'est-à-dire, avoir un modèle qui nous donne de petites erreurs, donc une petite Fonction Coût.

- **Algorithme d'apprentissage :**

Un algorithme d'apprentissage nous permet de trouver les paramètres du modèle qui minimisent la Fonction Coût, l'exemple le plus courant est l'algorithme de Gradient Descent.

Avec l'apprentissage supervisé on peut développer des modèles pour résoudre 2 types de problèmes : **Classification** ou bien **Régression**

- **Classification :**

les données de l'ensemble d'apprentissage appartiennent à deux ou plusieurs classes ou catégories ; alors, les données, déjà étiquetées, nous permettent d'apprendre au système à reconnaître les caractéristiques qui distinguent chaque classe.

Lorsque nous aurons besoin de considérer une nouvelle valeur inconnue du système, le système évaluera sa classe en fonction de ses caractéristiques [31].

- **Régression :**

Dans les problèmes de régression, on cherche à prédire la valeur d'une variable continue, c'est-à-dire une variable qui peut prendre une infinité de valeurs. Le cas le plus simple à comprendre est celui où l'on veut trouver la ligne qui décrit la tendance d'une série de points représentés dans un nuage de points.

2.3.2 Apprentissage Non Supervisé

L'apprentissage non supervisé est le cas où l'ensemble d'apprentissage est constitué d'une série de valeurs d'entrée x sans aucune valeur cible correspondante.

- **Clustering :**

Leur but est de découvrir et regrouper des données selon leur structure. on peut faire clustering utilisant des clusters comme l'algorithme de K-Mean. (voir plus de détails sur le fonctionnement de K-mean ici [27])

- **Réduction de la dimensionnalité :**

La réduction d'un ensemble de données de grande dimension à un seul avec seulement deux ou trois dimensions est utile non seulement pour la visualisation des données, mais aussi pour convertir des données de très haute dimensionnalité en données de dimensionnalité beaucoup plus faible, de sorte que chacune des dimensions inférieures transmet beaucoup plus informations.

En plus de ces deux méthodes principales, il existe un autre groupe de méthodes qui ont pour but la validation et l'évaluation des modèles.(figure2.2)

2.4 Techniques d'apprentissage automatique

Il existe de nombreuses techniques que l'apprentissage automatique utilise pour faire le travail de modèle requis. Parmi ces techniques nous pouvons citer :

- **Réseaux de Neurones**

Les Réseaux de Neurones sont des modèles bien plus complexes que tous les autres modèles de l'apprentissage automatique dans le sens où ils représentent des fonctions mathématiques avec des millions de coefficients (les paramètres)[34] (Figure 2.3).

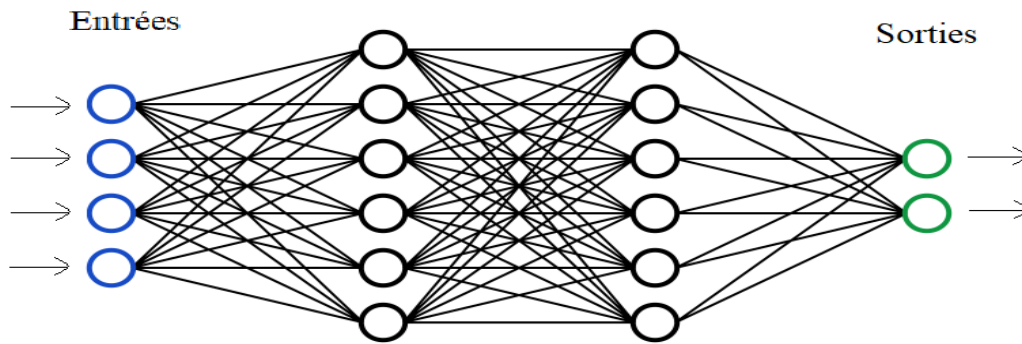


FIGURE 2.3 – Structure de modèle de réseau de neurones artificiels

Il y a un niveau d'entrées (input layer) à gauche, un niveau de sorties (output layer) à droite, et plusieurs niveaux cachés entre deux. Un "neurone" est une fonction algébrique (fonction d'activation) dont la valeur dépend de paramètres appelés coefficients ou poids. Les variables de cette fonction sont habituellement appelées "entrées" du neurone, et la valeur de la fonction est appelée sa "sortie".

Un neurone est donc avant tout un opérateur mathématique, dont on peut calculer la valeur numérique par quelques lignes de logiciel. [11]

- **Machines à vecteurs de support (SVM)**

L'algorithme de la machine à vecteurs de support (SVM), inventé par Vapnik et Cortes en 1998, est un puissant classificateur discriminant qui a été appliqué efficacement à de nombreux problèmes de reconnaissance de formes ou de classification et a obtenu des résultats positifs.

Il est considéré comme l'outil le plus avancé pour résoudre les problèmes de classification linéaires et non linéaires en raison de sa simplicité, de sa flexibilité, de sa capacité de prédiction et de son optimalité globale.

Les techniques de SVM sont basées sur la minimisation du risque structurel au lieu de la minimisation du risque empirique qui est traditionnellement utilisée pour les réseaux neuronaux artificiels.

2.5 Machines à vecteurs de support

Les machines à vecteurs de support (SVM) utilisent des méthodes d'apprentissage supervisé pour les applications de classification. En plus des problèmes de classification, les SVM peuvent être utilisés pour les problèmes de régression et de détection des valeurs aberrantes. Les principaux avantages des SVM sont l'efficacité de leur mémoire et leur efficacité dans les espaces de grande dimension. De plus, ces modèles ML conviennent à la formation d'échantillons avec des cas de grande dimension.

Les SVM apportent très peu d'inconvénients, par exemple, les estimations de probabilité ne sont pas disponibles, pour lesquelles une validation croisée k-fold doit être utilisée. Scikit-

learn SVM est très populaire parmi la communauté des développeurs de logiciels python dans le domaine de la science des données [17, 31, 9]

2.5.1 Concepts de base :

Avant d'en savoir plus sur SVM, nous devons connaître la signification de certaines notions de base dans SVM : (figure 2.4

- **Hyperplan** : est le séparateur optimale
- **Vecteurs de support** : sont les points les plus proches de l'hyperplan de chaque classe
- **Hyperplan de support** : sont des séparateurs parallèles à l'hyperplan passent par vecteurs de support
- **Marge (margin)** : est la distance entre l'hyperplan et les vecteurs de support

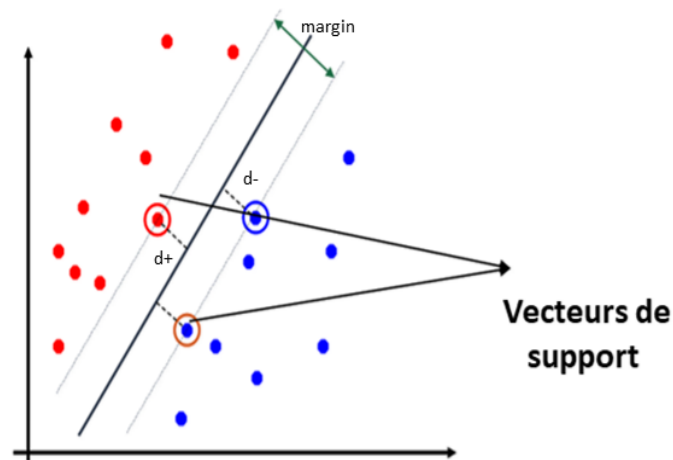


FIGURE 2.4 – principe de séparateur SVM

2.6 Types de classification SVM

Support Vector Machine (SVM) un classificateur binaire qui crée un hyperplan ou un groupe d'hyperplans pour séparer les données de différentes classes.(figure 2.5) .

SVM est un outil relativement récent pour le classement des données. Il est plus facile à utiliser que les réseaux de neurones. Une tâche de classification implique la formation et le test des données.

Chaque instance de l'ensemble d'apprentissage contient une *valeur cible* (étiquettes de classe) et plusieurs *attributs* (caractéristiques). L'objectif de SVM est de produire un modèle qui prédit la valeur cible des instances de données dans l'ensemble de test en fonction uniquement des attributs [21].

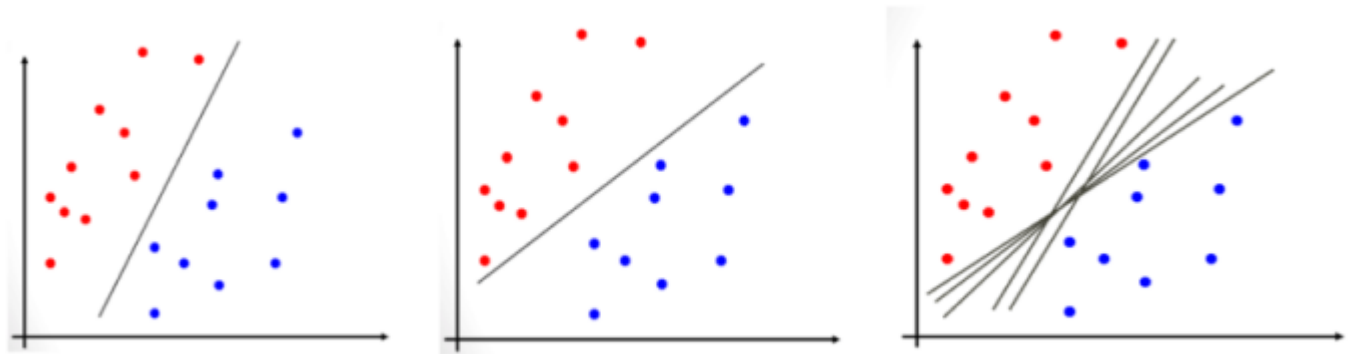


FIGURE 2.5 – les différents séparateurs que les données peut avoir

Les machines à vecteurs de support peuvent trouver la meilleure option pour séparer deux groupes différents en maximisant la marge entre l'hyperplan de séparation. La fonction de l'hyperplan, appelée classificateur, est généralement spécifiée par un petit sous-ensemble d'ensembles d'apprentissage, qui sont les vecteurs de support. La séparation peut se réaliser dans deux cas selon l'ensemble de données :

- Cas 1, linéairement séparable. (a) dans la figure 2.6
 - Cas 2, non linéairement séparable. (b) dans la figure 2.6
- Pour des raisons de visualisation seulement, nous allons considérer le cas d'un espace d'entrée à deux dimensions, c'est-à-dire $x \in \mathbb{R}^2$.

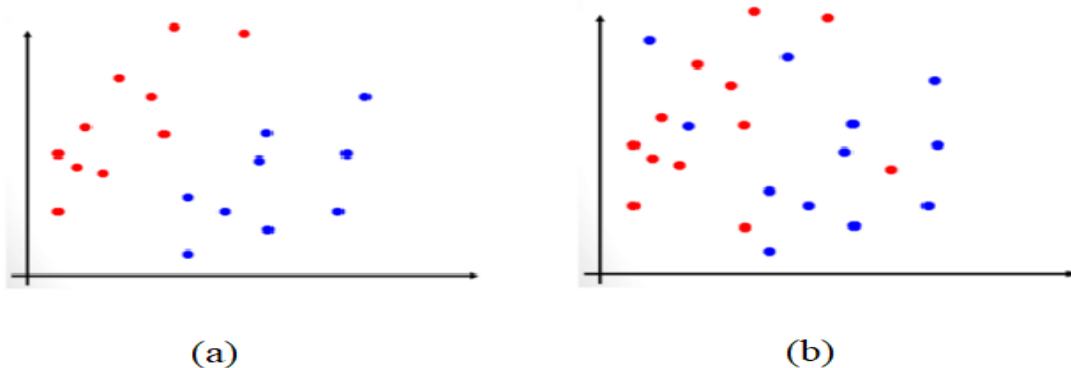


FIGURE 2.6 – les deux cas de séparation : (a) linéaire et (b) non linéaire

2.6.1 Linéairement séparable.

Le meilleur hyperplan est celui qui a la distance maximale des deux classes. En fait, pour ($\mathbf{x} \in \mathbb{R}^2$), la ligne de séparation dans l'espace d'entrée est définie par l'équation [16]

$$\mathbf{w}_1 \mathbf{x}_1 + \mathbf{w}_2 \mathbf{x}_2 + \mathbf{b} = 0$$

Un ensemble de modèles d'apprentissage étiquetés $(\mathbf{y}_1, \mathbf{x}_1) \dots (\mathbf{y}_m, \mathbf{x}_m)$ où, $\mathbf{y}_i \in \{-1, 1\}$ est dit linéairement séparable s'il existe un vecteur \mathbf{w} et un scalaire \mathbf{b} tel que les inégalités :

$$\mathbf{w}x_i + b \geq 1 \quad \text{si} \quad y_i = 1 \quad (2.1)$$

$$\mathbf{w}x_i + b \leq 1 \quad \text{si} \quad y_i = -1 \quad (2.2)$$

sont valables pour tous les éléments de l'ensemble d'apprentissage. Ainsi, l'hyperplan optimal :

$$\mathbf{w}_0 \cdot \mathbf{x} + b_0 = 0 \quad (2.3)$$

est l'unique qui sépare les données d'apprentissage avec une marge maximale :

L'hyperplan doit être aussi éloigné que possible des instances des deux classes. La distance qui doit être maximisée est

$$m = \frac{2}{\|\mathbf{w}\|} \quad (2.4)$$

Combiner les 3 équations 2.1 , 2.2 et 2.4 , on arrive donc au problème d'optimisation suivant (appelé problème **primal**) :

$$\begin{cases} \min_{\mathbf{w}, b} = \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{tel que } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1..n (\text{nombre de données}) \end{cases} \quad (2.5)$$

Un tel problème d'optimisation est résolu par le point selle de la fonctionnelle de Lagrange (Lagrangien) :

$$L(\mathbf{w}, \mathbf{b}, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^T \cdot \mathbf{w} - \sum_{i=1}^n \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1) \quad (2.6)$$

où les α_i sont des multiplicateurs de Lagrange.

L doit être minimisé par rapport à \mathbf{w} et b , et maximisé par rapport à α_i non négatif (c'est-à-dire que $\alpha_i \geq 0$ doit être trouvé). on utilise les conditions de Karush-Kuhn-Tucker (KKT) pour l'optimum de la fonction contrainte . au point de selle $(\mathbf{w}_0, \mathbf{b}_0, \boldsymbol{\alpha}_0)$, les dérivées du lagrangien L par rapport aux variables primales doivent s'annuler, ce qui conduit à :

$$\frac{\partial L}{\partial \mathbf{w}_0} = 0, \quad \mathbf{w}_0 = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \quad (2.7)$$

$$\frac{\partial L}{\partial b_0} = 0, \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (2.8)$$

les conditions de complémentarité KKT ci-dessous doivent également être remplies :

$$\alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1), i = 1..n \quad (2.9)$$

En substituant (2.7) et (2.8) dans un Lagrangien de variables primales $L(\mathbf{w}, \mathbf{b}, \boldsymbol{\alpha})$ (2.6), on passe aux variables duales Lagrangien $L_d(\boldsymbol{\alpha})$:

$$L_d(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j \mathbf{x}_i \mathbf{x}_j^T. \quad (2.10)$$

Afin de trouver l'hyperplan optimal, un double lagrangien $L_d(\boldsymbol{\alpha})$ doit être maximisé par rapport à α_i non négatif (c'est-à-dire que α_i doit être dans le quadrant non négatif) et par rapport à la contrainte d'égalité comme suit :

$$\alpha_i \geq 0, i = 1..n \quad (2.11)$$

$$\sum_{i=1}^n \alpha_i y_i x_i = 0 \quad (2.12)$$

2.6.2 Non linéairement séparable.

Dans le cas d'une séparabilité linéaire, le SVM essaie de trouver l'hyperplan qui maximise la marge, à condition que les deux classes soient classées correctement. Mais en réalité, les ensembles de données ne sont probablement jamais linéairement séparables.

L'idée clé dans laquelle SVM traite les cas de séparabilité non linéaire est la transformation des données. nous pouvons utiliser une certaine fonction de sorte que l'entrée de l'échantillon de données original produise l'ensemble de données linéairement séparables et c'est là qu'on nous présente la notion de "Kernel"

2.7 Fonctionnement de Kernel

Nous ne pouvons pas savoir quel hyperplan peut classer correctement les données qui ne sont pas séparées linéairement, pour cela SVM peut fonctionner avec Dataset non linéaire en utilisant "Kernel Trick" qui facilite la classification des données en essayant de convertir l'espace dimensionnel à un espace dimensionnel plus élevé (figure 2.7) . [33]



FIGURE 2.7 – L'idée de base du fonctionnement du kernel [38]

Pour faire ça, nous utilisons des fonctions quadratiques $\phi(x)$ qui nous permettent de trouver une frontière de décision qui divise clairement les points de données. Ces fonctions sont appelées Kernel qu'on peut s'écrire en équation 2.13

$$K(x_1, x_2) = \phi(x_1) + \phi(x_2) \quad (2.13)$$

L'hyperplan optimal dans un espace non linéaire peut être déterminé par le vecteur W qui minimise la fonction [2] :

$$\phi(W, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad (2.14)$$

Où ξ est une variable de relâchement et C une valeur pré-spécifiée qui est utilisée pour contrôler la quantité de régularisation.

Le choix du Kernel à utiliser est déterminé par le réglage des hyperparamètres.

2.7.1 Différentes fonctions du kernel

Voici quelques fonctions noyau que vous pouvez utiliser dans le SVM sont données ci-dessous :

- **Kernel linéaire :**

$$f(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1 \cdot \mathbf{x}_2$$

- **Kernel polynomial :**

Voici la formule du noyau polynomial :

$$f(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1^T \cdot \mathbf{x}_2 + 1)^d$$

Avec « d » est le degré du polynôme, que nous devons spécifier manuellement.

$$\begin{aligned} \mathbf{x}_1^T \cdot \mathbf{x}_2 &= \left| \begin{array}{c} \mathbf{x}_1 \end{array} \right| \left[\mathbf{x}_1 \mathbf{x}_2 \right] \\ &= \left| \begin{array}{cc} \mathbf{x}_1^2 & \mathbf{x}_1 \mathbf{x}_2 \\ \mathbf{x}_1 \mathbf{x}_2 & \mathbf{x}_2^2 \end{array} \right| \end{aligned}$$

Nous devons donc trouver \mathbf{x}_1^2 , \mathbf{x}_2^2 et $\mathbf{x}_1 \cdot \mathbf{x}_2$, maintenant nous pouvons voir que 2 dimensions ont été converties en 5 dimensions.

- **Kernel sigmoïde :**

Nous pouvons l'utiliser comme proxy pour les réseaux neuronaux. Son équation est la suivante :

$$f(\mathbf{x}_1, \mathbf{x}_2) = \tanh(\alpha \mathbf{x}^T \mathbf{y} + x)$$

Il s'agit simplement de prendre votre entrée, de la mettre en correspondance avec une valeur de 0 et de 1 afin qu'elle puisse être séparée par une simple ligne droite.

- **Kernel RBF :**

Il crée des combinaisons non linéaires de nos caractéristiques pour élever vos échantillons dans un espace de caractéristiques de plus haute dimension où nous pouvons utiliser une limite de décision linéaire pour séparer nos classes.

$$f(\mathbf{x}_1, \mathbf{x}_2) = e^{\left(\frac{-\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{\sigma^2}\right)}$$

Où :

1. σ est la variance .
2. $\|\mathbf{x}_1 - \mathbf{x}_2\|$ est la distance euclidienne entre deux points \mathbf{x}_1 et \mathbf{x}_2

2.8 SVM multiclass

L'adaptation des SVM bi classes au cas multiclass peut se faire de façons différentes tel que "One Vs One" ou "One Vs All". Le choix dépend de la taille du problème

2.8.1 One Vs One

Un contre un : consiste à entraîner $\frac{K(K-1)}{2}$ SVM sur chacun des couples de classes, puis à décider la classe gagnante soit par un vote majoritaire soit en post traitement les résultats grâce à l'estimation de probabilités a posteriori. Le nombre de classifieurs SVM à entraîne peut être

réduit en utilisant un codage astucieux pour les classes à travers un code correcteur d'erreur ou un graphe directe acyclique (DAGSV)

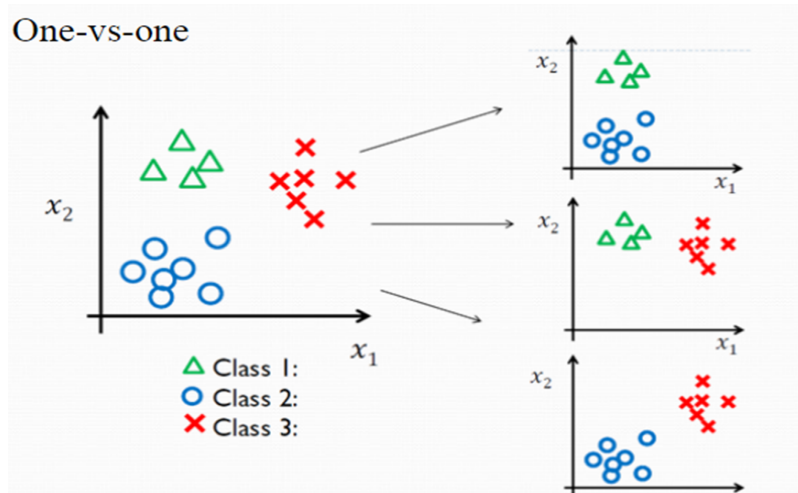


FIGURE 2.8 – Exemple de l'approche OVO

Nombre de classifieurs binaires est $K(K-1)/2$, où K est le nombre des classes

2.8.2 One Vs All

L'approche un contre tous consiste à entraîner un SVM binaire en utilisant les éléments d'une classe contre tous les autres

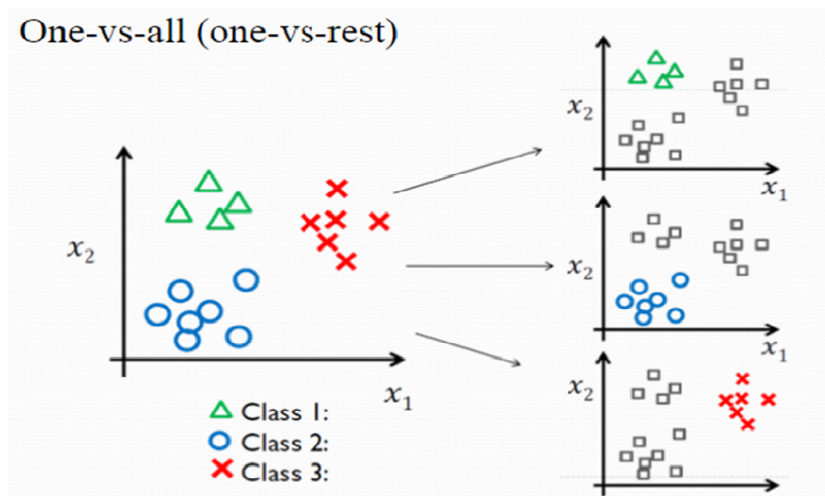


FIGURE 2.9 – Exemple de l'approche OVA

Il s'agit de résoudre de l'ordre de K problèmes où k est le nombre de classes

2.9 Conclusion :

Dans ce chapitre, on a présenté de manière simple l'apprentissage automatique et ses techniques en se concentrant sur les machines à vecteurs de support (SVM) et l'utilisation du "kernel trick" pour résoudre les problèmes de données non linéairement séparables.

Chapitre 3

Implementation

3.1 Introduction :

Dans ce chapitre nous présentons l'environnement de notre travail et l'utilisation des méthodes de classification SVM que nous avons présentées dans le chapitre précédent pour classifier les chiffres manuscrits de la base de données MNIST. Nous présentons aussi la mise en oeuvre de notre application en donnant les outils nécessaires tel-que le langage et Dataset utilisés. Ensuite, nous présenterons quelques résultats des différents modèles créés pour la classification des chiffres manuscrits .

3.2 Environnement de travail

Dans ce travail nous avons utilisé la langage de programmation MATLAB et la base de données MNIST pour classifier les chiffres manuscrits en 10 classes (de 0 à 9) utilisant les méthodes Svm pour la classification multiclass

3.2.1 MATLAB

Matlab est une plate-forme de programmation et de calcul numérique utilisée par des millions d'ingénieurs et de scientifiques pour analyser des données, développer des algorithmes et créer des modèles. Les boîtes à outils MATLAB sont développées par des professionnels, rigoureusement testées et entièrement documentées. cela nous permet d'exprimer directement les mathématiques matricielles .

3.2.2 LIBSVM

LIBSVM est une bibliothèque pour les machines à vecteurs de support (SVM). il se développe activement depuis l'an 2000. son objectif est d'aider les utilisateurs à appliquer facilement SVM à leurs applications.

LIBSVM a acquis une grande popularité dans l'apprentissage automatique et dans de nombreux autres domaines. Dans cet article [13], vous pouvez voir la présentation de tous les détails d'implémentation de LIBSVM où des questions telles que la résolution de problèmes d'optimisation SVM, la convergence théorique, les estimations de probabilité de classification multiclass et la sélection de paramètres sont discutées en détail. (détail de téléchargement de LIBSVM ici[8])

3.2.3 MNIST

La base de données (dataset) utilisé pour l'application dans ce travail est la base de données MNIST, constitué à l'origine de 60 000 images d'apprentissage, et 10 000 images de test. Différents classificateurs prouvés sur cette base de données [20, 3, 25, 19, 24] avaient montré un taux d'erreur de 1,00 à 0,42% [20]

les images de MNIST sont 28 x 28 en niveaux de gris (0-255) étiquetées et ont un format bitmap. (Figure 3.1)

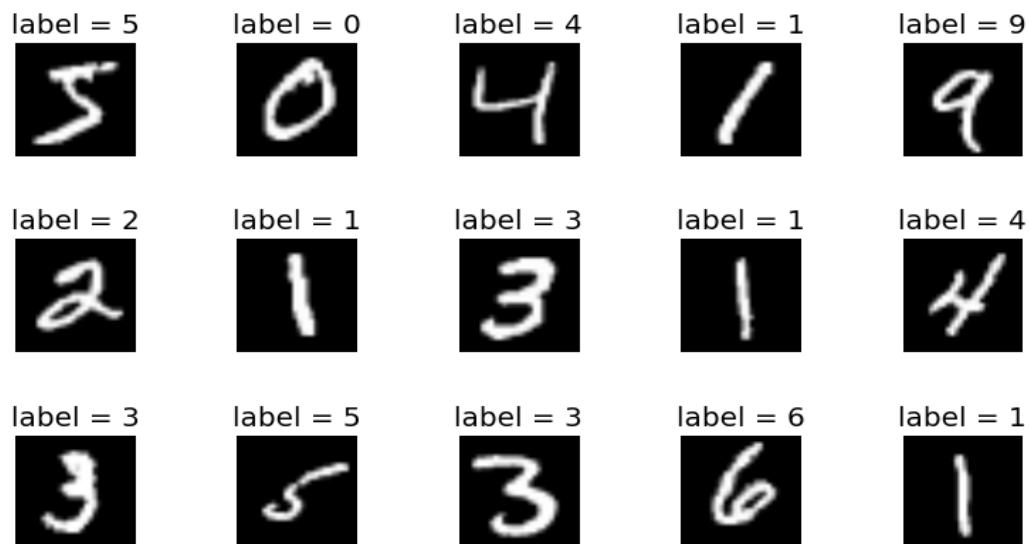


FIGURE 3.1 – Certains chiffres de la base de données MNIST

Les raisons pour lesquelles nous utilisons la base de données de chiffres manuscrits MNIST sont diverses. il s'agit d'une norme qui est une base de données relativement simple pour tester rapidement des théories et des algorithmes. Les chiffres manuscrits dans la base de données MNIST ont déjà été prétraités, y compris la segmentation et la normalisation, il devront non seulement économiser le temps de traitement, mais aussi réduire la probabilité de mauvaise classification [36]. donc cela peut être un bon début en dépensant un minimum d'efforts pour le pré-traitement.

3.3 Architecture de système de classification des chiffres manuscrit

La figure suivante montre l'architecture de notre système :

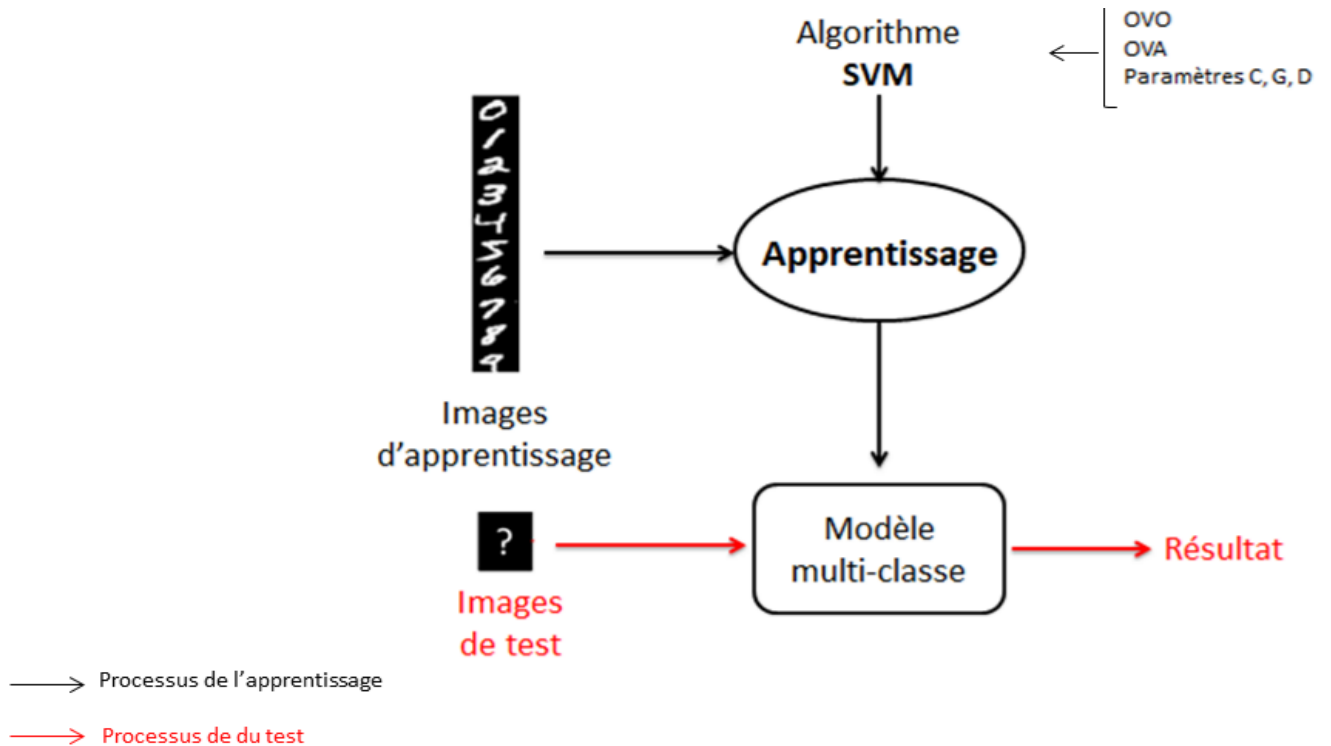


FIGURE 3.2 – Architecture de notre système de classification des chiffres manuscrits

Comme nous utilisons l'un des méthodes du l'apprentissage automatique supervisé, notre système contient deux processus :

- **Processus d'apprentissage**

Le processus d'apprentissage commence par le choix de l'algorithme SVM.

La phase d'apprentissage est lancée, en utilisant de 60000 images des chiffres manuscrits de la base de données MNIST . Durant cette phase, plusieurs paramètres sont réglés , à savoir : le paramètre C, le gamma g du kernel gaussien, et le de degré D du kernel polynomiale. Une fois l'apprentissage terminé, un modèle Multiclasse est créé.

- **Processus de test**

Nous donnons au modèle créé de nouvelles 10000 images des chiffres manuscrits du MNIST Dataset pour le test .

Lorsque le test est terminé, nous recevons un résultat sous forme de temps pris dans le test et un rapport de précision du test .

3.3.1 Présentation de l'application

La figure 3.3 représente l'interface principale de notre application, on peut la considérer comme un menu générale.

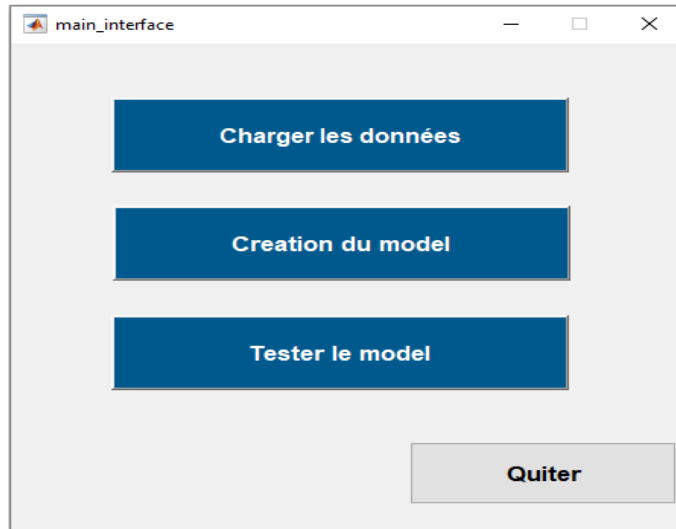


FIGURE 3.3 – Interface principale

Dans cette interface nous avons trois boutons pour créer un projet.

- Cliquant sur le bouton "charger les données" nous amènera à une nouvelle interface montré sur la figure 3.4

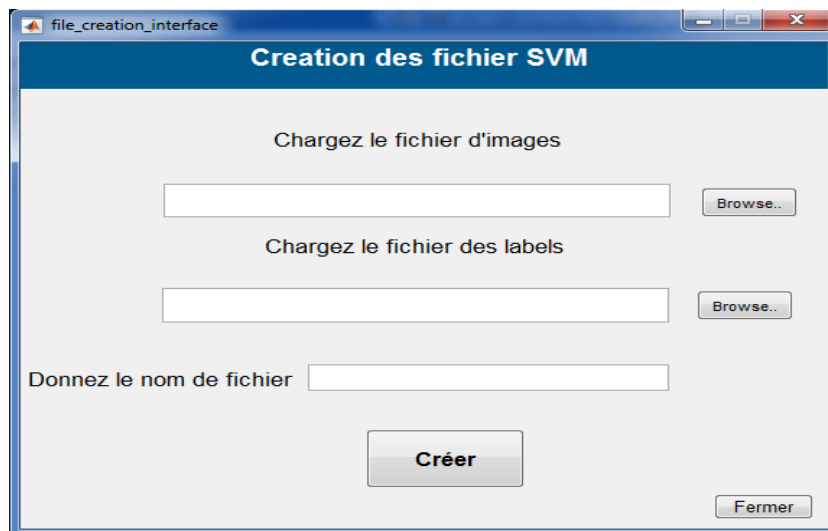


FIGURE 3.4 – Interface pour charger les données

- Cliquant sur le bouton "creation du model" nous amènera à une nouvelle interface montré sur la figure 3.5

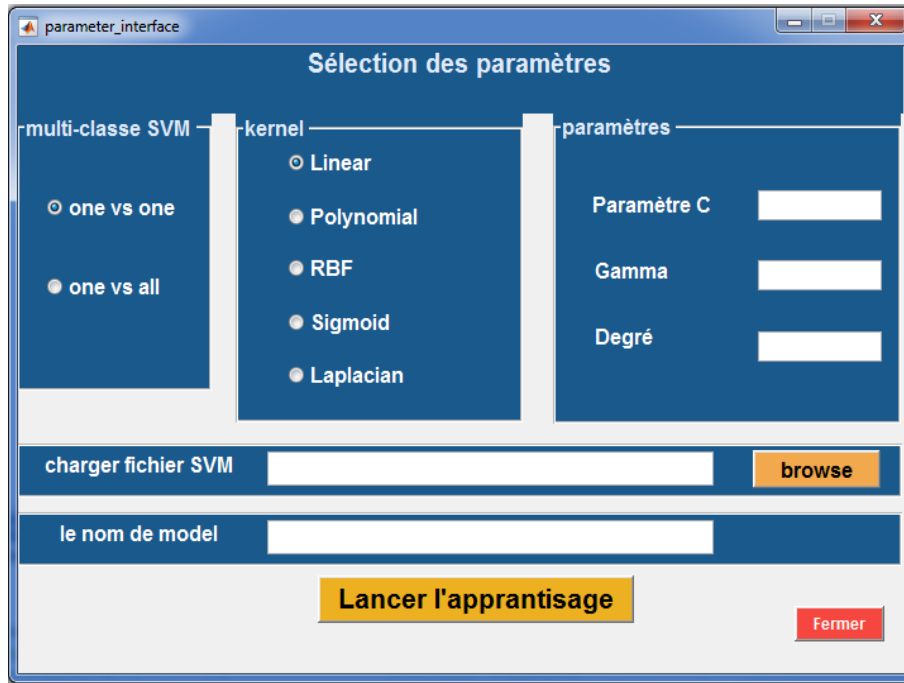


FIGURE 3.5 – Interface pour créer des modèles

- Cliquant sur le bouton "tester le modèle" nous amènera à une nouvelle interface montrée sur la figure 3.6

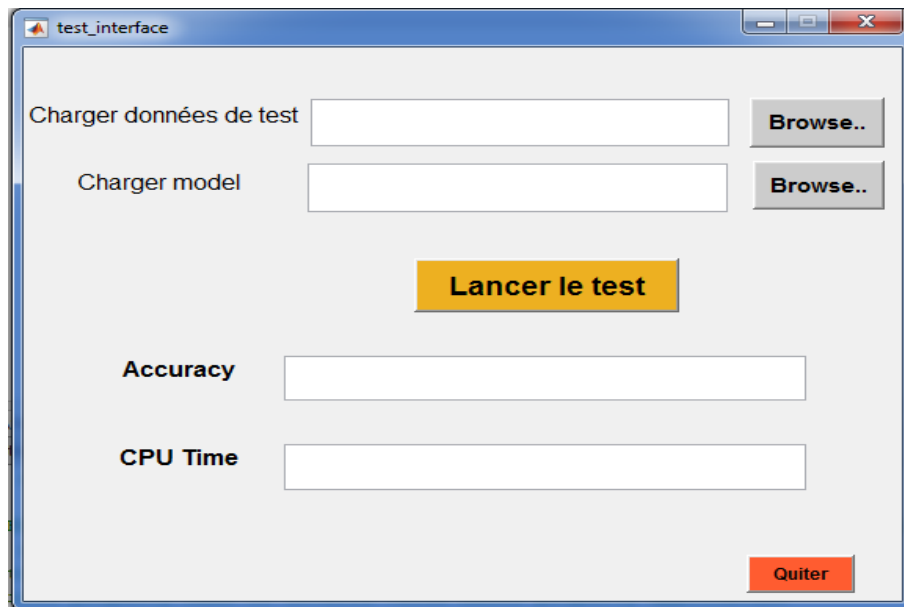


FIGURE 3.6 – Interface pour tester les modèles

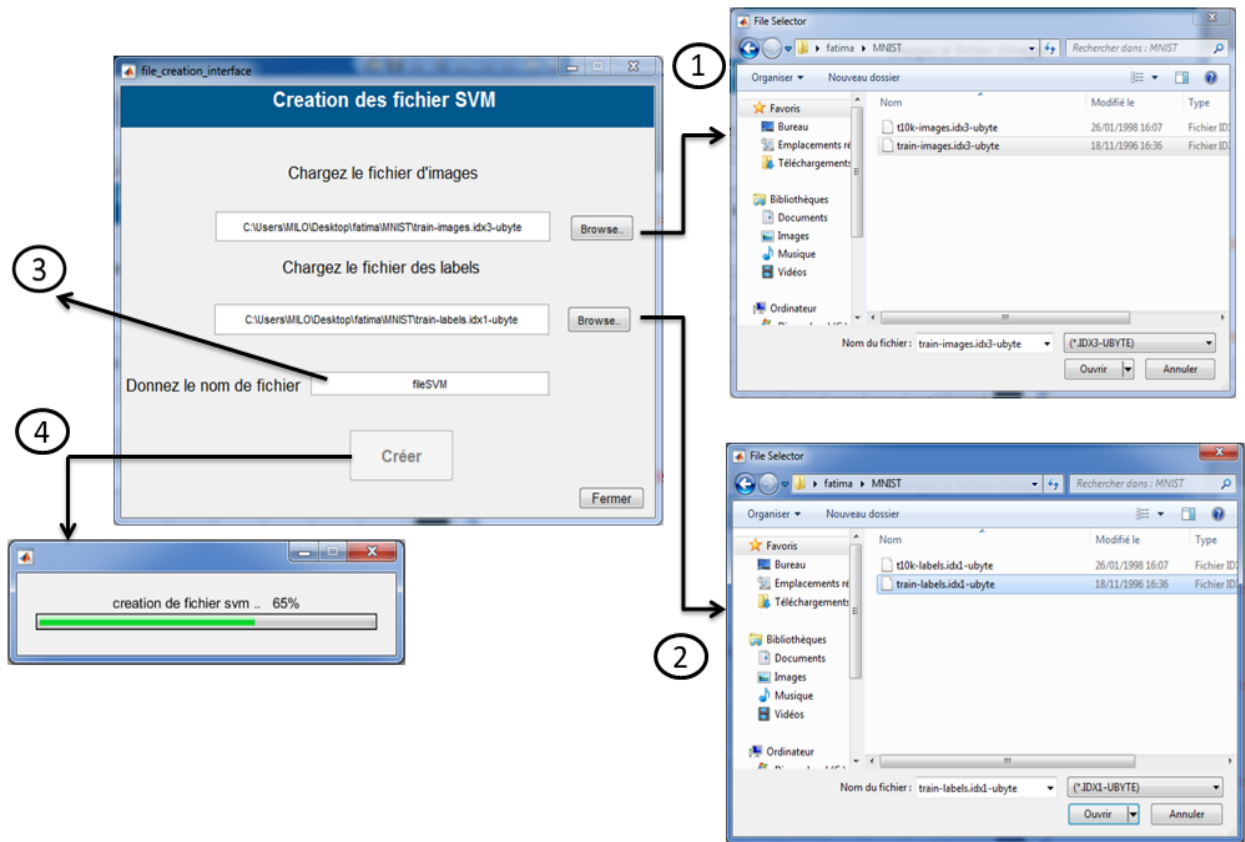


FIGURE 3.7 – l'utilisation de la fenêtre de création des fichiers SVM

Dans la figure 3.7, on a :

- (1) : Bouton "Browse" pour charger les images d'apprentissage
- (2) : Bouton "Browse" pour charger les labels des images d'apprentissage
- (3) : Nommer le fichier des données d'apprentissage à créer
- (4) : Lorsqu'on appuie sur le bouton "Créer", on lance la création du fichier affichant un bar de progression.

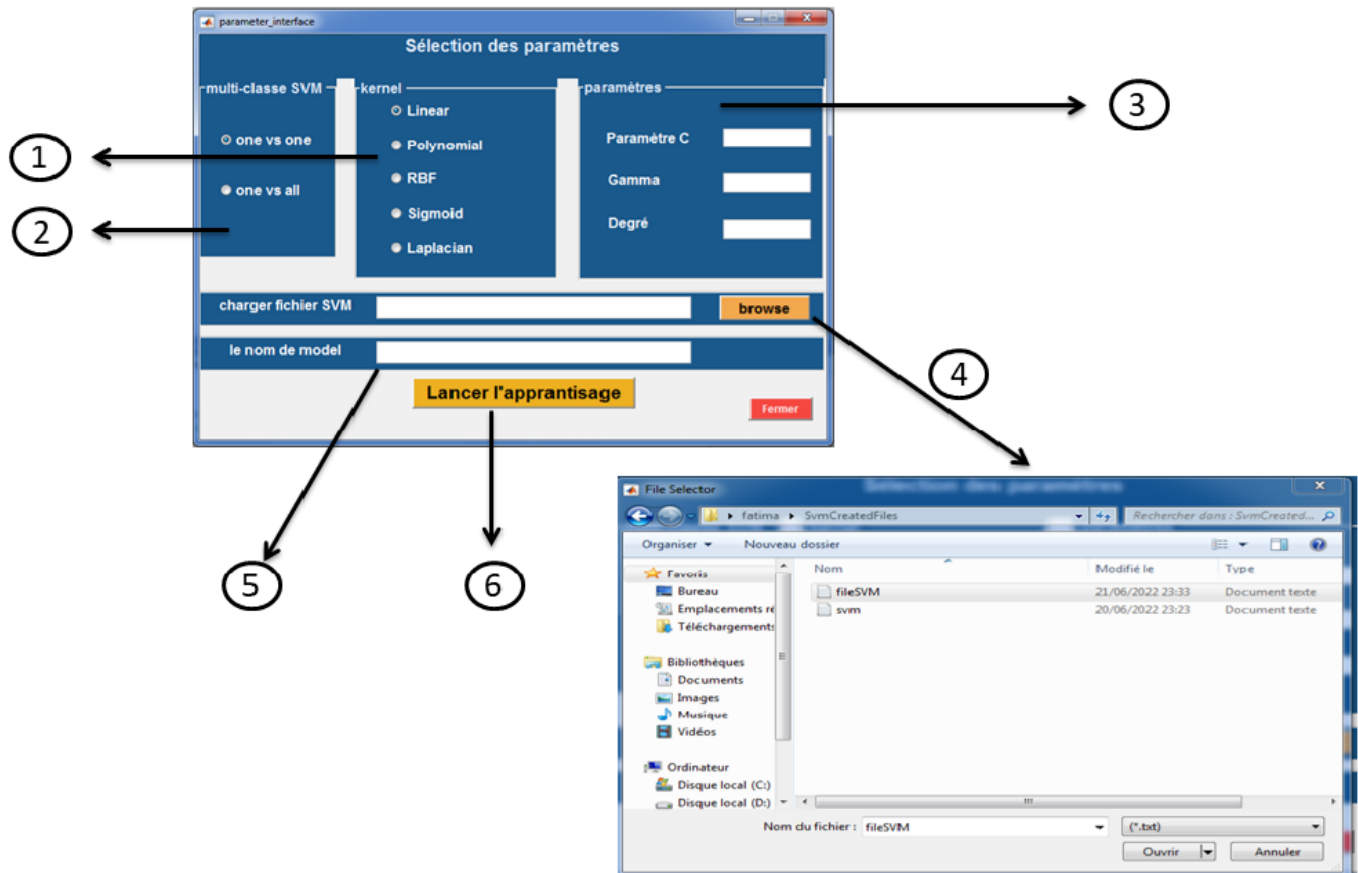


FIGURE 3.8 – La fenêtre pour créer les modèles

- (1) : Le choix de fonction du kernel entre :
- Linéaire kernel
 - Kernel polynomiale
 - Kernel de RBF
 - Kernel de sigmoid
 - Kernek de laplacien
- (2) : Le choix de l'approche de classification utilisant par svm pour les problemes multclasse , le choix est entre :
- One Vs One
 - One Vs All
- (3) : le choix de l'hyperparametre de la fonction kernel choisée, on a :
- Paramètre C (paramètre de régularisation).
 - Gamma γ (de RBF, sigmoid et laplacian kernels).
 - Degré D (pour polnmiale kernel)
- (4) : Lorsqu'on clique sur le bouton "Browse" , une fenêtre de répertoire apparaîtra pour charger le fichier de données d'apprentissage.

(5) : Pour donner le nom voulu de modèle.

(6) : Cliquez sur ce bouton pour lancer l'apprentissage (création d'un modèle qui sera enregistré dans un dossier).

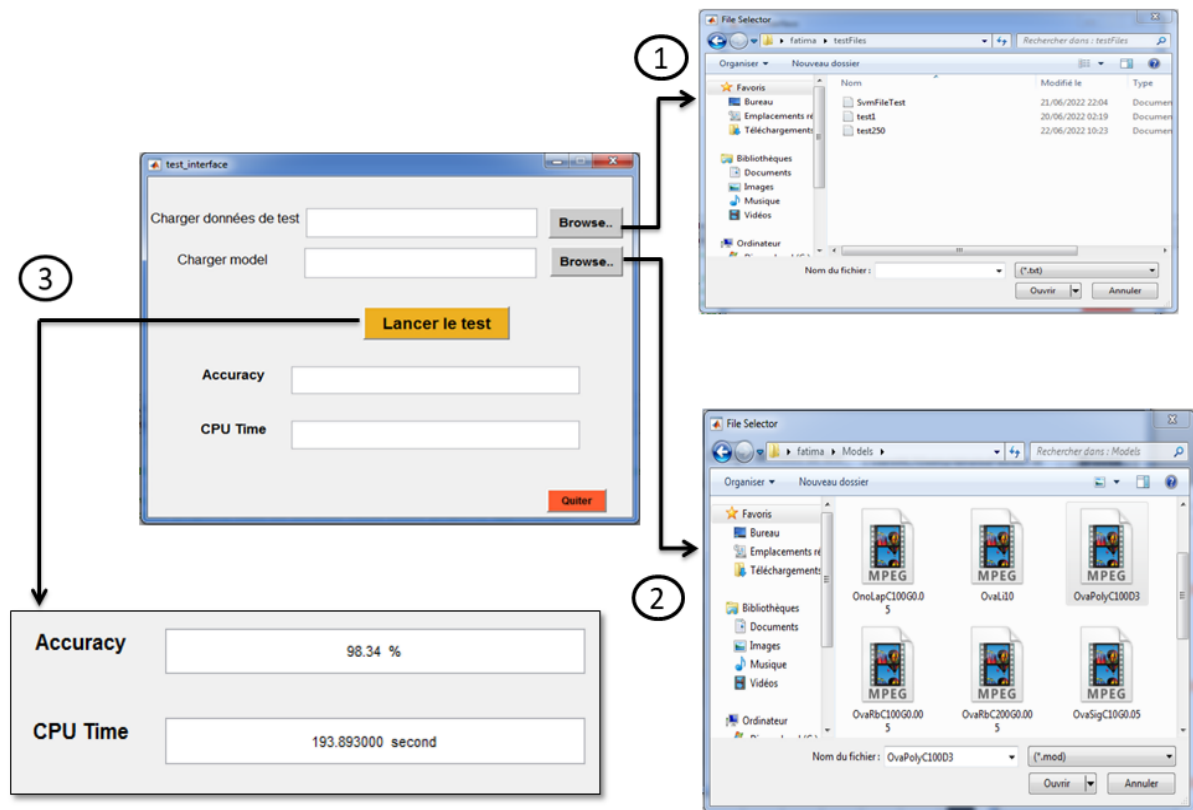


FIGURE 3.9 – La fenêtre pour tester les modèles

(1) : Lorsqu'on clique sur le bouton "Browse", on charge les images de test à partir de répertoire.

(2) : Lorsqu'on clique sur le bouton "Browse", on charge un modèle à tester qu'on a déjà créé .

(3) : Lorsqu'on clique sur le bouton "Lancer le test", le test du modèle démarre. Une fois le test est terminé, il nous donne les résultats en termes de précision et de temps pris pour terminer le test.

3.3.2 Expérimentation et discussion

Avec la base de données MNIST d'images de chiffres manuscrits, on a créé des modèles utilisant différentes approches de SVM multiclasse classification et différentes fonctions de kernel. basé sur les résultats de choix des paramètres dans ce mémoire [5], on a choisit des exemples de kernels avec leurs paramètres associés.

- Dans le tableau 3.1 on a choisi le kernel polynomial, fixé le paramètre $C=100$ et changé le degré D de chaque modèle, remarqué que le kernel polynomial prend beaucoup de temps d'apprentissage utilisant l'approche OVA que l'approche OVO.

	t	c	g	d	train CPU time (sec)
OVO	Polynomiale	100	-	2	822
OVO	Polynomiale	100	-	4	1102
OVA	Polynomiale	100	-	3	14157
OVA	Polynomiale	100	-	2	10234

TABLE 3.1 – Exemple d'apprentissage de 4 différents modèles du kernel polynomiale

- Lorsqu'on a fait un test sur les modèles mentionnés dans le tableau 3.1, nous avons obtenu de bons résultats qu'on se trouve dans le tableau 3.2.

	t	c	g	d	TEST CPU time (sec)	Accuracy
OVO	Polynomiale	100	-	2	113	98,05%
OVO	Polynomiale	100	-	4	95	97,37%
OVA	Polynomiale	100	-	3	194	98,34%
OVA	Polynomiale	100	-	2	169	98,07%

TABLE 3.2 – Exemple de test du 4 modèles de kernel polynomiale

- On a fait la même chose utilisant RBF kernel, voila le résultat d'apprentissage de 4 différents modèles dans le tableau 3.3

	t	c	g	d	train CPU time (sec)
OVO	RBF	1000	0.005	-	1065
OVO	RBF	2000	0.0005	-	1372
OVA	RBF	100	0.005	-	15425
OVA	RBF	200	0.005	-	12136

TABLE 3.3 – Résultats d'apprentissage utilisant le kernel RBF

- Le tableau 3.4 mention le résultat de test des modèles RBF.

	t	c	g	d	TEST CPU time (sec)	Accuracy
OVO	RBF	1000	0.005	-	763	100%
OVO	RBF	2000	0.0005	-	230	97,04%
OVA	RBF	100	0.005	-	294	98,4%
OVA	RBF	200	0.005	-	295	98,43%

TABLE 3.4 – Resultats de le test des modeles de kernel RBF

- Comme pour le kernel linéaire, on a pris 4 modèles d'exemples utilisant différentes valeurs de valeurs de C pour les deux approches OVO et OVA (tableau 3.5 remarquant que le temps d'apprentissage est plus grand lorsque $C=100$ que lorsque $C=10$).

	t	c	g	d	train CPU time (sec)
OVO	Linear	100	-	-	69420
OVO	linear	10	-	-	9401
OVA	linear	10	-	-	1906
OVA	linear	100	-	-	85460

TABLE 3.5 – Résultats d'apprentissage de 4 modèles de kernel linéaire

En remarquant que le noyau linéaire prend plus de temps d'apprentissage que le le noyau RBF et le noyau polynomial (selon les modèles que nous avons créés), peut-on dire que c'est à cause du paramètre C ?

en remarquant que lorsque C=100, cela prend beaucoup de temps comparé à lorsque C=10, mais ceci est basé sur nos remarques, on peut se tromper totalement. Donc cela nécessite une étude plus approfondie.

- Voila le résultat du test du modèles de kernel linéaire :

	t	c	g	d	TEST CPU time (sec)	Accuracy
OVO	Linear	100	-	-	125	92,57%
OVO	linear	10	-	-	166	93,08%
OVA	linear	10	-	-	70	97,16%

TABLE 3.6 – Test des modèles linéaire

3.4 Résultat du classification

Après une estimation globale et une comparaison des performances des modèles mentionnés ci-dessus, nous rapportons ici les résultats des classificateurs suivants :

- A partir de résultats de notre exemples,nous avons remarqué dans la figure 3.10 que les deux approches ont donné des résultats proches, mais en calculant la moyenne (figure 3.11), nous pouvons dire que l'approche OVA nous a donné de meilleurs résultats que l'approche OVO avec des moyennes de **98,08** pour OVA et **97.108** pour OVO, bien sûr ce n'est que selon les modèles que nous avons créés.

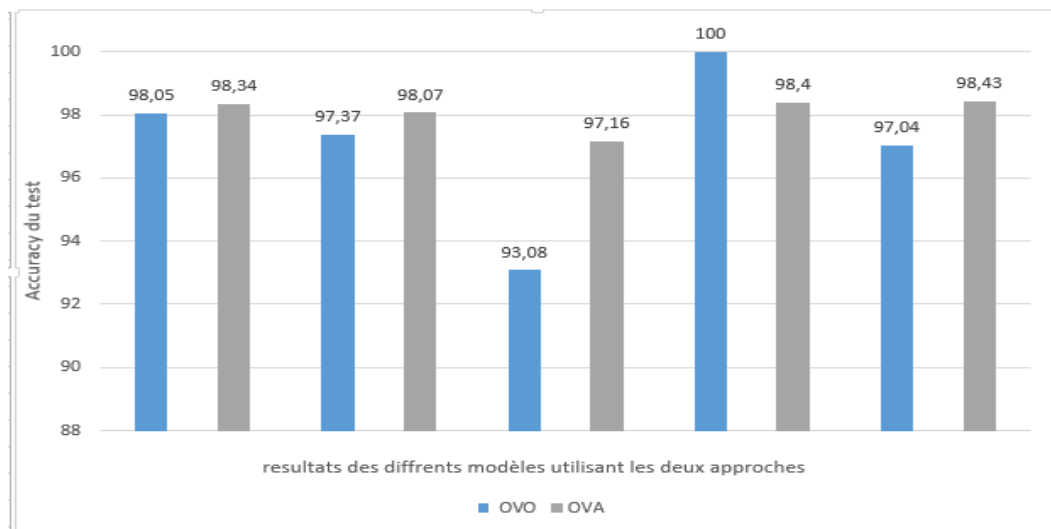


FIGURE 3.10 – graphe du précision basé sur les résultats des modèles qu'on a testés

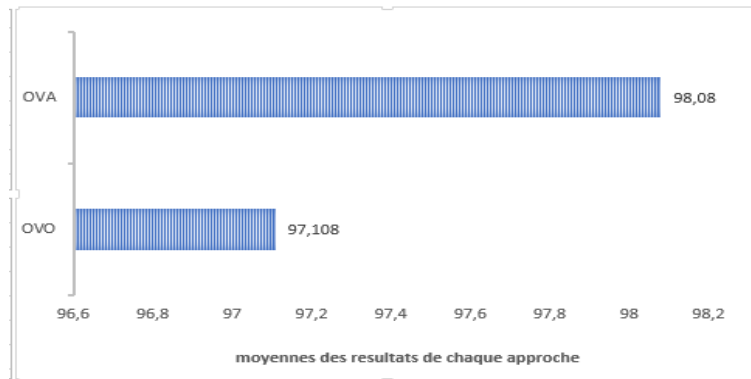


FIGURE 3.11 – Moyenne des résultats des tests de deux approches utilisées

- Concernant les fonctions du kernel, en remarquant dans les figures 3.12 , 3.13. RBF a donné de meilleurs résultats en termes de précision, puis polynomial et enfin linéaire. mais peut-on dire que RBF est meilleur ?
 en fait, tout dépend du choix des paramètres, peut-être que la création d'un autre modèle avec des paramètres différents peut modifier les résultats. mais à partir des modèles que nous avons créés avec les paramètres choisis ,en effet la fonction noyau RBF donne la meilleure précision de classification.

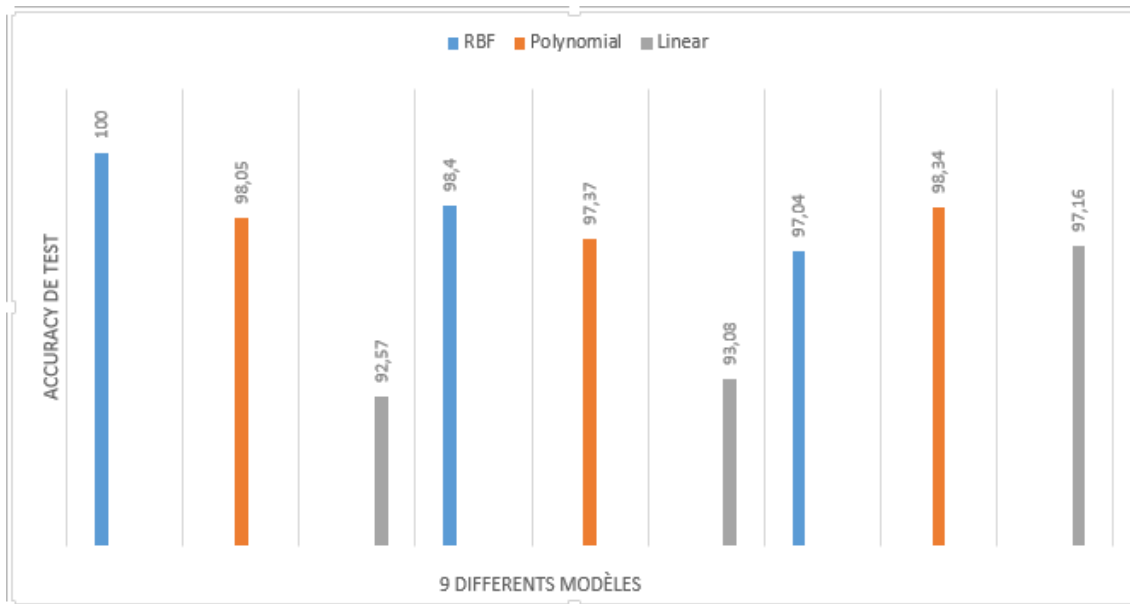


FIGURE 3.12 – Résultats d'accuracy du test de 9 différents modèles

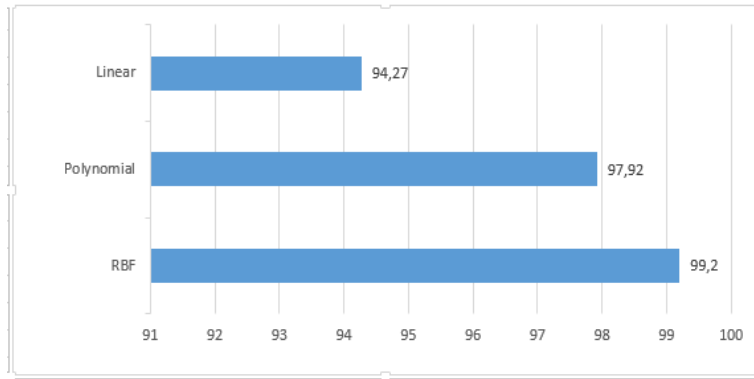


FIGURE 3.13 – Moyennes des résultats des tests de kernels utilisés

- En sachant que nous travaillons avec MNIST dataset pour l'apprentissage et le test , ce qui signifie que nous travaillons sur des images conditionnelles pré-traitées, nous pouvons conclure que les résultats précédents correspondent au choix de paramètres, nous avons expérimenté en créant d'autres modèles avec un choix aléatoire de paramètres, les résultats sont donnés dans le tableau 3.7 :

	t	c	g	d	Train CPU time (sec)	TEST CPU time (sec)	Accuracy
OVO	Polynomial	4	-	4	1018	89	97,45 %
OVO	Linear	2	-	-	2927	235	93,76 %
OVA	RBF	1000	0.005	-	14546	156	98,40 %
OVO	RBF	10	0.75	-	62153	791	18,49 %
OVO	RBF	10	0.5	-	43917	726	27,38 %
OVO	Laplacian	10	0.05	-	8476	528	98,05 %
OVA	Laplacian	10	0.05	-	22423	360	98,30 %
OVO	Sigmoid	10	0.5	-	13309	1012	10,22 %
OVA	Sigmoid	10	0.5	-	13042	1447	8,10 %
OVA	Sigmoid	10	0.05	-	13101	865	11,50 %

TABLE 3.7 – Résultats d'apprentissage et test de modèles choisis au hasard

Remarquer le noyau Rbf avec l'approche OVO nous a donné un très mauvais résultat de précision tandis que précédemment (dans le tableau 3.4) le noyau RBF utilisant l'approche OVO nous a donné le meilleur résultat , donc le choix des paramètres est en effet très important et un facteur clé dans l'interprétation des résultats.

En remarquant que les modèles que nous avons créés en utilisant le noyau sigmoïde ont tous donné les pires résultats, alors quelle valeur de C ou G nous permet d'obtenir de meilleurs résultats en utilisant le noyau sigmoïde ?

Il faudra faire d'autres expériences en créant d'autres modèles avec des paramétrages différents pour obtenir le résultat que nous voulons, encore une fois, cela ne fait que confirmer l'importance du choix des paramètres.

3.5 Conclusion :

Nous avons présenté dans ce chapitre le système proposé pour la classification des chiffres manuscrits, ainsi que les résultats obtenus avec quelques remarques.

Conclusion générale

SVM a été utilisé dans la reconnaissance de l'écriture manuscrite par de nombreux autres chercheurs auparavant. Dans ce travail nous avons étudié l'application de la méthode SVM pour la classification des chiffres manuscrits.

Comme décrit dans le chapitre 3, nous n'avons pas effectué de prétraitement des données, nous avons plutôt utilisé le dataset MNIST pour faire l'apprentissage de nos algorithmes proposés et tester nos modèles créés, comme cela nous pouvons concentrer plus sur les résultats basés sur le choix des paramètres sans ajouter les effets des données éventuellement mal traitées sur les résultats parce que, comme mentionné dans le chapitre 1, le prétraitement des données est une phase importante dans le système de reconnaissance qui peut changer les résultats de l'algorithme utilisé pour la classification.

Notre mémoire s'inscrit dans le cadre général des approches conçus pour la classification des données multi-classes (c'est le cas de la majorité des problèmes réelles).

Nous avons choisi les deux approches SVM multi classes one-versus-one (OVO) et one-versus-all (OVA) pour la classification des 10 chiffres(de 0 à 9)

Nous n'avons pas eu le temps de créer et d'expérimenter beaucoup en créant d'autres modèles et en sélectionnant différents paramètres pour voir comment ils fonctionnent et leur effet sur la classification des chiffres manuscrits, mais d'après les résultats obtenus, Nous avons trouvé que les SVM avec un noyau RBF sont les plus appropriés à l'utilisation dans le problème de la reconnaissance des chiffres manuscrits ainsi que les noyaux polynomiale donne des résultats mieux que le noyau linéaire.

On peut aussi conclure que l'approche multi classe OVA donne meilleures résultats que l'approche multi classe OVO,

Bibliographie

- [1] Abdul Ahmad. On-line handwriting recognition using support vector machines and hidden markov models approaches. page 11, 12 2008.
- [2] Rimah Amami, Dorra Ben Ayed, and Nouredine Ellouze. Practical selection of svm supervised parameters with different feature representations for vowel recognition. *arXiv preprint arXiv :1507.06020*, 2015.
- [3] Serge Belongie, Jitendra Malik, and Jan Puzicha. Matching shapes. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, page 459. IEEE, 2001.
- [4] Abdelhak Boukharouba and Abdelhak Bennia. Novel feature extraction technique for the recognition of handwritten digits. *Applied Computing and Informatics*, 13(1) :19–26, 2017.
- [5] Sdaoui Boutkhil. Contribution à la es èvolution des probl èmes de classification multiclass par les techniques d’optimisation combinatoire. page 5, 2017.
- [6] Sing-Tze Bow. *Pattern recognition and image preprocessing*. CRC press, 2002.
- [7] Romain Cendre. *Classification par méthodes d’apprentissage supervisé et faiblement supervisé d’images multimodales pour l’aide au diagnostic du lentigo malin en dermatologie*. PhD thesis, 02 2021.
- [8] Chih-Chung Chang and Chih-Jen Lin. LIBSVM : A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2 :27 :1–27 :27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [9] Liyuan Chen, Changjun Zhou, and Liping Shen. Facial expression recognition based on svm in e-learning. *Ieri Procedia*, 2 :781–787, 2012.
- [10] Hala Djerouni. Développement d’un système de reconnaissance de chiffres manuscrits.
- [11] Gérard Dreyfus, JM Martinez, M Samuelides, MB Gordon, F Badran, S Thiria, and L Héroult. *Réseaux de neurones*, volume 39. Eyrolles Paris, 2002.
- [12] Linwei Fan, Fan Zhang, Hui Fan, and Caiming Zhang. Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, 2(1) :1–12, 2019.
- [13] Rong-En Fan, Pai-Hsuen Chen, Chih-Jen Lin, and Thorsten Joachims. Working set selection using second order information for training support vector machines. *Journal of machine learning research*, 6(12), 2005.
- [14] Frédéric Grandidier. *Un nouvel algorithme de sélection de caractéristiques : application à la lecture automatique de l’écriture manuscrite*. PhD thesis, École de technologie supérieure, 2003.
- [15] Toufik Guedire. Reconnaissance des chiffres manuscrits à base des machines à vecteurs de supports. page 10, 2017.
- [16] Vojislav Kecman. *Support Vector Machines – An Introduction*, volume 177, pages 605–605. 05 2005.

- [17] Janith Kodithuwakku, Dilki Dandeniya Arachchi, and Jay Rajasekera. An emotion and attention recognition system to classify the level of engagement to a video conversation by participants in real time using machine learning models and utilizing a neural accelerator chip. *Algorithms*, 15(5) :150, 2022.
- [18] Oliver Kramer. *Machine learning for evolution strategies*, volume 20. Springer, 2016.
- [19] E Kussul, T Baidyk, L Kasatkina, and V Lukovich. Rosenblatt perceptrons for handwritten digit recognition. In *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, volume 2, pages 1516–1520. IEEE, 2001.
- [20] Ernst Kussul and Tatiana Baidyk. Improved method of handwritten digit recognition tested on mnist database. *Image and Vision Computing*, 22(12) :971–981, 2004.
- [21] Min Min Kyaw, Syed Khaleel Ahmed, and Zainul Abidin Md Shariff. Shape-based sorting of agricultural produce using support vector machines in a matlab/simulink environment. In *2009 5th International Colloquium on Signal Processing & Its Applications*, pages 135–139. IEEE, 2009.
- [22] Pierre-Michel LALLICAN and Christian VIARD-GAUDIN. Reconstruction de la chronologie d'un tracé manuscrit hors-ligne. In *16° Colloque sur le traitement du signal et des images, FRA, 1997*. GRETSI, Groupe d'Etudes du Traitement du Signal et des Images, 1997.
- [23] Fabien Lauer, Ching Y Suen, and Gérard Bloch. A trainable feature extractor for handwritten digit recognition. *Pattern Recognition*, 40(6) :1816–1824, 2007.
- [24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86 :2278–2324, 1998.
- [25] Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, and Hiromichi Fujisawa. Handwritten digit recognition using state-of-the-art techniques. *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 320–325, 2002.
- [26] Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, and Hiromichi Fujisawa. Handwritten digit recognition : investigation of normalization and feature extraction techniques. *Pattern Recognition*, 37(2) :266, 2004.
- [27] Shi Na, Liu Xumin, and Guan Yong. Research on k-means clustering algorithm : An improved k-means clustering algorithm. In *2010 Third International Symposium on intelligent information technology and security informatics*, pages 63–67. Ieee, 2010.
- [28] Meni Babakidi Narcisse. Développement d'un filtre gaussien pour la réduction du bruit dans les images médicales [development of a gaussian filter for noise reduction in medical images]. *International Journal of Innovation and Applied Studies*, 36(2) :594, 2022.
- [29] Asma Ouji. *Segmentation et classification dans les images de documents numérisés*. PhD thesis, Lyon, INSA, 2012.
- [30] Dr. Firoj Parwej. The state of the art recognize in arabic script through combination of online and offline. *International Journal of Computer Science and Telecommunications (IJCST), UK, London ,ISSN 2047-3338*, Volume 4 :60, 03 2013.
- [31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12 :2825–2830, 2011.
- [32] BOULEKCHER Rachida, KABOUR Oussama, and Chebout Mohamed Sedik. Une approche basée machine learning pour la prédiction du covid-19 en algérie. 2021.

- [33] Anshul Saini. Support Vector Machine(SVM) : A complete guide for beginners. 2021. https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/#h2_6.
- [34] Guillaume Saint-Cirgue. Apprendre le machine learning en une semaine. page 100, 2019. Book available at <https://machinelearnia.com/>.
- [35] Shivani Sharma and Maninder Kaur. Medial axis transformation based skeletonization of image patterns using image processing techniques. *IOSR Journal of Computer Engineering*, e-ISSN, pages 2278–0661.
- [36] Emmanuel Tonye. Le traitement des images de télédétection par l'exemple. page 183, 01 2000.
- [37] Øivind Due Trier, Anil K Jain, and Torfinn Taxt. Feature extraction methods for character recognition-a survey. *Pattern recognition*, 29(4) :642, 1996.
- [38] Yu Wang. Handwritten digit recognition by multi-class support vector machines. page 10, 2018.