

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

UNIVERSITE M'HAMED BOUGARA-BOUMERDES



**Faculté de Technologie**

**Département Ingénierie des Systèmes Electriques**

**Mémoire de Master**

Présenté par

**Chekiane Mohamed Ayoub**

**Boukhetaia sami**

**Filière : Télécommunications**

**Spécialité : Réseaux et Télécommunications**

---

**Deep learning VS Machine learning  
pour la vérification des signatures Hors ligne**

---

HAMMADOUCHE	Mhamed	PR	UMBB	Président
MESSAOUDI	Nouredine	MCA	UMBB	Examineur
GUERBAI	Yasmine	MCA	UMBB	Rapporteur

**Année Universitaire : 2021/2022**

## ***Remerciements***

*Avant tout, nous rendons grâce à DIEU, de nous avoir donné la force, la patience, la volonté afin d'accomplir ce modeste travail qui présente le fruit de plusieurs années de sacrifices.*

*Nous avons l'honneur et le plaisir de présenter notre profonde gratitude et nos sincères remerciements à notre promotrice Mme. Yasmine Guerbai pour tout le soutien, l'aide, l'orientation, la guidance qu'elle nous à apportés durant le mémoire, ainsi que pour ses précieux conseils et ses encouragements lors de la réalisation de notre mémoire le temps qu'elle a accordée à notre encadrement.*

*Nous remercions par ailleurs vivement les membres du professeurs de nous avoir fait l'honneur d'examiner notre travail.*

*Que tous les enseignants ayant contribué à notre formation trouvent ici l'expression de notre profonde reconnaissance.*

*Un grand merci à nos mamans et nos papas, pour leur amour, leurs conseils ainsi que leur soutien inconditionnel à la fois moral, ainsi que nos sœurs et frères et toutes nos familles.*

*Nous tenons à remercier tous nos collègues de promo 2017 à 2022.*

*Enfin , Nous tenons à remercier également l'ensemble de département ingénierie des Systèmes électriques.*



## *Dédicace*

*Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à ma Mère.*

*À mon Père qui a été mon ombre durant toutes mes années d'études, et qui a veillé tout au long de ma vie à m'encourager, à me donner l'aide et à me protéger.*

*À Mon cher unique frère ALLA*

*À mes sœurs de cœur*

*À tous les gens m'aiment ' oussama , Zinou , Raouf , youcef , slimane , abdenour bentir , hichem ,amine , nasro , nassim, abderaouf '*

*À tous ceux qui me sont chers.*

*À tous ceux que j'aime et qui m'aiment et tous les étudiants et les étudiantes de la promotion et tous mes enseignants.*

*Mohamed Ayoub*

## *Dédicace*

*Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à ma Mère.*

*À mon Père qui a été mon ombre durant toutes mes années d'études, et qui a veillé tout au long de ma vie à m'encourager, à me donner l'aide et à me protéger.*

*À Mon cher unique frère*

*À ma chère sœur de cœur*

*À tous les gens m'aiment ' Nadjib , sami , ahmed '*

*À tous ceux qui me sont chers.*

*À tous ceux que j'aime et qui m'aiment et tous les étudiants et les étudiantes de la promotion et tous mes enseignants.*

*SAMI*

### *Liste des abréviations*

**PDA** : Personal Digital Assistant

**OCR** : reconnaissance optique de caractères

**FRR** : False Rejection Rate

**FAR** : False Acceptance Rate

**EER** : Equal Error Rate

**HTER** : Half Total Error Rate

**ROC** : Receiver Operating Characteristic

**SVM** : Support Vector Machines

**BC-SVM** : Multiclass ( bi class ) SVM

**RF** : Radial Basis Function

**OC-SVM** : One Class Support Vector Machine

**SVDD** : Support Vector Data Description

**ANN** : Réseaux neuronaux artificiels

**DNN** : Réseaux neuronaux profonds

**FCNN** : Les réseaux neuronaux entièrement connectés

**CNN** : Réseaux de neurones convolutifs

**LIBSVM** : la bibliothèque de SVM

**HSVS** : Handwritten Signature Verification system

**PDF** : la fonction de densité de probabilité

***N<sub>p</sub>*** : les signatures authentiques

***N<sub>t</sub>*** : les signatures authentiques fictives

**CSV** : Comma Separated Values



*Sommaire*

# Sommaire

Introduction : .....	1
Chapitre 1 : Généralités sur la vérification de signature	
<b>1.1. Introduction:</b> .....	3
<b>1.2. Generalities sur la Biométrie:</b> .....	3
1.2.1. Définition de la Biométrie:.....	3
1.2.2. Définition et fonctionnalités d'un système biométrique .....	4
1.2.3. Protocoles de réalisation d'un système biométrique .....	4
a. Dépendant de l'utilisateur .....	5
b. Independent de l'utilisateur.....	5
1.2.4. Définition de la signature manuscrite .....	5
1.2.5. Functionalities de la signature manuscrite .....	5
1.2.6. Authentification de la signature manuscrite .....	6
1.2.7. Mode d'acquisition de la signature manuscrite.....	6
a) Mode d'acquisition en-ligne .....	6
b) Mode d'acquisition hors-ligne .....	7
1.2.8. Classes des signatures .....	7
A - Classe des signatures authentiques.....	7
B – Classe des signatures fausse .....	8
1.2.9. Structure générale d'un système d'authentification de signatures hors-ligne .....	9
1.2.10. Acquisition de l'image.....	9
1.2.10.1. Pré-traitement:.....	9
1.2.10.2. Génération des caractéristiques .....	11
1.2.11. Classification.....	12
1.2.11.1. L'apprentissage supervisé.....	12
1.2.11.2. L'apprentissage non-supervisé.....	15
1.2.12. Décision.....	15
1.2.13. Mesure de robustesse d'un système d'authentification biométrique .....	15
1.3. Conclusion .....	17

## Chapitre 2 : Méthodes utilisées dans la vérification de signature

<b>2.1. Introduction</b> .....	18
<b>2.2. Machine à vecteurs supports SVM</b> .....	18
2.2.1. Notions de base .....	19
2.2.1.1. Hyperplan.....	19
2.2.1.2. Vecteurs de support .....	19
2.2.1.3. Marge .....	20
2.3.1. Principe de fonctionnement général des SVMs : .....	21
2.3.2. Bi-Class Support Vector Machine (BC-SVM) .....	21
2.3.2.1. La fonction noyau.....	22
2.3.2.2. La Fonction de décision.....	24
2.3.3. One Class Support Vector Machine (OC-SVM).....	24
2.3.3.1. La fonction noyau .....	24
2.3.4. Mesures de distance du noyau OC-SVM .....	25
2.3.4.1. Fonction de décision.....	26
2.4. Deep learning .....	27
2.4.1. Généralité : deep learning .....	27
2.4.1.1. Définition.....	27
2.4.2. Les types de méthodes d'apprentissage .....	28
2.4.2.1. Apprentissage supervisé.....	28
2.4.2.2. Apprentissage non supervisé.....	28
2.4.3. Les réseaux de neurones.....	29
2.4.3.2. Réseaux neuronaux profonds DNN .....	30
2.4.4. Les différences architectures d'apprentissage approfondi .....	31
2.4.4.1. Réseau neuronal entièrement connecté .....	31
2.4.4.2. Réseaux de neurones convolutifs (CNN).....	31
2.5. Conclusion.....	32

## Chapitre 3 : Résultats et analyses

3. Introduction .....	33
3.1. OC-SVM Matlab .....	33

3. 1. 1. Conception du HSVS individual .....	33
3. 1. 2. caractéristique génération .....	35
3.1.2.1. classification et décision : .....	36
3.1.2.2. Mesures de distance du noyau OC-SVM .....	38
3.1.3. Conception du system HSVS combine ( Handwritten Signature Verification system)39	
3.1.3.1. déclaration de problème .....	39
3.1.3.2. Description des paramètres HSVS combinés indépendants de scripteur .....	39
3.1.3.3. Réglage du seuil.....	40
3.1.4. résultats expérimentaux .....	41
3.1.4.1. Description des ensembles de données et critères d'évaluation .....	41
3.1.4.2. Evaluation of the individual writer independent HSVS based on genuine and forged signatures. ....	42
3.1.4.3. Évaluation du HSVS indépendant de l'auteur individuel et combiné sur la base de signatures authentiques et fictives.....	46
3.1.4.4. Performance evaluation .....	50
3.2. Python Deep learning .....	53
3.2.1. Présentation des outils de développement.....	53
3.2.1.1. Matériel.....	53
3.2.2. Base de données.....	54
3.2.2.1. Présentation de la base de données: .....	54
3.2.2.2. Acquisition de données.....	54
3.2.3. Prétraitement .....	55
3.2.3.1. Conversion d'image en niveaux de gris .....	55
3.2.3.2. Redimensionnement de l'image .....	56
3.2.3.3. L'extraction de caractéristiques (Feature extraction) .....	57
3.2.3.4. Enregistrement des caractéristiques (Saving the Features ) .....	58
3.2.3.5. évaluation de précision et perte ( accuracy & loss ) : .....	60
3.2.3.6. TF Model .....	64
3.2.4. Le travail.....	64
3.2.4.1. Le résultat ( output ) :.....	65
3.2.5. Interface .....	68
3.3. Comparaison entre SVM MATLAB et DEEP LEARNING PYTHHON .....	73

3.3. 1. Langage utilisé dans Python et Matlab.....	73
3.3. 2. Bibliothèques.....	73
3.3. 3. Environnement.....	74
3.3. 4. Indexage.....	74
3.3. 5. Performances de Python Vs. Matlab.....	74
3.4. Conclusion.....	76
Conclusion.....	77
Références.....	78



*Liste des figures*

## Chapitre 1 : Généralités sur la vérification de signature

Figure 1. 1: Different moyens biométriques.....	4
Figure 1. 2: Exemples d'acquisition de signatures en-ligne.....	7
Figure 1. 3: Exemples d'acquisition de signatures hors-ligne.....	7
Figure 1. 4: Structure générale d'un système de reconnaissance d'une écriture. ....	9
Figure 1. 5: Exemple de binarisation, (a): Image brute, (b): Image binaire.....	10
Figure 1. 6: Exemple de segmentation, (a): Image originale, (b): Image segmentée. ....	11
Figure 1. 7: représente un exemple de normalisation .....	11
Figure 1. 8: Différentes approches de la classification supervisée. ....	12
Figure 1. 9: Principe général de fonctionnement du neurone formel. ....	13
Figure 1. 10: Principe général d'un classifieur SVM .....	14
Figure 1. 11: Distributions des taux de vraisemblances des modalités biométriques authentiques et celles erronées dans un système biométrique. ....	16
Figure 1. 12: Courbe ROC.....	16
Figure 1. 13: Taux d'erreurs FRR et FAR en fonction du seuil optimal de décision. ....	17

## Chapitre 2 : Méthodes utilisées dans la vérification de signature

Figure 2. 1: Exemple d'un hyperplan séparateur .....	19
Figure 2. 2: Les vecteurs de support. ....	20
Figure 2. 3: Hyperplan, vecteurs de support et marge maximale. ....	20
Figure 2. 4: Discrimination correcte d'un ensemble de deux classes. ....	21
Figure 2. 5: Exemple d'une difficulté à séparer deux classes .....	22
Figure 2. 6: (a) Données non linéairement séparables, (b) Données séparées par la fonction noyau. ....	23
Figure 2. 7: - (a) Distribution des données d'apprentissage, (b) Projection des données dans l'espace de redistribution. ....	25
Figure 2. 8: Le deep learning .....	28
Figure 2. 9: architecture du réseau neurone ANN. ....	30
Figure 2. 10: architecture du réseau neuronal profond .....	30
Figure 2. 11: Architecture d'un réseau de neurones convolutif [8].....	31

## Chapitre 3 : Résultats

Figure3. 1: L'architecture globale du HSVS individuel.....	34
Figure3. 2: conception du HSVS .....	35
Figure3. 3:L'architecture globale du HSVS individuel.....	36
Figure3. 4: L'architecture globale du HSVS individuel.....	37
Figure3. 5: L'architecture globale de l'auteur combiné indépendant HSVS .....	40
<b>Figure3. 6:</b> Quelques échantillons de signatures de utilisateur num "42". ....	41

Figure3. 7: Conception du rédacteur indépendant basée sur des signatures authentiques et falsifiées .....	43
Figure3. 8: Courbes FAR et FRR pour la base de données CEDAR. ....	44
Figure3. 9: Taux d'erreur en fonction du nombre d'auteurs en utilisant la base de données CEDAR. ....	44
<b>Figure3. 10:</b> PDF des signatures authentiques (figure de gauche) et fictives (figure de droite) pour différentes distances utilisées dans le noyau RBF de l'OC-SVM pour la base de données CEDAR .....	50
<b>Figure3. 11:</b> Courbes FAR et FRR pour sélectionner le seuil optimal : base de données CEDAR .....	50
Figure3. 12: AER selon le nombre d'écrivains pour CEDAR. ....	51
Figure3. 13: Exemple de la signature hors ligne de l'utilisateur 01.....	55
Figure3. 14: Notre fichier CSV en format Excel.....	59
Figure3. 15: La création de fichiers CSV. ....	59
Figure3. 16: deux graphiques représentant les pertes de deux modèles différents .....	61
Figure3. 17: Notre programme d'entraînement du modèle utilisant la fonction fit ( ) .....	62
Figure3. 18: la courbe de la perte ( Loss ).....	62
Figure3. 19: la courbe de la précision ( Accuracy ).....	63
Figure3. 20: exécution de notre programme python.....	65
Figure3. 21: Entrer l'identifiant ( iD) de l'utilisateur 01. ....	65
Figure3. 22: Entrer le chemin de l'image de la signature.....	66
Figure3. 23: Exemple pour entrer le chemin de l'image "b-original_1_20.png".....	66
Figure3. 24: Le résultat obtenu pour le test d'image " b-original_1_20.png ".....	67
Figure3. 25: Le résultat obtenu pour le test d'image " b-forgeries_1_15.png". ....	67
Figure3. 26: Exécution de notre interface. ....	69
Figure3. 27: Entrer l'identifiant ( iD) de l'utilisateur 01. ....	70
Figure3. 28: Le résultat obtenu pour le test d'image " b-original_1_20.png ".....	71
Figure3. 29: Le résultat obtenu pour le test d'image " b-forgeries_1_15.png".....	72



*Liste des tableaux*

## **Chapitre 2 : Méthodes utilisées dans la vérification de signature**

Tableau 2. 1: Quelques exemples des noyaux.....	23
Tableau 2. 2: Différent mesure de la distance .....	26

## **Chapitre 3 : Résultats**

Tableau 3. 1: Mesures de distance appliquées au noyau du classificateur OC-SVM.....	39
Tableau 3. 2: Performances de reconnaissance obtenues pour les différents seuils de rédacteur indépendant à l'aide du base de données CEDAR. ....	45
Tableau 3. 3: Performances de reconnaissance obtenues pour les différents seuils sur le jeu de données CEDAR. ....	46
Tableau 3. 4: Protocole d'évaluation.....	47
Tableau 3. 5: Taux d'erreur (%) fourni par différentes distances utilisées dans le noyau OC-SVM et la combinaison de différents seuils sur l'ensemble de données CEDAR.....	51
Tableau 3. 6: AER (%) fourni par le système proposé par rapport à d'autres systèmes utilisant l'ensemble de données CEDAR. ....	52

## ملخص

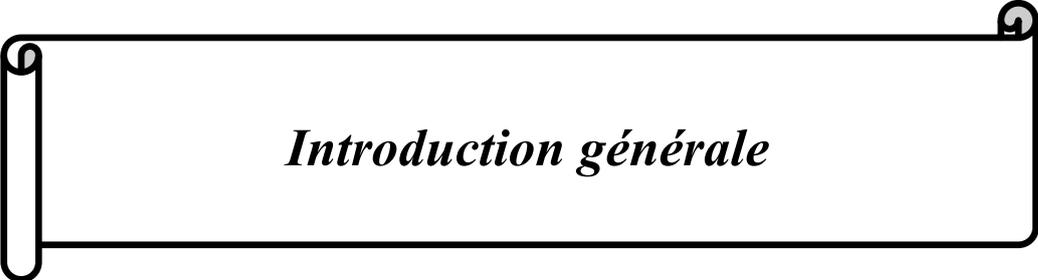
لكل شخص توقيع فريد الخاص به والذي يستخدم بشكل أساسي لأغراض التعريف الشخصي والتحقق من المستندات المهمة أو المعاملات القانونية. هناك نوعان للتحقق من صحة التوقيع: ثابت (غير متصل) وديناميكي (عبر الإنترنت). التحقق الثابت هو عملية التحقق توقيع إلكتروني أو مستند بعد إجرائه. التحقق من التوقيع دون اتصال غير فعال وبطء بالنسبة لعدد كبير من المستندات. للتغلب على عيوب التحقق من التوقيع دون اتصال بالإنترنت لقد شهدنا نمواً في التحقق الشخصي البيومتري عبر الإنترنت مثل بصمات الأصابع ومسح العين وما إلى ذلك. في هذا المشروع ، يُقترح التحقق من التوقيع دون اتصال بالإنترنت باستخدام آلة ناقلات دعم فئة واحدة أو لا الشبكة العصبية التلافيفية ثانياً. تسعى خوارزمية آلة ناقلات دعم فئة واحدة إلى تغليف القيم الداخلية الأساسية. الهدف هو فصل البيانات إلى فئتين (بناءً على وظيفة اتخاذ القرار) ، الفئة الإيجابية تعتبر فئة القيم المتطرفة والأخرى السلبية تعتبر فئة القيم المتطرفة. إلى جانب ذلك ، يجب أن تنتمي معظم بيانات التدريب إلى الفئة الإيجابية بينما يكون حجم الظرف ضئيلاً. الشبكة العصبية التلافيفية هي نوع من نماذج الشبكة العصبية التي تسمح لنا باستخراج تمثيلات أعلى للصورة المحتوى. تأخذ الشبكة العصبية التلافيفية بيانات البكسل الأولية للصورة ، وتدريب النموذج ، ثم تستخرج الميزات تلقائياً لتصنيف أفضل. الميزة الرئيسية للشبكة العصبية التلافيفية مقارنةً بسابقتها هي أنها تلقائية يكتشف الميزات المهمة دون أي إشراف بشري كما أنه يتمتع بأعلى دقة بين جميع الخوارزميات التي تتنبأ بالصورة.

## Résumé

Chaque personne a sa propre signature unique qui est utilisée principalement à des fins de identification personnelle et vérification de documents importants ou de transactions juridiques. Il existe deux sortes de vérification de signature : statique (hors ligne) et dynamique (en ligne). La vérification statique est le processus de vérification une signature électronique ou de document après qu'elle a été faite. La vérification de signature hors ligne n'est pas efficace et lent pour un grand nombre de documents. Pour surmonter les inconvénients de la vérification de signature hors ligne, nous avons constaté une croissance de la vérification personnelle biométrique en ligne, telle que les empreintes digitales, le scan des yeux, etc. Dans ce projet, il est proposé de vérifier la signature hors ligne à l'aide d'une machine vectorielle supportant une classe en premier, le réseau de neurones convolutionnels (CNN) en second. L'algorithme de La machine à vecteur de support à classe unique cherche à envelopper les inliers sous-jacents. L'objectif est de séparer les données en deux classes (basées sur une fonction de décision), la positive considérée comme la classe des inliers et la négative considérée comme la classe des outliers. De plus, la plupart des données d'apprentissage doivent appartenir à la classe positive alors que le volume d'enveloppe est minimal. CNN est un type de modèle de réseau de neurones qui nous permet d'extraire des représentations plus élevées pour l'image ,contenu. CNN prend les données de pixel brutes de l'image, forme le modèle, puis extrait automatiquement les caractéristiques pour une meilleure classification . Le principal avantage de CNN par rapport à ses prédécesseurs est qu'il détecte automatiquement les caractéristiques importantes sans aucune supervision humaine et qu'il a la plus grande précision parmi tous algorithmes qui prédisent les images.

## Abstract

Every person has his/her own unique signature that is used mainly for the purposes of personal identification and verification of important documents or legal transactions. There are two kinds of signature verification: static(offline) and dynamic(online). Static verification is the process of verifying an electronic or document signature after it has been made. Offline signature verification is not efficient and slow for a large number of documents. To overcome the drawbacks of offline signature verification, we have seen a growth in online biometric personal verification such as fingerprints, eye scan etc. In this project, it is proposed to verify the offline signature using a vector machine supporting one class first, convolutional neural network (CNN) second. The One-class Support Vector Machine (One-class SVM) algorithm seeks to envelop underlying inliers. The aim is to separate data into two classes (based on a decision function), the positive one considered as the class of inliers and the negative one considered as the class of outliers. Besides, most of the training data must belong to the positive class while the volume of envelope is minimal. CNN is a type of neural network model which allows us to extract higher representations for the image content. CNN takes the image's raw pixel data, trains the model, then extracts the features automatically for better classification . The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision also it has the highest accuracy among all algorithms that predicts images.



*Introduction générale*

## Introduction :

L'écriture de document est un mode de communication entre les humains à travers les siècles. Collecter et archiver des documents d'écriture permet de transmettre des informations dès la génération de génération en génération afin de préserver les souvenirs de l'humanité. De nos jours, chaque jour plus de un milliard de documents commerciaux sont délivrés par exemple par les banques ou les bureaux. Leur les traitements manuels nécessitent un grand effort de la part des opérateurs humains car ils doivent lire et retranscrire le contenu de chaque document, ce qui est fastidieux et prend beaucoup de temps. Par conséquent, il est nécessaire d'essayer le développement d'un système automatique qui remplace l'humain opérateur par un système de reconnaissance capable d'effectuer le même travail. Les avancées récentes dans le domaine de l'analyse et de la reconnaissance de documents, et la baisse des coûts de la puissance de calcul permettent désormais développement efficace de systèmes qui tentent de minimiser l'intervention humaine. Cependant, plusieurs obstacles doivent être surmontés pour parvenir à une reconnaissance manuscrite efficace et robuste système. Par conséquent, des travaux intensifs ont été menés au cours des dernières décennies afin de concevoir plusieurs systèmes de reconnaissance manuscrite tels que le tri du courrier, le chèque bancaire et le formulaire En traitement. le classificateur OC-SVM (One-Class Support Vector Machine) a été proposé d'avoir la capacité de distinguer l'objet de classe indépendamment des autres classes . Par conséquent, nous proposons dans cette thèse d'étudier la capacité de l'OC-SVM à résoudre certains de son utilisation dans de nombreuses applications comme la classification de documents , analyse intracrânienne, récupération d'images , audio surveillance , diarisation du locuteur , combinant différents traits biométriques et segmentation des images .

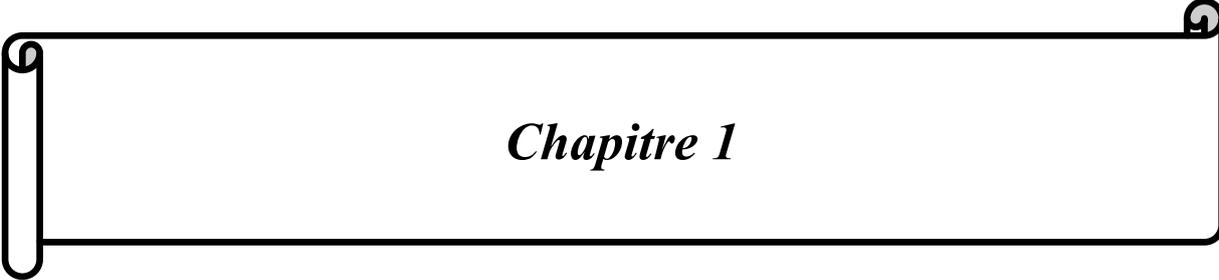
Dans cette mémoire , trois demandes de reconnaissance manuscrite sont envisagées pour évaluer la utilisation efficace de l'OC-SVM, qui sont la reconnaissance des chiffres manuscrits, la vérification de la signature et la reconnaissance du genre. Lors de l'utilisation de l'OC-SVM, nous sommes confrontés à trois problèmes principaux, qui sont les paramètres de réglage du classificateur OC-SVM lorsque le nombre d'échantillons est réduit, la sensibilité de la décision fonction au nombre d'échantillons et aussi la distance euclidienne utilisée dans le noyau comme un métrique. Ainsi, chacun de ces problèmes peut être résolu par les propositions suivantes.

- En règle générale, les modèles OC-SVM sont validés à l'aide de sous-ensembles de formation et de validation. Cependant, lorsqu'un échantillon réduit est disponible, l'étape de validation ne peut pas être effectuée. Par conséquent, diverses nouvelles techniques de validation du modèle OC-SVM sont proposées et comparées en utilisant uniquement la formation sans le sous-ensemble de validation. Plusieurs ensembles de données de référence pour l'évaluation et la comparaison, dédiés à la reconnaissance des chiffres manuscrits, à la vérification de la signature
- Cette technique est adoptée pour la vérification de la signature manuscrite. Dans un monde qui progresse vers l'innovation, la signature assume en effet la part la plus cruciale dans l'identification de un individu spécifique.

Cette section présente des données sur les avancées utilisées dans le projet qui sont le traitement d'image et Réseau de neurones convolutifs.

- Traitement des images
- Réseau de neurones artificiels
- Réseau de neurones convolutionnels Une organisation neuronale convolutive (CNN) est une sorte d'organisation neuronale contrefaite utilisée dans l'image l'accusé de réception et le traitement qui sont explicitement destinés à traiter les informations sur les pixels.

Ce mémoire présente une étude théorique et pratique en trois chapitres dont le premier est la généralités sur la vérification de la signature . le deuxième présente les méthodes utilisées dans la vérification de la signature et il est divisé en deux parties : la première partie présente Machine à vecteurs supports SVM et la 2<sup>ème</sup> présente le Deep learning . le troisième est divisé en deux parties : la première partie présente les résultats de la Machine à vecteurs supports SVM et la deuxième partie présente les résultats de deep learning avec la comparaison bien sûr .



*Chapitre 1*

## 1.1. Introduction:

La montée en puissance des technologies de l'information, avec pour corollaire, la dématérialisation des échanges, a accentué le besoin de disposer d'une technologie fiable permettant d'authentifier différents documents signés.

En effet, cette authentification consiste à numériser ces signatures pour pouvoir les analyser et les comparer de façon automatique par un logiciel approprié.

Cependant, la variabilité de ces signatures a confronté les algorithmes de classification et d'apprentissage à des problèmes difficiles et réalistes.

La nécessité d'assurer des performances élevées dans les applications réelles a poussé la recherche vers des modèles de classification de plus en plus sophistiqués. Ainsi, plusieurs générations de machines d'apprentissage ont vu le jour pour classifier ou catégoriser ces signatures.

## 1.2. Generalities sur la Biométrie:

### 1.2.1. Definition de la Biométrie:

La biométrie est une alternative aux deux précédents modes d'identification. Elle consiste à identifier une personne à partir de ses caractéristiques physiques ou comportementales. Le visage, les empreintes digitales, l'iris, etc. sont des exemples de caractéristiques physiques. La voix, l'écriture, le rythme de frappe sur un clavier, etc. sont des caractéristiques comportementales. Ces caractéristiques, qu'elles soient innées comme les empreintes digitales ou bien acquises comme la signature, sont attachées à chaque individu et ne souffrent donc pas des faiblesses des méthodes basées sur une connaissance ou une possession. En effet, un attribut physique ou comportemental ne peut être oublié ou perdu. En général, ils sont très difficiles à « deviner » ou à « voler » ainsi qu'à « dupliquer » [1]. La figure suivante différent moyenne biométriques:

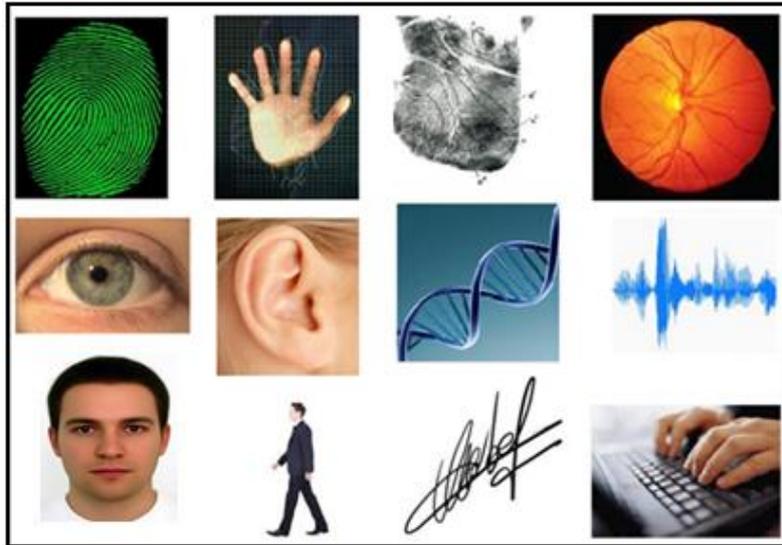


Figure 1. 1: Different moyens biométriques.

### 1.2.2. Définition et fonctionnalités d'un système biométrique

La biométrie permet l'identification et l'authentification d'une personne à partir de données reconnaissables et vérifiables, qui lui sont propres et qui sont uniques.

- **Authentification ou vérification biométrique**

Appelée également vérification, est le processus qui consiste à comparer les données caractéristiques provenant d'une personne, au modèle de référence biométrique de cette dernière (« Template »), afin de déterminer la ressemblance. Le modèle de référence est préalablement enregistré et stocké dans une base de données, dans un équipement ou objet personnel sécurisé. On vérifie ici que la personne présentée est bien la personne qu'elle prétend être [20].

- **Identification biométrique**

Il s'agit de saisir une donnée biométrique de cette personne, en prenant par exemple une photo de son visage, en enregistrant sa voix, ou en captant l'image de son empreinte digitale. Ces données sont ensuite comparées aux données biométriques de plusieurs autres personnes qui figurent dans une base [20].

### 1.2.3. Protocoles de réalisation d'un système biométrique

On distingue deux types de protocoles pour la réalisation d'un système biométrique et qui dépendent principalement de la manière dont les paramètres sont déterminés [2]:

**a. Dépendant de l'utilisateur**

Les paramètres sont adaptés pour chaque utilisateur de la base de données et la réalisation du système dépend de la variation des modalités biométriques, par conséquent et pour chaque nouvel utilisateur, les paramètres du système doivent être recalculés.

**b. Independent de l'utilisateur**

Les paramètres sont calculés pour toute la base de données indépendamment de l'utilisateur. Aussi pour l'ajout d'un nouvel utilisateur, il suffit juste d'établir son modèle. Dans ce cas, la procédure d'authentification sera élaborée comme suit:

1. Calcul des caractéristiques associées aux modalités biométriques.
2. Calcul des scores de similarités entre les caractéristiques associées aux modalités biométriques à tester et celles de références.
3. Comparaison des différents scores de similarités par rapport au seuil dans le but de déterminer la décision (modalité acceptée ou rejetée).

Dans le cadre de notre étude, nous nous sommes intéressés au protocole « Indépendant de l'utilisateur ».

**1.2.4. Définition de la signature manuscrite**

La signature manuscrite est le graphisme par lequel une personne s'identifie dans un acte et exprime son approbation au contenu de ce document, elle est le moyen biométrique d'authentification le plus utilisé.

Elle peut être formée d'une suite de lettres, de caractères, de chiffres ou de tous autres signes ou symboles dessinée avec la main de diverses manières et associée à des effets de style (traits, courbes, points) dans le but de la personnaliser et de la rendre unique.

**1.2.5. Functionalities de la signature manuscrite**

Un système d'authentification biométrique basé sur la signature manuscrite doit assurer les fonctions suivantes [3] :

- Adhésion au contenu: culturellement le fait d'apposer sa signature manuscrite signifie que l'on adhère au contenu indépendamment du support.
- Garantie de l'intégrité du document: elle peut être assurée par une fonction de hachage.
- Construction d'un original: la signature manuscrite sur papier ou sur interface graphique reste toujours unique et ne peut jamais être reproduite exactement de la même façon.

- Psychologique: l'utilisation de la signature manuscrite en amont de la signature électronique offre l'avantage de capter l'attention de l'individu.

Dans le cadre de l'élaboration de notre mémoire, nous nous sommes intéressés à l'authentification de la signature manuscrite.

### **1.2.6. Authentification de la signature manuscrite**

La signature manuscrite peut être considérée comme une méthode biométrique comportementale, du fait qu'elle repose sur des mouvements instinctifs. Ce postulat entraîne la stabilité de certaines caractéristiques de la signature pour chaque signataire.

Afin de réaliser un système d'authentification de la signature manuscrite, les caractéristiques dites stables doivent être isolées de celles qui ne le sont pas, du fait que la variation existante entre les signatures d'une même personne soit inférieure à celle entre les signatures de deux personnes différentes [4].

### **1.2.7. Mode d'acquisition de la signature manuscrite**

Les systèmes automatiques d'authentification des signatures manuscrites peuvent être classifiés, selon le mode d'acquisition de données utilisées, en deux catégories Système d'authentification automatique de signatures manuscrites en-ligne et système d'authentification automatique de signatures manuscrites hors-ligne [3].

#### **a) Mode d'acquisition en-ligne**

Dans un système en-ligne, la signature est acquise par le biais d'un dispositif dédié et muni d'un stylet électronique, qui peut être une tablette PC (Personal Computer), un PC à écran tactile ou un PDA (Personal Digital Assistant) qui fournit des propriétés dynamiques telles que les coordonnées, la pression, la vitesse, l'accélération et l'inclinaison du stylo à chaque point de la signature.

Les différentes expérimentations ont démontré que les dispositifs actuels d'acquisition de l'écriture manuscrite en-ligne sont loin d'être fiables en raison de la modification des usagers de leur manière de signer lors du passage sur un support numérique [3].

La figure suivante illustre un exemple de signature acquise en mode en-ligne:

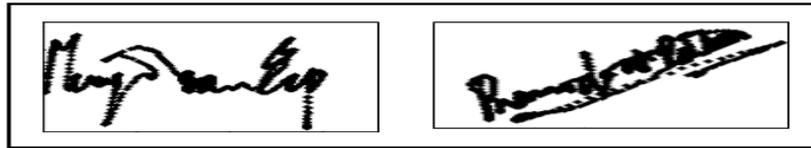


Figure 1. 2: Exemples d'acquisition de signatures en-ligne.

### b) Mode d'acquisition hors-ligne

Dans un système hors-ligne, la signature est réalisée sur un support papier puis numérisée par l'intermédiaire d'un scanner, d'où l'absence d'information directe telle que la dynamique. Par contre, d'autres informations sont disponibles, telles que l'épaisseur du trait ou la variation d'intensité du niveau de gris de l'image signature [3].

La figure suivante illustre un exemple de signatures acquises en mode hors-ligne:

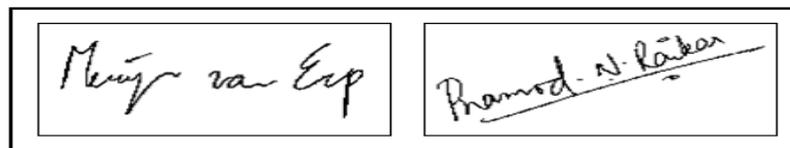


Figure 1. 3: Exemples d'acquisition de signatures hors-ligne.

### 1.2.8. Classes des signatures

L'authentification de la signature manuscrite s'avère incertaine, même pour les experts humains, du fait qu'elle représente une entité fortement variable [5]

Selon la littérature, on distingue deux classes de signatures: la classe des signatures authentiques et celle des signatures fausses [6].

#### A - Classe des signatures authentiques

Étant donné les différences d'origine et de culture entre les signataires à travers le monde, on distingue deux types de signatures :

- **La signature de type Américain:** Ces signatures s'apparentent à l'écriture cursive. Il est donc possible, pour un expert qui traite ce type de signatures d'utiliser un texte manuscrit écrit par le signataire pour comparer la forme de l'écriture
- **La signature de type Européen:** Ces signatures possèdent une composante graphique importante qui oblige à un traitement global des signatures.

Il existe d'autres types de signatures telles que le type arabe, chinois, etc..., cependant, nous ne traiterons dans ce mémoire que les signatures dites «latines».

**B – Classe des signatures fausse**

Nous consacrons ce paragraphe à l'énumération des types de faux. Suivant le type de faux recherché, l'expert utilise des traitements spécifiques, tel que les proportions de la signature, le nombre de parties des signatures et les projections, dont nous pourrions nous inspirer ultérieurement pour la mise au point d'un système d'authentification automatique. Généralement subdivisée en plusieurs groupes et qui sont les suivants:

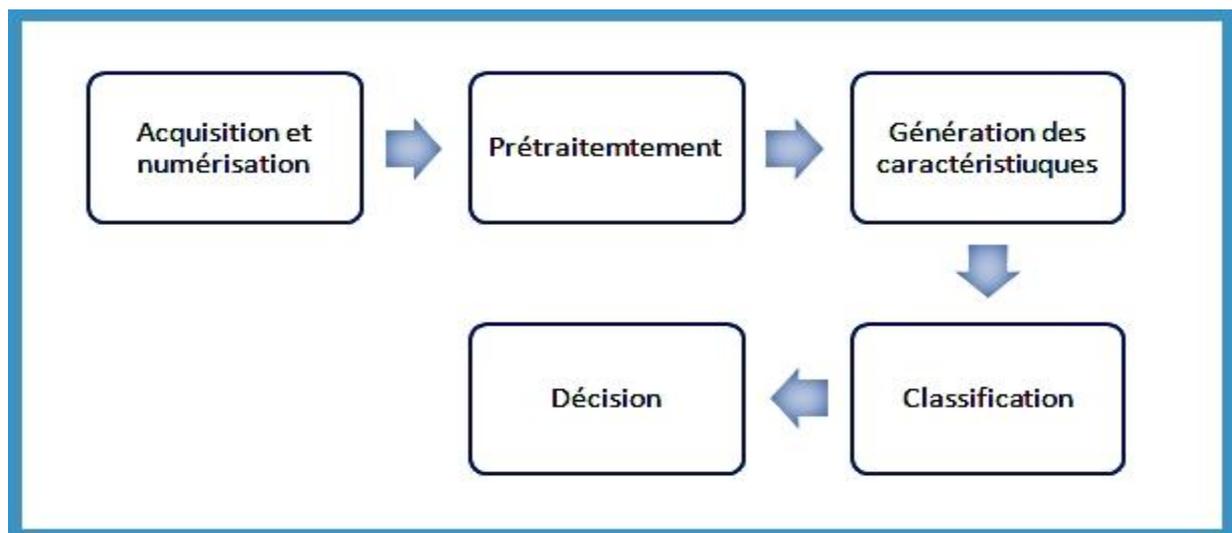
- **Les faux par déguisement :** Le faux par déguisement est particulier car il correspond à une signature faite par le signataire présumé (d'origine) mais déguisée délibérément dans le but de pouvoir renier celle-ci ultérieurement. Ces signatures sont généralement ressemblantes malgré une perte d'harmonie dans le tracé et des changements de vitesse. Une étude fine faite par un expert permet de les détecter.
- **Les faux par imitation servile :** Dans le cas d'un faux par imitation servile, le faussaire doit posséder un exemplaire de la signature authentique. Ce faux quoique ressemblant à l'original présente des différences dans les espacements, dans les inclinaisons. De plus le tracé est lent et hésitant d'où des variations visibles de la pression.
- **Les faux par imitation libre :** Pour ce faux le faussaire étudie soigneusement la signature authentique et s'entraîne à la reproduire de mémoire jusqu'à être satisfait du résultat. Ce sont, de l'avis des experts, les faux les plus difficiles à détecter car ils sont très ressemblants et le tracé est spontané. Ils diffèrent des originaux par les proportions relatives des éléments de la signature et par l'alternance des pleins et de déliés.
- **Les faux par calque :** Le faux par calque reproduit fidèlement l'image d'une signature authentique sur un document par l'utilisation d'un moyen de recopiage par exemple : une copie par transparence, avec du carbone, par photocopie. Ce faux est évidemment très difficile à détecter même pour les experts.
- **Les faux grossiers :** Dans le cas d'un faux grossier, le faussaire n'essaie pas de faire un faux ressemblant à un original. Ce faux est fréquent et il est le plus facile à détecter. Ce sont ces faux que nous espérons pouvoir détecter facilement à l'aide de la méthodologie développée dans ce mémoire.
- **Les faux aléatoires :** Les signatures des personnes autres que le signataire présumé sont appelées des faux aléatoires. Pour tester les systèmes, c'est le faux le plus simple à simuler. En effet, il est difficile d'obtenir des banques de faux réels en quantité

suffisante alors qu'il suffit de prendre les signatures des autres personnes dans une base de signatures. Les faux aléatoires font partie des faux grossiers et on suppose que les résultats obtenus sur ce type de faux sont identiques à ceux que l'on obtiendrait sur des faux grossiers.

- **Le faux simple:** le faussaire fabrique une signature à partir du nom du signataire sans imiter un original. Ce faux est généralement peu ressemblant à l'original mais reste plus difficile à détecter que le faux grossier.

### 1.2.9. Structure générale d'un système d'authentification de signatures hors-ligne

Comme tout système de reconnaissance de forme, le système d'authentification de signatures hors-ligne permet de passer de l'espace de représentation vers celui d'interprétation grâce aux modules présentés sur la figure 1.4 [7]:



**Figure 1. 4:** Structure générale d'un système de reconnaissance d'une écriture.

#### 1.2.10. Acquisition de l'image

Le but de cette étape consiste à obtenir une image numérique. Les données du système de vérification de signature statique sont acquises de différentes manières par exemple caméras ou scanners. Les échantillons de signature sont numérisés, puis les images numérisées sont stockées numériquement pour un traitement ultérieur.

##### 1.2.10.1. Pré-traitement:

Le prétraitement consiste essentiellement à réduire le bruit superposé aux données et tenter de préserver l'information significative de la forme représentée. Le bruit peut être dû aux

conditions d'acquisition (éclairage, mise incorrecte du document, ...) ou encore à la qualité du document d'origine. Effectuer une banalisation, segmentation et une normalisation si c'est nécessaire.

### *A -Binarisation*

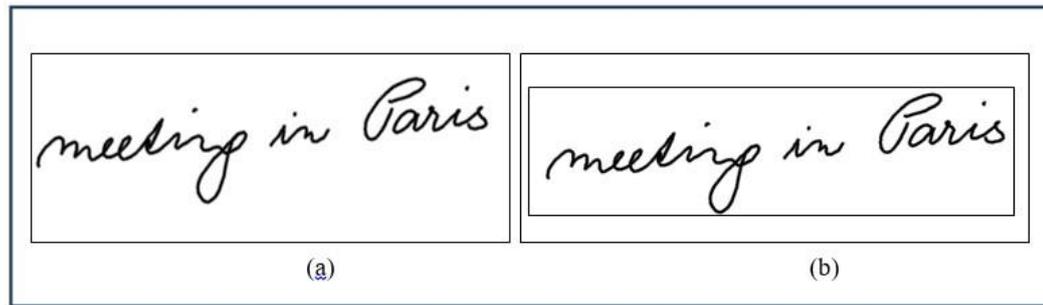
La binarisation est un filtrage de l'image doit donc être effectué, de manière à réduire le nombre de ces pixels parasites. Cette étape a pour but de faciliter l'extraction du caractère ou du texte manuscrit qui constitue le premier-plan de l'image, de son arrière-plan et de réduire le bruit. La binarisation se fait par le biais d'un seuil qui a une relation directe avec le caractère homogène ou complexe de l'arrière-plan. Pour un arrière-plan homogène, un seuil global unique suffit tandis que dans le cas contraire ou la distinction entre l'arrière-plan et le texte devient plus difficile, un seuillage adaptatif est utilisé. La Figure 1.5 représente un exemple de binarisation.[8].



**Figure 1. 5:** Exemple de binarisation, (a): Image brute, (b): Image binaire.

### *B -Segmentation*

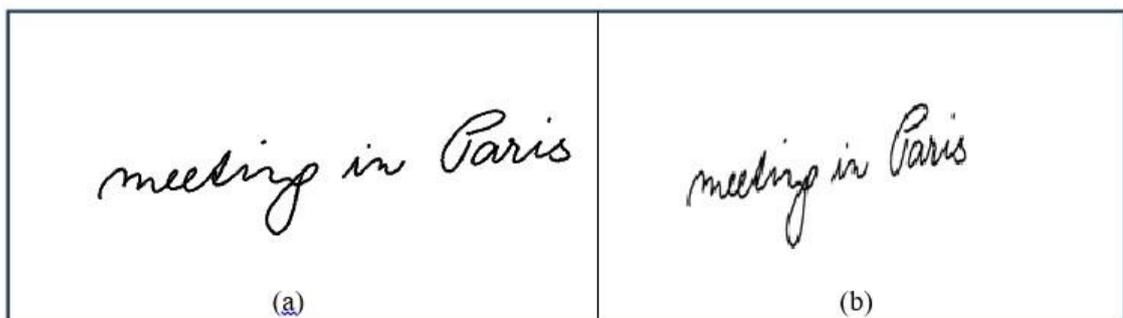
Cette technique consiste à découper le bloc textuel en plusieurs segments et les séparer par détection des zéros successifs afin de localiser la zone du texte dans l'image en indiquant ses limites gauche, droite, haute et basse. La Figure 1.6 représente un exemple de segmentation [9].



**Figure 1. 6:** Exemple de segmentation, (a): Image originale, (b): Image segmentée.

### *C -Normalisation*

L'aspect variant de l'écriture entre les personnes, ainsi que la diversité des systèmes d'acquisitions donnent des images de taille variable. Parfois, afin de pouvoir établir une base de comparaison robuste, un redimensionnement de la taille des images s'impose. Une image de petite taille optimise le temps de calcul, sauf que cette dernière doit être bien choisie par expérimentation afin de ne pas engendrer des pertes d'informations. La Figure 1.7 représente un exemple de normalisation [10].



**Figure 1. 7:** représente un exemple de normalisation

#### **1.2.10.2. Génération des caractéristiques**

L'efficacité d'un système de vérification de signature dépend principalement de l'étape d'extraction des caractéristiques. Les techniques d'extraction de caractéristiques doivent être rapides et faciles à calculer pour que le système soit optimal et ait une faible puissance de calcul. Les caractéristiques sélectionnées doivent faire la distinction entre une signature authentique et une signature falsifiée. Les caractéristiques extraites pour la vérification de signature statique peuvent être divisées en caractéristiques globales, locales et géométriques.

### 1.2.11. Classification

La classification consiste à affecter chaque écriture à sa classe correspondante, cette affectation est établie suivant une règle de décision. Les approches de classification se subdivisent en deux grandes familles, la classification supervisée et classification non-supervisée.

#### 1.2.11.1. L'apprentissage supervisé

Cette approche suppose que toutes les classes sont connues ou prédéfinies. La problématique consiste à associer la forme testée à la classe la plus adaptée. Pour cela, le classifieur doit effectuer un processus appelé « apprentissage » qui consiste à déterminer les frontières de séparation entre les différentes classes, selon deux approches : classification par séparation et classification par modélisation comme le montre la Figure 1.8 suivante [11] :

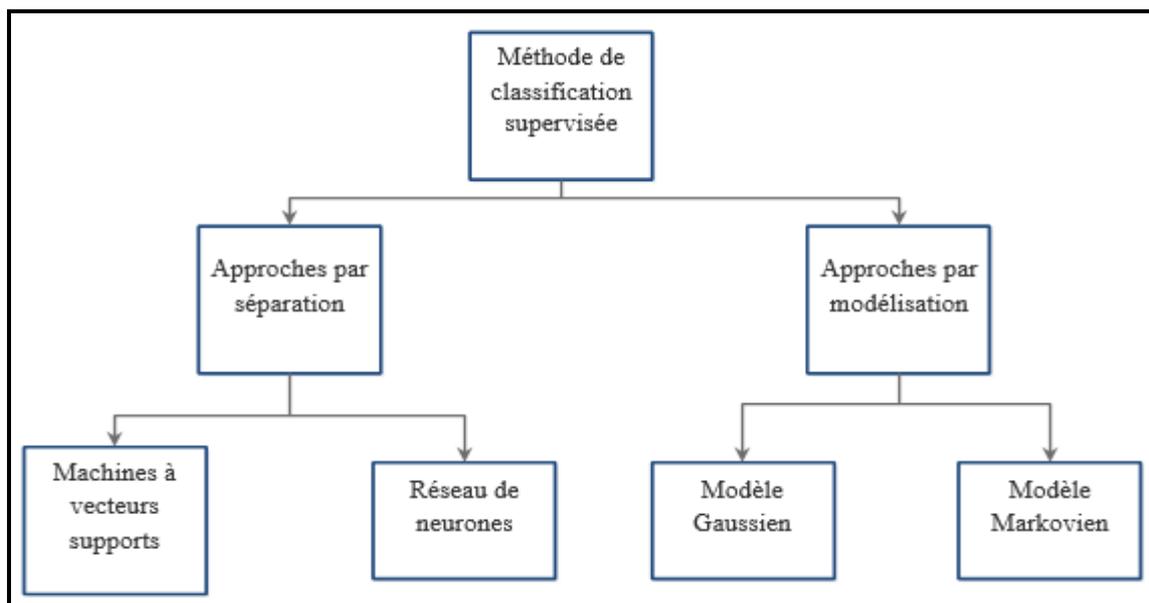


Figure 1. 8: Différentes approches de la classification supervisée.

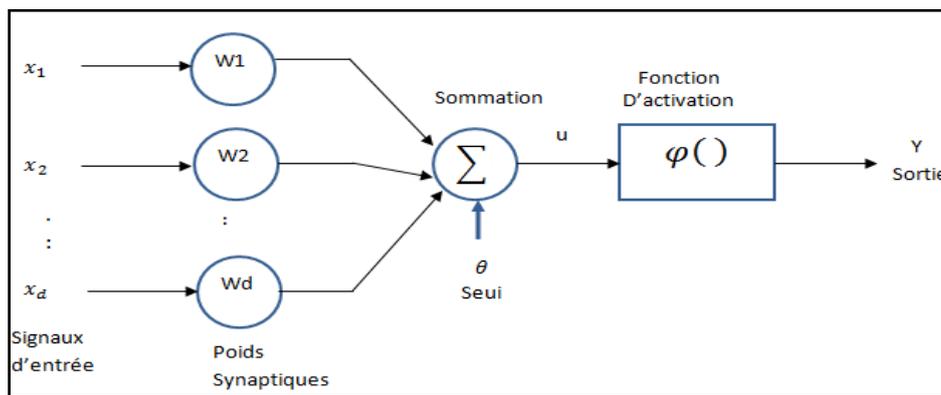
#### *A - Approches par séparation*

- **Réseau de neurones**

L'idée de cette approche était de comprendre le mécanisme de fonctionnement du cerveau humain et s'en inspirer pour établir une modélisation mathématique utile pour diverse domaine [12].

Le réseau de neurones artificiels est un ensemble de neurones formels interconnectés, où chaque neurone réalise une somme pondérée de ses entrées qui sera par la suite, comparée à un seuil, dans le but de déterminer une fonction algébrique [13].

Les réseaux de neurones ont dominé plusieurs années jusqu'à l'apparition des machines à vecteur de support qui ont été rapidement adaptées en raison de leurs capacités et du nombre de paramètres restreints à régler. La Figure 1.9 représente le principe général de fonctionnement du neurone formel.



**Figure 1. 9:** Principe général de fonctionnement du neurone formel.

$$\text{Avec } u = \sum_{i=1}^d w_i x_i \quad (1.1)$$

$$\text{et } y = \varphi(u) \quad (\varphi : \text{fonction d'activation}). \quad (1.2)$$

- **Machine à vecteurs supports (SVM)**

Elle est définie comme étant une famille d'algorithmes d'apprentissage permettant de résoudre efficacement les problèmes de classification. Le principe de ce modèle discriminant binaire consiste à minimiser les erreurs de généralisation tout en maximisant la marge séparant les données des différentes classes (voir la figure 1.11). Ce processus produit un ensemble réduit de prototypes appelés communément vecteurs de support [14].

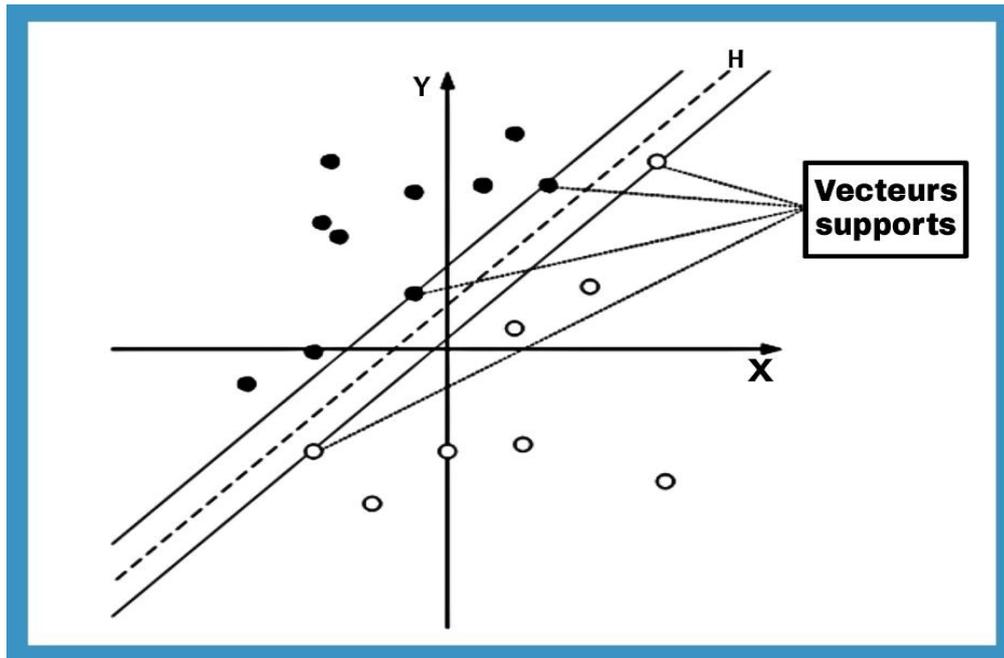


Figure 1. 10: Principe général d'un classifieur SVM

### *B - Approches par modélisation*

- **Modèle Markovien**

Ce modèle utilise essentiellement des données unidimensionnelles, ce qui a permis leur application directe en traitement de la parole, et ensuite pour l'OCR. L'inconvénient est que les HMMs sont monodimensionnels alors que les images de caractères sont en dimension 2, ce qui donne des résultats parfois peu satisfaisants pour une application directe des HMMs [15]. L'intérêt d'un tel modèle est de pouvoir prendre des « décisions » (classifications, estimations, filtrages,...) localement à partir de toute l'information disponible dans l'image [16].

- **Modèle Gaussien**

Ce modèle est basé sur une estimation paramétrique des fonctions de répartition des vecteurs caractéristiques. Il suppose que les éléments de chaque classe possèdent une distribution gaussienne multi-variable. Dans la mesure où cette hypothèse s'avère exacte, le modèle gaussien permet d'obtenir des frontières optimales de décision de Bayes [17].

### 1.2.11.2. L'apprentissage non-supervisé

Cette approche suppose que les formes à identifier sont inconnues. Ainsi, le classifieur doit calculer les scores de similarités entre ces dernières en tenant compte de la classe la plus adéquate, avec la méthode la plus connue qui est la classification par partition (k-means) [11].

### 1.2.12. Décision

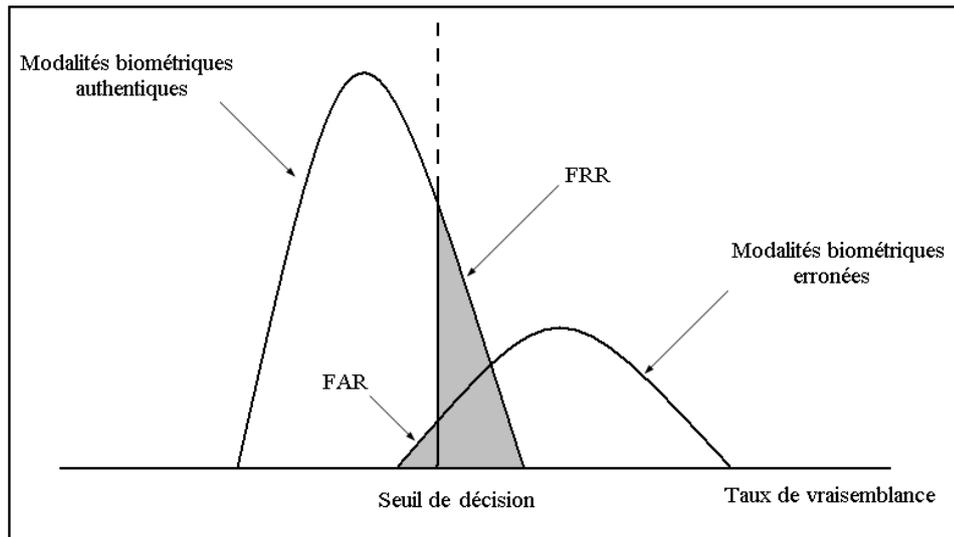
La décision permet de sélectionner les réponses appropriées fournies par le classifieur afin de les affecter à la classe appropriée [18].

### 1.2.13. Mesure de robustesse d'un système d'authentification biométrique

Un système d'authentification biométrique peut accepter des modalités biométriques erronées ou rejeter d'autres authentiques [19]. Pour cela, quatre taux d'erreurs peuvent être calculés afin d'évaluer ses performances:

- **FRR** (False Rejection Rate): il définit le taux des modalités biométriques authentiques rejetées par le système alors qu'elles devraient être acceptées.
- **FAR** (False Acceptance Rate): il indique le taux des modalités biométriques erronées acceptées alors qu'elles devraient être rejetées par le système.
- **EER** (Equal Error Rate): ce taux correspond à la valeur où le FRR est égal au FAR. Plus la valeur du taux EER est faible, plus le système est performant comme l'illustre la figure 1.17.
- **HTER** (Half Total Error Rate): il correspond à la moyenne du taux des faux rejets (FRR) et du taux des fausses acceptations (FAR).

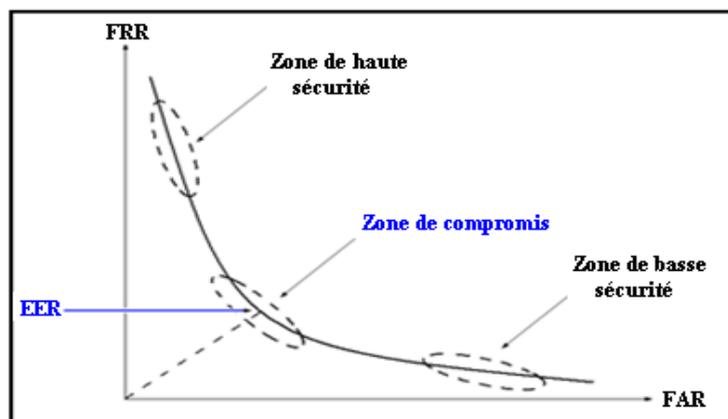
La Figure 1.12 présente la distribution des taux de vraisemblances des modalités biométriques authentiques et celles erronées dans un système biométrique



**Figure 1. 11:** Distributions des taux de vraisemblances des modalités biométriques authentiques et celles erronées dans un système biométrique.

Le choix du seuil de décision dépend du niveau de sécurité de l'application. Pour un système très sécurisé, on minimise les fausses acceptations. Par contre, pour un système où la sécurité n'est pas critique, on minimise les faux rejets [19].

Il existe plusieurs manières de représenter graphiquement les mesures de performance d'un système d'authentification dont la plus répandue est: la courbe ROC (Receiver Operating Characteristic) utilisée pour tracer le taux des faux rejets (FRR) en fonction du taux des fausses acceptations (FAR) (figure 1.13) [19].



**Figure 1. 12:** Courbe ROC.

Dans le cadre de notre travail, nous utiliserons une des variantes de la courbe ROC qui consiste à tracer les taux des fausses acceptations (FAR) et des faux rejets (FRR) en fonction du seuil de décision. L'intersection des courbes FAR et FRR constitue le seuil optimal où les deux taux sont égaux (figure 1.14).

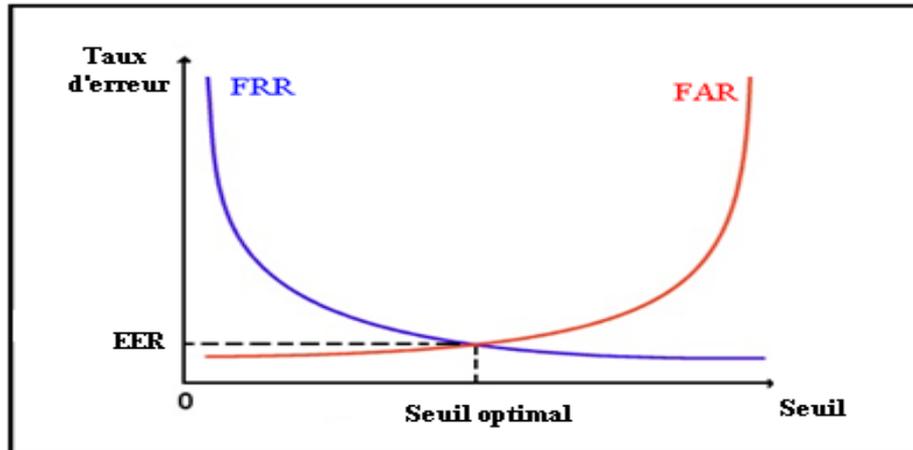
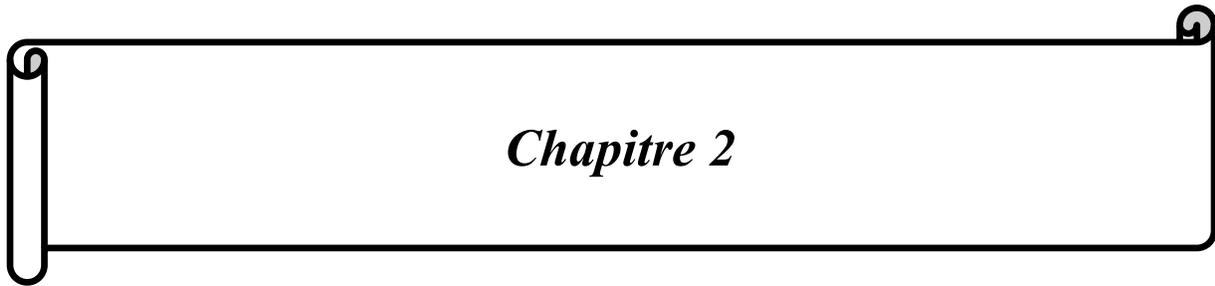


Figure 1. 13: Taux d'erreurs FRR et FAR en fonction du seuil optimal de décision.

### 1.3. Conclusion

Dans ce chapitre nous avons pu retenir que la signature manuscrite est une caractéristique propre à chaque être humain mais qu'elle n'a pas été à l'abri de falsifications. C'est dans ce sens que sont apparus les systèmes de vérification statique (vérification effectuée à partir d'une image brute numérisée) et de vérification dynamique (vérification à partir d'éléments dynamiques recueillis pendant que la signature est effectuée tels que le temps, la pression, la vitesse etc.). On a également donné les généralités sur l'apprentissage automatique, sa manière de fonctionner et l'évaluation de ses performances en mettant un accent sur les réseaux de neurones : sa conception et sa manière d'apprendre.



*Chapitre 2*

### 2.1. Introduction

L'apprentissage de la classification à partir d'échantillons de données en est un des tâches les plus importantes en intelligence artificielle. Il appartient aux approches d'apprentissage supervisé, qui est la forme la plus courante en apprentissage automatique.

Ce chapitre comportera les machine d'apprentissage notamment les machines à supports Vecteur (Support Vector Machines SVM) et les réseaux de neurones approfondies (Deep learning)

### 2.2. Machine à vecteurs supports SVM

Les machines à vecteurs de support ou support vector machines sont des algorithmes d'apprentissage utilisés en machine learning pour résoudre des problèmes de discrimination et de régression. Les SVM sont une généralisation des classifieurs linéaires.

Le SVM est une solution à ce problème de classification. Le SVM appartient à la catégorie des classifieurs linéaires (qui utilisent une séparation linéaire des données), et qui dispose de sa méthode à lui pour trouver la frontière entre les catégories.

Pour que le SVM puisse trouver cette frontière, il est nécessaire de lui donner des données d'entraînement. En l'occurrence, on donne au SVM un ensemble de points, dont on sait déjà si ce sont des ronds rouges ou des ronds bleus. A partir de ces données, le SVM va estimer l'emplacement le plus plausible de la frontière : c'est la période d'entraînement, nécessaire à tout algorithme d'apprentissage automatique.

Une fois la phase d'entraînement terminée, le SVM a ainsi trouvé, à partir de données d'entraînement, l'emplacement supposé de la frontière. En quelque sorte, il a « appris » l'emplacement de la frontière grâce aux données d'entraînement. Qui plus est, le SVM est maintenant capable de prédire à quelle catégorie appartient une entrée qu'il n'avait jamais vue avant, et sans intervention humaine. Pour notre problème le SVM a choisi une ligne droite comme frontière. C'est parce que, comme on l'a dit, le SVM est un classificateur linéaire. Bien sûr, la frontière trouvée n'est pas la seule solution possible, et n'est probablement pas optimale non plus.

Cependant, il est considéré que, étant donné un ensemble de données d'entraînement, les SVM sont des outils qui obtiennent parmi les meilleurs résultats. En fait, il a même été prouvé

que dans la catégorie des classificateurs linéaires, les SVM sont ceux qui obtiennent les meilleurs résultats.

Un des autres avantages des SVM, et qu'il est important de noter, est que ces derniers sont très efficaces quand on ne dispose que de peu de données d'entraînement : alors que d'autres algorithmes n'arriveraient pas à généraliser correctement, on observe que les SVM sont beaucoup plus efficaces. Cependant, quand les données sont trop nombreuses, le SVM a tendance à baisser en performance

### 2.2.1. Notions de base

#### 2.2.1.1. Hyperplan

Le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan. On détermine un hyperplan qui sépare les deux ensembles de points (classe 1/ classe 2) comme le montre la Figure 2.1

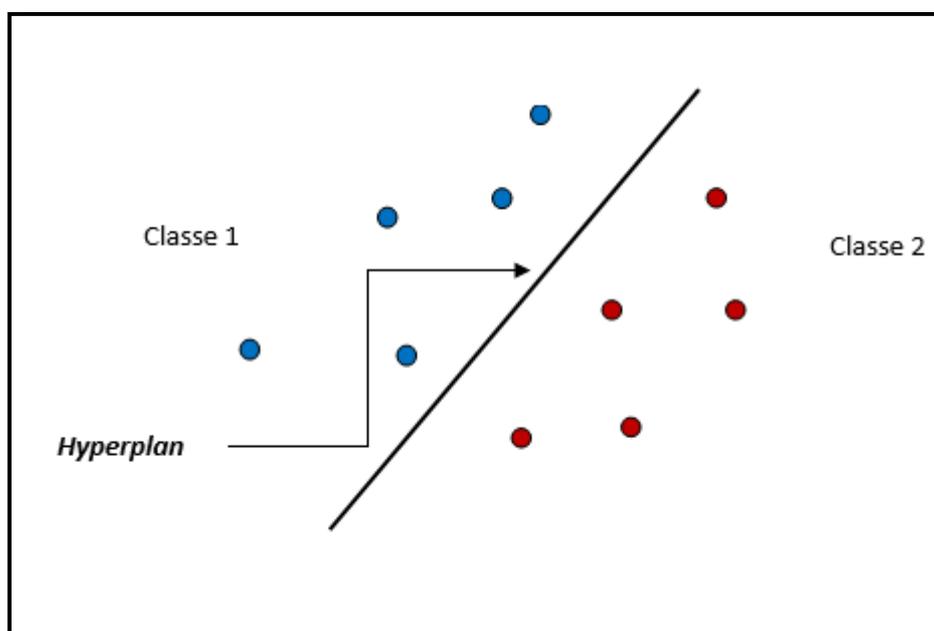


Figure 2. 1: Exemple d'un hyperplan séparateur

#### 2.2.1.2. Vecteurs de support

Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support comme le montre la Figure 2.2

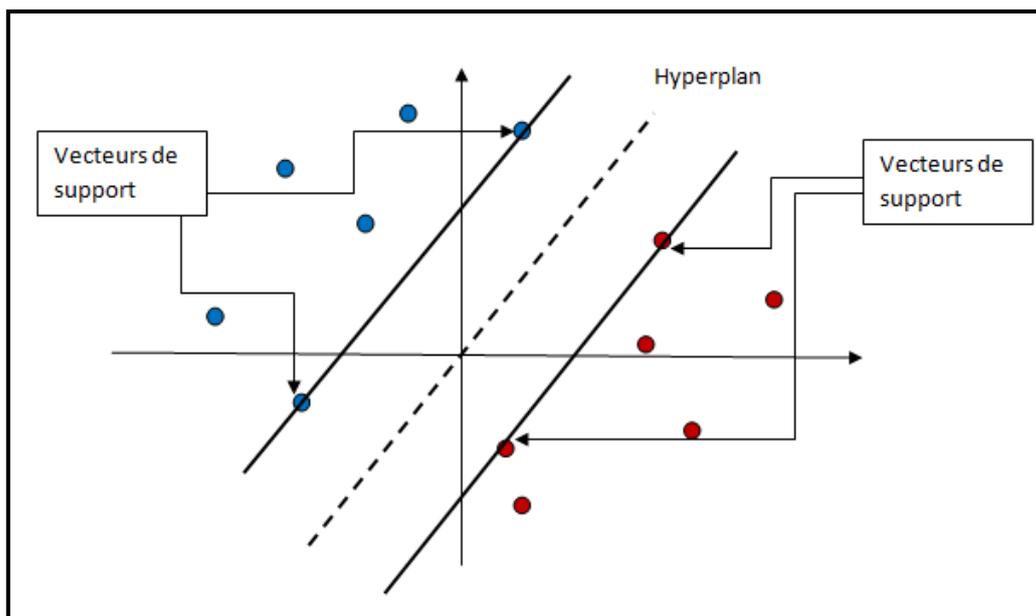


Figure 2. 2: Les vecteurs de support.

### 2. 2.1.3. Marge

Il existe plusieurs hyper plan permettant de séparer entre les deux classes mais nous allons en plus chercher dans la distance entre les points des deux classes. On appelle cette distance « marge » entre l'hyperplan et les exemples. L'hyperplan séparateur optimal est celui qui maximise la marge. Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge comme le montre la Figure 2.3

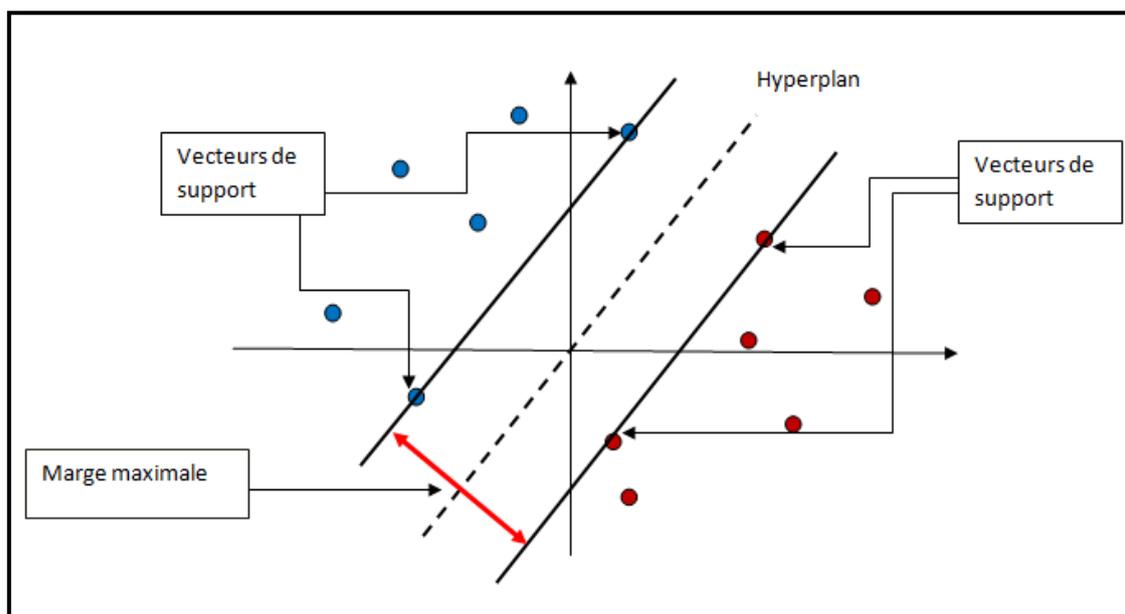


Figure 2. 3: Hyperplan, vecteurs de support et marge maximale.

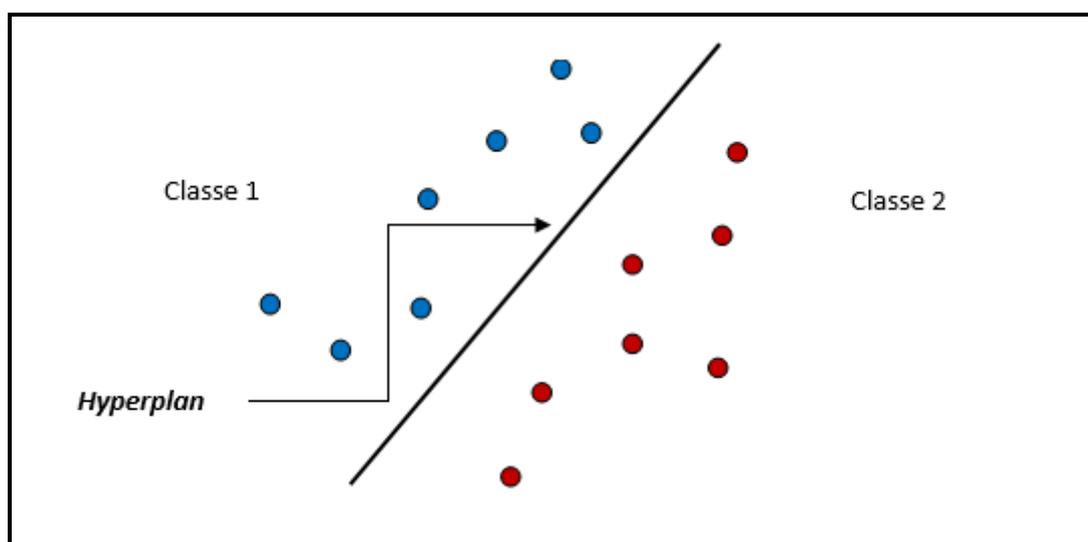
Intuitivement, le fait d'avoir une marge plus large procure plus de sécurité lorsque l'on classe un nouvel exemple. De plus, si l'on trouve le classificateur qui se comporte le mieux vis-à-vis des données d'apprentissage, il est clair qu'il sera aussi celui qui permettra au mieux de classer les nouveaux exemples.

### 2.3.1. Principe de fonctionnement général des SVMs :

La méthode SVM est une méthode de classification linéaire qui repose sur l'hypothèse qui dit que, étant donné un espace approprié, dans un cas linéairement séparable, il existe un classifieur linéaire appelé hyperplan permettant de séparer deux classes de l'espace (classe 1/ classe 2). Le but de cette méthode est d'apprendre, à partir d'un ensemble d'exemples d'apprentissage (entraînement), une fonction qui prédit les classes pour de nouveaux objets. Plus concrètement, il s'agit de trouver l'hyperplan optimal, qui sépare au mieux les données et maximise la distance entre les deux classes. La classification d'un nouvel exemple inconnu est donnée par sa position par rapport à l'hyperplan optimal [1].

### 2.3.2. Bi-Class Support Vector Machine (BC-SVM)

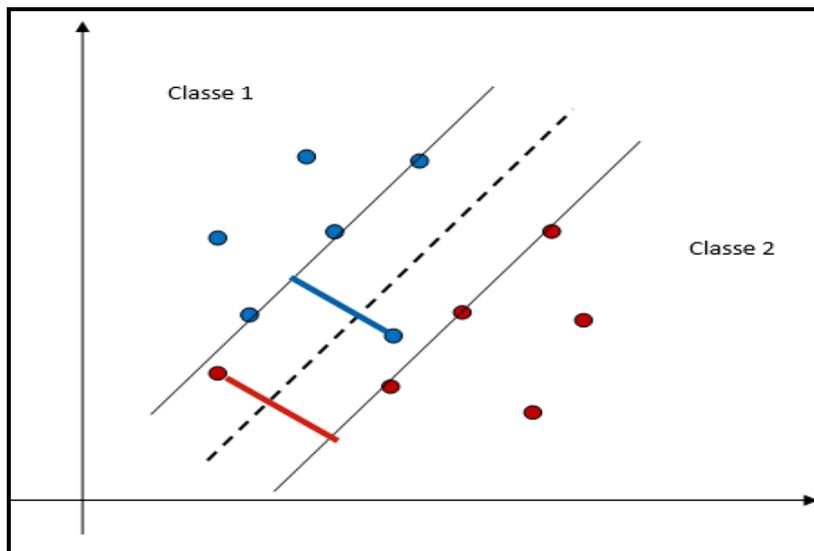
On utilise Ce classifieur pour la reconnaissance, il cherche à séparer un espace de descripteurs en deux classes. Si on dit que les données d'apprentissage sont linéairement séparables, c'est à dire qu'il existe un hyperplan qui sépare les données sans erreur comme le montre la figure 2.4 [4].



**Figure 2. 4:** Discrimination correcte d'un ensemble de deux classes.

Souvent il arrive que même si le problème est linéaire, les données sont affectées par un bruit et les deux classes se retrouvent mélangées autour de l'hyperplan de séparation dans ce

cas nos données d'entraînement ne sont pas linéairement séparables. Comme le montre la figure 2.5.

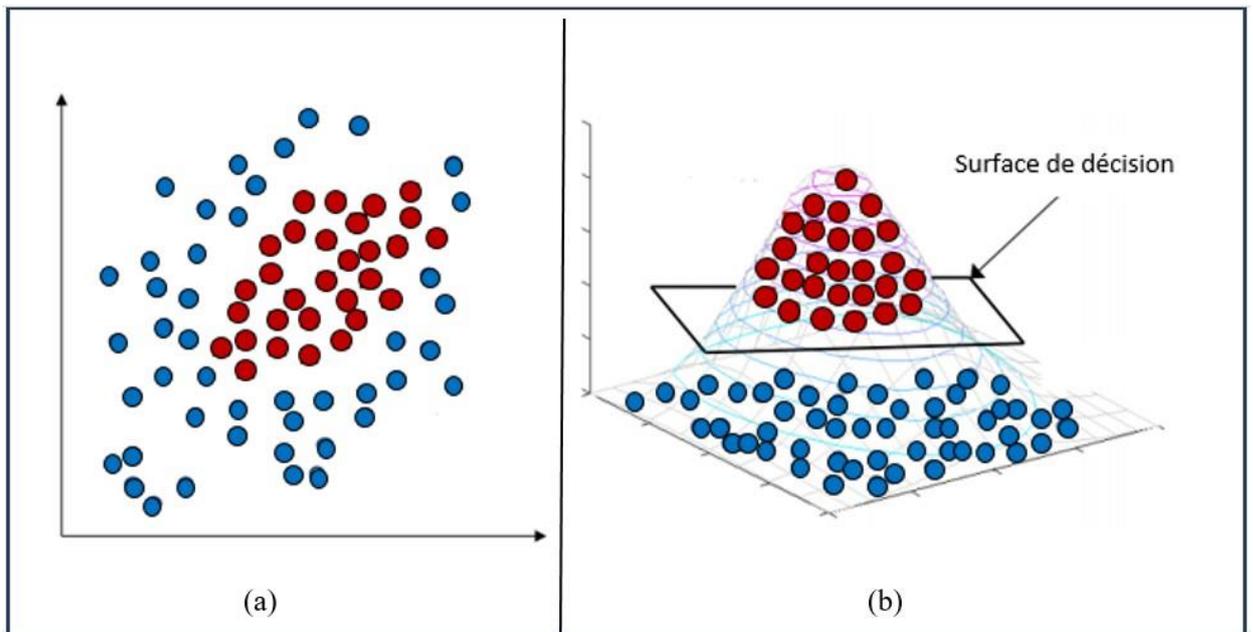


**Figure 2. 5:** Exemple d'une difficulté à séparer deux classes

Pour gérer ce type de problème on utilise une technique qui est appelée Kernel Trick (Fonction noyau) pour classer correctement nos échantillons de part et d'autre de l'hyperplan.

### 2.3.2.1. La fonction noyau

une classe de méthodes qui permettent de projeter les données dans un espace de plus grande dimension, appelé « Espace de redistribution ». Il est clair que plus la dimension de l'espace de re-description est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples est élevée. Les noyaux garantissent le bon fonctionnement d'un ensemble de caractéristiques mathématiques qui permet de calculer le produit scalaire entre les projections de deux vecteurs via une fonction noyau [2]. Comme le montre la figure 2.6 suivante :



**Figure 2. 6:** (a) Données non linéairement séparables, (b) Données séparées par la fonction noyau.

Le tableau 1 illustre quelques exemples de noyaux :

Noyau	$k(x,x')$
<i>RBF(Radial Basis Function)</i>	$\text{Exp}\left(\frac{-\ x-x_i\ ^2}{2 \times \sigma^2}\right)$
<i>Linéaire</i>	$x \cdot x'$
<i>Polynomial</i>	$(x \cdot x')^d$ ou $(c + x \cdot x')^d$
<i>Gaussien</i>	$e^{-\ x-x'\ ^2/\sigma}$
<i>Sigmoïde</i>	$\tanh(\alpha_0(x \cdot x') + b_0)$ .

**Tableau 2. 1:** Quelques exemples des noyaux.

**Remarque :**  $\|x - x_i\|$  dans le tableau X est la distance euclidienne entre deux échantillons.

Le paramètre  $\sigma$  appelé « largeur de bande » dont le réglage est souvent critique pour le bon fonctionnement de la méthode.

Dans notre travail, nous avons utilisé la fonction noyau gaussien K qui est un exemple de noyau à fonction de base radiale RBF puisqu'elle est la plus adaptée et la plus utilisée dans les systèmes de reconnaissances des formes.

$$k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \quad (2.1)$$

### 2.3.2.2. La Fonction de décision

$$f(x) = w^T x + b \quad (2.3)$$

Avec :  $w$  : vecteur de pondération et  $b$  : biais.

Cependant, dans certaines applications, lorsqu'un seul ensemble d'échantillons appartenant à la classe cible est disponible, le BC-SVM n'est pas possible à utiliser car il nécessite des exemples et des contre-exemples pour sa formation. Afin d'utiliser un classifieur SVM efficacement, il existe le classifieur OC-SVM (One-Class Support Vector Machine).

### 2.3.3. One Class Support Vector Machine (OC-SVM)

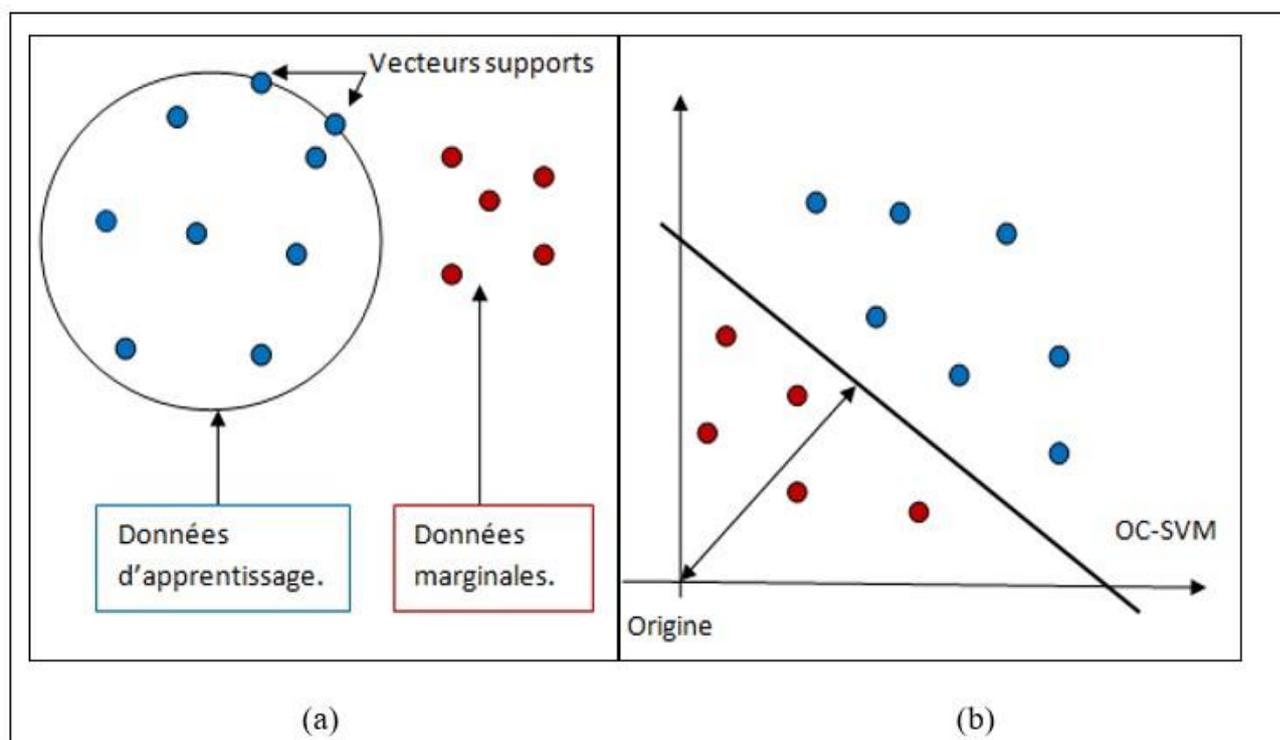
Nous savons tous que SVM a été une approche très populaire pour les problèmes de classification non linéaire. Cependant, dans les énoncés de problèmes, où il n'y a qu'une seule classe, comme dans la détection non supervisée des valeurs aberrantes, l'approche actuelle doit être modifiée.

One Class SVM, qui est une version modifiée de SVM, est venu comme un sauveur.

La solution pour OC-SVM est essentiellement dérivée de la solution de SVDD (Support Vector Data Description) [3].

#### 2.3.3.1. La fonction noyau

Le même principe est appliqué sur ce classifieur comme le montre la figure suivante.



**Figure 2. 7:** - (a) Distribution des données d'apprentissage, (b) Projection des données dans l'espace de redistribution.

### 2.3.4. Mesures de distance du noyau OC-SVM

Le résultat de la fonction de décision ne repose que sur le produit scalaire des vecteurs dans l'espace des caractéristiques, il n'est pas nécessaire d'effectuer une projection explicite dans cet espace. Tant qu'une fonction a les mêmes résultats, elle peut être utilisée à la place. C'est ce qu'on appelle l'astuce du noyau et c'est ce qui donne aux SVM une si grande puissance avec des points de données séparables non linéaires ; l'espace des caractéristiques peut être de dimension illimitée et donc l'hyperplan séparant les données peut être très complexe. Dans nos calculs cependant, nous évitons cette complexité. Nous nous sommes basées sur les cinq distances qui sont présentées dans le tableau ci-dessous :

<i>Mesure de la Distance</i>	$d(x, x_i)$
<i>Euclidienne</i>	$\sqrt{\sum_{k=1}^d (x_k - x_{i,k})^2}$
<i>Cityblock</i>	$\sum_{k=1}^d  x_k - x_{i,k} $
<i>Chebychev</i>	$\max_{k=1, \dots, d}  x_k - x_{i,k} $
<i>Correlation</i>	$\frac{\text{COV}((x, x_i))}{\sigma_x \sigma_{x_i}}$
<i>Spearman</i>	$\frac{1 - 6 \sum_{i=1}^d (\text{rank}(x) - \text{rank}(x_i))^2}{n(n^2 - 1)}$

**Tableau 2. 2:** Different mesure de la distance

### 2.3.4.1. Fonction de décision

Théoriquement, un motif est correctement classé et donc accepté lorsque la fonction de décision  $f(x)$  est positive. Implicitement, le seuil est fixé à zéro. Cette approche peut être considérée comme un seuil dur. En effet, certains échantillons de signature proches de l'hyper plan dans l'espace de fonctions ne sont pas acceptés et sont donc rejetés.

Afin de réduire cette contrainte, nous proposons un seuil doux pour réduire les erreurs de classification. Nous adoptons donc la règle de décision suivante:

$$x \in \begin{cases} \text{Accepté} & \text{si } f(x) \geq t \\ \text{Sinon Rejeté} & \end{cases} \quad (2.3)$$

- Dans le cadre de notre travail, la conception de chaque classifieur OC-SVM est réalisée selon les étapes suivantes :

- **La phase d'apprentissage** : Dans cette phase le classifieur apprend uniquement les formes de la classe des écritures authentiques dans le but de déterminer un modèle qui servira au calcul d'un seuil de décision optimal lors d'une phase intermédiaire dite de validation.
- **La phase de validation** : Cette phase permet de récupérer un modèle et un seuil optimal (les paramètres  $\mu$  et  $\gamma$ ) suivant lesquels elle permet de réaliser la phase suivante.
- **La phase de test** : Cette phase consiste à évaluer la robustesse du classifieur en utilisant le modèle et le seuil optimal sélectionné lors de la validation afin d'établir une décision.

### 2.4. Deep learning

#### 2.4.1. Généralité : deep learning

##### 2.4.1.1. Définition

Le deep learning ou apprentissage profond est un sous-ensemble de l'apprentissage machine où les réseaux neuronaux artificiels, des algorithmes inspirés du cerveau humain, apprennent à partir de grandes quantités de données. De la même manière que nous apprenons par expérience, l'algorithme d'apprentissage profond exécuterait une tâche de manière répétitive, en la modifiant chaque fois un peu pour améliorer le résultat.

Nous parlons d'apprentissage profond parce que les réseaux neuronaux ont plusieurs couches (profondes) qui permettent l'apprentissage. Tout problème qui nécessite une "réflexion" pour être résolu est un problème que l'apprentissage profond peut apprendre à résoudre.

La quantité de données que nous générons chaque jour est stupéfiante - actuellement estimée à 2,6 quintillions d'octets et c'est cette ressource qui rend l'apprentissage profond possible. Comme les algorithmes d'apprentissage approfondi nécessitent une tonne de données pour apprendre. En plus de la création de données, les algorithmes d'apprentissage approfondi bénéficient d'une puissance de calcul plus importante aujourd'hui, ainsi que de la prolifération de l'intelligence artificielle (IA) en tant que service [5].

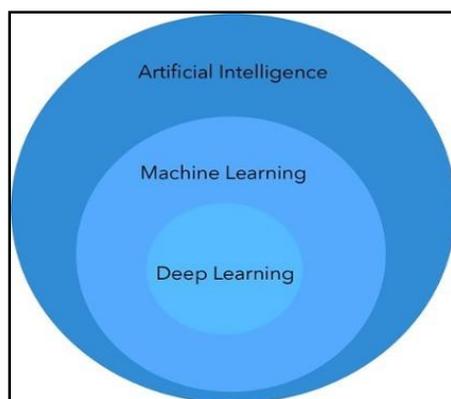


Figure 2. 8: Le deep learning

### 2.4.2. Les types de méthodes d'apprentissage

Dans l'apprentissage approfondi, les modèles prédictifs utilisent divers algorithmes fondamentaux pour déduire des relations mathématiques à partir des données de formation. Il existe principalement trois types de méthodes d'apprentissage, à savoir :

#### 2.4.2.1. Apprentissage supervisé

Dans l'apprentissage supervisé, le modèle est alimenté par un ensemble de données de formation contenant à la fois les observations (c'est-à-dire les entrées) et les résultats correspondants (c'est-à-dire les sorties). Le modèle déduit ensuite la transposition mathématique des entrées aux sorties, qu'il peut utiliser pour classer les futurs points de données de test d'entrée.

#### 2.4.2.2. Apprentissage non supervisé

Dans l'apprentissage non supervisé, le modèle est alimenté par des données de formation non classifiées (c'est-à-dire uniquement les entrées). Ensuite, le modèle classe les points de données de test dans différentes classes en trouvant des points communs entre elles.

#### 2.4.2.3. Apprentissage semi-supervisé

Comme son nom l'indique, l'apprentissage semi-supervisé hérite des propriétés de l'apprentissage supervisé et de l'apprentissage non supervisé.

Un ensemble de données semi-supervisées contient principalement des points de données de formation non classifiés ainsi que de petites quantités de données classifiées.

Les modèles semi-supervisés présentent deux avantages importants. Premièrement, ils sont sensiblement plus précis que les modèles non supervisés avec l'ajout de quelques points de données classifiées. Deuxièmement, ils sont nettement moins laborieux et demandent moins de temps que l'apprentissage supervisé.

L'apprentissage semi-supervisé peut se référer soit à l'apprentissage transductif, soit à l'apprentissage inductif.

### **2.4.3. Les réseaux de neurones**

Les réseaux de neurones sont des imitations simples des fonctions d'un neurone dans le cerveau humain pour résoudre des problématiques d'apprentissage de la apprentissage profond

Le neurone est une unité qui est exprimée généralement par une fonction sigmoïde.

#### **2.4.3.1. Réseaux neuronaux artificiels (ANN)**

Les réseaux neuronaux artificiels (ANN) cherchent à simuler ces réseaux et à faire en sorte que les ordinateurs agissent comme des cellules cérébrales interconnectées.

Différentes parties du cerveau humain sont responsables du traitement de différentes informations, et ces parties du cerveau sont disposées hiérarchiquement, ou en couches. Ainsi, lorsque l'information entre dans le cerveau, chaque niveau de neurones traite l'information, fournit des informations et les transmet à la couche suivante, plus élevée.[6] C'est cette approche en couches du traitement de l'information et de la prise de décision que les ANN tentent de simuler. Dans sa forme la plus simple, un ANN ne peut avoir que trois couches de neurones :

- la couche d'entrée :où les données entrent dans le système.
- la couche cachée :où les informations sont traitées.
- la couche de la sortie : où le système décide de ce qu'il doit faire en fonction des données.

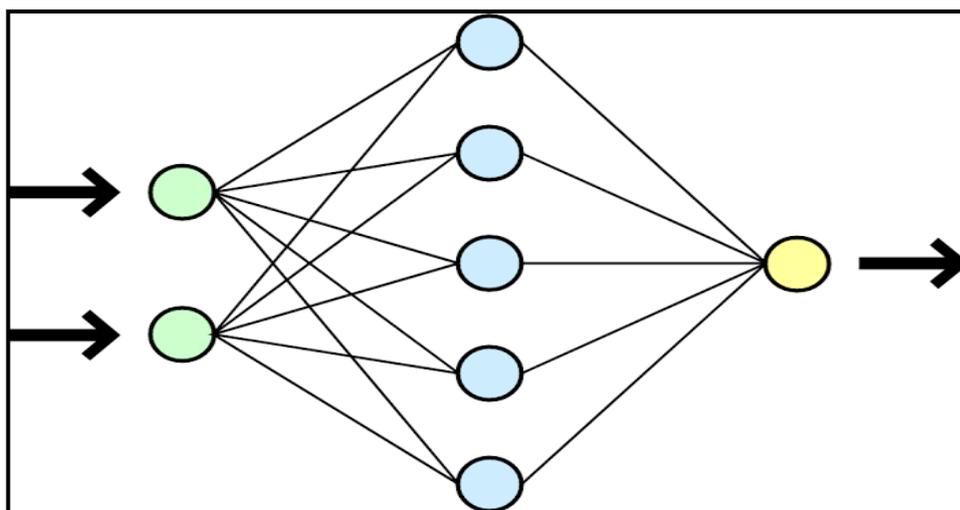


Figure 2. 9: architecture du réseau neurone ANN.

### 2.4.3.2. Réseaux neuronaux profonds DNN

Un réseau neuronal profond (DNN) est un réseau neuronal artificiel (ANN) avec plusieurs couches entre les couches d'entrée et de sortie, le DNN trouve la manipulation mathématique correcte pour transformer l'entrée en sortie, qu'il s'agisse d'une relation linéaire ou non linéaire. Le réseau se déplace à travers les couches en calculant la probabilité de chaque sortie [7].

L'utilisateur peut examiner les résultats et sélectionner les probabilités que le réseau doit afficher (au-dessus d'un certain seuil, etc.) et renvoyer l'étiquette proposée.

Chaque manipulation mathématique en tant que telle est considérée comme une couche, et les DNN complexes ont de nombreuses couches, d'où le nom de réseaux "profonds".

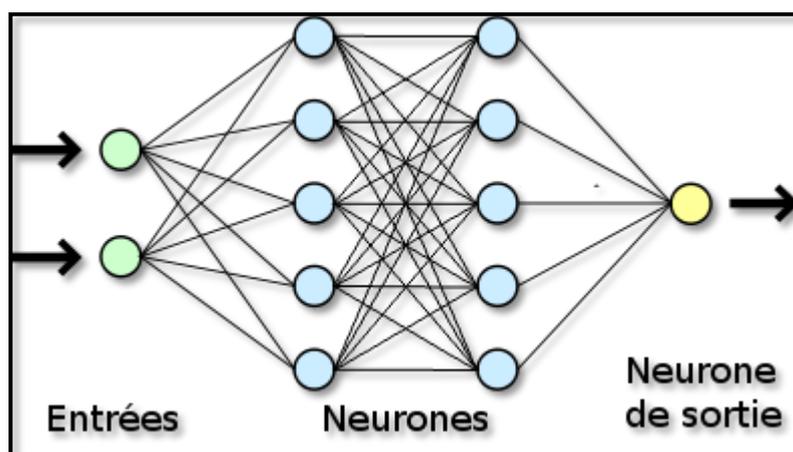


Figure 2. 10: architecture du réseau neuronal profond

### 2.4.4. Les différences architecturales d'apprentissage profond

#### II.4.4.1. Réseau neuronal entièrement connecté

Les réseaux neuronaux entièrement connectés (FCNN) sont un type de réseau neuronal artificiel dont l'architecture est telle que tous les nœuds, ou neurones, d'une couche sont connectés aux neurones de la couche suivante. qui permet de toute valeur d'entrée peut affecter toute valeur de sortie

#### 2.4.4.2. Réseaux de neurones convolutifs (CNN)

Les réseaux neuronaux convolutifs (CNN) sont l'une des variantes des réseaux neuronaux très utilisés dans le domaine de la vision par ordinateur, Il tire son nom du type de couches cachées dont il est constitué.

Les couches cachées d'un CNN sont généralement constituées de couches convolutionnelles, de couches de regroupement, de couches entièrement connectées et de couches de normalisation.

Ici, cela signifie simplement qu'au lieu d'utiliser les fonctions d'activation normales définies ci-dessus, les fonctions de convolution et de mise en commun sont utilisées comme fonctions d'activation.

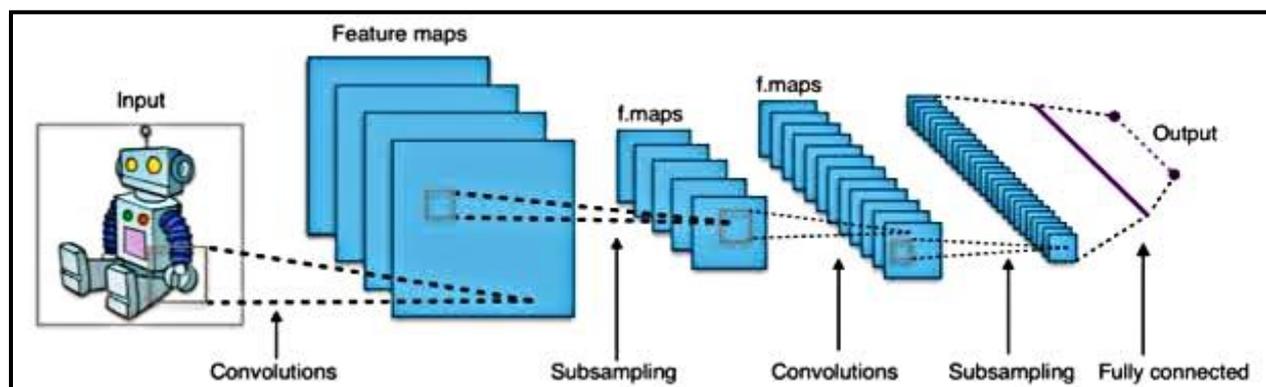


Figure 2. 11: Architecture d'un réseau de neurones convolutif [8].

- **Convolution**

La convolution fonctionne sur deux signaux (en 1D) ou deux images (en 2D) : vous pouvez considérer l'un comme le signal (ou l'image) "d'entrée", et l'autre (appelé le noyau) comme un "filtre" sur l'image d'entrée, produisant une image de sortie (la convolution prend donc deux images en entrée et en produit une troisième en sortie).

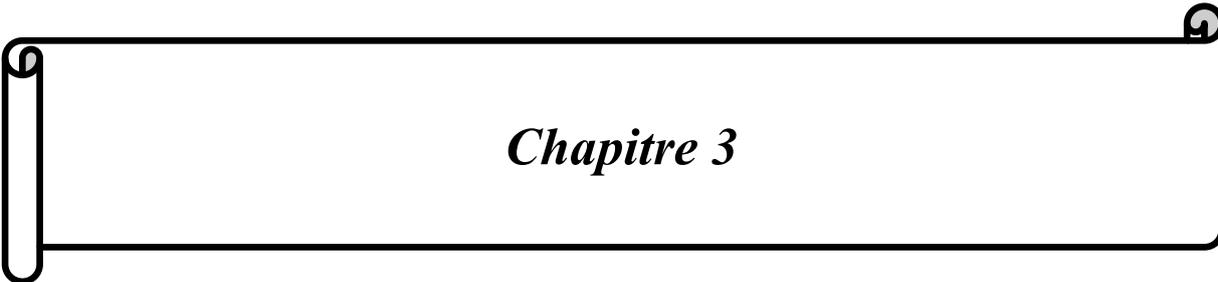
- **Pooling**

La mise en commun est un processus de discrétisation basé sur un échantillon. L'objectif est de sous-échantillonner une représentation d'entrée (image, matrice de sortie à couche cachée, etc.), en réduisant sa dimensionnalité et en permettant de faire des hypothèses sur les caractéristiques contenues dans les sous-régions mises en commun.

### 2.5. Conclusion

Dans ce chapitre, nous avons présenté les récents développements dans Machines d'apprentissage basé sur les Supports Vector Machine et deep learning. Ces deux derniers ont été choisis car elles sont les mieux adaptées en termes de reconnaissance.

Une étude comparative entre ces deux méthodes « SVM » et « deep learning » fera l'objet du chapitre suivant en présentant les résultats obtenus



*Chapitre 3*

### 3. Introduction

La phase d'implémentation de notre système d'authentification reprend la phase de conception décrite dans le chapitre précédent tout en leur donnant une forme expérimentale. Le système est basé sur le classificateur mono-classe OC-SVM mis en œuvre à l'aide de différents descripteurs dont les caractéristiques sont générées à partir d'images de signature. Son implémentation est réalisée dans l'environnement Matlab avec la bibliothèque LIBSVM.

Tout d'abord, nous présenterons la base de données et le protocole expérimental pour valider et tester notre système d'authentification. Nous aborderons ensuite l'évaluation des résultats obtenus à partir du classifieur OC-SVM associé aux un seul descripteur. Enfin, la dernière étape sera consacrée à expérimenter plusieurs combinaisons pour améliorer les performances par rapport à chaque classifieur mono-source pris seul, justifiant ainsi notre choix.

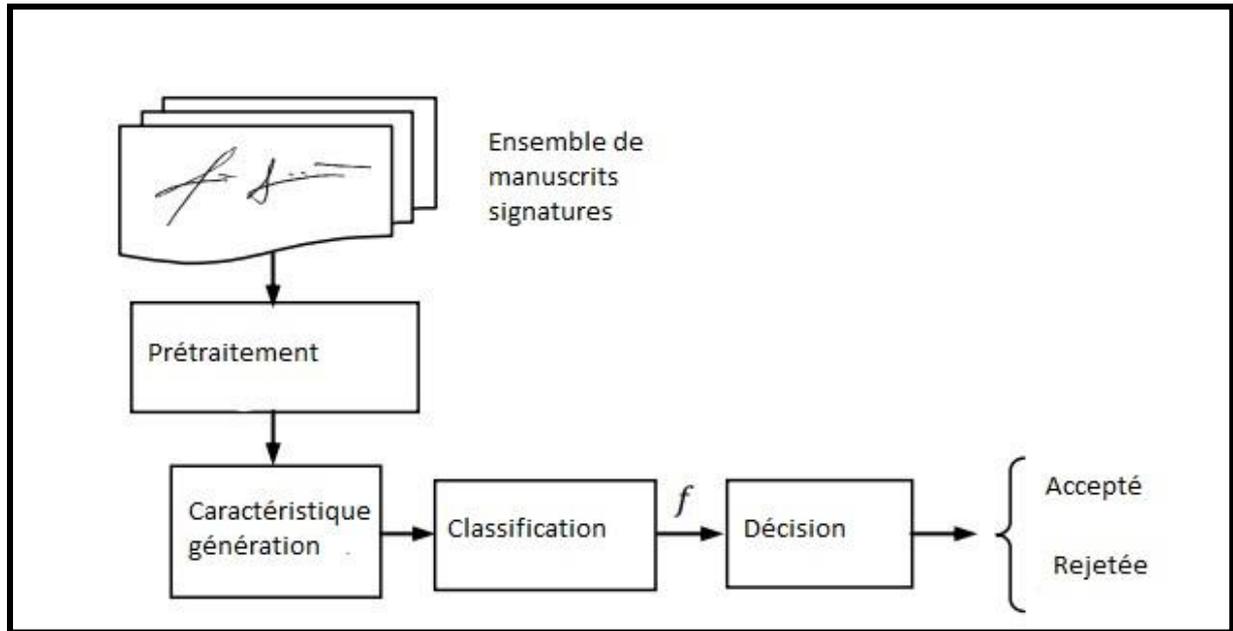
Aussi ce chapitre a été alloué pour présenter les résultats obtenus pour la validation du système vérification de signature. Nous faire la conception d'application en utilisant le langage Python. Dans ce chapitre, nous utilisons les réseaux des neurones Convolutif qui peut être appliqué efficacement à la vérification de signature utilisant l'apprentissage profond, en utilisant la base de données "CEDAR bd\_b"

#### 3. 1. OC- SVM Matlab

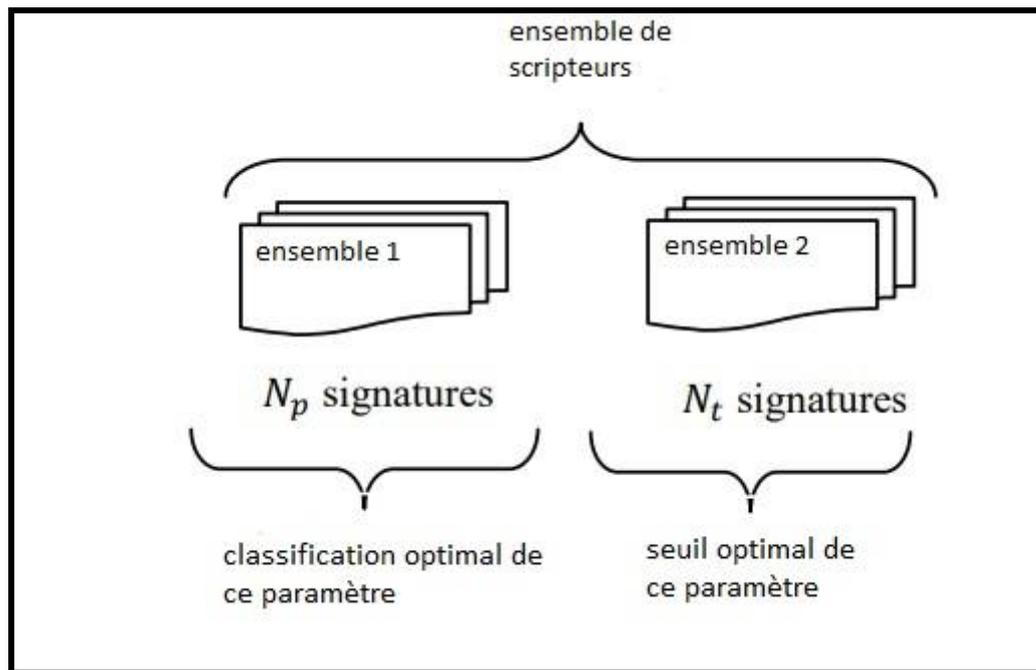
##### 3. 1. 1. Conception du HSVS individual

La conception proposée de chaque HSVS, comme le montre la figure 3.1, comporte quatre étapes: le prétraitement de chaque signature manuscrite, la génération de caractéristiques, la classification et la décision d'accepter ou de rejeter une signature. Au cours de la conception, deux approches sont adoptées pour sélectionner les paramètres du classier. La première approche est basée sur le HSVS dépendant de chaquescripteur, qui consiste à générer un modèle pour chaque auteur, chacun ayant ses propres paramètres. La seconde approche est basée sur le paramètre indépendant de chaque scripteur [30], qui consiste à concevoir un HSVS pour chaque scénariste mais tous les scénaristes ont les mêmes paramètres. Dans ce cas, la conception du système peut être effectuée en trois étapes: la sélection d'un ensemble descripteurs, la construction des modèles de signature et de trouver le seuil de décision optimale. Plus précisément, l'ensemble des auteurs est choisi au hasard à partir d'un ensemble de données ayant chacun N signatures authentiques ou falsifiées. Pour construire les modèles de signature, l'ensemble de signatures pour chaque auteur est divisé en deux sous-ensembles,

à savoir l'ensemble 1 et l'ensemble 2, respectivement. Le premier ensemble (ensemble 1) contenant les signatures authentiques de l'AN est utilisé pour trouver les paramètres de l'OC-SVM pendant l'étape d'instruction. Alors que le deuxième ensemble (ensemble 2) contenant des signatures authentiques ou falsifiées de NT est utilisé pour trouver le seuil de décision optimal. La figure 3.2 montre le concept de sélection des paramètres optimaux du HSVS.



**Figure3. 1:** L'architecture globale du HSVS individuel



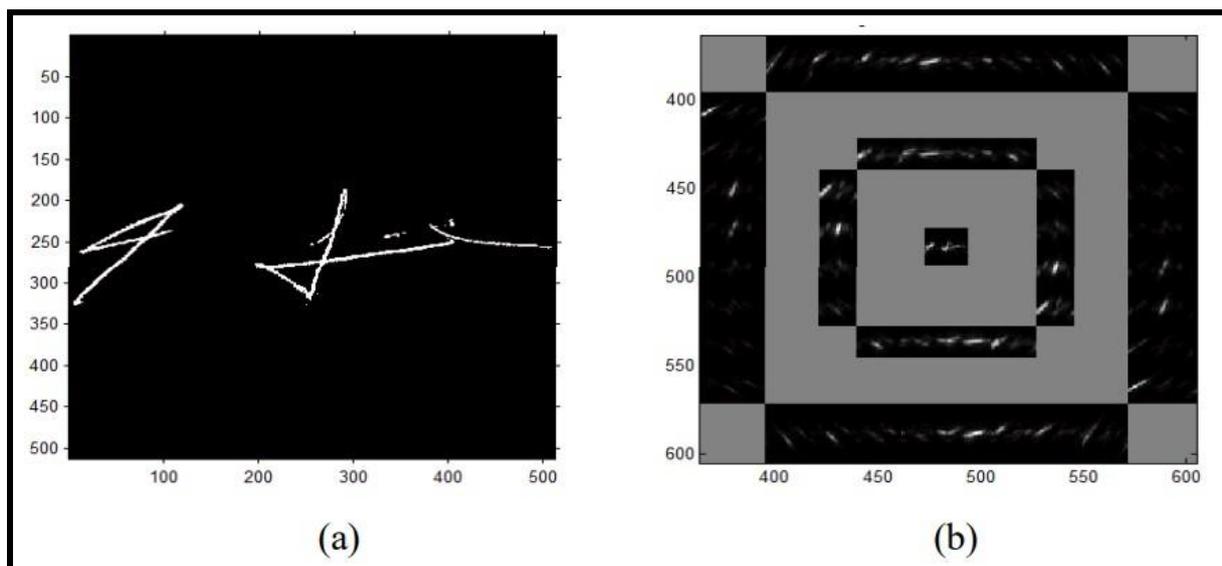
**Figure3. 2:** conception du HSVS

Dans la section suivante, nous décrivons d'abord la technique de génération des caractéristiques. Deuxièmement, nous développons la méthodologie adoptée pour ajuster le seuil de décision lors de l'utilisation du classificateur OC-SVM.

### 3. 1. 2.caractéristique génération

La transformation curvelet est une nouvelle transformation multi-échelle mise au point par Candès et Donoho [31]. Dans notre système, nous n'utilisons que l'énergie du coefficient de courbure calculé à partir de l'ensemble de l'image de signature manuscrite. Plus précisément, pour représenter les images de signature, nos descripteurs sont générés en calculant les coefficients curvelet via l'utilisation de la technique d'emballage à différentes échelles et différentes orientations pour chaque image. Par conséquent, pour une caractérisation efficace de l'image de signature, nous calculons l'énergie de coin indiquée par  $E$  de l'échelle  $l$  et l'orientation  $r$  de Curvelet transformer comme suit :

$$(l, r) = \sum \sum |Cl,r (i,j)| \quad (3.1)$$

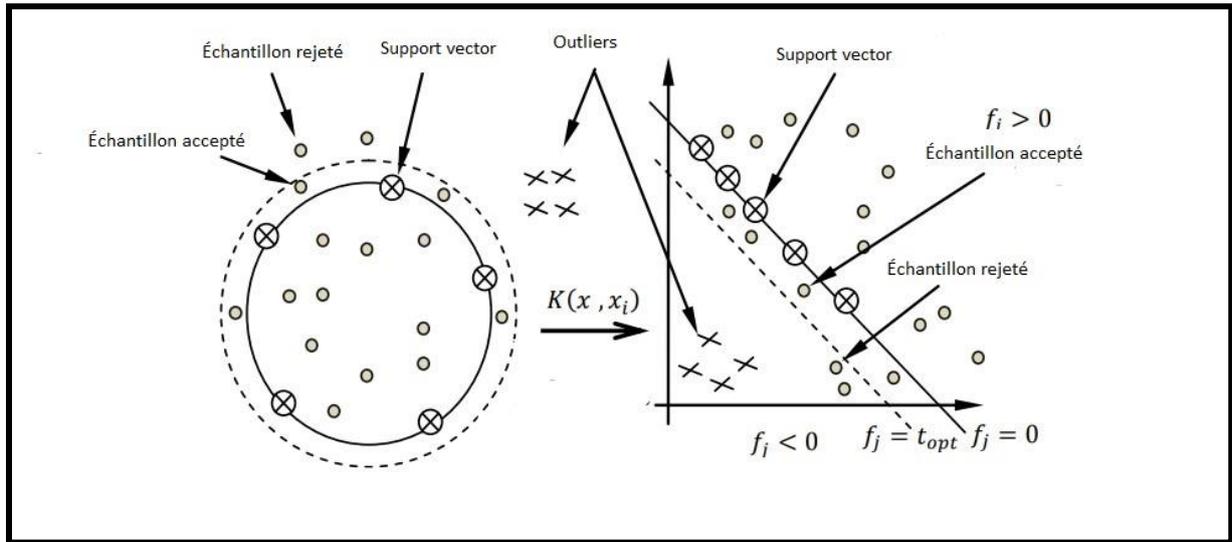


**Figure3. 3:**L'architecture globale du HSVS individuel

La figure 3.3 présente un exemple d'image de signature utilisée pour l'évaluation de l'œuvre proposée et de ses transformations Curvelet correspondantes. Malgré la représentation effective de la courbe la transformant sensible à la variation d'orientation. Par conséquent, nous calculons la moyenne de chaque quatre (4) coin adjacent qui conduit à améliorer la robustesse du descripteur utilisé. Le vecteur caractéristique est obtenu par la concaténation de toutes les énergies de calage calculées. Par conséquent, il est composé de dix composants.

### 3.1.2.1. classification et décision :

L'OC-SVM permet de fournir des performances précises lorsque de nombreux échantillons d'entraînement sont disponibles. Cependant, lors de la vérification de la signature manuscrite, peu d'échantillons sont souvent disponibles et, par conséquent, l'étape de formation du OC-SVM n'est pas bien exécutée, ce qui génère un rendement inexact. La figure 3.4 illustre un exemple d'échantillons rejetés par l'OC-SVM, qui se trouvent à proximité et à l'extérieur de l'hypersphère dans l'espace d'origine, et à proximité de l'hyper-plan dans l'espace caractéristique.



**Figure3. 4:** L'architecture globale du HSVM individuel

Théoriquement, un motif  $x$  est correctement classé et donc accepté lorsque la fonction de décision  $f(x)$  est positive. Implicitement, le seuil est fixé à zéro. Cette approche peut être considérée comme un seuil ferme. En effet, certains échantillons de signature proches de l'hyper plan dans l'espace de fonctionnalité ne sont pas acceptés et donc rejetés.

Afin d'assouplir cette contrainte, nous proposons un seuil doux pour réduire le défaut de classification. Par conséquent, nous adoptons la règle de décision suivante :

$$x \in \begin{cases} \text{accepté} & \text{if } f(x) \geq t \\ \text{rejeté} & \text{ailleurs} \end{cases} \quad (3.2)$$

$f(x)$  est la fonction de décision choisie lors de l'utilisation du vote majoritaire et définit le seuil calculé selon l'équation suivante [Guerbai et al, 2012a., Guerbai et al, 2012b] :

$$t = m_f + k\sigma_f \quad (3.3)$$

$m_f$  et  $\sigma_f$  sont la moyenne et l'écart-type respectifs calculés à partir de toutes les valeurs de la fonction de décision des cinq distances pendant l'étape de formation inspirée de la distribution gaussienne qui tient compte de la moyenne et de l'écart-type. Le paramètre  $k$  permet de contrôler le seuil de décision . Le seuil de décision optimal ( $t_{opt}$ ) est déduit des courbes FRR

et FAR en utilisant le taux d'erreur total de moitié (HTER) tel que défini dans (Bengio et Mariéthoz, 2004) :

$$HTER = \frac{FRR+FAR}{2} \quad (3.4)$$

Telque :

- Le FRR est le taux de faux rejet, qui correspond aux signatures authentiques rejetées par le système.
- FAR est le taux de fausse acceptation, qui correspond aux signatures falsifiées ou fictives, acceptées par le système.

### 3.1.2.2. Mesures de distance du noyau OC-SVM

La formation de l'OC-SVM nécessite de définir la distance appropriée utilisée dans le noyau RBF. Habituellement, la distance euclidienne est la métrique la plus utilisée. Cependant, cette métrique ne fournit pas des résultats satisfaisants dans tous les cas. Considérant  $f(x)$  comme la fonction de décision d'un OC-SVM ayant sa propre distance :

$$f(x) = \text{sgn} \left\{ \sum_{i=1}^m a_i k(x, x_i) - \rho \right\} \quad (3.5)$$

De sorte  $k(x, x_i) = \exp(-\gamma d(x, x_i))$ ,  $a_i$  sont les multiplicateurs Lagrange calculés pour chaque distance. Il est donc possible de sélectionner différentes distances, qui sont définies dans la littérature [32]. Dans notre cas, nous utilisons cinq mesures de distance comme indiqué dans le tableau 3.1 [33].

mesure de distance $d(x, x_i)$
Euclidien $\sqrt{\sum_{k=1}^d (x_i - x_{i,k})^2}$

Cityblock	$\sum_{k=1}^d  x_i - x_{i,k} $
Chebychev	$\max_{k=1,\dots,d}  x_i - x_{i,k} $
Corrélation	$\frac{cov((x, x_i))}{\sigma_x \sigma_{x_i}}$
Spearman	$\frac{1 - 6 \sum_{i=1}^d (\text{rank}(x) - \text{rank}(x_i))^2}{n(n^2 - 1)}$

**Tableau 3. 1:** Mesures de distance appliquées au noyau du classificateur OC-SVM.

### 3.1.3. Conception du system HSVS combine ( Handwritten Signature Verification system)

#### 3.1.3.1. déclaration de problème

En général, la conception de chaque système ne donne pas des résultats satisfaisants dans tous les cas. Par conséquent, il est simple et intéressant de combiner plusieurs systèmes individuels afin d'améliorer la robustesse d'une application particulière. Dans notre cas, nous proposons de combiner plusieurs OC-SVM chacun ayant sa propre distance afin d'améliorer le processus de vérification.

Plusieurs règles de combinaison sont possibles pour atteindre la performance du système, comme la moyenne, min, max, produit, la pondération statique, la pondération dynamique et le vote majoritaire [34].

Dans notre système, nous utilisons le vote majoritaire comme règle de combinaison pour sa mise en œuvre simple et il permet également de fournir de bons résultats par rapport à d'autres règles de combinaison.

#### 3.1.3.2. Description des paramètres HSVS combinés indépendants de scripteur

La conception des paramètres HSVS indépendants de l'auteur consiste à générer un modèle pour chaque scripteur mais tous les écrivains ont les mêmes paramètres

Le HSVS combiné proposé, comme le montre la figure 3.5, se compose de quatre étapes : prétraitement de chaque signature manuscrite, génération de caractéristiques à l'aide de la transformation curvelet, classification à l'aide de plusieurs classificateurs OC-SVM ( $L$  définit le nombre de classificateurs OC-SVM)., une règle combinée et une règle de décision.

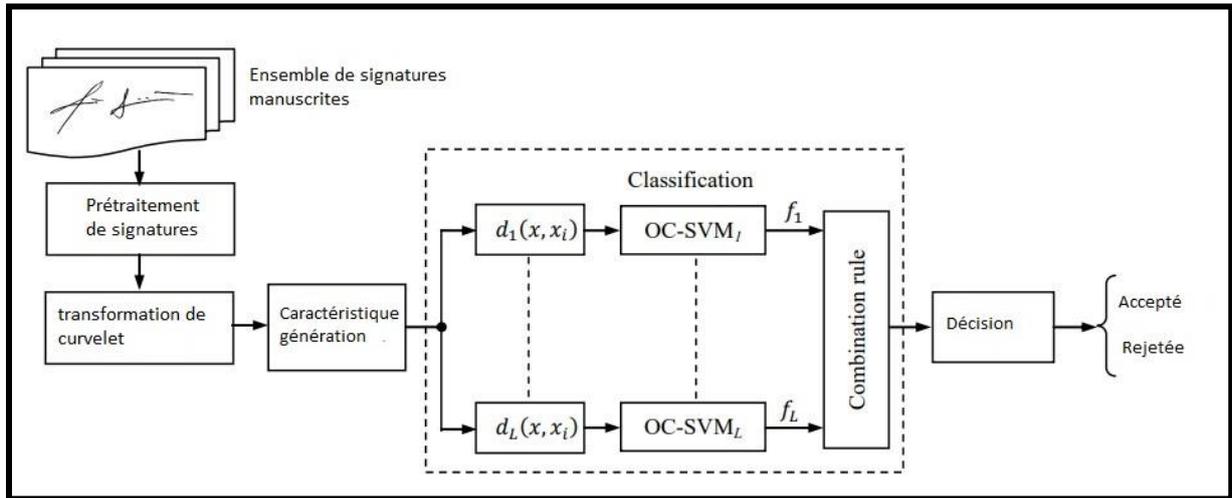


Figure3. 5: L'architecture globale de l'auteur combiné indépendant HSVS

### 3.1.3.3. Réglage du seuil

Pour l'ajustement du seuil comme nous l'avons présenté pour le système individuel. Pour le système combiné, un motif  $x$  est accepté ou rejeté selon la règle de décision suivante :

$$x \in \begin{cases} accepte , & f(x) \geq t \\ rejete , & \text{ailleurs} \end{cases} \quad (3.6)$$

La fonction décisionnelle d'un classificateur OC – SVM  $j$  est rédigée comme suit :

$$f_j(x) = sgn \left\{ \sum_j^m a_{ij} k_j(x, x_i) - \rho_j \right\} \quad (3.7)$$

telle que  $k_j(x, x_i) = \exp(-\gamma d_j(x, x_i))$ ,  $j=1, \dots, L$  où  $L$  est le nombre de classificateurs OC-SVM.  $a_{ij}$  sont les multiplicateurs Lagrange calculés pour chaque distance.

$f_j(x)$  est la fonction de décision sélectionnée lors de l'utilisation du vote majoritaire et  $t$  définit le  $threshold$  calculé en utilisant toutes les fonctions de décision selon l'équation 3.3.

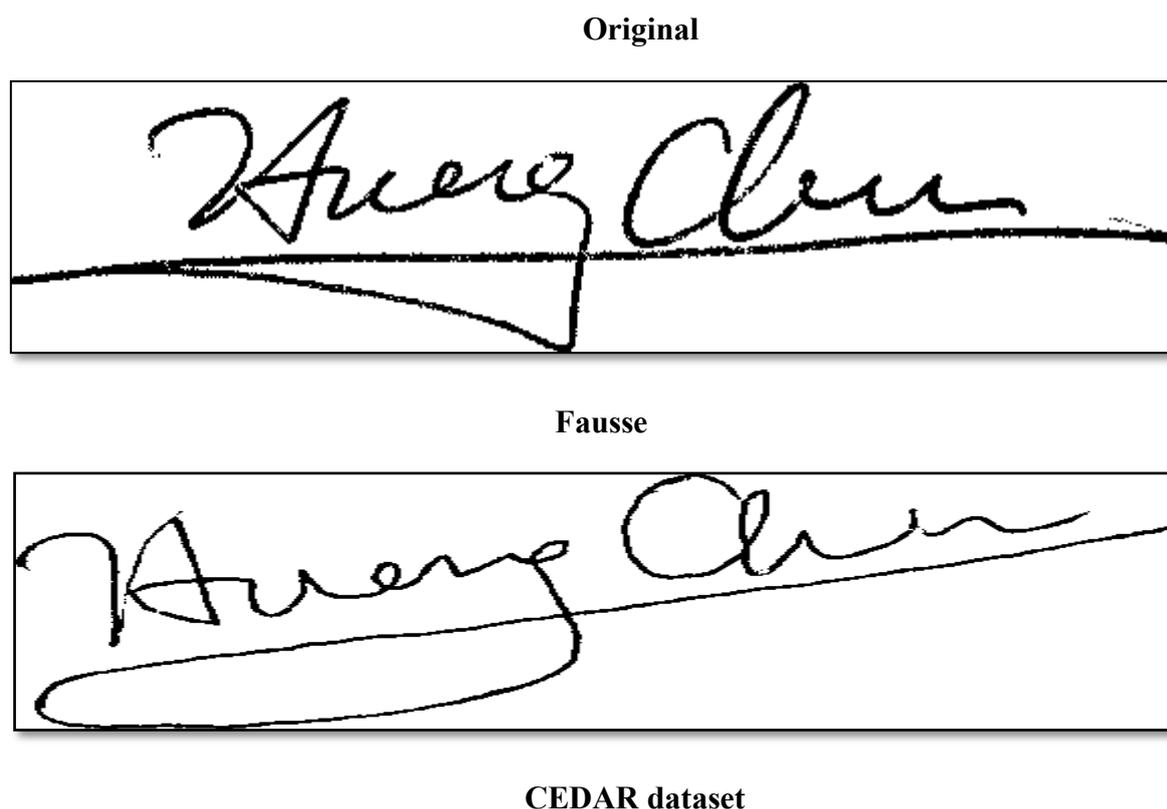
### 3.1.4. résultats expérimentaux

#### 3.1.4.1. Description des ensembles de données et critères d'évaluation

La signature du Centre d'excellence pour l'analyse et la reconnaissance de documents (CEDAR) L'ensemble de données est un ensemble de données couramment utilisé pour la vérification de signature hors ligne. La base de données CEDAR est composé de 55 rédacteurs ; chacun a 24 authentiques et 24 faux signatures. Les signatures des différents rédacteurs sont scannées à 300 dpi. Nous utilisons uniquement le jeu de données CEDAR dans le but d'évaluer la performance du

paramètres indépendants de l'écrivain systèmes de vérification de signature manuscrite.

La figure 3.6 montre quelques exemples de signatures, la première ligne représente la signature authentique, la deuxième ligne représente la signature fausses [35].



**Figure3. 6:** Quelques échantillons de signatures de utilisateur num “42”.

Afin d'évaluer le rendement de plusieurs HSVS proposés, deux mesures sont généralement utilisées, soit la moitié du taux d'erreur total (HTER) et le taux d'erreur moyen (AER), respectivement. Le HTER est défini comme suit [36]:

$$HTER = \frac{FAR+FRR}{2} \quad (3.8)$$

De sorte que le FAR est le taux de fausse acceptation, qui permet de prendre en compte uniquement les contrefaçons qualifiées; tandis que le FRR est le taux de faux rejet (FRR), qui permet de prendre en compte uniquement les signatures authentiques.

$$AER = \frac{\text{nombre de vraies signatures rejetées} + \text{nombre de fausses signatures acceptées}}{\text{nombre total de signature test}} \quad (3.9)$$

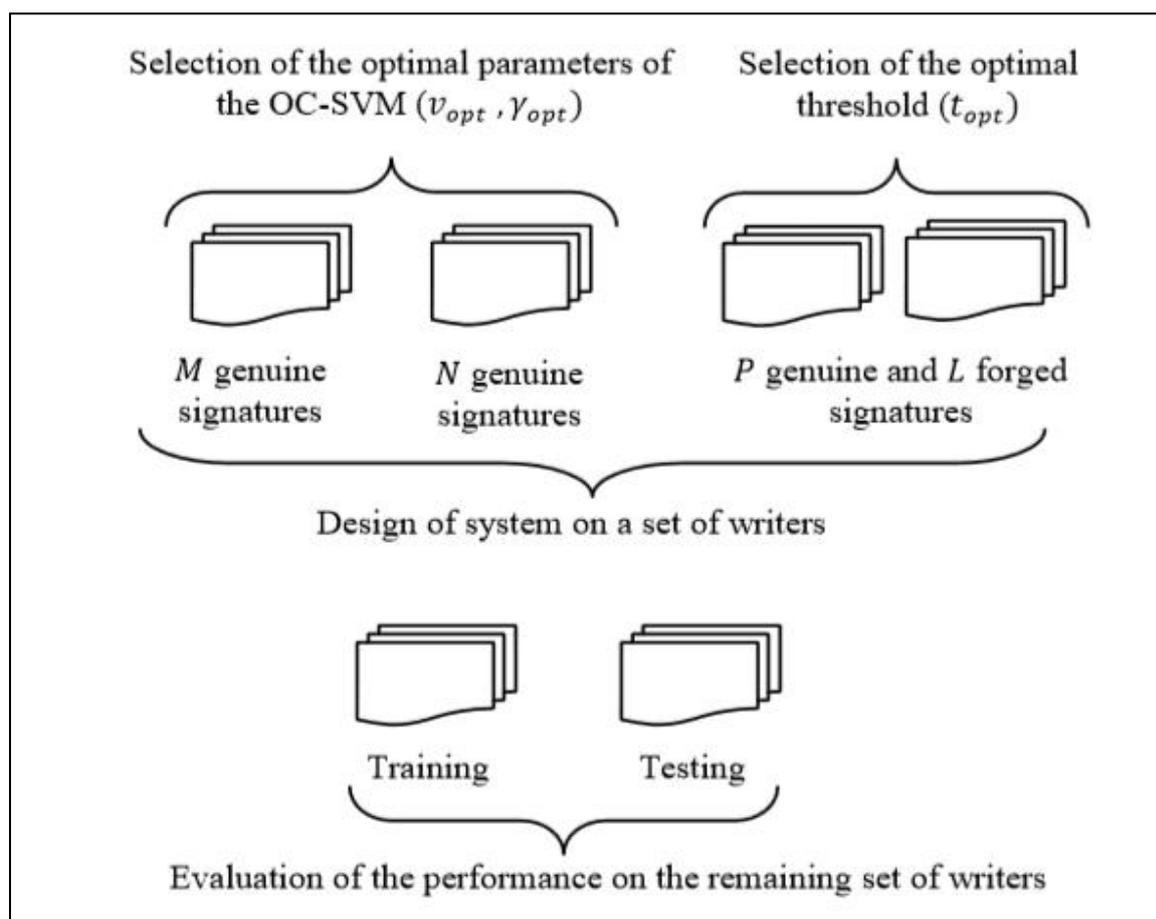
une méthode peut être considérée comme précise lorsque l'HTER ou l'AER sont aussi petits que possible.

Les expériences se déroulent en trois étapes. Tout d'abord, nous évaluons le rendement de chaque HSVS lorsque nous utilisons le classificateur OC-SVM, nous adoptons l'approche dépendante de l'auteur pour laquelle nous supposons que chaque utilisateur a ses propres paramètres. Par la suite, nous évaluons la performance de l'OC-SVM selon les paramètres indépendants de l'utilisateur en utilisant des signatures authentiques et forgées pour la conception de l'HSVS. Enfin, nous évaluons les performances des systèmes individuels et combinés en utilisant des signatures authentiques et des signatures fictives, qui sont authentiques appartenant à d'autres scripteurs.

### 3.1.4.2. Evaluation of the individual writer independent HSVS based on genuine and forged signatures.

#### 3.1.4.2.1. Protocole experimental

Lors de la conception du système, comme illustré à la figure 3.7, des signatures  $K$  authentiques et  $L$  falsifiées sont disponibles pour chaque auteur. Les signatures authentiques sont subdivisées au hasard en trois sous-ensembles, à savoir les signatures authentiques  $M$ ,  $N$  et  $P$ . Les signatures  $M$  et  $N$  sont utilisées pour former et valider le modèle de l'OC-SVM tandis que les signatures  $P$  authentiques et  $L$  contrefaites sont utilisées pour trouver le seuil optimal  $topt$  à partir des courbes FAR et FRR.

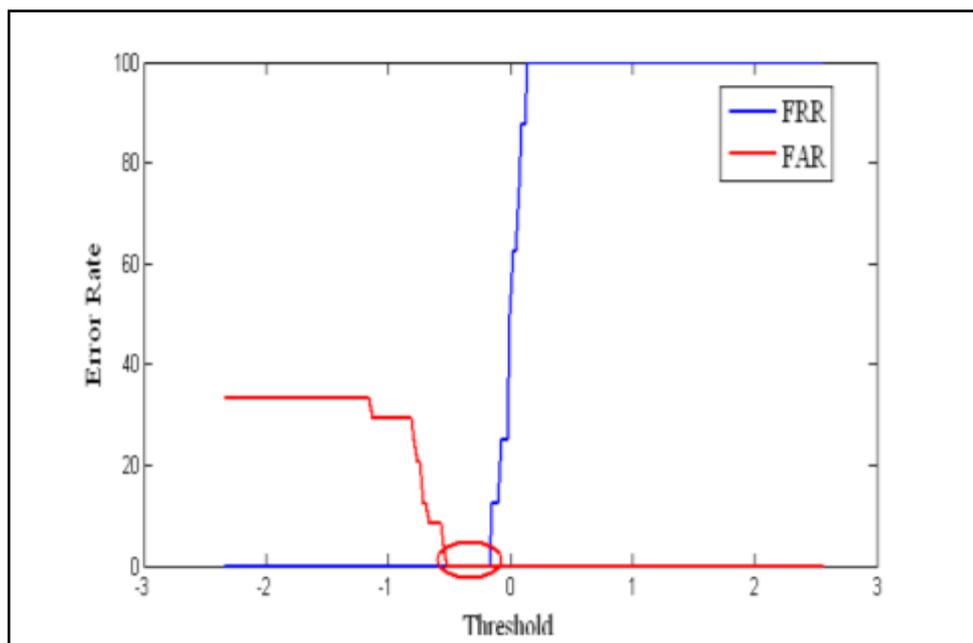


**Figure3. 7:** Conception du rédacteur indépendant basée sur des signatures authentiques et falsifiées .

#### 3.1.4.2.2. Trouver le seuil optimal

Pour chaque rédacteur, nous considérons 8 signatures authentiques pour entraîner l'OC-SVM, 8 signatures authentiques pour la validation et 8 signatures authentiques et toutes falsifiées pour trouver le seuil optimal. Le seuil optimal ( $t_{opt}$ ) est sélectionné à partir des courbes FAR et FRR par rapport aux courbes de seuil qui sont définies comme le taux d'erreur demi-total.

Le figure 3.8 montrent les courbes FAR et FRR pour la base de données CEDAR dans le but de sélectionner le seuil optimal.

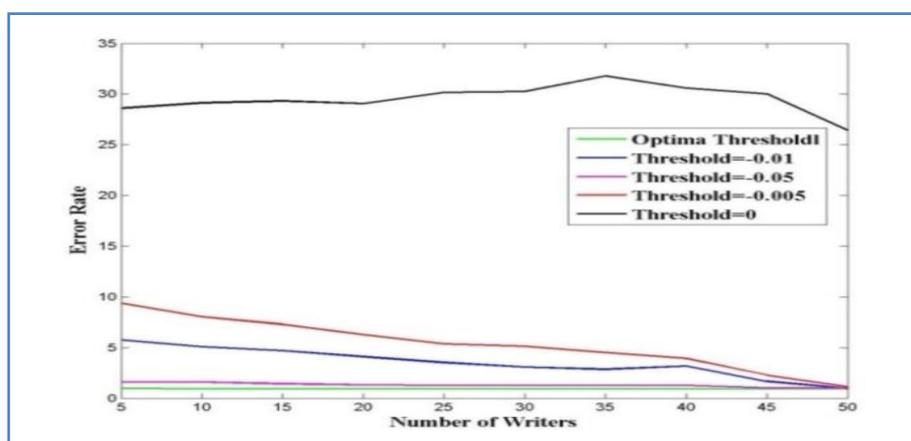


**Figure3. 8:** Courbes FAR et FRR pour la base de données CEDAR.

### 3.1.4.2.3. Influence du numéro de l'écrivain

Dans cette partie, nous étudions l'influence du nombre d'écrivains sur les performances du HSVS. Ainsi, l'étude expérimentale est menée comme suit pour chaque jeu de données :

La base de données CEDAR : Initialement, nous considérons 5 rédacteurs pour générer le modèle et le reste de l'ensemble de données est utilisé pour évaluer les performances de HSVS. Ensuite, pour chaque exécution, nous ajoutons 5 écrivains et nous répétons l'évaluation pour le reste des écrivains. Le figure 3.9 montrent l'influence du numéro du rédacteur sur le taux d'erreur pour la base de données CEDAR.



**Figure3. 9:** Taux d'erreur en fonction du nombre d'auteurs en utilisant la base de données CEDAR.

On constate clairement que le taux d'erreur le plus bas est obtenu par le seuil optimal qui représente le HTER le plus bas retenu dans la construction du modèle. De plus, il est indépendant du nombre d'écrivains. D'autre part, nous voyons que les valeurs de seuil de reste dépendent fortement du nombre d'écrivains et prennent leur erreur la plus faible avec le plus grand nombre d'écrivains, sauf le seuil dur, qui présente le taux d'erreur le plus élevé quel que soit le nombre d'écrivains utilisés pour générer maquette.

#### 3.1.4.2.4. Résultats pour un nombre fixe d'écrivains

La deuxième partie consiste à sélectionner un nombre fixe d'écrivains pour générer le modèle et stocker le paramètre, avec les mêmes paramètres, nous évaluons les performances sur d'autres écrivains afin d'obtenir les performances les plus faibles du classifieur. La base de données signature CEDAR est composé de 25 rédacteurs sélectionnés au hasard pour générer le modèle et de 30 rédacteurs pour évaluer les performances du système.

Afin d'apprécier l'efficacité d'utilisation de la méthode proposée avec un nombre fixe d'écrivains, différents seuils sont sélectionnés pour être comparés ; qui sont le seuil dur et le seuil mou. Le seuil dur est calculé lorsque la fonction de décision est égale à 0, tandis que le seuil souple est sélectionné à partir des courbes FAR et FRR.

Threshold	FRR (%)	FAR (%)	AER (%)
<b>0.000 (Hard threshold)</b>	32.50	30.27	31.39
<b>-0.005</b>	9.55	1.14	5.34
<b>-0.010</b>	5.68	1.14	3.41
<b>-0.050</b>	0.68	1.82	1.25
<b>- 0.174</b>	<b>0.00</b>	<b>1.82</b>	<b>0.91</b>

**Tableau 3. 2:** Performances de reconnaissance obtenues pour les différents seuils de rédacteur indépendant à l'aide du base de données CEDAR.

On constate clairement que la meilleure performance est atteinte par le seuil calculé à partir de la courbe ROC (AER = 0,909% pour un seuil égal à - 0,174 et pour les autres bornes : - 0,05 et 0,01 pour la base de données CEDAR, l'AER = 0,173 lorsque le seuil est - 3,232. En

revanche, un seuil solide entraîne un rapport plus élevé car le seuil est de 0, l'AER de 31,385 pour l'ensemble de données CEDAR.

### 3.1.4.2.5. Influence du numéro de signature

Pour étudier l'influence des signatures sélectionnées utilisées pour la formation de l'OC-SVM, nous sélectionnons des échantillons de 4 à 7 pour générer le modèle pour chaque écrivain.

Les résultats sont rapportés dans le tableau 3.3 Différents seuils sont sélectionnés pour comparaison, qui sont les seuils durs et souples. On peut noter que le seuillage doux est plus adéquat comparativement au seuillage dur puisque le taux d'erreur diminue lorsque le seuil diminue.

De plus, on peut noter que le nombre de signatures utilisées pour la formation influence le choix du seuil. En effet, lorsque le seuil est fixé à 0, l'AER est plus élevé et lorsque le seuil est relâché de 0 au seuil optimal, l'AER diminue significativement. De plus, la sélection du seuil optimal permet d'atteindre le même AER quel que soit le nombre d'échantillons de signature utilisés pour générer le modèle de signature.

Threshold	4 signatures			5 signatures			6 signatures			7 signatures		
	FRR	FAR	AER	FRR	FAR	AER	FRR	FAR	AER	FRR	FAR	AER
0.000	75.79	8.93	42.36	71.83	0.06	35.90	67.79	12.00	39.89	63.25	13.68	38.46
-0.005	23.97	0.03	12.00	24.36	0.03	12.20	22.47	0.03	11.25	22.53	0.34	11.28
-0.010	16.22	0.05	8.14	17.81	0.04	8.92	16.66	0.03	8.35	17.11	0.34	8.75
-0.050	2.04	0.14	1.09	5.22	0.10	2.66	3.09	0.08	1.58	3.68	0.08	1.88
<b>Optimal</b>	0.00	0.34	0.17	0.00	0.34	0.17	0.00	0.34	0.17	0.00	0.34	0.17
<b>Threshlod</b>	$t_{opt} = -3.12$			$t_{opt} = -3.08$			$t_{opt} = -3.32$			$t_{opt} = -3.70$		

**Tableau 3. 3:** Performances de reconnaissance obtenues pour les différents seuils sur le jeu de données CEDAR.

### 3.1.4.3. Évaluation du HSVS indépendant de l'auteur individuel et combiné sur la base de signatures authentiques et fictives.

#### 3.1.4.3.1. protocole experimental

Afin d'évaluer l'utilisation efficace de l'OC-SVM dans le HSVS indépendant de l'écrivain, nous considérons un ensemble d'écrivains pour les systèmes de formation utilisant uniquement des signatures authentiques afin de trouver le paramètre optimal du classificateur OC-SVM et le seuil optimal. Cela suppose que chaque écrivain a son propre modèle de classificateur. Alors que toutes les signatures authentiques et falsifiées des écrivains restants

non inscrits au système sont utilisées pour évaluer la robustesse du système proposé, les mêmes paramètres de l'OC-SVM sont utilisés sans trouver le seuil optimal.

Pour construire les modèles de signature, l'ensemble des signatures authentiques de chaque écrivain est divisé en deux sous-ensembles : Le premier sous-ensemble ( $N_p$ ) est utilisé pour entraîner le classificateur OC-SVM afin de trouver ses paramètres qui sont le pourcentage de valeurs aberrantes ( $\nu_{opt}$ ) et le paramètre du noyau ( $\gamma_{opt}$ ). Le deuxième sous-ensemble ( $N_t$ ) est utilisé pour trouver le seuil optimal. Le tableau 3.4 présente le protocole d'évaluation dans lequel, écrivains (Writers),  $N_g$  et  $N_f$  définissent respectivement le nombre d'auteurs, de signatures authentiques et falsifiées pour évaluer les performances du système proposé.

Dataset	Design step			Evaluation step		
	#Writers	$N_p$	$N_t$	#Writers	$N_g$	$N_f$
CEDAR	30	8	16	25	24	24

**Tableau 3. 4:** Protocole d'évaluation.

### 3.1.4.3.2. Sélection du seuil de décision optimal

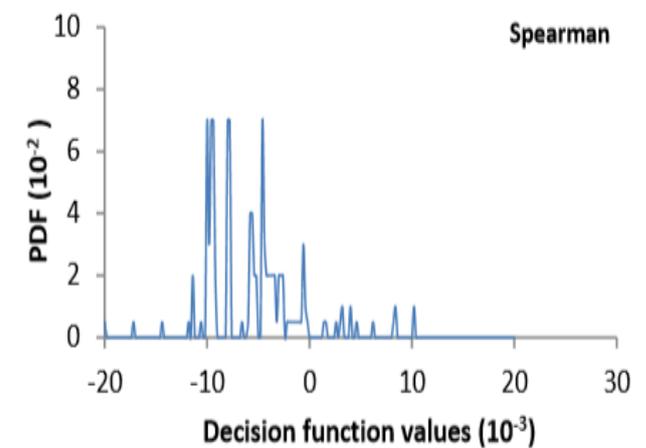
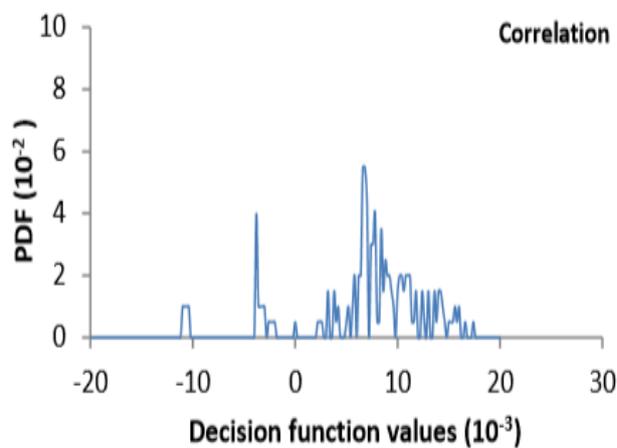
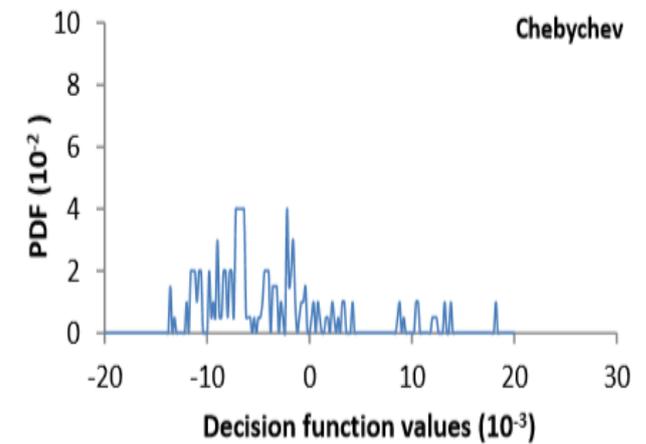
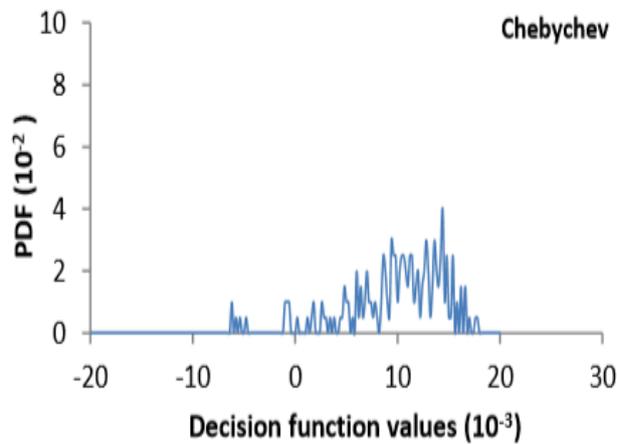
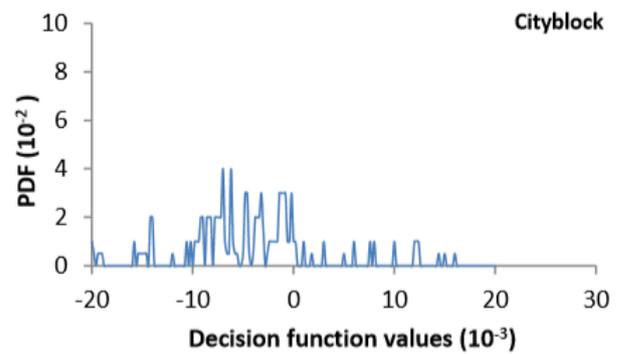
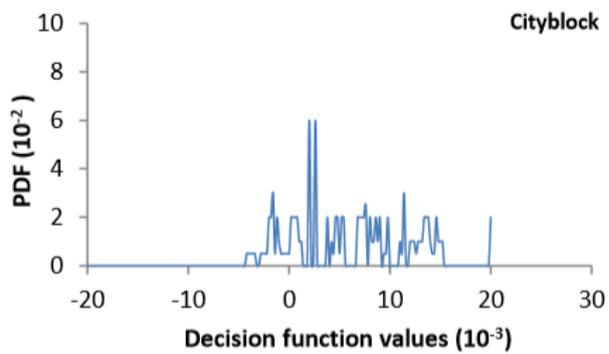
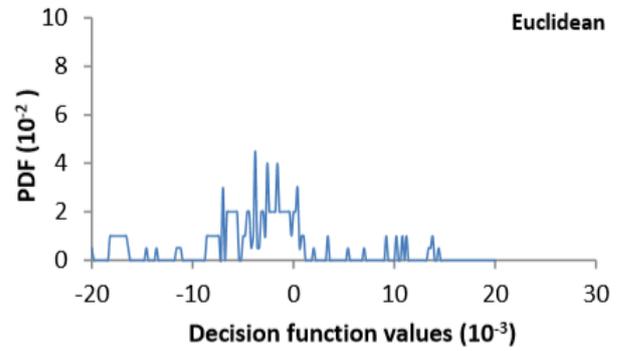
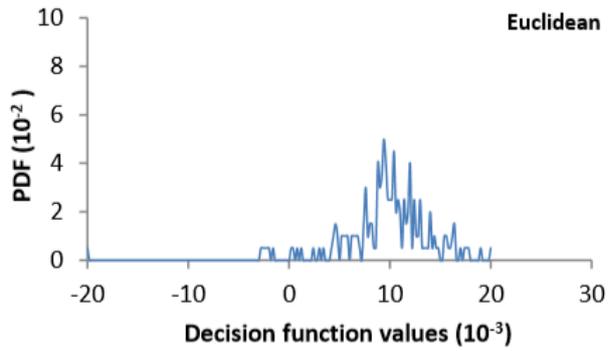
La sélection du seuil de décision optimal est une étape cruciale pour la conception d'un HSVS robuste. Dans notre cas, il est sélectionné parmi les courbes FAR et FRR utilisant des signatures authentiques ( $N_p$ ) et fictives ( $N_t$ ). Afin de montrer les réponses complémentaires des différents classificateurs, **les figures 3.10** représentent la fonction de densité de probabilité (PDF) des valeurs de décision (sans seuil) qui sont générées à différentes distances dans les noyaux du classificateur OC-SVM pour les signatures authentiques et fictives menées sur base de données CEDAR.

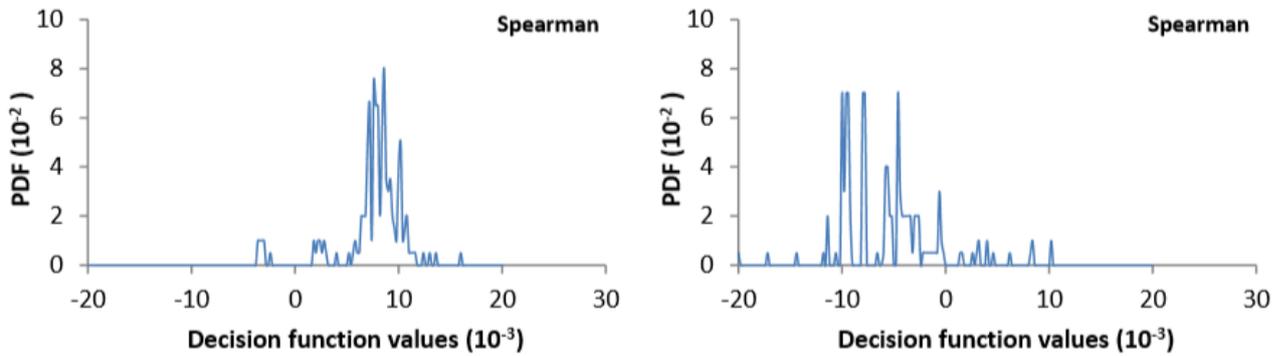
Comme nous pouvons le voir, les PDF montrent des complémentarités entre les signatures authentiques et fictives fournies par chaque classifieur pour l'ensemble de données. Cela justifie l'intérêt d'une sélection rigoureuse du seuil optimal pour accepter une signature authentique ou rejeter une contrefaçon. De plus, nous pouvons également noter qu'il existe des complémentarités entre les classificateurs ayant leur propre distance utilisée dans le noyau RBF (Euclidean, Manhattan, Chebychev, Correlation et Spearman) pour les signatures authentiques et fictives. Cela prouve la combinaison nécessaire de plusieurs OC-SVM. Dans

notre cas, nous utilisons le vote majoritaire comme règle de combinaison, ce qui permet d'améliorer les performances des distances à noyau unique.

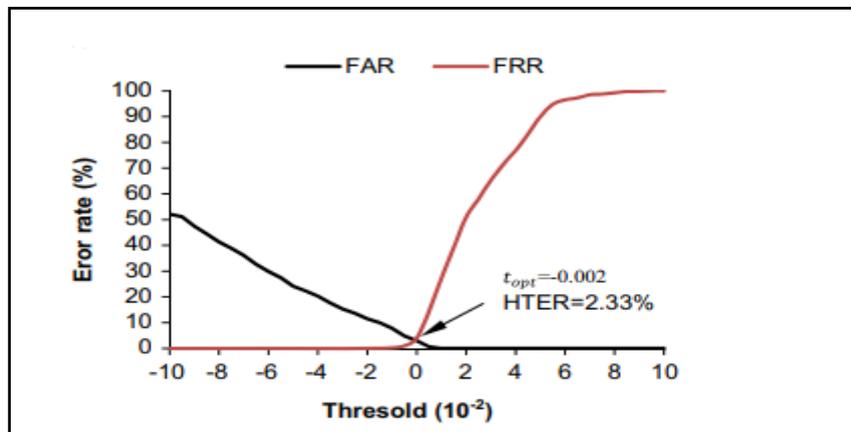
Pour trouver le seuil optimal, nous utilisons toutes les valeurs de fonction de décision fournies par différents OCSVM pour calculer respectivement la moyenne et l'écart type. Pour ajuster le seuil, les valeurs de  $k$  sont variées dans la plage  $[-3,+3]$ .

La valeur de  $k$  est choisie et par conséquent le seuil optimal ( $topt$ ) entre FAR et FRR lorsque le HTER est égal à  $(FAR+FRR)/2$ . La figure 3.18 montre les courbes FAR et FRR par rapport au seuil pour la base de données CEDAR. On peut observer que le seuil est de  $-0,002$  pour le jeu de données CEDAR lorsque le HTER est égal à  $2,33\%$ . De plus, il est intéressant de noter que les seuils sont proches de zéro mais ont des valeurs négatives pour l'ensemble de données.





**Figure3. 10:** PDF des signatures authentiques (figure de gauche) et fictives (figure de droite) pour différentes distances utilisées dans le noyau RBF de l'OC-SVM pour la base de données CEDAR



**Figure3. 11:** Courbes FAR et FRR pour sélectionner le seuil optimal : base de données CEDAR

**3.1.4.4. Performance evaluation**

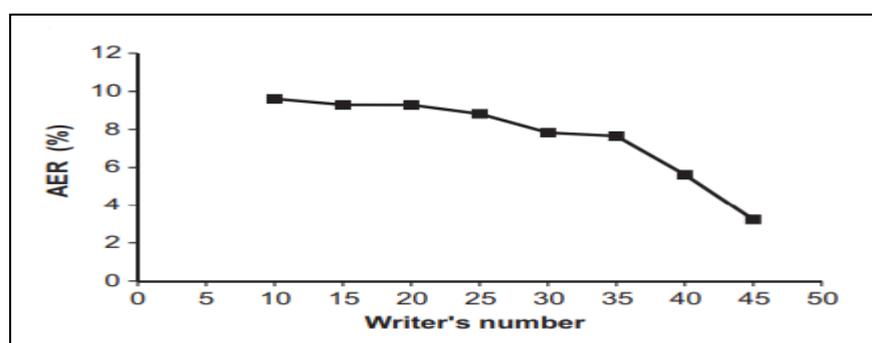
Le tableaux 3.5 rapportent les taux d'erreur en termes de FRR, FAR et AER fourni par différentes distances utilisées dans le noyau OC-SVM et leur combinaison pour la base de données CEDAR , pour les seuils souples  $t \approx t_{opt}$  et durs ( $t \approx 0$ ). Un résultat supplémentaire est également rapporté pour une valeur positive du seuil ( $t > 0$ ) déduit de la Fig. 3.11 lorsque FAR est égal à zéro. Cette valeur est sélectionnée dans afin de rejeter le maximum de signatures falsifiées. L'obtenu les résultats montrent différentes valeurs d'AER fournies par chaque classificateur pour ensembles de données en fonction du seuil sélectionné. Pour la base de données CEDAR, les ARE vont de 8,50% à 11,75%, En revanche, lorsque la valeur du seuil est dure, Les AER varient de 21,04 % à 32 % pour base de données CEDAR, . Pour un seuil positif, Les AER varient de 27,79 % à 40,71 % pour CEDAR, . Lors de la

combinaison de différents classificateurs à travers la majorité vote, les AER sont réduites quel que soit le seuil retenu pour les deux ensembles de données. En effet, lors de la sélection de seuils souples, durs et positifs, Les AER sont respectivement de 7,83 %, 15,83 % et 29,54 pour l'ensemble de données CEDAR. Cela prouve la nécessité de combiner les classificateurs afin de obtenir de meilleures performances. De plus, on peut noter que le seuil est très sensible et il doit être ajusté avec soin en fonction des exigences de le HSVS.

Distance	$t = t_{opt} = -0.002$			$t = 0$			$t = 0.01$		
	FRR	FAR	AER	FRR	FAR	AER	FRR	FAR	AER
Euclidean	5.33	14.00	9.66	36.50	27.50	32.00	77.92	3.50	40.71
Cityblock	0.00	21.00	10.50	7.83	34.25	21.04	66.25	0.50	38.87
Chebychev	20.50	3.00	11.75	49.00	12.16	30.58	70.30	3.74	37.02
Correlation	10.00	8.00	9.00	23.75	25.83	24.79	52.26	3.33	27.79
Spearman	9.00	8.00	8.50	35.00	27.17	31.08	55.21	7.93	31.57
<b>Combinaison</b>	<b>7.41</b>	<b>8.25</b>	<b>7.83</b>	<b>30.25</b>	<b>15.83</b>	<b>23.04</b>	<b>58.75</b>	<b>0.33</b>	<b>29.54</b>

**Tableau 3. 5:** Taux d'erreur (%) fourni par différentes distances utilisées dans le noyau OC-SVM et la combinaison de différents seuils sur l'ensemble de données CEDAR.

performance. Ensuite, on ajoute pour chaque run 25 écrivains et on répéter l'évaluation. La figure 12 illustre l'influence du numéro de l'écrivain pour CEDAR ensemble de données lors de la sélection du seuil souple (optimal). Nous avons clairement peut noter que les AER diminuent lorsque le nombre d'écrivains augmente. Par conséquent, les performances dépendent fortement de la sélection nombre de rédacteurs utilisés lors de l'étape de conception du projet proposé HSVS. Par conséquent, il est important de sélectionner un nombre approprié de écrivains au cours de l'étape de conception.



**Figure3. 12:** AER selon le nombre d'écrivains pour CEDAR.

- **Analyse**

Le tableaux 3.6 rapportent les AER fournies par divers systèmes utilisant base de données CEDAR. Comparaisons avec d'autres systèmes sont difficiles en raison du type de signature (authentique et falsifiée signature) et le nombre de signatures sélectionnées lors de la étape de conception pour chaque rédacteur du HSVS. Pour le base de données CEDAR, on peut noter que notre système proposé offre de meilleures performances en termes d'AER comparativement à d'autres systèmes, qui utilisent signatures authentiques et falsifiées lors de l'étape de conception. Dans En revanche, dans notre approche, nous n'utilisons que des informations authentiques et fictives. signatures pour la formation et la définition du seuil optimal. En effet, nous obtenons des AER de 8,70 %, 7,83 % et 5,60 % en utilisant 4, 8 et 12 signatures authentiques, respectivement. offrent les meilleures performances en termes d'AER car le type et le nombre de signatures sélectionnées lors de la formation. Lorsque comparant nos résultats à Batista et al., nous pouvons noter que notre système proposé fournissait un AER réduit quel que soit le Signature. En effet, on obtient des AER de 16,92%, 15,95% et 15,07% lorsque en utilisant respectivement 4, 8 et 12 signatures authentiques.

References	Classifier	Feature	<sup>a</sup> Signature for training	AER (%)
Srihari et al. [11]	Naive Bayes	Global, statistical	1 Genuine + 1 forged	26.60
	Distance statistics	Geometrical and	1 Genuine + 1 forged	27.70
	Naive Bayes	Topological	16 Genuine + 16 forged	23.50
	Distance statistics		16 Genuine + 16 forged	21.70
Kumar et al. [13]	Neural network	Surroundedness	24 Genuine + 24 forged	8.33
<b>Our approach</b>	OC-SVM	Curvelet transform	4 Genuine	8.70
			8 Genuine	7.83
			12 Genuine	5.60

**Tableau 3. 6:** AER (%) fourni par le système proposé par rapport à d'autres systèmes utilisant l'ensemble de données CEDAR.

La signature pour la formation est le nombre de signatures pour chaque rédacteur utilisé pour la formation des classificateurs OC-SVM.

## 3.2. Python Deep learning

### 3.2.1. Présentation des outils de développement

#### 3.2.1.1. Matériel

Le matériel réalisé est PC personnel HP I3 avec un 4GB capacité mémoire, et un processeur Intel® N2840 @2.16 GHz 2.16 GHz, avec Windows 10 professionnel, 64 bit type du système.

#### 3.2.1.2. Python

Python est un langage de programmation de haut niveau interprété (il n'y a pas d'étape de compilation) et orienté objet avec une sémantique dynamique, a été créé en 1989 par Guido van Rossum, aux Pays-Bas. Il est très sollicité par une large communauté de développeurs et de programmeurs. Python est un langage simple, facile à apprendre et permet une bonne réduction du cout de la maintenance des codes. Les bibliothèques (packages) python encouragent la modularité et la réutilisabilité des codes. Python et ses bibliothèques sont disponibles (en source ou en binaires) sans charges pour la majorité des plateformes et peuvent être redistribués gratuitement [37].

#### 3.2.1.3. Google Colab

Google Colaboratory (Colab) est un service de cloud basé sur Jupyter Notebooks destiné à diffuser des informations et des formations en apprentissage automatique. Colaboratory fournit des temps d'exécutions sur Python 2 et 3, préconfigurées avec les bibliothèques d'apprentissage automatique, en profondeur et d'intelligence artificielle essentielles telles que TensorFlow, Matplotlib,pandas,time,numpy,os,skyimage et Keras, et un accès gratuit à un GPU robuste. De plus, les documents Colab sont enregistrés directement votre compte Google Drive [38].

#### 3.2.1.4.TensorFlow

TensorFlow est un framework de programmation pour le calcul numérique qui a été rendu Open Source par Google en Novembre 2015. Depuis son release, TensorFlow n'a cessé de gagner en popularité, pour devenir très rapidement l'un des frameworks les plus utilisés pour le Deep Learning et donc les réseaux de neurones. Son nom est notamment inspiré du fait que les opérations courantes sur des réseaux de neurones sont principalement faites via des tables de données multi-dimensionnelles, appelées Tenseurs (Tensor). Un Tensor à deux dimensions

est l'équivalent d'une matrice. Aujourd'hui, les principaux produits de Google sont basés sur TensorFlow: Gmail, Google Photos, Reconnaissance de voix [39].

- Méthode proposée pour la vérification des signatures : Les étapes de la méthode sont décrites ici :
  - Étape 1: Obtenir les signatures de la base de données CEDAR bd\_b.
  - Étape 2: Effectuer le prétraitement sur les signatures.
  - Étape 3: Architecture de CNN.

### 3.2.2. Base de données

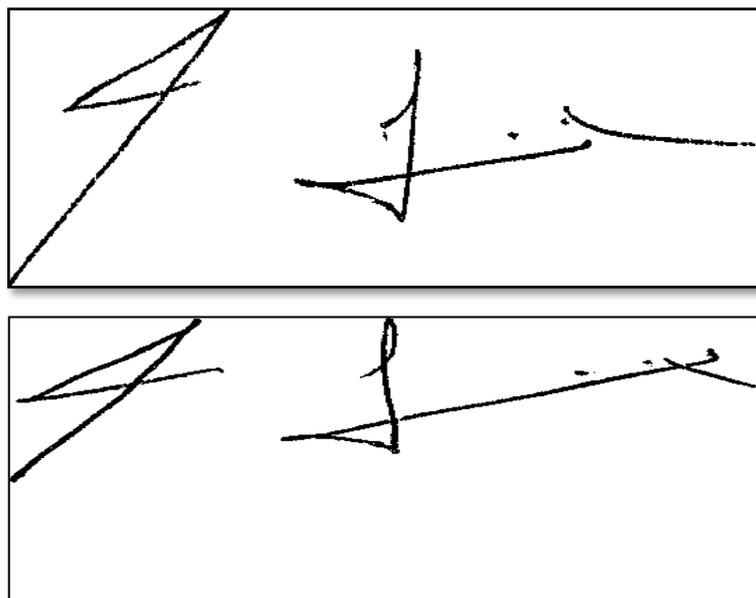
#### 3.2.2.1. Présentation de la base de données:

La signature du Centre d'excellence pour l'analyse et la reconnaissance de documents (CEDAR) L'ensemble de données (Kalera et al, 2004) est un ensemble de données couramment utilisé pour la vérification de signature hors ligne. La base de données CEDAR est composé de 55 rédacteurs ; chacun a 24 authentiques et 24 faux signatures. Les signatures des différents rédacteurs sont scannées à 300 dpi. Nous utilisons uniquement le jeu de données CEDAR dans le but d'évaluer la performance du

paramètres indépendants de l'écrivain systèmes de vérification de signature manuscrite.

#### 3.2.2.2. Acquisition de données

Dans la phase de test, les signatures peuvent être authentiques : écrites par l'auteur de référence, ou simulées par un autre auteur que l'auteur de référence, la collection contient des échantillons de signatures hors ligne et en ligne. En mode hors ligne, les signatures de vérification de signature seront constituées d'images PNG, numérisées à 300 dpi, couleur RVB (RGB). Chaque image de signature a une dimension de 502\*141 pixels et 476\*267



**Figure3. 13:** Exemple de la signature hore ligne de l'utilisateur 01.

### 3.2.3. Prétraitement

Puisque les signatures sont des images acquises par un scanner, ils ont des tailles différentes et ils sont représentés en RVB, donc, l'étape de prétraitement est primordiale. Le prétraitement regroupe l'ensemble des processus visant à améliorer les caractéristiques des signatures de la base de données. Cette étape consiste à préparer les données qui seront fournies au module d'authentification. Plusieurs traitements sont nécessaires, les deux suivants sont utilisés.

#### 3.2.3.1. Conversion d'image en niveaux de gris

Après l'acquisition numérique d'une signature, nous disposons d'une image RGB (à 16 millions couleurs). Utiliser des images en niveau de gris est plus fiable dans les systèmes de vérification de signature. Donc il faut calculer la valeur d'intensité lumineuse d'un pixel (valeur entre 0 et 255) au lieu de son couleur (valeur de 0 à 16 millions) En vision par ordinateur et traitement d'image, la méthode d'Otsu est utilisée pour effectuer un seuillage automatique à partir de la forme de l'histogramme de l'image, ou la réduction d'une image à niveaux de gris en une image binaire. L'algorithme suppose alors que l'image à binariser ne contient que deux classes de pixels, (c'est-à-dire le premier plan et l'arrière-plan) puis calcule le seuil optimal qui sépare ces deux classes afin que leur variance intra-classe soit minimale.

- **algorithme rgbgrey(img)**

```

1. // convert RGB values to grayscale at each pixel, then put in grayscale array
2.   for (int i = 0; i < iHeight; i++)
3.     for (int j = 0; j < iWidth; j++)
4.     {
5.         r = pImg[i*iWidth * 3 + j * 3 + 2];
6.         g = pImg[i*iWidth * 3 + j * 3 + 1];
7.         b = pImg[i*Wp + j * 3];
8.         r * 0.299;
9.         g * 0.587;
10.        b * 0.144;
11.        gray = std::round(r + g + b);
12.        pImgGS[i*Wp + j] = gray;
13.    }

```

- **algorithme grybin [40]**

```

1. int main(int argc, char* argv[])
2. {
3.     std::string gray, binary;
4.
5.     std::cout << "Enter the gray code ";
6.     std::cin >> gray;
7.     binary = gray[0];
8.     for (int i = 0; i < gray.length() - 1; i++)
9.     {
10.        /* XOR operation */
11.        if (binary[i] == gray[i+1])
12.            binary = binary + "0";
13.        else
14.            binary = binary + "1";
15.    }
16.    std::cout << "Gray Code : " << gray << std::endl
17.        << "Binary Code : " << binary << std::endl;
18.    return 0;
19. }

```

- **Output :**

```

- $ a.out
- Enter the gray code 1000
- Gray Code : 1000
- Binary Code : 1111
- $ a.out
- Enter the gray code 0101
- Gray Code : 0101
- Binary Code : 0110

```

### 3.2.3.2. Redimensionnement de l'image

Suite à différentes tailles d'images contenant les signatures et afin de normaliser la taille des images d'entrées du réseau de neurone convolutif, un processus de redimensionnement est

effectué. Les images sont d'abord recadrées dans une image de taille 502\*141 et 476\*267 (la plus grande image), en suite les images de résultats sont redimensionnées en taille standard 180\*180.

### 3.2.3.3. L'extraction de caractéristiques (Feature extraction)

L'extraction de caractéristiques fait partie du processus de réduction de la dimensionnalité, dans lequel un ensemble initial de données brutes est divisé et réduit à des groupes plus gérables. Ainsi, lorsque vous souhaitez traiter, ce sera plus facile. La caractéristique la plus importante de ces grands ensembles de données est qu'ils comportent un grand nombre de variables. Ces variables nécessitent beaucoup de ressources informatiques pour être traitées. Ainsi, l'extraction de fonctionnalités permet d'obtenir la meilleure fonctionnalité de ces grands ensembles de données en sélectionnant et en combinant des variables dans des fonctionnalités, réduisant ainsi efficacement la quantité de données. Ces fonctionnalités sont faciles à traiter, mais toujours capables de décrire l'ensemble de données réel avec précision et originalité. Pourquoi l'extraction de fonctionnalités est-elle utile ? La technique d'extraction des entités est utile lorsque vous disposez d'un grand ensemble de données et que vous devez réduire le nombre de ressources sans perdre d'informations importantes ou pertinentes. L'extraction de caractéristiques permet de réduire la quantité de données redondantes de l'ensemble de données. En fin de compte, la réduction des données aide à construire le modèle avec moins d'effort de la machine et augmente également la vitesse des étapes d'apprentissage et de généralisation dans le processus d'apprentissage automatique. Pour cette raison, (ratio , centroid, Eccentricity Solidity , SkewKurtosis doivent être calculées

Ensuite , Vous avez tous vu des ensembles de données. Parfois, ils sont petits, mais souvent parfois, ils sont extrêmement grands. Il devient très difficile de traiter les ensembles de données qui sont très volumineux, au moins suffisamment importants pour provoquer un goulot d'étranglement de traitement.

Alors, qu'est-ce qui rend ces ensembles de données aussi volumineux ? Eh bien, ce sont des fonctionnalités. Plus le nombre d'entités est élevé, plus les jeux de données seront volumineux. Eh bien, pas toujours. Vous trouverez des jeux de données où le nombre d'entités est très élevé, mais ils ne contiennent pas autant d'instances. Mais ce n'est pas le sujet de la discussion ici. Ainsi, vous pourriez vous demander avec un ordinateur de base en main comment traiter ce type d'ensembles de données sans tourner le pot.

Souvent, dans un ensemble de données de grande dimension, il reste des caractéristiques totalement non pertinentes, insignifiantes et sans importance. On a vu que la contribution de ces types de caractéristiques est souvent moindre vers la modélisation prédictive par rapport aux caractéristiques critiques. Ils peuvent également avoir une contribution nulle. Ces caractéristiques posent un certain nombre de problèmes qui, à leur tour, empêchent le processus de modélisation prédictive efficace -Allocation de ressources inutile pour ces fonctionnalités. Ces fonctionnalités agissent comme un bruit pour lequel le modèle d'apprentissage automatique peut très mal fonctionner. Le modèle de machine prend plus de temps à se former. Alors, quelle est la solution ici? La solution la plus économique est la sélection de fonctionnalités.

La sélection des caractéristiques est le processus de sélection des caractéristiques les plus significatives d'un jeu de données donné. Dans de nombreux cas, la sélection de fonctionnalités peut également améliorer les performances d'un modèle d'apprentissage automatique. Vous verrez une longue liste de fonctions disponibles. Pour l'instant, nous allons utiliser une fonction appelée `#getFeatures()` qui vous donnera la référence à toutes les fonctionnalités d'une couche. Dans notre cas, chaque entité sera un point représentant un aéroport. Vous pouvez taper la commande suivante pour parcourir chacune des entités de la couche actuelle. vous obtenez les fonctionnalités mentionnées précédemment, puis les combinez toutes dans un seul fichier.

Maintenant, vous apportez toutes les caractéristiques et les mettez dans le "CSV".

#### 3.2.3.4. Enregistrement des caractéristiques (Saving the Features)

Nous avons toutes les caractéristiques, nous faisons un fichier CSV en utilisant Excel

- C'est quoi CSV ?

Le module `csv` en Python permet d'analyser les fichiers CSV (Comma Separated Values). Un fichier CSV contient des valeurs séparées par des virgules, que l'on utilise pour stocker des données tabulaires. Il est le format le plus commun dans l'importation et l'exportation de feuilles de calculs et de bases de données. Le format fut utilisé pendant des années avant qu'aient lieu des tentatives de standardisation avec la RFC 4180. L'absence de format bien défini signifie que des différences subtiles existent dans la production et la consommation de données par différentes applications. Ces différences peuvent gêner lors du traitement de fichiers CSV depuis des sources multiples. Cependant, bien que les séparateurs et délimiteurs

varient, le format global est suffisamment similaire pour qu'un module unique puisse manipuler efficacement ces données, masquant au programmeur les détails de lecture/écriture des données. Le module csv implémente des classes pour lire et écrire des données tabulaires au format CSV. Il vous permet de dire « écris ces données dans le format préféré par Excel » ou « lis les données de ce fichier généré par Excel », sans connaître les détails précis du format CSV utilisé par Excel. Vous pouvez aussi décrire les formats CSV utilisés par d'autres applications ou définir vos propres specializations [41].

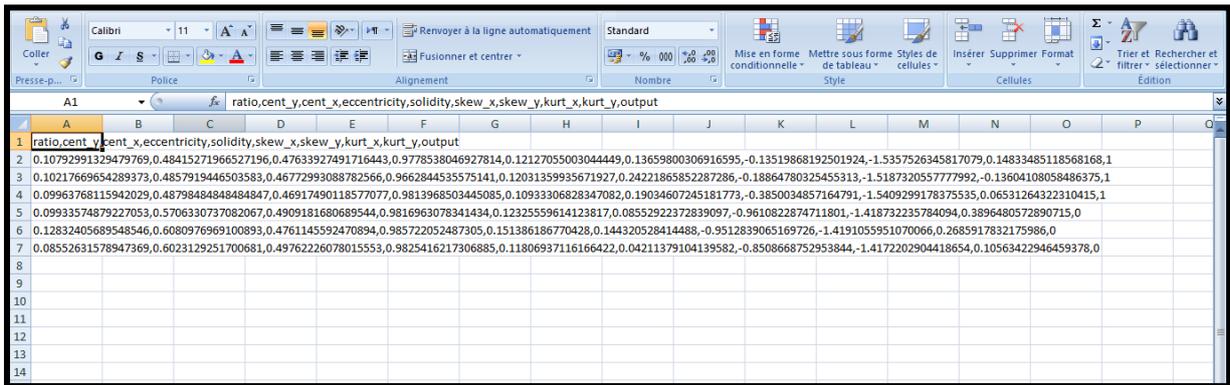


Figure3. 14: Notre fichier CSV en format Excel.

Ensuite ,nous exécutons le fichier CSV sur Google colab après il nous montre la création de fichiers CSV comme indiqué sur le capture suivant . Il offre des fonctions pour toutes les personnes , c’est-à-dire jusqu’à 55 utilisateurs .

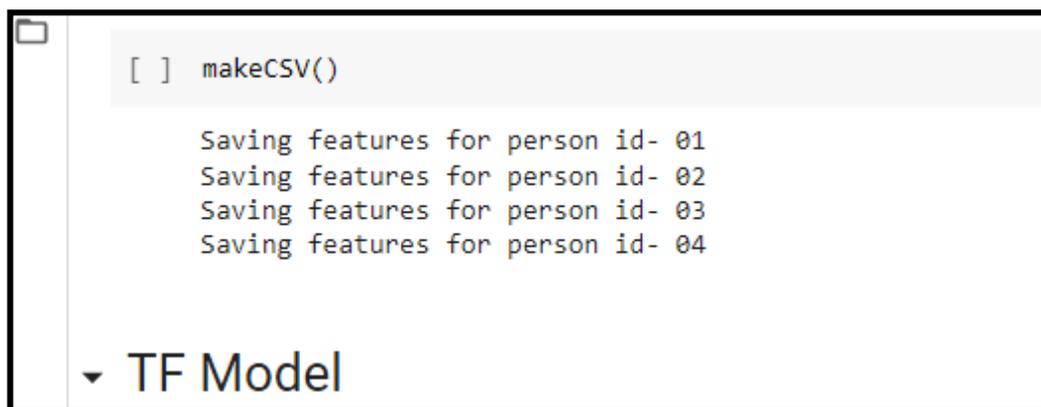


Figure3. 15: La creation de fichiers CSV.

### 3.2.3.5. évaluation de précision et perte ( accuracy & loss ) :

Lors de la formation de mon réseau de neurones à l'aide de Theano ou de tensorflow, une variable appelée "perte" par époque a été signalée. Maintenant la question est, comment interpréter cette variable ? Qu'est-ce que cela signifie pour la précision de mon réseau de neurones ?

Une fonction de perte est utilisée pour optimiser un algorithme d'apprentissage automatique. La perte est calculée sur la formation et la validation et son interprétation est basée sur la performance du modèle dans ces deux ensembles. C'est la somme des erreurs commises pour chaque exemple dans les ensembles d'apprentissage ou de validation. La valeur de perte implique à quel point un modèle se comporte mal ou bien après chaque itération d'optimisation.

Une métrique de précision est utilisée pour mesurer les performances de l'algorithme de manière interprétable. La précision d'un modèle est généralement déterminée après les paramètres du modèle et est calculée sous la forme d'un pourcentage. C'est la mesure de la précision de la prédiction de votre modèle par rapport aux données réelles.

Exemple Supposons que vous disposiez de 1000 échantillons de test et que votre modèle soit capable d'en classer 990 correctement, alors la précision du modèle sera de 99%.

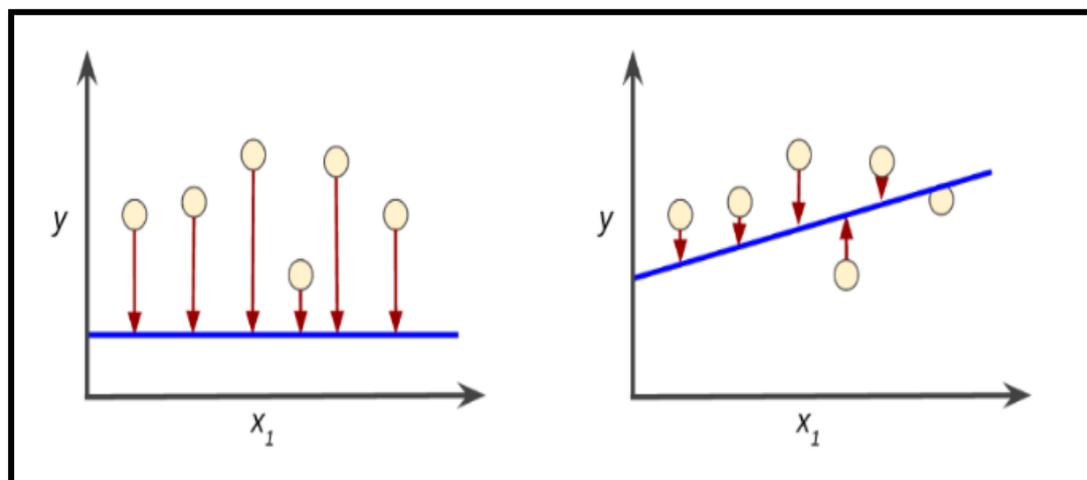
Former un modèle signifie simplement apprendre (déterminer) de bonnes valeurs pour tous les poids et le biais à partir d'exemples étiquetés.

La perte est le résultat d'une mauvaise prédiction. Une perte est un nombre indiquant à quel point la prédiction du modèle était mauvaise sur un seul exemple.

Si la prédiction du modèle est parfaite, la perte est nulle ; sinon, la perte est plus importante. L'objectif de la formation d'un modèle est de trouver un ensemble de pondérations et de biais qui ont une faible perte, en moyenne, dans tous les exemples. Une perte plus élevée est la pire (mauvaise prédiction) pour n'importe quel modèle.

La perte est calculée sur la formation et la validation et son interprétation est la performance du modèle pour ces deux ensembles. Contrairement à la précision, une perte n'est pas un pourcentage. Il s'agit de la somme des erreurs commises pour chaque exemple dans les ensembles d'apprentissage ou de validation.

Dans les schémas suivants, il y a deux graphiques représentant les pertes de deux modèles différents, le graphique de gauche a une perte élevée et le graphique de droite a une faible perte.



**Figure3. 16:** deux graphiques représentant les pertes de deux modèles different

- Les flèches représentent une perte.
- Les lignes bleues représentent les prédictions.

Dernière préparation avant d'utiliser le modèle.

Il faut séparer ces données en deux types :

les données d'entraînement vont servir à entraîner le modèle les données de validation vont valider son apprentissage En fait, cela permet d'éviter un problème fréquent en Machine Learning: l'overfitting.

L'overfitting, c'est le fait qu'un modèle se spécialise tellement sur ses données d'apprentissage qu'il devient inefficace sur d'autres données, des données réelles. On appelle aussi ce phénomène le surapprentissage.

C'est comme si vous vous entraîniez tous les jours au tennis mais seulement de votre revers. Dans un match réel vous serez très performant sur vos revers mais vos coups droits ne seront pas si bon...

L'idée c'est de bien entraîner notre modèle sur des données générales et de ne pas le surentraîner.

On peut ensuite compiler le modèle en choisissant l'optimizer, la loss function et la métrique. Puis entraîner le modèle avec la fonction fit().

La variable history nous permet de garder l'historique de l'apprentissage.

```
history = model.fit(train_data,
                    epochs=EPOCHS,
                    validation_data=valid_data,
                    shuffle=True,
                    verbose=1,
                    )

model.save('mod0.h5')

with open('history0', 'wb') as file_pi:
    pickle.dump(history.history, file_pi)

135/135 [=====] - 10s 76ms/step - loss: 0.0019 - acc: 0.9995 - val_loss: 0.0586 - val_acc: 0.9859
Epoch 73/100
135/135 [=====] - 10s 77ms/step - loss: 0.0023 - acc: 0.9995 - val_loss: 0.0588 - val_acc: 0.9859
```

Figure3. 17: Notre programme d'entraînement du modèle utilisant la fonction fit ()

- Dans la variable history on a les valeurs de la loss et de l'accuracy, la perte et la précision.

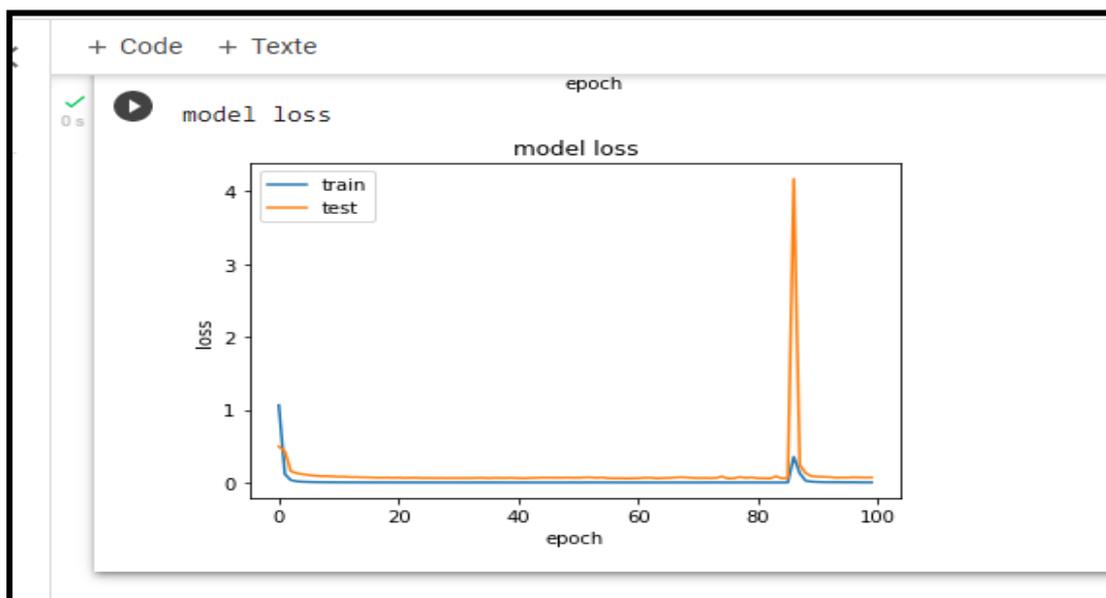


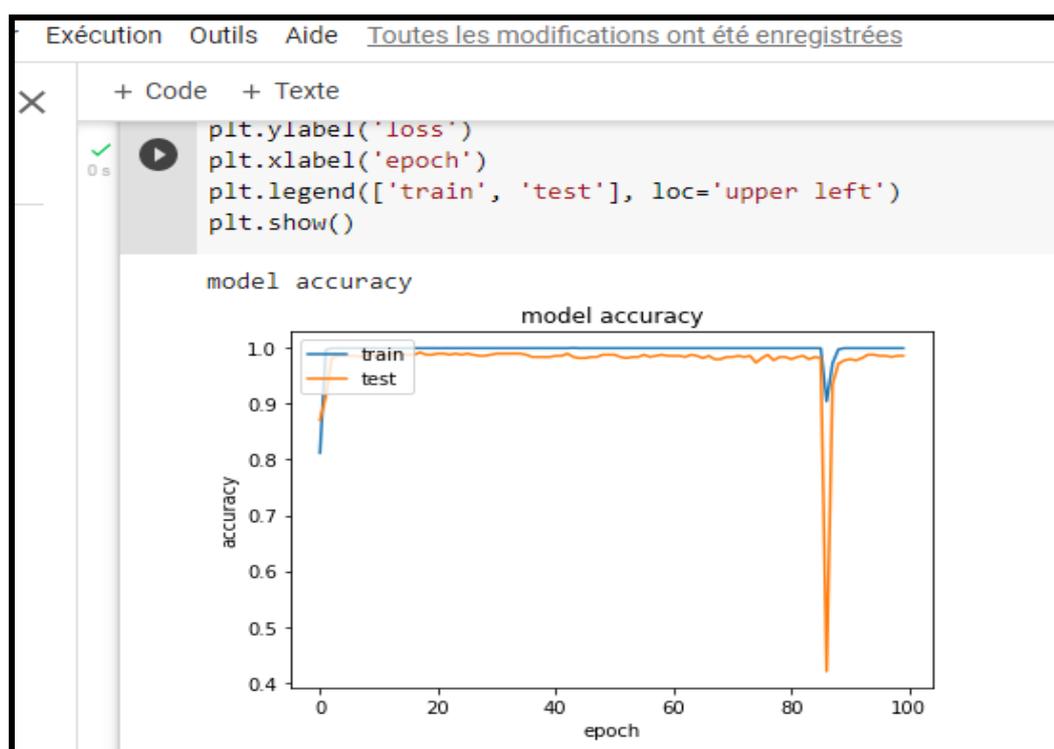
Figure3. 18: la courbe de la perte ( Loss )

On voit au niveau des premiers epoch que la perte sur les données d'entraînement et celle sur les données de validation diminue de manière similaire. Très vite les courbes ne diminuent plus aussi vite (ce qui est normal) et à un moment, à l'epoch 1, la perte sur les données de validation augmente et ne rebaisse plus. C'est exactement là qu'est l'overfitting.

Le modèle se spécialise sur les données d'entraînement donc la loss ne fait que diminuer pour ces données mais en se spécialisant autant il n'est plus capable d'être performant sur les données de validation et, du manière générale, sur les données réelles.

On peut d'ailleurs vérifier ce fait en traçant la courbe de la précision du modèle sur les données d'entraînement et sur les données de validation.

Effectivement, la précision diminue à partir de l'epoch 4 pour les données de validation alors qu'elle continue d'augmenter pour les données d'entraînement.



**Figure3. 19:** la courbe de la precision ( Accuracy ).

On peut aussi évaluer notre modèle sur les données de test.

On a à gauche la perte et à droite la précision.

Ici attention à ne pas confondre les deux métriques:

la précision peut être prise en pourcentage, si elle est de 0,85 alors 85%

la perte elle n'est pas un pourcentage, notre but est seulement de la faire tendre vers 0

#### 3.2.3.6. TF Model

Obtenez une signature à vérifier à partir de la base de données. Vérifiez la sortie émise à partir du réseau neuronal. Cette méthode commence par examiner les images sur PC avec des utilitaires de pointe, puis en modifiant leur qualité grâce à la mise à l'échelle de l'image, suivie de l'extraction des points lumineux et de la préparation du réel, pour célébrer enfin vérifier si la signature est vraie ou fausse. La signature hors ligne se fait sur papier.

Les images de réduction du son sont vendues en raison d'erreurs de traduction ou de canaux bruyants. A Les images sont également endommagées en raison des effets de bruit causés par l'éclairage et divers éléments de l'environnement. Supprimer le contexte - Nous avons engagé une stratégie de seuil pour reconnaître la signature du contexte. Après Seuil, les pixels de la marque seront à 1 et les différents pixels qui ont une position avec le fond seront à 0.

#### 3.2.4. Le travail

Dans ce projet, nous formons d'abord toutes les images de signature. Ces images formées et testées sont enregistrées dans le dossier de formation et de test.

Ensuite, il obtient le nom et le chemin de l'image en tant qu'entrée de l'utilisateur et commence la vérification étape par étape à l'aide de réseaux de neurones convolutifs.

Dans ce processus, le premier prétraitement d'image se produit. Dans binarisation du prétraitement de l'image, la segmentation est effectuée. L'image extraite est ensuite utilisée pour former le classifieur, et s'il correspond, il produit une sortie authentique

sinon, il produit une sortie frauduleuse (output).

### 3.2.4.1. Le résultat ( output ):

- Entrer l'identifiant de personne :

```
print("Training average-", train_avg/n)
print("Testing average-", test_avg/n)
print("Time taken-", time()-start)
return train_avg/n, test_avg/n, (time()-start)/n

evaluate(train_path, test_path, type2=True)

WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/pytho
Instructions for updating:
non-resource variables are not supported in the long term
Enter person's id : 
```

**Figure3. 20:** execution de notre programme python.

- Par exemple : 01 (utilisateur 01) :

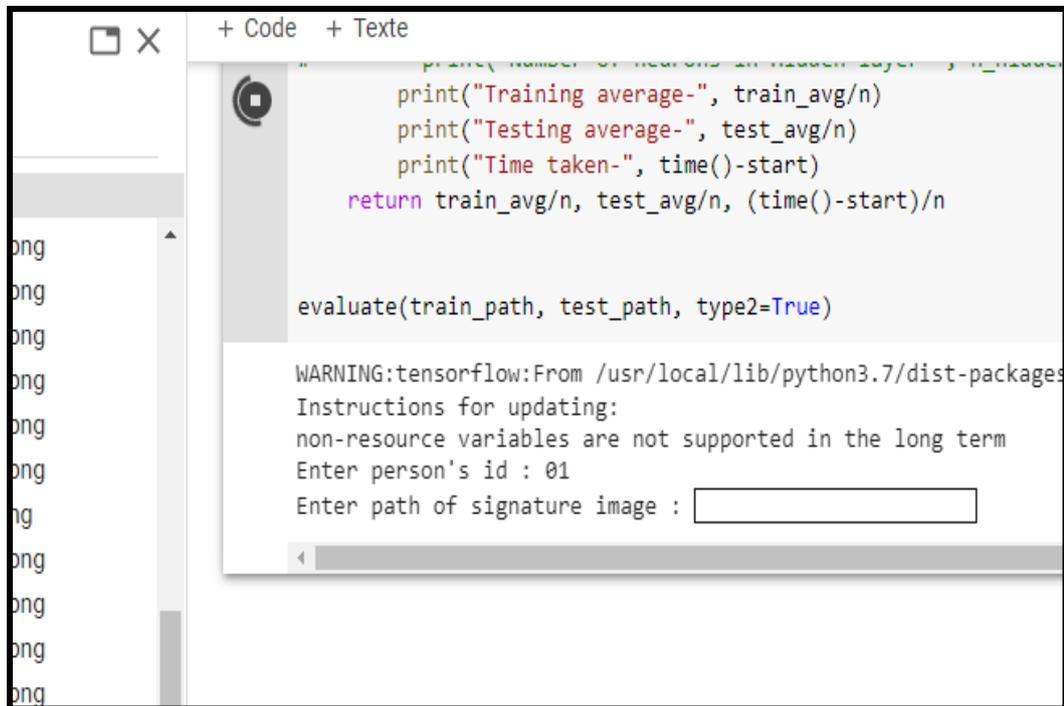
```
# print("Number of neurons in Hidden layer-", n_hidden)
print("Training average-", train_avg/n)
print("Testing average-", test_avg/n)
print("Time taken-", time()-start)
return train_avg/n, test_avg/n, (time()-start)/n

evaluate(train_path, test_path, type2=True)

WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages
Instructions for updating:
non-resource variables are not supported in the long term
Enter person's id : 01
```

**Figure3. 21:** Entrer l'identifiant ( id) de l'utilisateur 01.

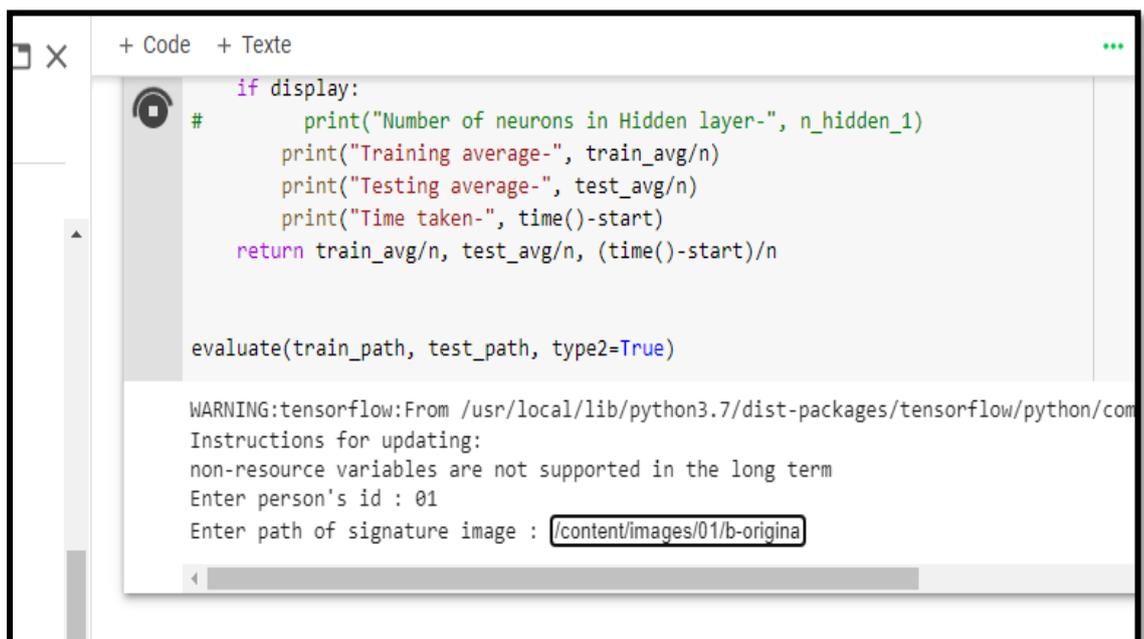
- Maintenant , entrer le chemin de l'image de la signature



```
+ Code + Texte  
print("Number of neurons in hidden layer-", n_hidden_1)  
print("Training average-", train_avg/n)  
print("Testing average-", test_avg/n)  
print("Time taken-", time()-start)  
return train_avg/n, test_avg/n, (time()-start)/n  
  
evaluate(train_path, test_path, type2=True)  
  
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages  
Instructions for updating:  
non-resource variables are not supported in the long term  
Enter person's id : 01  
Enter path of signature image : 
```

**Figure3. 22:** Entrer le chemin de l'image de la signature.

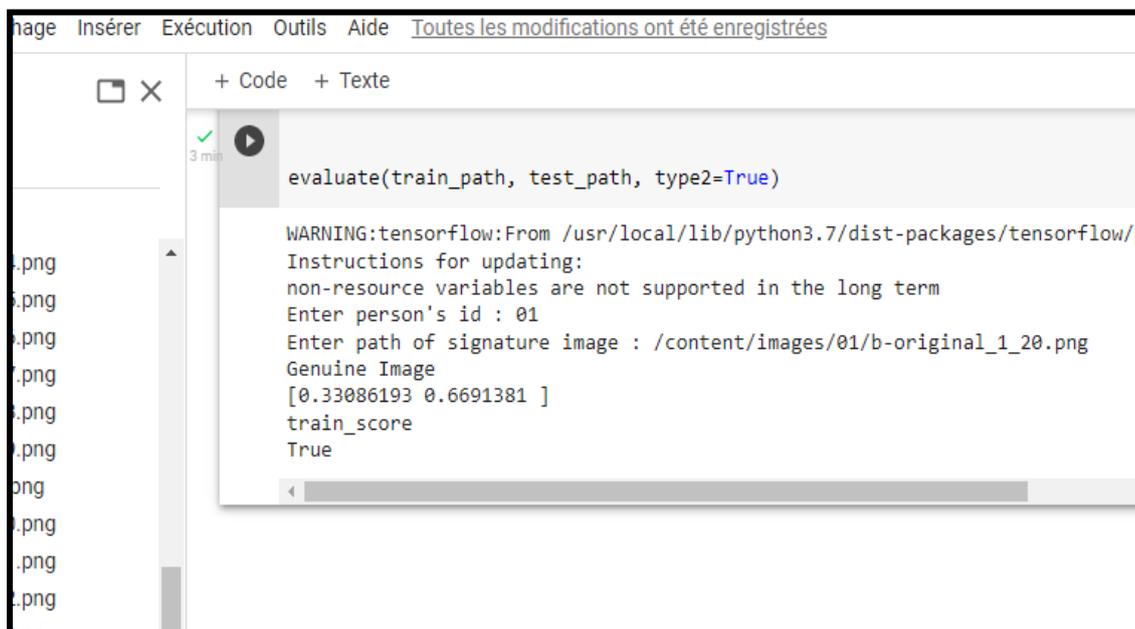
- Par ex : “ /content/images/01/b-original\_1\_20.png ”



```
+ Code + Texte  
if display:  
# print("Number of neurons in Hidden layer-", n_hidden_1)  
print("Training average-", train_avg/n)  
print("Testing average-", test_avg/n)  
print("Time taken-", time()-start)  
return train_avg/n, test_avg/n, (time()-start)/n  
  
evaluate(train_path, test_path, type2=True)  
  
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/com  
Instructions for updating:  
non-resource variables are not supported in the long term  
Enter person's id : 01  
Enter path of signature image : /content/images/01/b-original_1_20.png
```

**Figure3. 23:** Exemple pour entrer le chemin de l'image "b-original\_1\_20.png".

- Et puis vous obtenez le résultat suivant après avoir testé l'image " b-original\_1\_20.png "



```
evaluate(train_path, test_path, type2=True)

WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/...
Instructions for updating:
non-resource variables are not supported in the long term
Enter person's id : 01
Enter path of signature image : /content/images/01/b-original_1_20.png
Genuine Image
[0.33086193 0.6691381 ]
train_score
True
```

**Figure3. 24:** Le resultat obtenu pour le test d'image " b-original\_1\_20.png ".

- Nous donnerons aussi un exemple de la fausse image dont le nom " b-forgeries\_1\_15.png".Le résultat obtenu sur la capture suivant



```
evaluate(train_path, test_path, type2=True)

Enter person's id : 01
Enter path of signature image : /content/images/01/b-forgeries_1_15.png
Forged Image
[[0.84349114 0.15650882]]
train_score
False
```

**Figure3. 25:**Le resultat obtenu pour le test d'image " b-forges\_1\_15.png".

### 3.2.5. Interface

La programmation orientée objet convient particulièrement bien au développement d'applications avec interface graphique. Des bibliothèques de classes comme Tkinter ou wxPython fournissent une base de widgets très étoffée, que nous pouvons adapter à nos besoins par dérivation. Dans ce chapitre, nous allons utiliser à nouveau la bibliothèque Tkinter, mais en appliquant les concepts décrits dans les pages précédentes, et en nous efforçant de mettre en évidence les avantages qu'apporte l'orientation objet dans nos programmes.

Tk / Tcl fait depuis longtemps partie intégrée de Python. Il fournit un jeu d'outils robustes et indépendants de la plateforme pour gérer les fenêtres. Disponible aux développeurs via le paquet tkinter et ses extensions, les modules tkinter.tix et tkinter.ttk.

Le package tkinter est une fine couche orientée objet au-dessus de Tcl/Tk. Pour utiliser tkinter, vous n'avez pas besoin d'écrire de code Tcl, mais vous devrez consulter la documentation Tk, et occasionnellement la documentation Tcl. tkinter est un ensemble de wrappers qui implémentent les widgets Tk en tant que classes Python.

Les principales vertus de tkinter sont qu'il est rapide et qu'il est généralement fourni avec Python. Bien que sa documentation standard soit faible, du bon matériel est disponible, qui comprend : des références, des tutoriels, un livre et autres. tkinter est également célèbre pour son apparence et sa convivialité obsolètes, qui ont été considérablement améliorées dans Tk 8.5. De plus, il existe de nombreuses autres bibliothèques d'interface graphique qui pourraient vous intéresser. Le wiki Python répertorie plusieurs frameworks et outils GUI alternatives [42].

Nous avons créé une interface sous la forme d'un logiciel qui nous permet de tester des images au lieu de code et au lieu d'utiliser le Google colab .

- Voilà! C'est ça notre interface . Il contient le logo de l'université UMBB

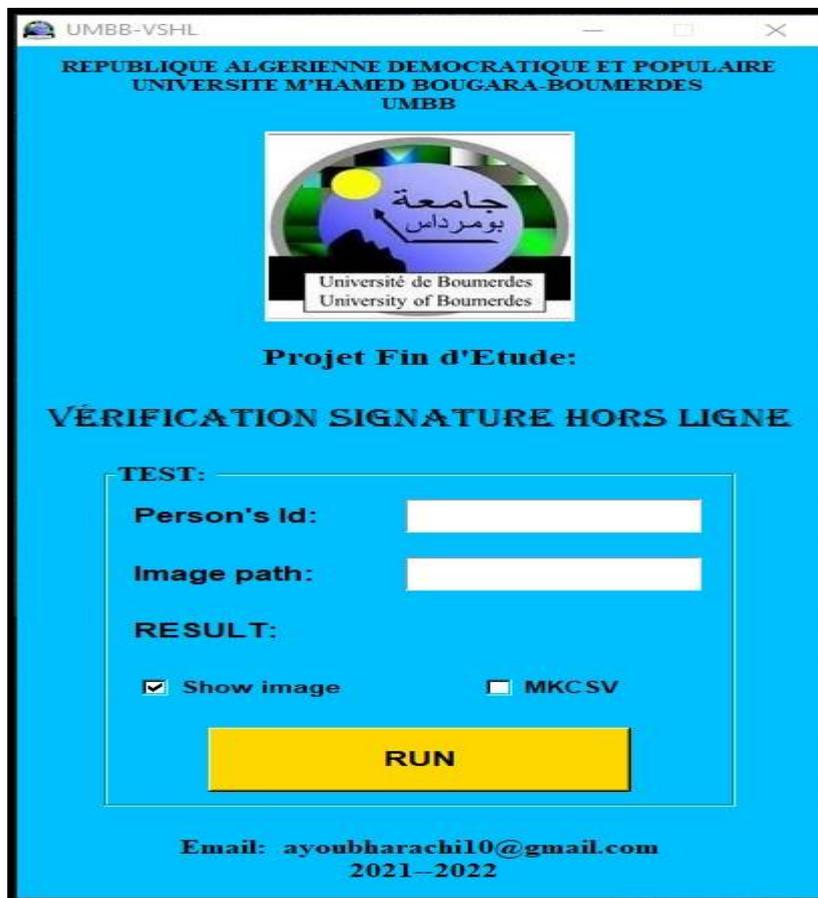
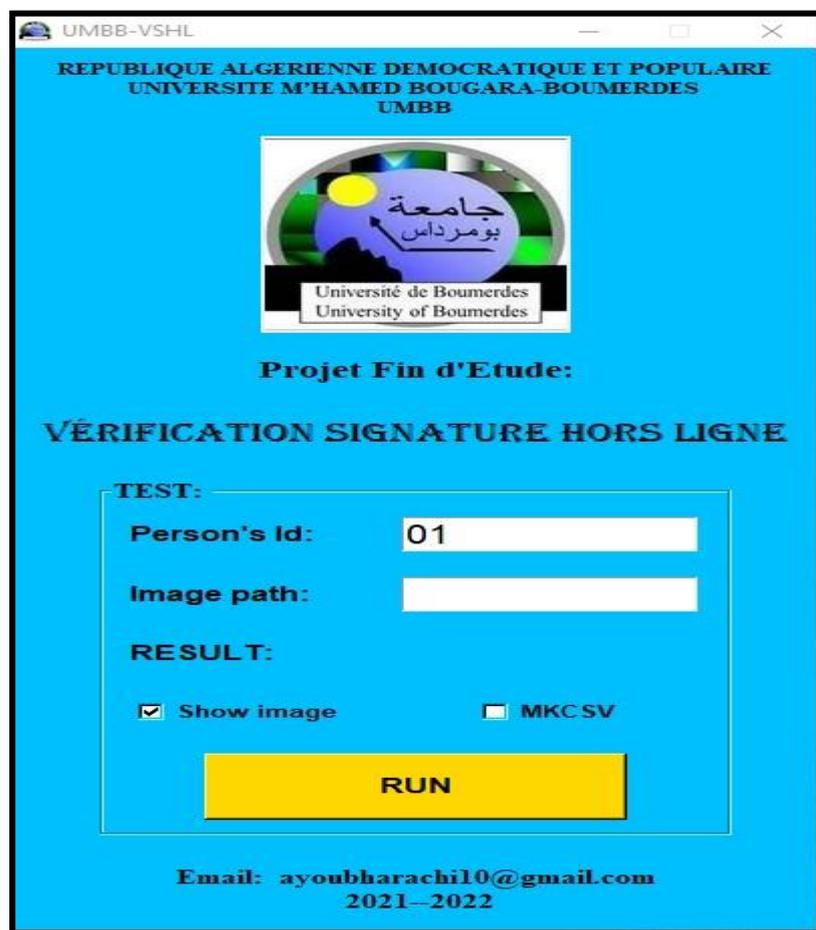


Figure3. 26: Execution de notre interface.

- Ensuite, Nous écrivons le numéro de l'utilisateur (person's id ) par ex : 01



The screenshot shows a web browser window titled "UMBB-VSHL". The page has a blue background and contains the following text and elements:

- Header: REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE, UNIVERSITE M'HAMED BOUGARA-BOUMERDES, UMBB
- Logo: A circular logo with Arabic text "جامعة بومرداس" (University of Boumerdes) and "Université de Boumerdes / University of Boumerdes" below it.
- Section: **Projet Fin d'Etude:**
- Section: **VÉRIFICATION SIGNATURE HORS LIGNE**
- Section: **TEST:**
  - Person's Id:
  - Image path:
- Section: **RESULT:**
  - Show image
  - MKCSV
- Button: **RUN** (yellow)
- Footer: Email: ayoubharachi10@gmail.com, 2021-2022

Figure3. 27: Entrer l'identifiant ( id) de l'utilisateur 01.

- Après nous copié le chemin de l'image "b-original\_1\_15"



Figure3. 28: Le resultat obtenu pour le test d'image " b-original\_1\_20.png ".

- Aussi nous avons copié le chemin de l'image " b-forgeries\_1\_15 " et l'avons mis dans la case (image path)



Figure3. 29: Le resultat obtenu pour le test d'image " b-forgeries\_1\_15.png".

### 3.3. Comparaison entre SVM MATLAB et DEEP LEARNING PYTHON

Nous avons présenté un système qui classe la signature et grâce à lui nous savons si cette signature est vraie ou fausse en utilisant Python et Matlab par contre, le but de tout cela est qu'il faut bien fixer le seuil et afin d'obtenir meilleurs résultats. Les résultats obtenus sont meilleurs par rapport au passé car le taux d'erreur a diminué de près de zéro, mais la question est quelle est la meilleure façon de faire cela ? Sachant que Matlab et Python sont des méthodes différentes pour beaucoup de choses.

- La différence réside dans beaucoup de choses. Nous en citons quelques-uns :

#### 3.3. 1. Langage utilisé dans Python et Matlab

Par définition, Python est un langage de programmation multiparadigme à usage général écrit en ANSI C portable et utilisé pour développer des applications à part entière ou d'autres outils logiciels. Le langage de base et les bibliothèques de Python fonctionnent de la même manière sur toutes les plates-formes, vous pouvez donc programmer dans des environnements mixtes. Matlab, d'autre part, est un langage commercial orienté vers les mathématiques et les matrices pour la programmation mathématique. Matlab dispose de boîtes à outils spécialisées développées par des professionnels et méticuleusement testées pour plusieurs applications scientifiques et techniques. Matlab a des fonctionnalités numériques plus complètes que Python.

#### 3.3. 2. Bibliothèques

Python est livré avec une bibliothèque standard étendue qui vous permet de créer un serveur Web, de télécharger un fichier, de décompresser une archive compressée, etc., avec seulement quelques lignes de code. La bibliothèque destinée à la programmation à usage général et contient des modules pour des éléments spécifiques au système d'exploitation tels que le threading, la mise en réseau, les bases de données, etc. La bibliothèque standard de Matlab, d'autre part, ne contient pas autant de fonctionnalités pour la programmation à usage général, mais elle a plusieurs boîtes à outils puissantes pour résoudre les problèmes les plus courants.

### 3.3. 3. Environnement

Python et Matlab ont de nombreux points communs, notamment une syntaxe simple et propre, aucune déclaration de variables, la création facile d'interfaces graphiques et le collage de la visualisation et de la simulation. Cependant, Python a clairement un avantage sur Matlab en termes d'environnement de développement. L'environnement Python est complètement ouvert et conçu pour l'intégration avec des outils externes, tandis que Matlab est un programme mathématique interactif utilisé pour l'analyse numérique, la visualisation graphique, le traitement du signal et de nombreuses autres applications. Matlab est indépendant de la plateforme, ce qui signifie que le programme peut s'exécuter invariablement sur toutes les plateformes.

### 3.3. 4. Indexage

Python utilise une indexation basée sur 0, ce qui signifie que les tableaux sont indexés à partir de 0 pour assurer la cohérence avec d'autres familles de langages, notamment C, C++ et Java. L'indexation basée sur 0 signifie que le premier élément est à l'index 0, plutôt qu'à 1, et que le deuxième élément est à l'index 1, et ainsi de suite. Cependant, les tableaux peuvent également être indexés à l'aide de nombres négatifs, ce qui simplifie la forme des algorithmes. Matlab, d'autre part, utilise l'indexation basée sur 1 comme la plupart des vecteurs et des matrices, ce qui signifie simplement qu'il a une indexation des tableaux à partir de 1 au lieu de 0.

### 3.3. 5. Performances de Python Vs. Matlab

Python est plus expressif et lisible que Matlab, lorsqu'il s'agit de problèmes à grande échelle et les bibliothèques de Python sont plus sophistiquées pour écrire des programmes utilitaires et des scripts CGI. Cependant, les capacités graphiques de Matlab sont plus pratiques que celles de Python car ses graphiques reposent sur des packages externes qui doivent être installés séparément. Le code Python nécessite des modules complémentaires orientés développeur pour obtenir des performances améliorées telles que la programmation parallèle explicite et la compilation juste-à-temps. Cependant, Matlab est faible en tant que langage de programmation à usage général [43].

- Nous avons déjà choisi de travailler avec Python, et ce n'est pas une preuve que nous n'aimons pas travailler avec Python ou même sous-estimons sa valeur. Pour plusieurs raisons :  
pourquoi python ?

La première étape de l'utilisation de Python consiste à décider de l'utiliser. Il existe de nombreuses raisons pour lesquelles on pourrait envisager d'utiliser Python, et j'ai énuméré les principales pour moi dans ce qui suit. De même, il y a aussi de bonnes raisons pour lesquelles Matlab est supérieur à Python. Ceux-ci sont également répertoriés ci-dessous.

- C'est gratuit

Celui-ci est simple, mais l'un des plus importants pour quiconque travaille en dehors du milieu universitaire. Comme Matlab est un logiciel coûteux, de nombreuses entreprises n'ont qu'une seule licence (le cas échéant). Pour cette raison, l'utilisation d'une alternative gratuite pourrait être intéressante pour de nombreuses entreprises.

- C'est open source

Avoir un langage où chacun peut contribuer à son développement signifie que de nouvelles fonctionnalités seront constamment ajoutées et que des bogues seront corrigés. Cela signifie également que vous pouvez aller inspecter exactement le fonctionnement des fonctions que vous utilisez. Cela contribue également à la croissance du langage et contribue à créer une grande communauté d'utilisateurs et de contributeurs avancés.

- C'est l'avenir

La popularité de Python n'a cessé d'augmenter de façon exponentielle au cours des dernières années. Avoir un langage plus populaire signifie qu'il est plus facile de trouver des réponses à toutes vos questions et de trouver des exemples de code de ce dont vous avez besoin. Il a plus de fonctionnalités

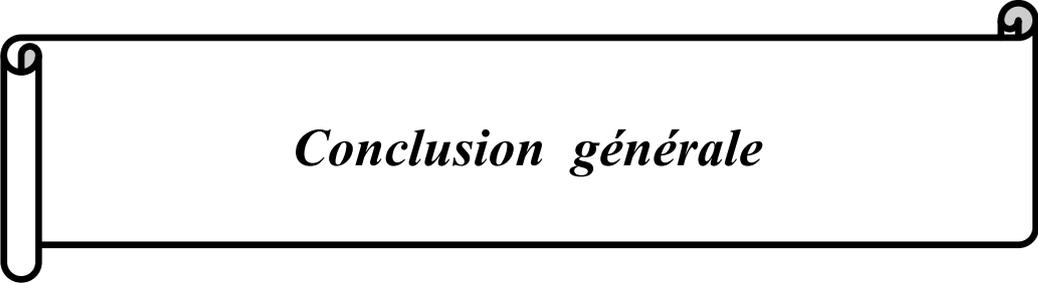
Avec la popularité croissante de l'apprentissage automatique et de l'IA, Python a des années-lumière d'avance sur Matlab, car tous les principaux frameworks sont basés sur Python : Tensorflow, Keras, PyTorch, Scikit-learn. Et puisque toutes les recherches sur l'IA sont effectuées à l'aide de ces frameworks, il est beaucoup plus facile de trouver des algorithmes de pointe pour Python que Matlab [44].

### 3.4. Conclusion

Dans ce chapitre, nous avons présenté différentes formulations pour l'apprentissage des représentations pour la signature hors ligne vérification.

Pour un système individuel basé sur un HSVS dépendant de l'écrivain, le classificateur OC-SVM est plus précis que le classificateur BC-SVM. Le principal avantage des paramètres combinés indépendants de l'écrivain HSVS est qu'il permet de concevoir le HSVS en utilisant peu d'écrivains et de signatures. Lorsqu'un nouvel écrivain est présenté au système, les mêmes paramètres de l'OC-SVM sont utilisés sans trouver le seuil optimal.

L'analyse des résultats ci-dessus montre que les réseaux de neurones convolutifs sont meilleurs pour le classement des signatures. L'intuition de classer entre signatures authentiques et contrefaçons (indépendamment de l'utilisateur) en apprenant les repères visuels a amélioré la précision. L'amélioration significative de la la précision est également due à la nouvelle architecture inspirée de GoogleNet, qui a fonctionné plus largement que d'aller Plus profond. Par conséquent, à partir des résultats expérimentaux ci-dessus, il est clair que l'architecture InceptionSVGNet est plus efficace pour identifier les modèles dans les images en utilisant les réseaux plus larges. Ce modèle se poursuivra pendant travaux futurs, les chercheurs procédant à l'étude de meilleurs ensembles de fonctionnalités (à l'aide de réseaux d'apprentissage en profondeur) et des approches pour améliorer l'arrangement avec un nombre limité de tests. Techniques basées sur des ensembles de les classificateurs, en particulier les méthodes de choix dynamique, sont également encourageants dans ce domaine.



*Conclusion générale*

### Conclusion

Nous avons proposé d'étudier l'utilisation du classifieur OC-SVM qui permettait de classer une seule classe (objets) parmi tous les autres objets possibles grâce à trois reconnaissances manuscrites applications telles que la reconnaissance des chiffres manuscrits, la vérification des signatures manuscrites et reconnaissance du genre. En essayant d'utiliser l'OC-SVM, trois difficultés principales sont apparues, à savoir liées à l'ajustement de leurs paramètres, la sensibilité au nombre d'échantillons pour formation et distance utilisées comme métrique dans le noyau RBF.

Dans ce mémoire, nous avons présenté différentes formulations pour l'apprentissage des représentations pour la signature hors ligne. L'analyse des résultats ci-dessus montre que les réseaux de neurones convolutifs sont meilleurs pour le classement des signatures. L'intuition de classer entre signatures authentiques et contrefaçons (indépendamment de l'utilisateur) en apprenant les repères visuels a amélioré la précision. L'amélioration significative de la précision est également due à la nouvelle architecture inspirée de GoogleNet, qui a fonctionné plus largement que d'aller plus profond. Par conséquent, à partir des résultats expérimentaux ci-dessus, il est clair que l'architecture InceptionSVGNet est plus efficace pour identifier les modèles dans les images en utilisant les réseaux plus larges. Ce modèle se poursuivra pendant travaux futurs, les chercheurs procédant à l'étude de meilleurs ensembles de fonctionnalités (à l'aide de réseaux d'apprentissage en profondeur) et des approches pour améliorer l'arrangement avec un nombre limité de tests. Techniques basées sur des ensembles de classificateurs, en particulier les méthodes de choix dynamique, sont également encourageants dans ce domaine.

## Références

- [1] [S. Liu, M. Silverman, « A Practical Guide to Biometric Security Technology », IEEE Computer Society, IT Pro-Security, Janvier-Février 2001.]
- [2] [Morizet, N. « Reconnaissance Biométrique par Fusion Multimodale du Visage et de l'Iris ». Thèse présentée pour l'obtention du grade de Docteur en Electronique, Spécialité : Signal et Images, Ecole Nationale Supérieure des Télécommunications, France, 209 p, 2009.]
- [3] [Wirotius, M. « Authentification par signature manuscrite sur support nomade ». Thèse présentée pour l'obtention du grade de Docteur en Informatique, Université François Rabelais Tours, France, 161 p, 2005.]
- [4] [Santos, C., Justino, E., Bortolozzi, F., Sabourin, R. « An Off-Line Signature Verification Methode based on the Questioed Document Expert's Approach and a Neural Network Classifier ». International Workshop On Frontiers in Handwriting Recognition : IWFHR, Japan, p. 498-502, 2004.]
- [5] [Impedovo, S., Pirlo, G. « Verification of Handwritten Signature: an Overview ». 14th International Conference on Image Analysis and Processing : ICIAP 2007, IEEE, Italia, p. 191-196, 2007.]
- [6] [ Sabourin, R., Genest, G. « Definition and Evaluation of a Family of Shape Factor for Off-Line Signature Verification ». First Brazilian Symposium, BSDIA'97 Curitiba, Vol. 1339, p. 261-272, 1997. ] et [Chen, S. « Offline Automatic Signature Verification Techniques: F flexible Template Matching Approaches ». Doctoral thesis, Specialty: Signal Processing, Faculty of the Graduate School of the State University New York, USA, 250 p, 2006.]
- [7] [ Kholmatov, A. « Biometric Identity Verification Using On-Line & Off-Line Signature Verification », Sabanci University, 55 p, 2003.]
- [8] [MIOULET, L., « Reconnaissance de l'écriture manuscrite avec des réseaux récurrents », Université de rouen, 2015.]
- [9] [NACEREDDINE Nafaâ., «Segmentation d'images par Approches Statistiques et Recherche d'images par le Contenu. Application aux images radiographiques de soudures», Thèse, ENP , 2011]

- [10] [CHERGUI, A., « Etude comparative des descripteurs pour la Classification du Genre de Scripteurs », USTHB, 2013.]
- [11] [ Fergani, B. « Application des méthodes à Vecteurs de support pour l'indexation en locuteur de documents sonores ». Thèse présenté pour l'obtention du Diplôme de Doctorat d'état en Electronique, Spécialité : Electronique des Systèmes, 170 p, 2007 .]
- [12] [ BOUSLIMI, R., « Système de reconnaissance hors-ligne des mots manuscrits arabe pour multi-scripteurs », UNIVERSITE DE JENDOUBA, 2006.]
- [13] [B. Gosselin , « Application de réseaux de neurones artificiels à la reconnaissance automatique de caractères manuscrits » ,PFD , Fac. Polytech. Mons, 1996.]
- [14] [Ayat, N. « Sélection de modèle automatique des machines a vecteurs de support : application à la reconnaissance d'images chiffres manuscrits ». Thèse, Ecole de Technologie Supérieure de l'Université du Québec, Canada, 2007.]
- [15] [Reconnaissance des Formes - Intelligence Artificielle, « Reconnaissance de l'écriture en ligne par les réseaux de neurones dynamiques », UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE D'ORAN - MOHAMED BOUDIAF, 2015.]
- [16] [Wojciech PIECZYNSKI, « Modèles de Markov en traitement d'images », PFD, Evry, France, 2003.]
- [17] [Boukharouba Abdelhak, « Contribution à la segmentation et à la reconnaissance de l'écriture arabe manuscrite », Thèse , Université MENTOURI, 2011.]
- [18] [GUERBAI, Y., « Handwritten Recognition Based on One-Class Support Vector Machine », thèse, USTHB. ]
- [19] [Perronnin, F., Degelay, J. « Introduction à la biométrie, authentification des individus par traitement audio-vidéo ». Revue Traitement du signal, Vol. 19, p. 253-265, 2002.]
- [20] [ <https://www.thalesgroup.com/fr/europe/france/dis/gouvernement/inspiration/biometrie> consulté le 21 mars 2022.]
- [21] [GUERBAI, Y., « Handwritten Recognition Based on One-Class Support Vector Machine », thèse, USTHB.]

- [22] [ CHERGUI, A., « Etude comparative des descripteurs pour la Classification du Genre de Scripteurs », USTHB, 2013]
- [23] [ <http://cedric.cnam.fr/vertigo/cours/ml2/coursMethodesNoyaux.html>consulté le 21 mars 2022.]
- [24] [ <https://medium.com/@mail.garima7/one-class-svm-oc-svm-9ade87da6b10>]consulté le 21 mars 2022.]
- [25] [B. MARR, What Is Deep Learning AI ? A Simple Guide With 8 Practical Examples, 2018.]
- [26] [GUERBAI, Y.,«The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters » Article ,UMBB, 2015]
- [27] [S AGATONOVIC-KUSTRIN et R BERESFORD, “Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research”, Journal of pharmaceutical and biomedical analysis, t. 22, no 5, p. 717-727, 2000.]
- [28] [R. MIIKKULAINEN, J. LIANG, E. MEYERSON, A. RAWAL, D. FINK, O. FRANCON, B. RAJU, H. SHAHRZAD, A. NAVRUZYAN, N. DUFFY et al., “Evolving deep neural networks”, in Artificial Intelligence in the Age of Neural Networks and Brain Computing, Elsevier, 2019, p. 293-312. ]
- [29] [J. DEJASMIN, Les réseaux de neurones convolutifs. 2018.]
- [30] [Guerbai et al, 2014a. Guerbai et al, 2014b,Guerbai et al, 2015.]
- [31] [ Donoho et Duncan, 1999. Candés et coll., 2004]
- [32] [Cha et Srihari, 2002]
- [33] [ Guerbai et Chibani, 2012 <<One-class versus bi-class SVM classifier for off-line signature verification>> conference , morocco]
- [34] [Guerbai et al, 2014a.b, <<One-class versus bi-class SVM classifier for off-line signature verification>> Article , Stockholm, Sweden 2014,]

[35] [Kalera et al, , « OFFLINE SIGNATURE VERIFICATION AND IDENTIFICATION USING DISTANCE STATISTICS» , article, International Journal, 2004]

[36] [ Bengio et Mariéthoz, , « A STATISTICAL SIGNIFICANCE TEST FOR PERSON AUTHENTICATION», Article , rue du Simplon 4 1920 Martigny, Switzerland,2004 ]

[37] [<https://python.sdv.univ-paris-diderot.fr/cours-python.pdf>]

[38] [<https://ledatascientist.com/google-colab-le-guide-ultime/>]

[39] [<https://www.tensorflow.org/?hl=fr>]

[40][<https://stackoverflow.com/questions/45604567/cvisual-studio-change-rgb-to-grayscale> ]

[41] [ <https://docs.python.org/fr/3/library/csv.html> ]

[42] [ <https://python.developpez.com/cours/TutoSwinnen/> ]

[43] [<http://www.differencebetween.net/technology/internet/difference-between-python-and-matlab/> ]

[44] [ <https://medium.com/swlh/python-for-matlab-users-part-1-why-python-python-vs-matlab-959d92d702ef> ]