UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À

L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE APPLIQUÉES

PAR

ABDOULAYE TINE

# VÉRIFICATION DE SIGNATURES MANUSCRITES BASÉE SUR L'APPRENTISSAGE PROFOND

Juin 2023

## REMERCIEMENTS

Tout d’abord, je remercie Dieu, le Tout-Puissant, de m’avoir donné la force et le courage d’accomplir ce travail.

Ensuite je voudrais exprimer ma profonde gratitude à monsieur Nouboud Fathallah, mon directeur de recherche, qui s'est toujours montré à l'écoute et très disponible tout au long de ce mémoire, pour son soutien et ses encouragements continuels ainsi que pour sa patience. Ce que j'ai appris en travaillant avec professeur Nouboud ne se limite pas à l'aspect scientifique, mais aussi humain et relationnel. Ô combien, il serait difficile de vouloir remercier Nouboud. Car remercier une personne, c'est la remercier à sa juste valeur. En effet, comme le disait le penseur français Pierre Dac : « la façon de remercier dépend de ce que l'on reçoit ». De ce que j'ai reçu de Nouboud, je ne pourrais jamais en estimer la valeur. La réalisation de ce mémoire a été une expérience vraiment gratifiante pour moi, au point que j'ai maintenant envie de poursuivre mes recherches. Cette expérience restera gravée dans ma mémoire pour toujours. Bien que je pourrais continuer à écrire davantage, je tiens simplement à dire : Nouboud, merci infiniment pour tout ce que vous avez fait pour moi !

Ma gratitude va aussi aux professeurs Alain Goupil et Maxime Bérubé qui, de par leurs minutieuses corrections, m’ont permis d’améliorer ce mémoire.

Ma gratitude va également envers tous les professeurs des départements d’informatique, de mathématiques et de statistiques qui ont grandement contribué à ma formation en me transmettant des connaissances inestimables que je garderai à jamais en mémoire.

Pour finir, j'exprime ma gratitude envers ma famille ainsi que tous ceux qui m'ont soutenu et encouragé tout au long de mon parcours académique, que ce soit de près ou de loin.

## DÉDICACES

Je dédie ce travail :

- À mon défunt père.
- À ma maman, ma vie, qui a été toujours là pour moi.
- À mon défunt ami et frère Aliou Seck.
- À mes frères et sœurs.
- À mes amis du Sénégal et du Canada

# RÉSUMÉ

La signature est un moyen pour une personne de s'identifier et d'approuver le contenu d'un document. Elle est utilisée fréquemment dans des documents officiels et privés tels que les conventions et les chèques, ce qui en fait un élément important dans tous les domaines. Cependant, les signatures peuvent être falsifiées, il est donc nécessaire de protéger les signatures personnelles.

Des systèmes de vérification de signatures ont été développés, basés sur des techniques d'apprentissage automatique, mais ils ne sont pas toujours efficaces, car les caractéristiques utilisées pour la vérification ne révèlent pas toujours des informations cruciales permettant d'identifier une signature authentique.

C'est dans ce cadre que nous adoptons, dans ce mémoire, l'approche des réseaux de neurones siamois pour mettre en place un système performant de vérification de signatures manuscrites statiques. Pour cela, nous faisons appel à l'apprentissage profond (Deep Learning) afin d'apprendre à identifier les signatures authentiques et falsifiées à partir d'une base de données de signatures manuscrites.

Grâce aux recherches que nous avons menées, nous avons mis en place un système de vérification de signature basé sur un réseau de neurones siamois, qui est capable de prédire avec une grande précision si une signature est falsifiée ou pas.



**Mots clés** : Signature, Apprentissage profond, Réseaux de neurones siamois

## ABSTRACT

The signature is a mean for a person to identify themselves and approve the content of a document. It is commonly used in official and private documents such as contracts and checks, making it an important element in various fields. However, signatures can be forged, so it is necessary to protect personal signatures.

Signature verification systems have been developed based on machine learning techniques, but they are not always effective as the features used for verification do not always reveal crucial information for identifying an authentic signature.

It is in this context that we adopt, in this thesis, the approach of siamese neural networks to establish an efficient system for static handwritten signature verification. For this purpose, we utilize deep learning techniques to learn how to identify authentic and forged signatures from a database of handwritten signatures.

Through our research, we have implemented a signature verification system based on a siamese neural network, which is capable of accurately predicting whether a signature is forged or not.



**Key words**: Signature, Deep Learning, Siamese Neural Networks

# Table des matières

## LISTE DES FIGURES

## LISTE DES TABLEAUX

# LISTE DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES

| | |
|---|---|
| AER | Acceptance Error Rate |
| FRR | False Rejection Rate |
| FAR | False Acceptance Rate |
| DTW | Dynamic Time Warping |
| PIN | Personal Identifier Number |
| SVM | Support Vector Machine |
| IA | Intelligence Artificielle |
| IBM | International Business Machines |
| CNN | Convolutional Neural Networks |
| RNN | Recurrent Neural Network |
| ReLU | Rectifier Linear Unit |
| AWS | Amazon Web Services |
| RMSProp | Root Mean Square Propagation |
| HMM | Hidden Markov Models |
| HSV | Handwritten Signature Validation |
| WI | Writer Independent |
| WD | Writer Dependent |
| DT | Dichotomy Transformations |
| HOG | Histogram of Oriented Gradients |
| ICP | Infrastructure à Clé Publique |
| RAM | Random Access Memory |
| GPU | Graphics Processing Unit |
| TPU | Tensor Processing Unit |
| CSV | Comma Separated Values |

# CHAPITRE 1 : INTRODUCTION GÉNÉRALE

## 1.1 Introduction

De nos jours, il existe plusieurs manières qui sont utilisées pour identifier les humains. Ces techniques sont notamment la reconnaissance des yeux, la reconnaissance des visages, la reconnaissance des empreintes digitales, mais on se rend compte que malgré tous les progrès technologiques, la signature reste le moyen le plus utilisé pour authentifier un document, valider un contrat ou une transaction financière [1] [2].

La signature manuscrite d'un individu est un compromis avantageux, car elle est à la fois relativement fiable, facile à acquérir et largement acceptée socialement comme moyen d'identification. Depuis longtemps, la signature a été utilisée comme moyen d'authentification des documents, d'engagement des individus dans des contrats, etc. En conséquence, la signature est largement reconnue comme un moyen de validation lié à l'identité d'une personne. [3].

La signature d'une personne est un tracé unique qui résulte d'un mécanisme complexe propre à cette personne. On suppose que chaque signature d'un individu soit unique et caractérise cette personne. Il est évident que plusieurs facteurs tels que son humeur et sa forme physique peuvent influencer la forme d'une signature. En raison de cela, des variations peuvent apparaître dans les signatures d'une même personne, il est donc nécessaire de prendre en compte ces paramètres pour caractériser une signature. La variabilité au sein et entre les individus, ainsi que le nombre et les types de signatures différentes (liées à des aspects culturels), nécessitent la mise en place de méthodologies pour vérifier l'authenticité des signatures. En conséquence, les systèmes judiciaires et bancaires font appel à des experts pour déterminer l'authenticité d'un document ou d'une signature lors de litiges. Ces experts utilisent des méthodes comme:

- Observation visuelle : Les experts examinent attentivement la signature pour détecter des indices visuels tels que la forme générale, les traits distinctifs, les caractéristiques particulières, les angles, la pression exercée sur le papier, etc. Ils comparent la signature en question à des exemples de signatures connues du même individu ou à d'autres signatures pour détecter les similitudes ou les différences.

- Analyse graphologique: L'analyse graphologique implique l'étude des traits graphiques de la signature, tels que la taille, la forme, la pression, la vitesse, l'inclinaison, l'espace entre les lettres, etc. Les graphologues cherchent à déterminer des caractéristiques spécifiques qui peuvent indiquer l'authenticité ou la falsification de la signature [4].

- Analyse forensique: L'analyse forensique des signatures peut impliquer l'utilisation de diverses techniques scientifiques et technologiques, notamment la comparaison d'images numériques, la photographie infrarouge, la spectroscopie, l'examen des encres et des supports d'écriture, etc. Ces techniques peuvent révéler des altérations, des traces de contrefaçon ou des indications de falsification [5].

- Analyse chronologique : L'analyse chronologique des signatures est basée sur l'idée que la signature d'une personne change généralement avec le temps en raison de l'évolution de son style d'écriture, de son âge, de son état de santé, etc. Les experts comparent des signatures présumées authentiques à différentes périodes de la vie de l'individu pour détecter des variations significatives.

- Analyse statistique : L'analyse statistique utilise des méthodes quantitatives pour évaluer la probabilité que deux signatures proviennent de la même personne. Elle peut impliquer l'utilisation de logiciels spécialisés qui mesurent des paramètres spécifiques des signatures et les comparent pour déterminer leur similarité ou leur dissemblance.

Notre méthode de recherche s'inspire sur cette dernière analyse afin de mener à bien notre travail de recherche en utilisant les réseaux de neurones siamois. En effet, notre approche consiste à évaluer la dissimilitude des signatures en produisant un score de dissimilarité. Ce score permettra de quantifier la différence entre deux signatures et de déterminer si elles sont authentiques ou fausses.

L’appel à des experts est généralement efficace, mais elle est coûteuse en matière de temps et d’expertise. En effet, l’authentification d’une signature peut prendre plusieurs mois et nécessite diverses recherches. Dès lors, les banques ne font appel à une telle expertise qu’en présence d’un doute, pour des chèques impliquant des sommes élevées.

Un système de vérification de signatures statiques performant avec les réseaux de neurones siamois pourrait éliminer ces difficultés. Il permettrait une vérification rapide,

systématique et efficace des signatures, ce qui réduirait les coûts de vérification considérablement avec les risques de contrefaçon.

Les contrefaçons peuvent être classées en trois catégories [6] :

- Faux simples : Le faussaire connait seulement le nom de l'utilisateur et ne voit aucune signature authentique
- Contrefaçons aléatoires : Le faussaire n'a aucune information sur le nom et la signature de l'utilisateur
- Contrefaçons habiles : Le faussaire connait à la fois le nom et la signature authentique

Un problème se pose lorsqu'une personne utilise l'identité d'une autre personne pour commettre une fraude ou une fausse déclaration. Donc, il est important de protéger les signatures personnelles adéquatement afin de garantir l'authenticité d'une signature, car l'œil humain ne peut pas être efficace face à ces situations, car soumis à plusieurs paramètres.

Il existe des systèmes de vérification de signatures manuscrites qui ont été développés récemment. Ces systèmes sont répartis en deux classes suivant le mode d'acquisition de l'image des signatures :

- Le système de vérification statique « Off-Line » : La vérification statique de signatures se concentre sur l'analyse visuelle des signatures manuscrites en utilisant des caractéristiques statiques telles que la forme, la taille, les angles, les traits, etc[3]. Les experts comparent la signature en question à des exemples de signatures connues pour détecter des similitudes ou des différences.

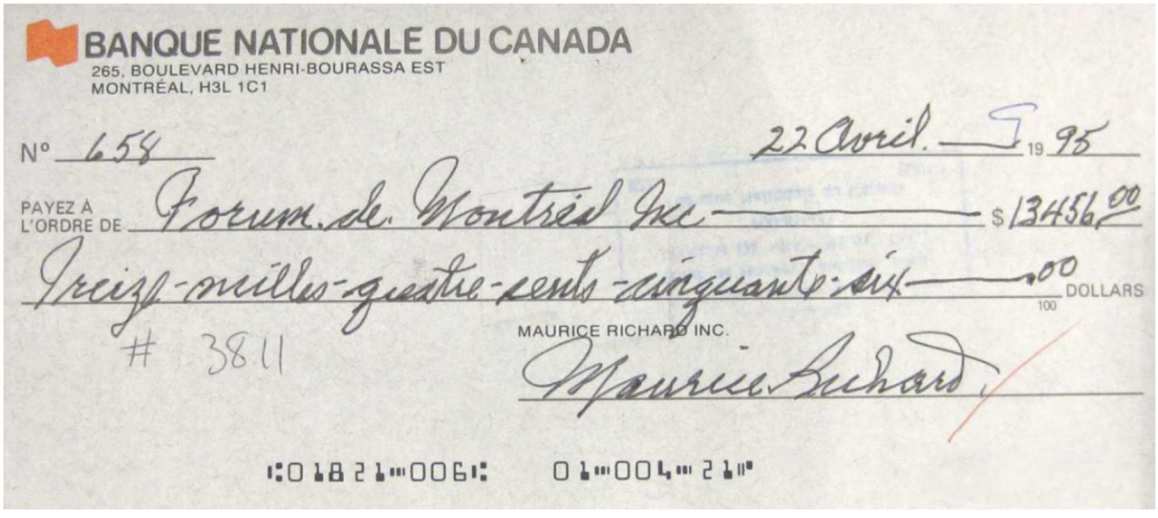


Figure 1 : Exemple de signature hors ligne sur des chèques collectés sur Internet

- Le système de vérification dynamique « On-Line » utilise une tablette électronique et un stylet connecté à un ordinateur pour extraire des informations sur une signature, telles que la pression exercée, la vitesse d'écriture, etc., pour vérifier l'authenticité de la signature.[7]

Figure 2 : Dispositif de capture de signature en ligne

Dans ce mémoire, nous avons choisi d'utiliser la vérification statique en raison des avantages suivants:

- Coût réduit : La vérification statique peut être réalisée sans nécessiter de matériel spécialisé ou coûteux, contrairement à certaines méthodes de vérification dynamique qui peuvent nécessiter l'utilisation de tablettes graphiques ou d'autres dispositifs spécifiques.

- Accessibilité des données : La vérification statique peut être réalisée à partir d'images numériques ou de copies de signatures, ce qui les rend facilement accessibles et utilisables. Il est généralement plus simple d'obtenir des exemples de signatures pour effectuer une comparaison statique plutôt que d'acquérir des données dynamiques spécifiques (comme avec la vérification dynamique).

- Moins sensible aux changements : La vérification statique peut être moins affectée par les variations naturelles des signatures d'une personne au fil du temps. Elle se concentre sur les caractéristiques de base qui sont moins susceptibles de changer considérablement, ce qui peut être un avantage lorsque les signatures à comparer ont été réalisées à différents moments.

La plupart de ces systèmes sont élaborés sur la base des algorithmes d'apprentissage machine et possèdent néanmoins beaucoup de faiblesses comme la sensibilité aux variations, le manque de données d'entrainement qui ne permettent pas de procéder à une vérification efficace. C'est dans cette perspective que nous visons la mise en place d'un système de vérification de signature statique, en explorant l'approche des réseaux de neurones siamois qui permettra d'identifier de manière efficace si une signature est authentique ou falsifiée, tout en fournissant un score de dissimilarité précis.

## 1.2 Problématique

Face à la problématique croissante de la falsification de signatures, les institutions telles que les services juridiques et les banques nécessitent des systèmes informatiques de vérification de signatures efficaces pour protéger les biens des clients et la garantie de leur sécurité. Toutefois, les systèmes actuels présentent souvent des faiblesses telles que la sensibilité à la variabilité, le manque d'échantillons de signatures suffisants, ce qui limite la plupart leur efficacité. Dans cette optique, il est important de mettre en place une solution plus performante en adoptant une approche efficace pour l'application de la vérification de signature, qui est capable d'identifier et de confirmer une signature manuscrite en peu de temps. Comment les réseaux de neurones siamois, en surmontant les faiblesses des systèmes actuels, peuvent-ils être utilisés pour relever ce défi et améliorer l'efficacité et la précision des systèmes de vérification de signature statiques, en fournissant une solution plus fiable?

## 1.3 Objectifs

L'objectif principal de notre mémoire est de mettre en place un système de vérification de signature manuscrite statique performant capable d'identifier une signature authentique ou falsifiée en utilisant les réseaux de neurones siamois. Pour aboutir à cet objectif, on doit procéder comme suit :

- Faire une revue de littérature des systèmes de vérifications de signature existants.
- Examiner tous les aspects liés à la structure des réseaux de neurones ainsi que les modèles mathématiques associés.
- Analyser les langages de programmation et les outils pertinents pour la mise en place de notre modèle.
- Implémenter et réaliser des tests.
- Comparer nos résultats avec les travaux d'autres chercheurs.

## 1.4 Plan du mémoire

Nous organisons le mémoire comme suit :

Dans le chapitre suivant, nous présentons les généralités sur la signature, les définitions et les méthodes utilisées et des systèmes de vérifications de signature existants. Le chapitre 3 sera consacré à la revue de littérature. Dans le quatrième chapitre intitulé cadre de recherche, nous parlerons de l'architecture de notre réseau de neurones ainsi que des outils de développements utilisés. Le chapitre 5 consistera en une présentation de notre méthodologie ainsi que l'ensemble des données utilisées. Nous allons présenter dans le chapitre 6 l'ensemble des résultats que nous avons pu avoir tout au long de ce mémoire, puis une discussion générale de ces résultats et enfin terminer par une comparaison avec les travaux d'autres chercheurs. Une conclusion générale mettra un terme à ce mémoire.

# CHAPITRE 2 : GÉNÉRALITÉS SUR LES SIGNATURES

## 2.1 Introduction

Le but de ce chapitre est de présenter les caractéristiques des signatures et les généralités sur les systèmes de vérification de signatures manuscrites. Nous allons présenter aussi les différents types de signatures manuscrites, les types de faux, l'utilité de la signature manuscrite et les caractéristiques de la signature humaine. Nous allons aussi, dans ce chapitre, parler des techniques de vérification de signatures. Nous concluons ce chapitre en évoquant l'apprentissage en profondeur, une technique qui utilise les réseaux de neurones et qui sera la méthode d'apprentissage que nous utiliserons pour les signatures.

## 2.2 Généralités sur la signature

La signature est un moyen d'identification de l'auteur d'un document, d'une œuvre ou la cause d'un phénomène : ainsi un auteur signe ses écrits [1]. La signature peut également servir à indiquer l'approbation d'un document par une personne qui n'en est pas nécessairement l'auteur. C'est fréquent dans les contrats et autres documents commerciaux qui sont signés par toutes les parties concernées. La signature a donc pour but une identification.

Il existe plusieurs types de signatures, parmi lesquels les plus utilisées sont :

- La signature manuscrite :

Les signatures dans les cultures utilisant un alphabet se présentent généralement sous la forme d'une écriture à la main personnalisée des prénoms et des noms de la personne (bien que l'ordre puisse varier). Cette écriture peut être simplifiée, calligraphiée, dessinée de diverses manières, et associée à des effets de style (traits, courbes, points) qui rendent la signature unique et difficile à reproduire par d'autres.

- La signature numérique

La signature numérique représente une « empreinte digitale » électronique. Grâce à un codage spécial, la signature numérique associe en toute sécurité un signataire à un document dans une transaction enregistrée. La signature numérique utilise aussi un format standard accepté, appelé infrastructure à clé publique (ICP), pour fournir les plus hauts niveaux de sécurité et d'acceptation universelle [8].

Figure 3: Exemple de signature numérique

- La signature par cachet

Dans certaines cultures, comme celles d'Asie, la notion de signature manuscrite n'a pas la même signification qu'en occident. En effet, pour ces cultures, écrire ou signer son nom revient au même, car les caractères utilisés sont les mêmes. Les gens utilisent plus tôt un genre de sceau nominatif où leur nom est écrit avec une graphie de cachet à la place d'une signature manuscrite [1].

- La signature aveugle

Selon le cryptographe américain David Chaum, la signature aveugle est une signature effectuée sur un document qui a été masqué avant d'être signé [9].

- La signature en peinture

En matière de peinture, la signature sert à inclure des informations telles que la date ou d'autres détails, dans une inscription simulée qui ressemble à une gravure sur pierre.

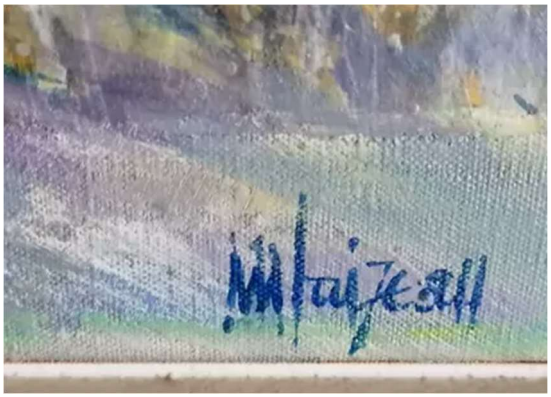

Figure 4: Exemple de signature en peinture

- La signature rythmique

Une signature rythmique, également appelée signature de mesure, est un concept utilisé en musique pour indiquer la structure rythmique d'une composition. Elle est représentée par une fraction placée à la clé de la musique et se compose de deux chiffres : le chiffre supérieur indique le nombre de temps dans une mesure, tandis que le chiffre inférieur indique la valeur de chaque temps. Par exemple, dans une signature rythmique 4/4, il y a quatre temps dans une mesure et chaque temps est une noire. La signature rythmique est essentielle pour les musiciens pour comprendre et interpréter correctement le rythme d'une pièce musicale. [10].

## 2.3 Aperçu de la signature manuscrite

### 2.3.1 Les caractéristiques de la signature humaine.

D'après le dictionnaire « American Heritage », une signature peut être décrite comme « le nom d'une personne écrit de sa propre main ; l'action de signer son nom » [11]. Cette définition suggère que la signature est une image bidimensionnelle statique qui ne contient pas d'informations temporelles.

Les signatures peuvent prendre diverses formes : certaines personnes utilisent leur nom comme signature, d'autres préfèrent utiliser leurs initiales ou des signatures qui ne sont pas directement liées à leur nom [12].

D'après Gubta [12], la similarité entre deux signatures de la même personne pourrait être interprétée comme une tentative de contrefaçon par traçage.

### 2.3.2 Importance de la signature manuscrite

Malgré tous les progrès technologiques, la signature reste le moyen le plus utilisé pour authentifier un document, valider un contrat ou une transaction financière [2]. La signature peut être aussi utilisée comme un mot de passe pour consulter des documents confidentiels, ou tout simplement pour accéder à son bureau. Elle est aussi fiable que l'identification vocale ou rétinienne [13]. L'auteur d'une signature est appelé un signataire. Il ne faut pas confondre une signature avec un autographe. L'autographe est une signature d'un artiste qui est destinée à être exposée au public, tandis que la signature est habituellement gardée confidentielle.

La signature dans de nombreux contrats a pour fonction non seulement de prouver l'identité de la partie contractante, mais aussi de fournir une preuve de la délibération et de l'accord. Cela indique que la partie contractante était présente et a accepté les termes de l'accord. Dans plusieurs pays, les signatures doivent être établies devant un notaire pour

valider tout document juridique. Pour les documents légaux, une personne non instruite peut faire une « marque », tandis qu'une personne instruite signe le même document. Dans certains pays, les personnes analphabètes utilisent leur empreinte digitale sur des documents juridiques plutôt qu'une signature manuscrite [1].

### 2.3.3 Types de signatures manuscrites

On peut identifier différents types de signatures manuscrites, notamment les signatures authentiques et les signatures falsifiées. Les signatures authentiques sont généralement stables, tandis que les signatures falsifiées peuvent varier considérablement selon les compétences du faussaire.

- Signature authentique

Les signatures d'une même personne présentent des variations régulières, notamment en matière de forme, de taille, etc. Divers facteurs tels que le pays d'origine, l'âge, l'humeur, les habitudes, l'état psychologique ou physique de la personne affectent également la signature [14].

- Signature falsifiée

La falsification d'une signature réussie demande un processus complexe pour le faussaire, qui doit non seulement copier les caractéristiques de l'écriture imitée, mais également masquer les caractéristiques personnelles de l'auteur. En réalité, c'est l'excès de similitude entre la signature originale et la copie qui rend cette dernière une contrefaçon. Certains experts en signatures soulignent que si deux signatures d'une même personne écrites sur papier étaient identiques, elles pourraient être considérées comme un faux par copie, et il y'a également des faux très bien réalisés qui peuvent tromper le système.

### 2.3.4 Type de faux

Parmi les types de faux qui existent, nous pouvons citer :

- Les faux par déguisement

Les signatures dissimulées sont des signatures qui sont créées dans des situations où une personne authentique signe des documents dans le but de les renier plus tard [15]. Ces signatures sont produites par des utilisateurs authentiques et ont des similitudes avec les signatures authentiques, mais possèdent également des caractéristiques communes aux contrefaçons.

- Le faux par imitation servile

Dans ce type de faux, le faussaire doit disposer d'un exemplaire de la signature originale [3]. Ce type de faux est caractérisé par sa lenteur et sa complexité, avec des interruptions dans le mouvement de la plume qui révèlent les hésitations et les tentatives du faussaire.

- Les faux par imitation libre

Pour ce type de faux, le faussaire étudie attentivement une signature authentique pour la reproduire de mémoire jusqu'à atteindre un résultat satisfaisant [3]. Les experts considèrent que ce type de falsification est le plus difficile à détecter, car la ressemblance est très forte. Les contrefaçons de ce type se distinguent des originaux par les proportions relatives des éléments de la signature et l'alternance entre les traits pleins et déliés.

- Les faux par calque

Le faux par calque reproduit avec exactitude l'image d'une signature authentique sur un document en utilisant un moyen de reproduction, tel qu'une copie par transparence, avec du carbone ou par photocopie [3]. Ce genre de faux est extrêmement difficile à détecter, même pour les experts.

- Les faux aléatoires

Le faux aléatoire est produit en utilisant la signature du faussaire plutôt que celle de l'individu à imiter. Contrairement au faux simple, le contenu d'un faux aléatoire est clairement différent de celui de l'authentique, ce qui le rend relativement facile à détecter.

- Les faux simples

Le faussaire connait seulement le nom de l'utilisateur et ne voit aucune signature authentique [16].

Pour concrétiser ce que représentent les différents types de faux, la figure 5 présente un échantillon de chaque catégorie.

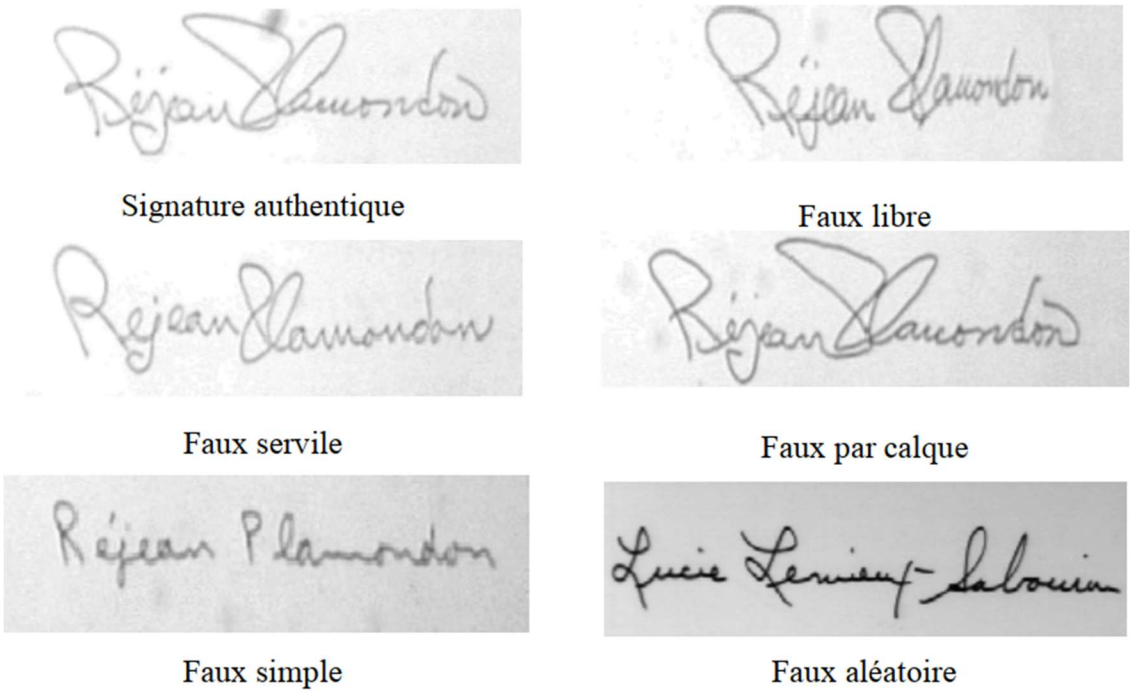


Figure 5 : Les types de faux

## 2.4 Méthodes de vérification de signatures

### 2.4.1 Généralité

Les systèmes automatisés de vérification de signature fonctionnent en prenant une signature, qu'elle soit authentique ou falsifiée, comme entrée, et produisent en sortie le statut de l'image, c'est-à-dire si elle est authentique ou falsifiée. Ils utilisent pour cela des algorithmes d'apprentissage tels que les réseaux de neurones, les machines à vecteurs de support, etc. La figure 6 présente un exemple de cette procédure ;

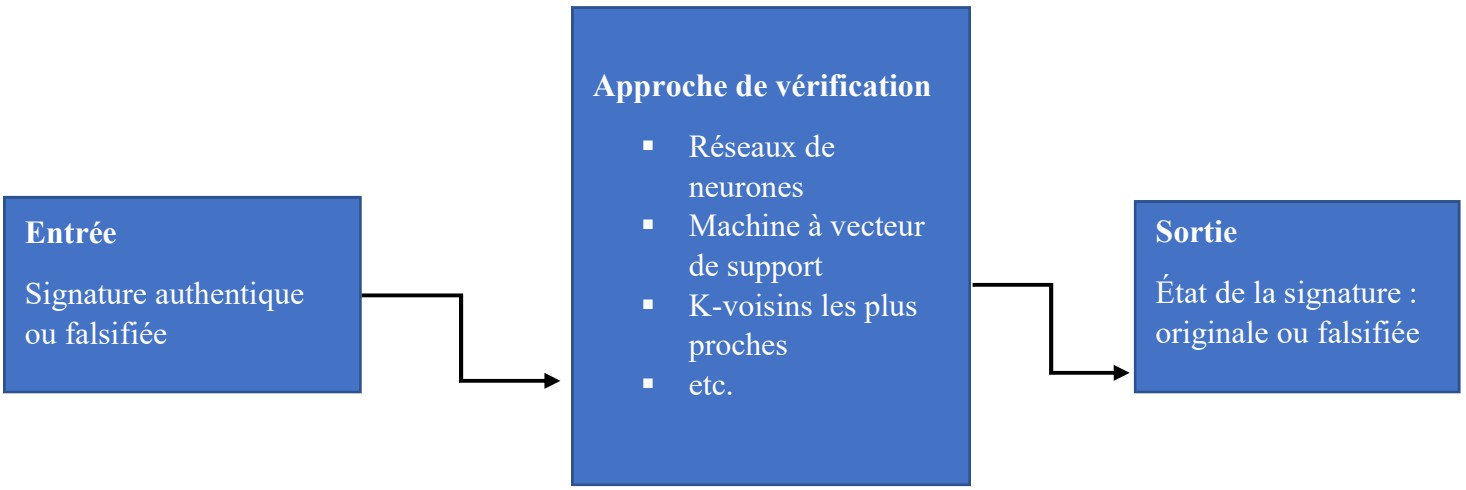


Figure 6 : Schéma fonctionnel d'un système de vérification de signature

Les systèmes de vérification de signatures sont classés en deux catégories : les systèmes de vérification statique et les systèmes de vérification dynamique. Dans l'approche statique, les images de signatures sont capturées par un appareil photo ou un scanneur. Les systèmes de vérification de signatures statiques utilisent des caractéristiques telles que les niveaux de pixels, le centre de masse, le ratio de l'image, etc., extraits de l'image de signature numérisée pour la vérification [1][3]. En revanche, l'approche dynamique implique l'utilisation d'une tablette électronique et d'un stylet, le tout connecté à un ordinateur pour extraire des informations sur la signature. Elle prend en compte des informations dynamiques telles que la pression, le temps et la vitesse d'écriture à des fins de vérification.

### 2.4.2 Système de vérification statique

La vérification de signature statique est la première approche utilisée pour résoudre le problème de la vérification de signature. Elle permet de distinguer les signatures authentiques des signatures falsifiées en utilisant des images statiques. Pour ce faire, seules des caractéristiques basées sur la forme de la signature sont extraites, sans aucune connaissance sur le processus de signature. [17]. La vérification de signature statique présente des avantages comme :

- Applications étendues : La vérification de signature statique est utilisée dans divers domaines tels que la banque, les services juridiques, l'administration gouvernementale et la sécurité pour garantir l'authenticité des documents et des transactions.
- Réduction de la fraude : En détectant les signatures falsifiées avec précision, la vérification de signature statique contribue à réduire les cas de fraude et d'utilisation abusive de signatures.
- Amélioration de l'efficacité : La vérification de signature statique permet de réduire le temps et les ressources nécessaires pour vérifier manuellement les signatures.

La vérification de signature statique se réalise sans contrôle électronique en utilisant des images de signatures capturées par un scanneur ou une caméra. Les caractéristiques extraites de l'image de la signature numérisée sont utilisées pour vérifier l'authenticité, et peuvent être classées en deux catégories principales :

- Les caractéristiques globales sont faciles à extraire et incluent généralement la zone de signature, le centre de gravité, etc.

- Les caractéristiques locales sont obtenues à partir de petites sections de l'image de la signature. Elles sont plus précises que les caractéristiques globales et incluent la

densité de pixels locaux, les caractéristiques des zones inclinées, les points critiques, etc.

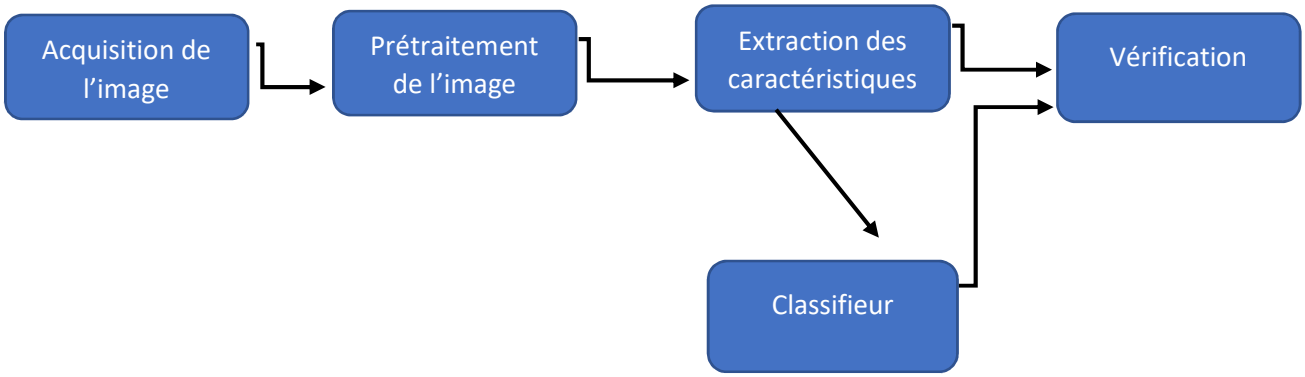


Figure 7 : Présentation générale d'un système de vérification de signature statique

**Acquisition de l'image** : Il est nécessaire que les signatures à vérifier soient sous forme numérique. Elles peuvent être obtenues de diverses manières, telles que la capture photographique ou la numérisation. Les signatures sont converties en format numérique, puis sauvegardées en tant qu'images pour une utilisation ultérieure.

**Prétraitement de l'image** : L'objectif du prétraitement des signatures est de les normaliser en vue de faciliter l'extraction des caractéristiques. Cette étape est indispensable pour améliorer la précision de l'extraction et de la vérification des caractéristiques. Avant de procéder à l'extraction des caractéristiques à partir de l'image, divers algorithmes de traitement d'image sont appliqués à l'image numérisée, notamment la binarisation et la réduction du bruit.

**Extractions des caractéristiques** : La performance d'un système de vérification de signatures repose principalement sur l'étape d'extraction des caractéristiques. Les techniques utilisées pour extraire les caractéristiques doivent être à la fois rapides et faciles à calculer pour optimiser le système et minimiser la puissance de calcul requise. Les caractéristiques sélectionnées doivent être suffisamment discriminantes pour différencier une signature authentique d'une signature contrefaite. Les caractéristiques extraites pour la vérification de la signature statique peuvent être classées en caractéristiques globales, locales et géométriques.

**Vérification** : Au cours de la phase de vérification, les caractéristiques des signatures de test sont comparées à celles des signatures authentiques à l'aide de différentes techniques de classification de motifs. Cette étape permet de déterminer si la signature est authentique ou falsifiée. La décision définitive est prise en fonction des résultats de la comparaison.

**Classifieur** : Son rôle est de classer l'état de la signature : soit authentique ou falsifiée

L'étude de la vérification de signatures manuscrites utilise plusieurs technologies, en particulier pour la méthode statique. Ce domaine est en constante évolution. Les méthodes utilisées par les chercheurs varient selon les caractéristiques extraites, la méthode de formation des données, ainsi que le modèle de classification et de vérification choisi.

### 2.4.3 Vérification de signature : exemples d'applications

La méthode de vérification de signature statique peut être utilisée dans de nombreux cas :

- Les transactions financières : La signature est souvent considérée comme le moyen le plus pratique d'authentification, mais les cas de falsification ont entraîné une augmentation récente des pertes financières. Selon A. Khomiakov [18], la fraude aux cartes de crédit entraîne des pertes annuelles évaluées à 450 millions de dollars pour MasterCard.

- Authentification de documents : La signature manuscrite statique peut être utilisée pour authentifier des documents tels que des contrats, des accords, des chèques ou des formulaires. La comparaison entre la signature présente sur le document et une signature de référence peut aider à déterminer l'authenticité du document.

- Accès sécurisé : La signature manuscrite statique peut servir de méthode d'identification pour accéder à des espaces sécurisés tels que les systèmes informatiques, les salles de données ou les coffres-forts. La comparaison de la signature avec celle enregistrée dans le système peut permettre de vérifier l'identité de l'individu.

- Contrôle d'accès physique : Dans les environnements sécurisés, tels que les entreprises, les centres de données ou les installations gouvernementales, la signature manuscrite statique peut être utilisée pour autoriser l'accès physique à des zones restreintes. La comparaison de la signature avec celle enregistrée dans le système peut permettre de contrôler les entrées et les sorties.

- Contrôle des présences : Dans les contextes tels que les établissements scolaires, les entreprises ou les événements, la signature manuscrite statique peut être utilisée pour enregistrer la présence des individus. Les personnes peuvent signer un registre ou un document pour attester de leur présence à un endroit ou à un événement spécifique.

## 2.5 Les réseaux de neurones

### 2.5.1 Machine Learning (Apprentissage automatique) : Généralités

L'apprentissage automatique ou le Machine Learning est un sous domaine de l'intelligence artificielle. Le Machine Learning est au centre du métier de Data Scientist, il permet aux algorithmes de s'adapter et d'améliorer leur performance en analysant les données qu'ils reçoivent. D'après Laure Audubon [19], « Nous sommes en interaction constante avec des applications de Machine Learning, que ce soit lorsque nous utilisons les réseaux sociaux, interagissons avec un chatbot, ou lorsque nous consultons les moteurs de recommandations. »

#### 2.5.1.1 Historique du Machine Learning

En 1959, Arthur Samuel a utilisé le terme « Machine Learning » pour la première fois pour son programme de jeu de dames créé en 1952, qui avait la capacité d'apprendre au fil des parties. Bien que les débuts du Machine Learning soient souvent associés à l'intelligence artificielle, cela remonte à la création du test de Turing en 1950.

En 1957, Frank Rosenblatt a développé le « perceptron », le premier réseau de neurones capable de classifier des données binaires. Cependant, les résultats insuffisants de ses travaux ont provoqué une crise dans le domaine de l'IA, appelé « l'hiver de l'IA », qui a entraîné une interruption des recherches sur l'intelligence artificielle et le Machine Learning.

Au cours des années 1990, l'intérêt pour l'IA et le Machine Learning a été renouvelé, en grande partie grâce à l'émergence d'Internet, qui a offert aux chercheurs de nouvelles opportunités d'interaction et un accès à des quantités importantes de données nécessaires pour des applications de Machine Learning efficaces.

#### 2.5.1.2 But du Machine Learning

Le Machine Learning fait partie de l'IA, mais il n'est pas obligatoire dans tous les projets d'IA, même si la plupart des modèles d'IA actuels l'utilisent.

L'objectif de l'intelligence artificielle (IA) est de reproduire de manière logique le comportement humain, ordonnée et raisonnée, mais le Machine Learning permet à l'application d'apprendre et de répondre de manière adaptative, appelée Intelligence Artificielle neuronale [19]. Cette forme d'IA forte nécessite une grande quantité de données

pour s'entraîner, apprendre et se développer, ce qui constitue la base du Machine Learning. Cependant, l'exploitation de grandes quantités de données n'est pas facile, car le cerveau humain n'est pas capable de les traiter efficacement. Le Machine Learning intervient en permettant de détecter et d'analyser les modèles cachés dans ces grandes quantités de données. Cela aide les entreprises à mieux comprendre leurs données et à relever les défis de leur marché comme les prévisions et analyses prédictives, la détection des fraudes grâce à la communication machine à machine [20].

### 2.5.1.3 Type d'apprentissage automatique

La figure ci-dessous illustre les différents types d'algorithmes de Machine Learning :

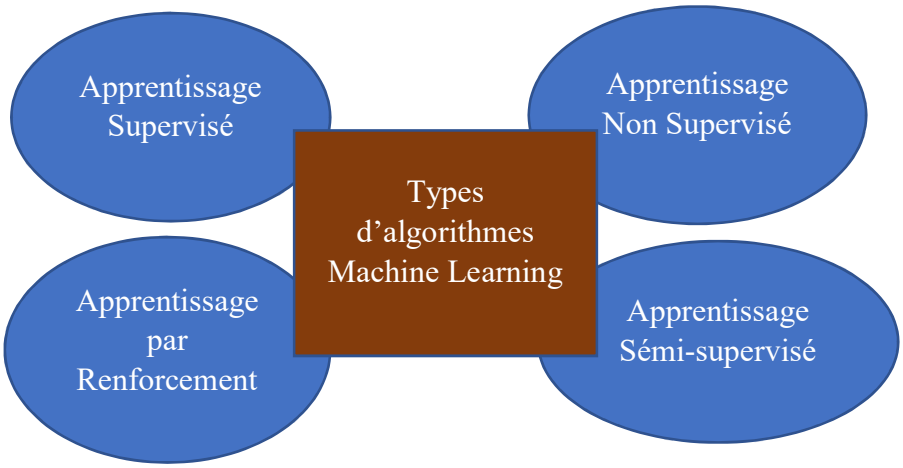


Figure 8 : Les différents types d'apprentissage automatique.

- Apprentissage non supervisé

Le Feature Learning, également appelé apprentissage non supervisé, consiste à classer les données en fonction de modèles historiques sans avoir une connaissance préalable des résultats souhaités, contrairement au Machine Learning supervisé. Dans le cadre de la détection de fraudes, par exemple, l'ordinateur apprend à identifier les comportements frauduleux enregistrés précédemment par l'entreprise.

On distingue deux types d'apprentissages non supervisés :

– **Clustering** : L'ordinateur a pour but de créer des groupes de données.

– **Association de données** : l'ordinateur a pour objectif d'identifier les règles permettant de créer des groupes de données [19].

Le clustering et l'association de données visent à découvrir des structures et des relations dans les données sans l'utilisation de labels préalables. Ils peuvent être utilisés individuellement ou de manière complémentaire en fonction des objectifs et des caractéristiques spécifiques des données

- Apprentissage supervisé

L'apprentissage supervisé implique l'utilisation d'une base de données d'apprentissage pour prédire des résultats. Les données sont étiquetées et les classes sont déjà connues, le système doit simplement déterminer à quelle catégorie appartient le nouvel élément.
On distingue deux types d'apprentissages supervisés :

– **Classification** : la sortie attendue est une variable qualitative.
(ex. : chat ou chien)

– **Régression** : la sortie attendue est une variable quantitative.
(ex. : prix d'une maison)

- Apprentissage par renforcement

On parle d'apprentissage par renforcement, lorsque l'agent interagit avec son environnement en effectuant des actions pour recevoir des récompenses. La récompense peut être négative, positive, ou nulle. Si l'on prend par exemple, un robot qui joue au jeu de Tetris, quand il gagne il reçoit une récompense positive, et dans le cas contraire une récompense négative.

- Apprentissage semi-supervisé

L'apprentissage semi-supervisé est une alternative entre l'apprentissage supervisé et non supervisé. Il y a des cas où certaines données sont étiquetées, mais pas toutes les données. On peut utiliser l'apprentissage non supervisé pour segmenter une grande quantité de données en deux clusters. On pourra donc labelliser ces données en fonction des caractéristiques de chaque cluster. Une fois les données labellisées, l'apprentissage supervisé pourra servir pour la classification [20].

La figure 9 donne quelques exemples d'utilisation du Machine Learning :

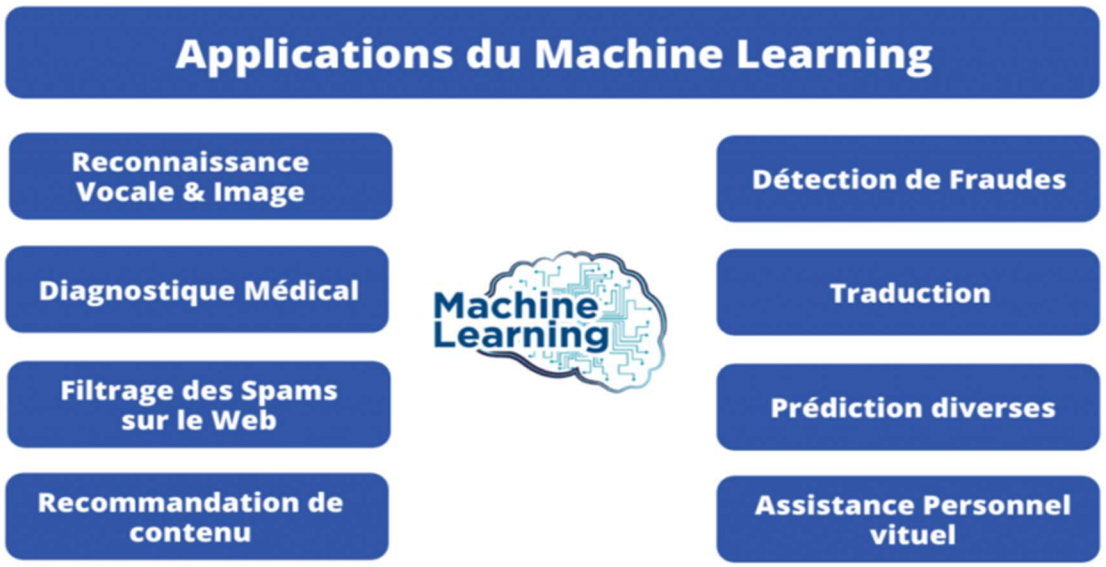


Figure 9 : Illustrations de l'application du Machine Learning

### 2.5.2 Deep Learning (Apprentissage en profondeur) : Généralités

#### 2.5.2.1 Historique du Deep Learning

En 1943, le premier modèle de « neurone formel » a été proposé par Warren McCulloch et Walter Pitts. Il s'agissait d'un neurone binaire, dont la sortie était soit 0 soit 1, ce qui a été la première représentation informatique du cerveau humain. Ensuite, avec l'introduction du « perceptron » en 1957, on a commencé à utiliser le terme « réseau de neurones artificiels » [21].

Après l'hiver de l'IA en 1973, les années 1980 ont vu des avancées notables telles que le « perceptron multicouche ». Le terme « Deep Learning » a été popularisé par Yann LeCun, chercheur français, grâce à sa technique de réseaux de neurones convolutifs appartenant à la classe des réseaux de neurones artificiels, qui s'inspirent du cortex visuel des animaux. Cependant, cette technologie nécessite une grande puissance de calcul pour entraîner les modèles, ce qui a entraîné une perte d'intérêt pour cette technologie.

C'est en 2012, à l'occasion du concours ImageNet Large Scale Visual Recognition Challenge, un défi de recherches informatiques organisé par l'université de Stanford, que le Deep Learning fait son grand retour, grâce au programme vainqueur qui s'appuie sur des techniques d'apprentissage profond [21].

De nos jours, les entreprises font principalement appel aux techniques d'apprentissage en profondeur, que ce soit pour l'analyse d'images, de textes, de sons, ou autres domaines d'application.

#### 2.5.2.2 C'est quoi le Deep Learning ?

L'apprentissage en profondeur est un sous-ensemble de l'apprentissage automatique, qui est essentiellement un réseau de neurones à 3 couches ou plus. Selon la grande société multinationale américaine IBM [22], « Ces réseaux de neurones tentent de simuler le comportement du cerveau humain, bien que loin de correspondre à ses capacités, lui permettant d'apprendre à partir de grandes quantités de données. Alors qu'un réseau de neurones avec une seule couche peut toujours faire des prédictions approximatives, des couches cachées supplémentaires peuvent aider à optimiser et à affiner la précision ».

L'utilisation de l'apprentissage en profondeur est à l'origine de nombreuses applications et services d'intelligence artificielle (IA), qui permettent une automatisation des tâches analytiques et physiques sans intervention humaine. Cette technologie soutient des produits et services courants tels que les assistants numériques, les télécommandes de télévision à commande vocale, ainsi que des technologies émergentes comme les voitures autonomes et la détection de fraudes par cartes de crédit.

#### 2.5.2.3 Fonctionnement du Deep Learning

Les réseaux de neurones d'apprentissage en profondeur sont conçus pour reproduire le fonctionnement du cerveau humain en utilisant une combinaison de données d'entrée, de pondérations et de biais. Ces éléments travaillent ensemble pour identifier, classer et décrire avec précision les objets présents dans les données.

Ces réseaux se composent de plusieurs couches de nœuds interconnectés. Chaque couche utilise les résultats de la couche précédente pour améliorer et optimiser la prédiction ou la classification, dans un processus appelé propagation avant. Les couches d'entrée et de sortie sont appelées couches visibles, où la couche d'entrée reçoit les données à traiter et la couche de sortie produit la prédiction ou la classification finale.

Pour entraîner le modèle, un processus appelé rétropropagation utilise des algorithmes comme la descente de gradient pour évaluer les erreurs dans les prévisions et ajuster les poids et les biais de la fonction. La propagation avant et la rétropropagation travaillent ensemble pour permettre au réseau neuronal de faire des prédictions et de corriger les erreurs en conséquence, devenant de plus en plus précis au fil du temps.

Il existe différents types de réseaux de neurones pour résoudre des problèmes :

- Les réseaux de neurones convolutifs (CNN) sont largement employés dans le domaine de la vision informatique et de la classification d'images. Ils sont capables de détecter des modèles et des caractéristiques dans une image, facilitant ainsi des tâches telles que la détection d'objets ou leur reconnaissance. En 2015, un CNN a surpassé un être humain pour la première fois dans un défi de reconnaissance d'objets [21].

- Les réseaux de neurones récurrents (RNN) sont utilisés généralement dans les applications de langage naturel et de reconnaissance vocale, car ils exploitent des données séquentielles ou chronologiques.

#### 2.5.2.4 Applications du Deep Learning

Parmi les exemples d'applications du Deep Learning, on peut citer :

- **La santé** : Le domaine de la santé a considérablement bénéficié du Deep Learning, notamment à travers la numérisation des dossiers médicaux et des images. Les algorithmes de reconnaissance d'images peuvent servir aux spécialistes en imagerie médicale et les radiologues, en leur permettant d'analyser et d'évaluer un plus grand nombre d'images en un temps plus court [21].

- **La sécurité informatique** : Le Deep Learning peut offrir une détection proactive des spams dans les courriels, de fraude et d'intrusion.

- **Les forces de l'ordre** : Dans le secteur de la sécurité, on peut tirer parti des algorithmes d'apprentissage en profondeur pour détecter des activités potentiellement frauduleuses ou criminelles. Les techniques de reconnaissance vocale, de vision par ordinateur et d'autres applications d'apprentissage en profondeur peuvent améliorer les processus d'investigation en extrayant des modèles et des preuves à partir de données audio et vidéo, d'images et de documents. Cela aide les forces de l'ordre à analyser plus rapidement et avec plus de précision des grandes quantités de données.

- **Les services clients :** Le Deep Learning est de plus en plus fréquent pour les processus de service client dans les organisations. Les chatbots, qui peuvent être utilisés dans diverses applications et services, sont une forme simple d'IA. Les chatbots traditionnels basés sur le langage naturel ou la reconnaissance visuelle sont couramment utilisés dans les centres d'appels. Cependant, les solutions plus avancées de chatbots utilisent l'apprentissage pour déterminer les différentes réponses possibles à des questions complexes. En fonction des réponses reçues, le chatbot essaie alors de fournir une réponse immédiate ou de transférer la

conversation à un représentant humain [21]. Les assistants virtuels comme Siri d'Apple ou Google Assistant étendent l'idée d'un chatbot en activant la fonctionnalité de reconnaissance vocale. Cela crée une nouvelle méthode pour engager les utilisateurs de manière personnalisée.

Toutefois certaines applications du Deep Learning peuvent poser des menaces à la démocratie et aux droits individuels. Ci-dessous quelques exemples de ces menaces :

- Surveillance de masse : Les systèmes de surveillance basés sur le Deep Learning, tels que la reconnaissance faciale, peuvent permettre une surveillance de masse sans précédent. Cela peut compromettre le droit à la vie privée et la liberté individuelle des citoyens.

- Discrimination algorithmique : Les modèles de Deep Learning peuvent être influencés par des biais inconscients présents dans les données d'entraînement, ce qui peut entraîner une discrimination injuste dans les décisions automatisées, comme la sélection des candidats pour un emploi ou le profilage des individus.

- Manipulation de l'opinion publique : Les algorithmes de Deep Learning peuvent être utilisés pour manipuler l'opinion publique en diffusant des informations biaisées ou en ciblant des messages personnalisés. Cela peut affecter la libre circulation des idées et la formation d'une opinion publique éclairée.

- Automatisation des décisions politiques : L'utilisation du Deep Learning pour prendre des décisions politiques, telles que la prédiction des résultats électoraux ou l'élaboration de politiques publiques, peut entraîner une perte de contrôle démocratique et de responsabilité. Les décisions basées sur des modèles de Deep Learning peuvent être difficiles à comprendre et à remettre en question.

- Manipulation des médias : Les techniques de Deep Learning peuvent être utilisées pour créer des médias synthétiques, tels que des vidéos ou des images falsifiées, qui peuvent être utilisées à des fins de désinformation ou de manipulation.

Ainsi, il est essentiel de mettre en place des réglementations et des mécanismes de contrôle adéquats afin de réduire ces risques et de veiller à ce que les applications du Deep Learning respectent les principes démocratiques fondamentaux ainsi que les droits individuels. Cela requiert la transparence des algorithmes, la préservation de la vie privée, la lutte contre la discrimination, ainsi que la promotion de la responsabilité et de la participation du public dans les décisions automatisées.

## 2.6 Définitions de quelques algorithmes d'apprentissages

Il existe de nombreux algorithmes d'apprentissage supervisés et non supervisé, et chacun ont une méthode distincte pour l'apprentissage. Il n'y a pas de méthode ou de solution unique. La détermination de l'algorithme approprié nécessite beaucoup d'essais et d'erreurs. Même les experts en science des données ne peuvent pas prédire si un algorithme fonctionnera avant de le tester. Cependant, la sélection de l'algorithme dépend également de la quantité et du type de données à traiter, des résultats souhaités et de l'utilisation finale des informations. Parmi ces algorithmes, nous avons :

### 2.6.1 SVM (Support Vector Machine)

Les machines à vecteurs de support sont considérées comme des classificateurs, ce qui signifie qu'elles peuvent être utilisées pour séparer un groupe d'éléments d'un autre groupe. Plus concrètement, les SVM sont un ensemble de techniques d'apprentissage supervisé qui ont pour objectif de trouver, dans un espace de dimension N>1, l'hyperplan qui divise au mieux un jeu de donnée en deux. Les SVM sont des séparateurs linéaires, c'est-à-dire que la frontière séparant les classes est un hyperplan [23]. La marge entre les classes fait référence à la distance entre l'hyperplan de séparation et les exemples de chaque classe les plus proches. Dans un problème de classification binaire, la marge est la distance minimale entre l'hyperplan et les exemples de chaque classe qui se trouvent le plus près de l'hyperplan. Une marge plus grande indique une meilleure séparation entre les classes et peut être considérée comme une mesure de la robustesse du modèle de classification.

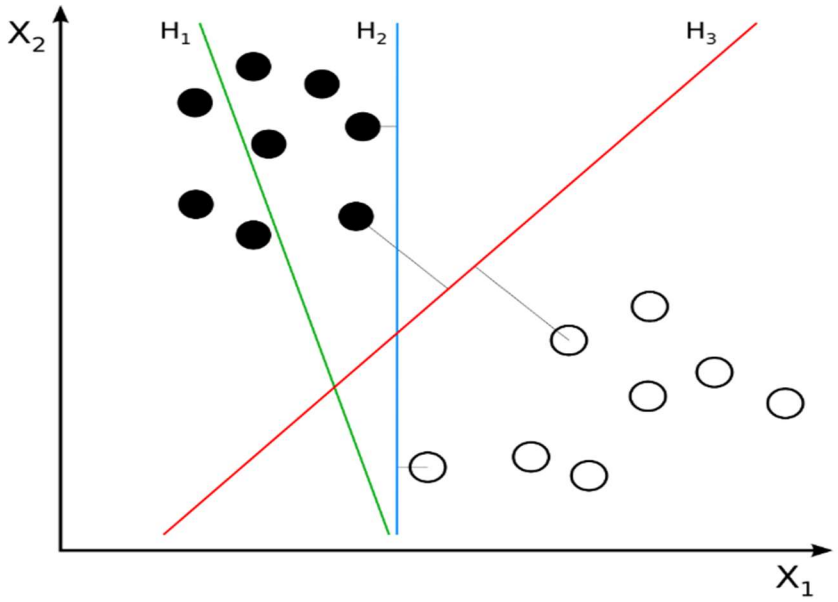


Figure 10 : Exemple illustratif de différentes séparations en deux classes

Dans la figure 10, ci-dessus nous avons un exemple d'hyperplan séparateur pour N=2. H1 ne sépare pas correctement le jeu de donnée ; H2 le sépare bien, mais pas de façon optimale ; H3 sépare le jeu de donnée avec la marge maximale.

Les SVM (Support Vector Machines) sont considérés comme des outils très performants dans l'apprentissage automatique, particulièrement pour des cas avec une base de données de petite ou moyenne taille. Il existe deux principales catégories de SVM : les SVM linéaires et les SVM non linéaires :

- Les SVM linéaires utilisent un hyperplan de séparation linéaire pour séparer les classes.
- Les SVM non linéaires utilisent des fonctions non linéaires pour créer des frontières de décision courbes ou surfaces, permettant ainsi de séparer les classes.
- Les SVM à marge maximale sont une sous-catégorie de SVM non linéaire qui cherchent à maximiser la marge entre les classes. La marge représente la distance entre la frontière de décision (l'hyperplan séparateur) et les échantillons les plus proches de chaque classe.

### 2.6.2 Les K plus proches voisins

Selon la grande entreprise américaine IBM : « L'algorithme des k plus proches voisins, également connu sous le nom de KNN ou k-NN, est un discriminant d'apprentissage supervisé non paramétrique, qui utilise la proximité pour effectuer des classifications ou des prédictions sur le regroupement d'un point de données individuel. Bien qu'il puisse être utilisé pour des problèmes de régression, il est généralement utilisé comme algorithme de classification, en partant de l'hypothèse, des caractéristiques similaires peuvent être trouvés les uns à côté des autres. » [24].

Cette approche peut exiger d'importantes capacités de stockage ou de mémoire pour les données, mais nécessite uniquement des calculs lorsqu'une prédiction est requise. Les données d'apprentissage peuvent être mises à jour et gérées au fil du temps afin de maintenir la précision des prévisions. Toutefois, la mesure de distance ou de proximité peut devenir inefficace pour de très grandes dimensions, c'est-à-dire lorsque le nombre de variables d'entrée est élevé, ce qui peut affecter les performances de l'algorithme. Ce phénomène est connu sous le nom de « malédiction de la dimensionnalité ». Il est donc recommandé de n'utiliser que les variables d'entrée les plus pertinentes pour prédire la variable de sortie.

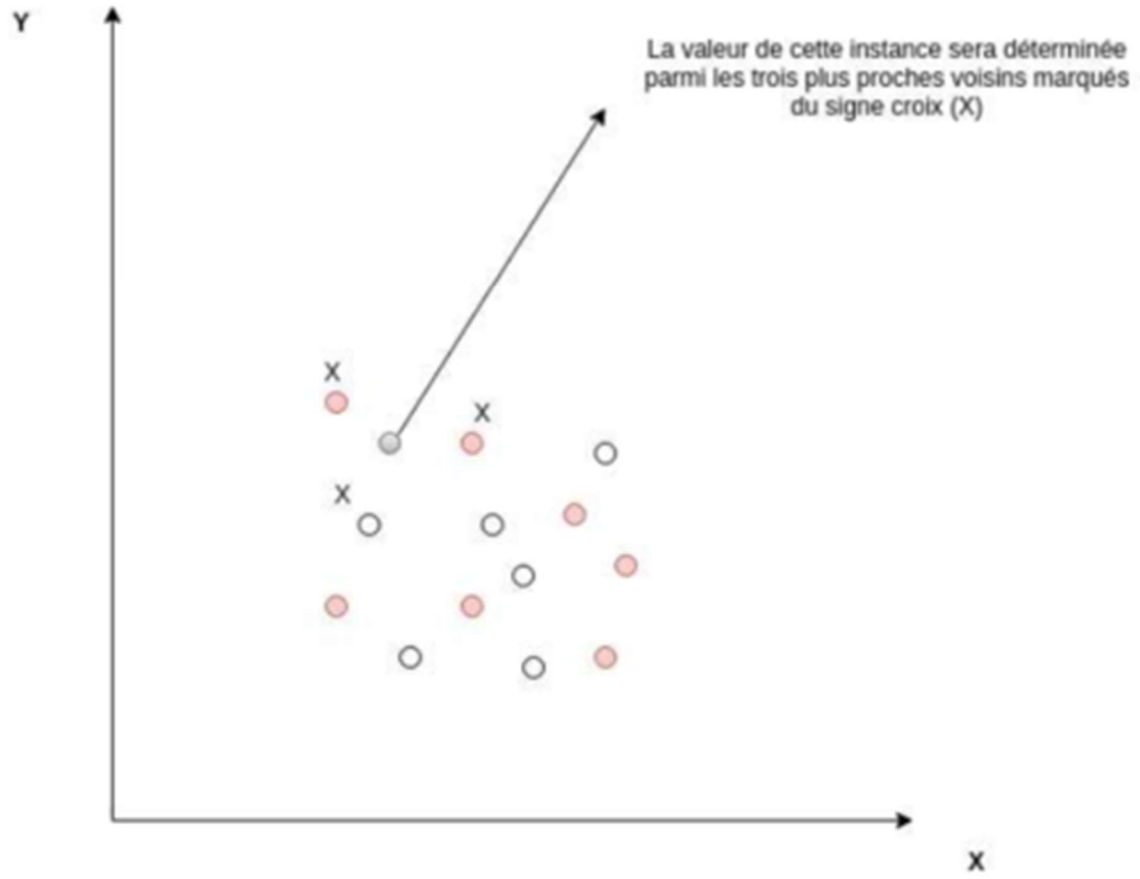


Figure 11 : Illustration de 3 voisins les plus proches

## 2.7 Conclusion

Au cours de ce chapitre, nous avons constaté que chaque personne a une signature manuscrite unique, mais qu'elle peut être contrefaite (falsifiée). C'est pourquoi des systèmes de vérification ont été développés, utilisant soit des images statiques numérisées, soit des informations dynamiques collectées pendant la signature, tels que le temps, la pression, la vitesse, etc. Nous avons également discuté des concepts généraux de l'apprentissage automatique et en profondeur, de leur mode de fonctionnement et de leurs exemples d'applications courantes. Dans le chapitre suivant, nous allons présenter quelques travaux de recherches portant sur la vérification de signature.

# CHAPITRE 3 : REVUE DE LITTÉRATURE

## 3.1 Introduction

Dans ce chapitre, nous allons présenter la revue de la littérature qui recense des travaux qui sont liés à ce travail. Nous allons récapituler les contributions qui ont été faites sur la vérification de la signature manuscrite en utilisant la méthode statique.

## 3.2 Système de vérification statique

La vérification de signature manuscrite est une tâche cruciale dans le domaine de la biométrie et de la sécurité des documents. Elle vise à distinguer les signatures authentiques des signatures falsifiées, ce qui donne une importance capitale dans de nombreux domaines tels que la banque, les transactions juridiques et l'identification personnelle. Au fil des années, de nombreuses approches ont été développées pour aborder ce problème, allant des méthodes basées sur des caractéristiques manuelles aux techniques plus avancées utilisant l'apprentissage automatique.

Parmi ces dernières, l'utilisation des réseaux de neurones siamois a suscité un intérêt croissant en raison de leur capacité à capturer les similitudes et les différences subtiles entre les signatures. Cette revue de littérature se concentre sur l'examen des travaux antérieurs qui ont utilisé différentes approches comme les SVM, KNN, CNN ou réseaux de neurones siamois pour la vérification de signature manuscrite. L'accent sera mis sur l'évaluation des performances de ces méthodes, permettant ainsi de mieux comprendre leur efficacité dans la détection des signatures authentiques et falsifiées.

Bien que les méthodes d'apprentissage en profondeur soient maintenant privilégiées par rapport aux méthodes basées sur des caractéristiques créées manuellement, ces dernières ont été étudiées dans des recherches antérieures dans ce domaine. Par exemple, Cavalcanti et al [25] ont utilisé trois méthodes (structurelle, invariante et pseudo dynamique) pour extraire des caractéristiques et ont choisi le sous-ensemble de caractéristiques ayant le résultat le plus élevé selon les classificateurs Bayes et K-NN.

Oliveira et al [26] ont également étudié les signatures de chèques bancaires et ont découvert que les caractéristiques pseudos dynamiques ainsi que les caractéristiques statiques sont utiles. L'histogramme des gradients orientés (HOG) est une méthode populaire en raison de sa capacité à détecter les contours et sa résistance au bruit et aux transformations dans les images, quelle que soit leur forme. C'est une technique utilisée en vision par ordinateur pour la description et la représentation des caractéristiques visuelles d'une image.

Plus tard, Zhang [27] et Harfiya et al [28] ont utilisé le Pyramid Histogram of Oriented Gradients, une extension de HOG. Certains ont également examiné la question sous un angle mathématique, fractal ou graphométrique.

Erick [1], de son côté, utilise 75 % des signatures pour l'entraînement, soit 30 signatures par utilisateur, et 25 % pour les tests, soit 10 signatures par utilisateur. Cela donne un total de 1200 signatures pour l'entraînement et 400 pour les tests pour tous les utilisateurs. Les résultats obtenus indiquent une précision de 87% pour les SVM et 80% pour les K-voisins les plus proches.

Blumenstein et al. [29] utilisent deux méthodes de classification et de vérification, à savoir le réseau de neurones à propagation résiliente et la fonction de base radiale, ont été comparées à l'aide d'une base de données de 2106 signatures, comprenant 936 signatures authentiques et 1170 signatures falsifiées. Les résultats ont montré des taux de précision de 91,21 % pour le premier et de 88 % pour le second. Un autre article décrit un modèle de vérification basé sur un classifieur de réseau neuronal. Les signatures de la base de données ont été prétraitées pour extraire les caractéristiques de l'image de signature, qui ont ensuite été utilisées pour former le réseau de neurones. L'étude a enregistré un taux de 82,66 % de signatures correctement classées sur un ensemble de 300 signatures [18].

B. Herbst.J et al [30], quant à eux, utilisent le modèle de Markov caché pour proposer un système de vérification de signatures basé sur des caractéristiques globales. La méthode calcule une transformée aléatoire discrète appelée « sinographe » pour chaque image de signature binaire dans une plage de 0 à 360, qui reflète la fonction du nombre total de pixels et de l'intensité par pixel. Ce système a obtenu un taux d'erreur d'acceptation (AER) de 18,4 % pour un ensemble de 440 signatures authentiques de 32 auteurs avec 132 contrefaçons.

T.S. Enturk et al [31] utilisent l'algorithme de machine à vecteur de support (SVM) pour la classification et la vérification à l'aide de caractéristiques globales, directionnelles et de grille de signature. Les résultats obtenus à partir d'une base de données de 1320 signatures ont montré un taux de fausse rejet (FRR) de 2 % et un taux de fausse acceptation (FAR) de 11 %.

Brittany Cozzens et al[32] présentent une méthode pour l'analyse de signatures de chèques à l'aide de réseaux de neurones convolutifs (CNN). L'architecture du CNN est simple et utilise la bibliothèque Keras en Python, basée sur TensorFlow, pour améliorer la qualité de l'image de la signature. La méthode utilise un système de comparaison d'images qui repose sur un système de classification d'images. Lorsqu'une signature est soumise au programme, elle est comparée à d'autres caractéristiques de signature portant la même étiquette. La principale contribution de cette méthode est la détection et la réduction de la

contrefaçon, en particulier dans le secteur bancaire. Ils avaient obtenu un taux de réussite de 83,93%. Cependant les CNN présentent des lacunes. Par exemple, si on considère un exemple classification de visage pour une entreprise qui a 5 employés, La classification du visage à l'aide d'un réseau de neurones convolutif (CNN) conventionnel serait généralement basée sur 5 probabilités de sortie. La formation d'un tel réseau nécessite de disposer de nombreux échantillons de chaque personne. Cependant, un problème se pose lorsque nous souhaitons ajouter une nouvelle classe, telle que le visage d'un nouvel employé. Dans ce cas, il serait nécessaire d'obtenir un grand nombre d'images pour cette nouvelle classe et de ré-entraîner l'ensemble du modèle à nouveau. Cela entraine beaucoup de frais financiers.

Jivesh Poddar et al[33] ont proposé une méthode pour la vérification de signatures. Cette méthode comprend un prétraitement pour faciliter la vérification de signature, l'utilisation d'un réseau de neurones convolutifs (CNN) et d'un modèle basé sur l'algorithme Crest-Trough pour le système de vérification de signature, ainsi qu'un modèle basé sur Harris et Surf pour la détection de la contrefaçon dans la signature. Leur système proposé a atteint une précision de 85-89 % pour la détection de falsification.

En 2022, Abolfazl [6] a utilisé les réseaux de neurones siamois pour développer un système de classification de signatures statique. Son approche repose sur la comparaison à N voies, où l'image de la requête est comparée avec N autres classes, y compris ses propres échantillons de classes. Au cours de cette comparaison, le modèle produit un score de similarité. Les résultats obtenus ont montré une précision d'environ 90% pour les réseaux de neurones siamois, en utilisant une base de données néerlandaise.

Ce dernier [6] a également utilisé un autre jeu de données composé de signatures chinoises. Pour chacune des 20 classes, il a attribué 6 images pour l'entraînement, 3 pour les tests et 3 images pour la validation. Les résultats ont montré une précision de 84% pour les réseaux de neurones siamois. La comparaison à N voies présente des inconvénients comme : la sensibilité aux variations intra-classe, le défi de l'évolutivité et le coût de calcul élevé.

## 3.3 Conclusion

En conclusion de ce chapitre, nous avons examiné plusieurs approches utilisées dans la vérification de signature manuscrite statique, notamment les SVM, KNN, CNN et réseaux de neurones siamois,etc. Nous avons observé que ces méthodes ont toutes démontré leur efficacité dans la détection des signatures authentiques et falsifiées, mais avec des performances variables. Les SVM et KNN ont été largement utilisés dans les premières études, offrant des résultats satisfaisants mais souffrant de limitations en termes de scalabilité et de complexité. Les CNN, en revanche, ont permis d'exploiter les

caractéristiques visuelles des signatures et ont obtenu de bons résultats dans plusieurs études. Les réseaux de neurones siamois, en particulier, se sont avérés prometteurs en exploitant les similarités et les différences entre les paires de signatures pour la vérification. Dans le prochain chapitre, nous proposerons notre approche basée sur les réseaux de neurones siamois pour la vérification de signature manuscrite, en tenant compte des leçons tirées de la revue de littérature et en cherchant à surmonter ces défis.

# CHAPITRE 4 : CADRE DE RECHERCHE

## 4.1 Introduction

Ces dernières années, l'utilisation de techniques de classification basées sur des réseaux de neurones a connu une croissance considérable, en particulier dans le domaine de la vision par ordinateur, où de nombreuses recherches sont en cours.

Dans ce chapitre, nous allons parler des réseaux de neurones convolutifs (CNN), les réseaux siamois (ou réseaux jumeaux) ainsi que les outils qui seront utilisés dans notre mémoire afin d'atteindre notre objectif, qui est de mettre en place un système de vérification de signatures manuscrites performant. Ainsi, nous allons commencer par parler des CNN, par la suite on parlera plus spécifiquement des réseaux siamois et de leurs caractéristiques et pourquoi c'est une méthode efficace pour nous aider à faire la classification même avec une faible quantité de données d'entrainement, et enfin on va terminer par les différents outils qui seront utilisés dans ce mémoire.

## 4.2 L'apprentissage profond avec les réseaux de neurones convolutifs (CNN)

Dans la section 5 du deuxième chapitre, nous avons évoqué les CNN, leur récente domination dans le monde de traitement des données, de la vision artificielle notamment. Dans ce qui suit, nous discuterons de l'origine des CNN, la description de leurs architectures et enfin terminer par des exemples d'architectures de CNN.

### 4.2.1 L'origine des CNN

Les réseaux de neurones à convolution sont basés sur le modèle humain et animal, notamment sur le système oculaire, tout comme les réseaux de neurones artificiels dans leur ensemble. L'inspiration pour les réseaux de neurones à convolution est venue des travaux de neurophysiologistes tels que David Hunter Hubel et Torsten Nils Wiesel [34] sur les structures du cortex visuel des animaux tels que les chats. C'est Fukushima [35] en 1980 qui a introduit la notion du « Neocognitron » (voir Figure 12), un modèle d'apprentissage automatique capable de reconnaître les formes géométriques similaires.

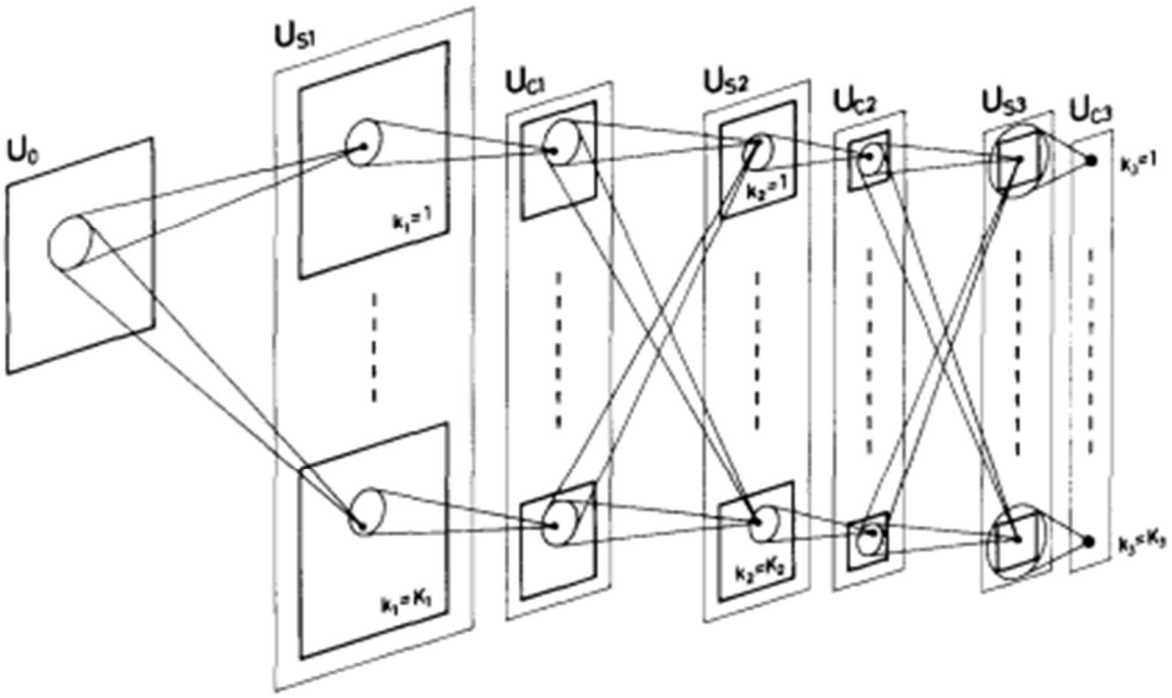


Figure 12 : Interconnexions des couches dans le Neocognitron.

Étant donné que les cellules du réseau sont interconnectées en cascade, plus la couche est profonde, plus le champ récepteur de chaque cellule de cette couche est large. La densité des cellules dans chaque plan de cellules est déterminée de manière à diminuer en fonction de l'augmentation de la taille des champs récepteurs. Par conséquent, le nombre total de cellules dans chaque plan de cellules diminue avec la profondeur du plan de cellules dans le réseau. Dans le dernier module, le champ récepteur de chaque cellule C devient suffisamment grand pour couvrir toute la zone de la couche d'entrée U0, et chaque plan de cellules C est déterminé de manière à n'avoir qu'une seule cellule C [35].

Au fil des années, les réseaux de neurones convolutifs ont subi une évolution graduelle pour aboutir à ce que nous connaissons aujourd'hui. Le travail des chercheurs, tels que Lecun, Bottou, Bengio et Haffner [36], a finalement donné lieu à l'architecture LeNet-5, qui est largement utilisée pour la reconnaissance des textes manuscrits. Cette évolution progressive a permis d'aboutir à des avancées significatives dans le domaine des réseaux de neurones convolutifs.

### 4.2.2 Architecture des CNN

Les réseaux de neurones à convolution (CNN) sont constitués de plusieurs couches, comme tous les réseaux de neurones. La première couche est la couche d'entrée, suivie d'un certain nombre de couches, et enfin d'une couche de sortie qui fournit le résultat de la classification. La structure typique d'un CNN comprend une succession de couches de

convolutions, de couches Relu (unité linéaire rectifiée) et de couches de pooling, suivies d'une couche entièrement connectée qui est une couche de type perceptron et d'une couche de perte identifiée sous le nom de LOSS. La Figure 13 donne un aperçu de l'architecture typique d'un CNN[37].

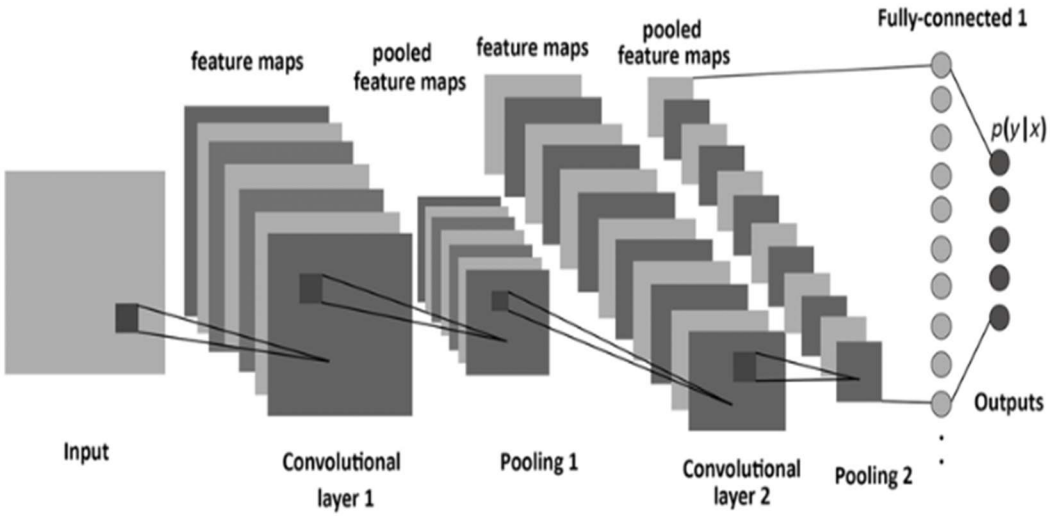


Figure 13 : Architecture classique d'un réseau de neurones convolutif

### 4.2.2.1 Les couches de convolution

Dans un réseau de neurones à convolution (CNN), la structure de base se compose de couches convolutives qui sont connectées de manière sélective aux couches précédentes. Ainsi, la première couche de convolution est connectée à la couche d'entrée, qui peut être une image dans le cas de la première couche de convolution, ou les propriétés extraites de la couche précédente [38]. Chaque neurone de la couche est connecté au pixel qui lui correspond en fonction des paramètres spécifiés, comme illustré dans la Figure 14.

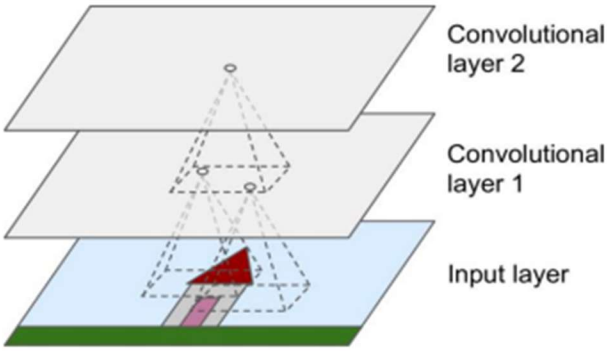


Figure 14 : Couches de convolutions avec les récepteurs locaux en rectangle

### 4.2.2.2 Les couches de pooling

La couche de pooling est un autre bloc constitutif d'un réseau de neurones convolutif. Cette couche effectue une opération de sous-échantillonnage qui permet de réduire considérablement le poids de calcul, l'utilisation de la mémoire et le nombre de paramètres[6]. Tout comme les couches de convolution, les couches de pooling sont connectées à un certain nombre de neurones des sorties des couches précédentes. Leur fonctionnement est encore plus simple, car ces couches n'ont pas de poids. Elles se contentent de renvoyer le résultat d'une fonction d'agrégation, soit maximal ou moyen, comme illustré sur la Figure 15. Cette opération permet de réduire la résolution spatiale des données tout en préservant les informations les plus importantes.

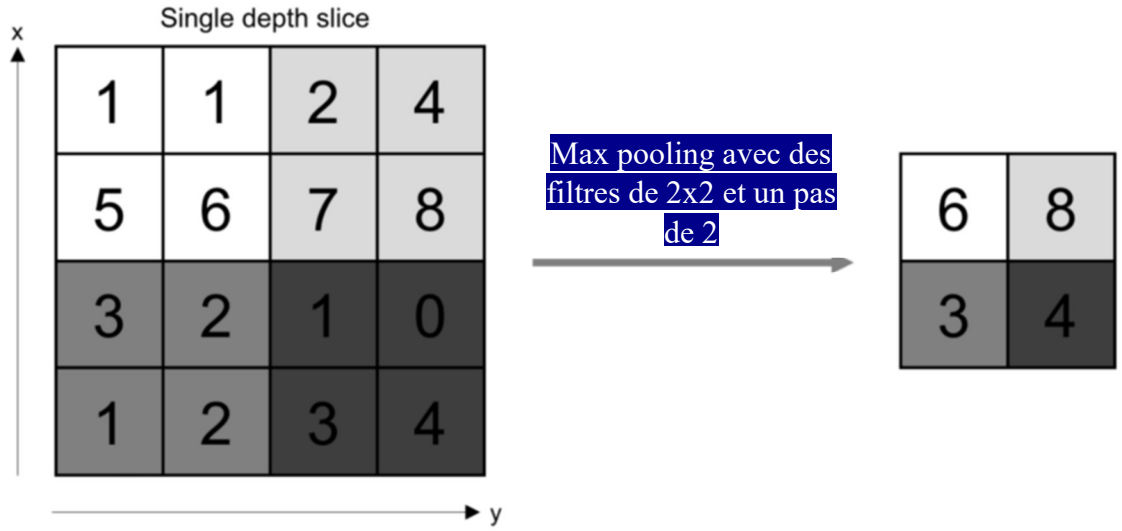


Figure 15 : Illustration de Max Pooling avec filtre 2 x 2 et un pas de 2

Le rôle de la fonction Max Pooling est de réduire la dimensionnalité des données en conservant les caractéristiques les plus importantes.

Un autre avantage de l'utilisation des couches pooling est l'élimination du surapprentissage. Toutefois, elles ont également un inconvénient, à savoir la perte d'informations causée par la réduction de la résolution spatiale des données. Pour minimiser cette perte, il est possible d'utiliser des filtres de petite taille, tels que des filtres 2 x 2, comme illustré dans l'exemple de la Figure 15.

### 4.2.2.3 *Les couches entièrement connectées*

Tout comme son nom l'indique, chaque couche est complètement connectée à la couche précédente. Pour générer des prédictions de classes, la dernière couche entièrement connectée peut être utilisée avec une fonction « sigmoïde » ou « softmax ».

### 4.2.2.4 Les fonctions d'activation

La fonction d'activation joue le rôle d'une passerelle entre l'entrée et la sortie du neurone. Elle est une fonction mathématique qui détermine la sortie en fonction d'un seuil.

Les fonctions non linéaires les plus utilisées sont :

- Sigmoïde ou logistique

$f(x) = \frac{1}{1+e-x}$ Df =] -∞, +∞ [; l'ensemble de tous les nombres réels

- Tangente hyperbolique

$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$

Df =] -∞, +∞ [

- Unité de rectification linéaire (ReLU)

$$f(x) = \begin{cases} 0 \text{ si } x < 0 \\ 1 \text{ si } x \geq 0 \end{cases}$$

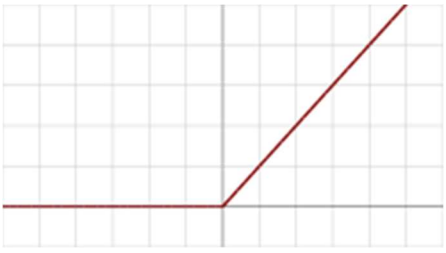

Figure 16 : Exemple d'une fonction d'activation ReLU pour l'intervalle (0; +∞)

Parmi les trois fonctions ci-dessus, la fonction ReLU (Rectified Linear Unit) est celle qui est la plus utilisée dans les réseaux de neurones.

#### 4.2.2.5 La fonction de perte (LOSS)

Afin d'améliorer l'apprentissage d'un réseau de neurones, il est nécessaire de réduire les erreurs. C'est là que la fonction de perte, également appelée fonction de coût ou d'erreurs, intervient pour définir son rôle [39]. Le rôle de la fonction de perte est d'évaluer à quel point les prédictions d'un modèle sont proches des valeurs réelles attendues.

Il existe plusieurs fonctions qui peuvent être utilisées pour déterminer les erreurs, mais dans le cadre du Deep Learning, les deux fonctions les plus utilisées sont :

- Le maximum de vraisemblance : L'approche du maximum de vraisemblance vise à déterminer les valeurs optimales des paramètres en maximisant une fonction de vraisemblance dérivée des données d'apprentissage [40].

- L'entropie croisée (Cross-Entropy) : Dans le cas des problèmes de classification utilisant les CNN, la fonction d'entropie croisée est souvent préférée. Elle s'exprime par la formule ci-dessous:

$$-\sum_{C=1}^{M}(y_C \log \hat{y}_C)$$

Où M est le nombre de classes, $y_C$ correspond à la vraie distribution, uniquement partiellement observée, $\hat{y}_C$ correspond à la distribution non naturelle à partir d'un modèle statistique construit.

### 4.2.3 Exemples d'architectures de CNN

Il existe différentes architectures de réseaux de neurones convolutifs qui sont utilisées en fonction du contexte. Ces architectures ont souvent fait leurs preuves dans des défis d'apprentissage en profondeur, ce qui les a rendues populaires et largement utilisées. Voici quelques-unes de ces architectures :

#### 4.2.3.1 AlexNet

AlexNet est une architecture de réseau de neurones convolutifs développée par Alex Krizhevsky, Ilya Sutskever et Geoffrey Hinton. Elle a remporté le concours ImageNet Large Scale Visual Recognition Challenge (ILSVRC) en 2012, marquant une avancée significative dans le domaine de la vision par ordinateur [41]. AlexNet se compose de plusieurs couches convolutives et entièrement connectées, avec une utilisation innovante

de la fonction d'activation ReLU (Rectified Linear Unit) et une régularisation par suréchantillonnage (dropout).

#### 4.2.3.2 GoogleNet

GoogleNet, également connu sous le nom de Inception v1, est une architecture de réseau de neurones convolutifs développée par l'équipe de recherche de Google. Elle a été présentée en 2014 et a remporté le concours ILSVRC la même année. GoogleNet est caractérisé par son utilisation d'un module d'inception, qui consiste en l'utilisation de différentes tailles de filtres de convolution en parallèle pour extraire des caractéristiques à différentes échelles. Cette architecture permet de réduire le nombre de paramètres du réseau tout en maintenant une performance élevée. De plus, GoogleNet intègre également des couches de réduction de dimension et des connexions résiduelles pour améliorer la stabilité et la capacité de généralisation du réseau [42].

#### 4.2.3.3 ResNet

ResNet, abréviation de Residual Network, est une architecture de réseau de neurones convolutifs introduite par Microsoft Research en 2015. Elle a été développée pour résoudre le problème du dégradement de la performance lors de l'ajout de couches profondes dans un réseau neuronal. L'idée centrale de ResNet réside dans l'utilisation de blocs résiduels qui permettent aux informations de contourner certaines couches du réseau, facilitant ainsi l'apprentissage en profondeur.

#### 4.2.3.4 MobileNet

MobileNet est une architecture de réseau de neurones convolutifs conçue spécifiquement pour les applications sur des appareils mobiles avec des ressources limitées en termes de puissance de calcul et de mémoire. Elle a été développée par Google en 2017 dans le but de fournir des modèles de réseaux de neurones légers et efficaces pour la classification d'images et d'autres tâches de vision par ordinateur sur des dispositifs mobiles.

#### 4.2.3.5 DenseNet

DenseNet est une architecture de réseau de neurones convolutifs qui se distingue par sa connexion dense ou "dense connectivity". Elle a été introduite en 2016 par Huang et al. dans leur article intitulé "Densely Connected Convolutional Networks". DenseNet a montré de très bonnes performances sur différentes tâches de vision par ordinateur, notamment la classification d'images, la détection d'objets et la segmentation sémantique.

## 4.3 Approche proposée

La figure 17 illustre d'une manière simplifiée notre système de vérification de signature. Ce dernier accepte deux entrées qui correspondent à deux signatures, et une sortie qui représente le score de dissimilarité entre celles-ci. Dans notre système de vérification de signature, nous utilisons une approche qui prédit à la fois le degré de dissimilarité entre les deux signatures en entrée et détermine si la signature est authentique ou frauduleuse. Les détails de cette approche seront explorés plus en profondeur dans la section 5.2 du chapitre 5.

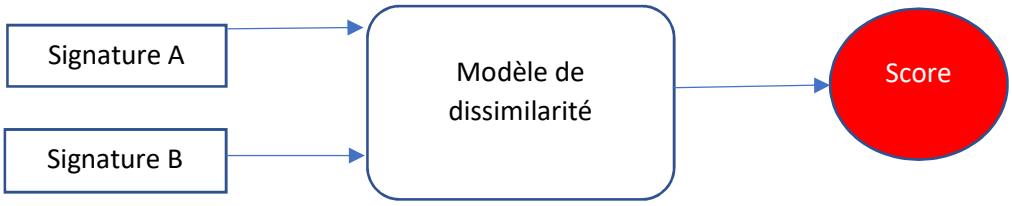


Figure 17 : Modèle simplifié du système de vérification de signature.

Dans la section 4.3.1, nous allons parler de notre motivation quant à l'utilisation des réseaux de neurones siamois ou réseaux jumeaux, qui constitue l'originalité de notre mémoire. La section 4.3.2 sera consacrée à l'architecture des réseaux de neurones siamois et pour finir à la section 4.3.3, nous allons parler de la classification des signatures avec ces réseaux de neurones.

### 4.3.1 Le choix des réseaux de neurones siamois pour notre mémoire

Dans un système de vérification de signature, il est courant de disposer d'un nombre limité d'exemples de signatures authentiques et frauduleuses. De plus, pour les grandes organisations comme les banques, nous pouvons avoir beaucoup de clients et nous ne pouvons pas recycler le modèle chaque fois qu'un nouveau client s'inscrit à la banque. C'est dans ce contexte que nous comptons utiliser l'architecture siamoise dans notre mémoire, qui est une solution efficace pour résoudre ce problème. Les avantages de l'architecture du modèle siamois sont qu'elles permettent d'avoir une seule signature par personne au lieu d'avoir besoin de plusieurs échantillons de signatures pour former le classificateur.

### 4.3.2 Architecture des réseaux de neurones siamois

Les architectures siamoises nous permettent intuitivement d'acquérir une mesure de similarité en utilisant deux entrées indépendantes ayant une relation abstraite de similarité. Les deux entrées peuvent être des images, des phrases, des vecteurs ou tout autre type de données. Elle est composée de deux branches identiques qui partagent les mêmes

poids et architectures de couches de neurones. Chaque branche traite une des deux entrées à comparer. Les sorties des deux branches sont ensuite comparées à l'aide d'une fonction de coût pour déterminer leur similarité (voir la figure 18). Cette architecture est souvent utilisée dans des applications telles que la reconnaissance faciale, l'authentification de signatures, la détection de plagiat, etc. Les premiers travaux sur l'utilisation de réseaux de neurones siamois font mention de la vérification de signatures effectuée par Bromley et al [43].

Les réseaux de neurones siamois sont des réseaux jumeaux qui sont une classe d'architectures de réseau qui contient généralement deux sous-réseaux identiques. Les CNN jumeaux ont la même configuration avec les mêmes paramètres et des poids partagés. L'architecture peut varier en fonction de la tache spécifique et de la complexité des données en entrée.

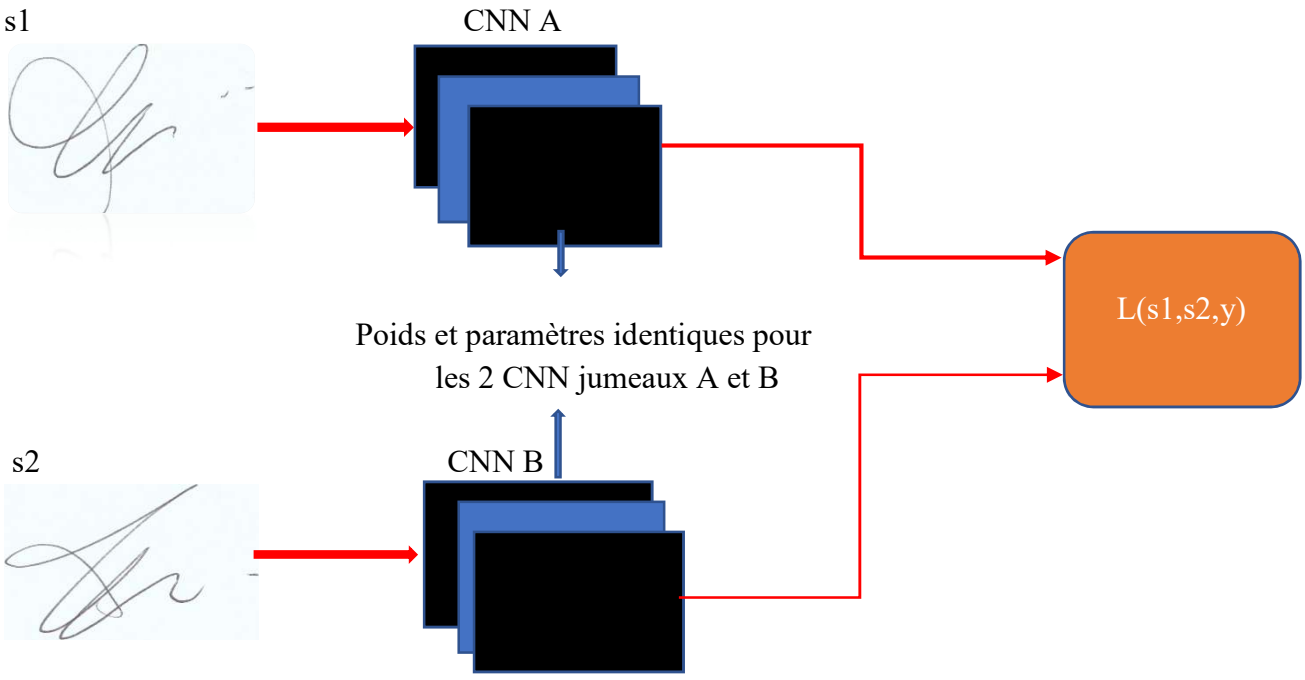


Figure 18 : Exemple d'architecture de réseaux siamois pour la vérification de signatures

$$\mathbf{L\ (s_1, s_2, y) = \alpha(1 - y)\ D^2{}_w + \beta y\ max\ (0; m - D_w)^2}$$

Avec :

- $s_1$ et $s_2$ sont deux échantillons de signatures, y est une fonction indicatrice binaire indiquant si les deux échantillons proviennent ou non de la même classe,

- α et β sont deux constantes et m est la marge égale à 1 dans notre cas.
- $\mathbf{D_w = \|f(s_1 ; w_1) - f(s_2 ; w_2)\|_2}$ est la distance euclidienne calculée dans l'espace des caractéristiques intégrées, f est une fonction d'intégration qui mappe une image de signature à l'espace vectoriel réel par CNN, $w_1$ et $w_2$ sont des pondérations apprises par une couche particulière du réseau sous-jacent.
- A et B sont des CNN identiques
- le choix de la distance euclidienne dans la fonction L est cohérent avec l'objectif de mesurer la similarité ou la dissimilarité entre les signatures en se basant sur la proximité de leurs caractéristiques dans un espace euclidien.

Traditionnellement, les réseaux de neurones sont conçus pour prédire plusieurs classes, ce qui peut poser des problèmes lorsqu'il est nécessaire d'ajouter ou de supprimer des classes de données. Chaque classe est constituée d'un ensemble de signature de la même personne. En outre, les réseaux de neurones profonds ont besoin d'un grand nombre de données pour s'entraîner. Les réseaux siamois, quant à eux, apprennent une fonction de similarité qui leur permet de déterminer si deux images sont identiques ou pas. Cette approche permet de classifier de nouvelles classes de données sans avoir besoin de réentraîner le réseau à nouveau.

### 4.3.3 Classification avec les réseaux de neurones siamois

Dans une application d'authentification de signature, nous n'avons généralement pas beaucoup d'échantillons de signatures authentiques et de fausses signatures. De plus, pour les grandes organisations comme les banques ou les assurances, nous pouvons avoir beaucoup de clients et nous ne pouvons pas recycler le modèle chaque fois qu'un nouveau client s'inscrit à la banque ou l'assurance.

Si nous considérons l'exemple de la classification du visage. Pour une entreprise qui a 5 employés, la classification du visage pour eux par la classification CNN normale aurait 5 probabilités de sortie. Pour former un tel réseau, nous avons besoin de grands échantillons de chaque personne. Un autre problème est que si nous voulons ajouter une nouvelle classe comme le visage d'un nouvel employé, nous devons obtenir une grande quantité d'images pour cette nouvelle classe et recycler à nouveau l'ensemble du modèle. Il existe plusieurs tâches et applications où nous avons de petits échantillons pour des classes données. De plus, parfois, les classes changent constamment. Un exemple de cela serait un nouvel employé pour l'entreprise. Par conséquent, si nous avons utilisé le modèle de classification normal. Non seulement nous avons besoin de plus d'échantillons de visages de nouveaux employés, mais aussi, l'ensemble du modèle devrait être recyclé. Cela entraîne beaucoup de frais financiers. Les réseaux siamois sont utiles pour résoudre ce genre de problèmes. En effet, ces réseaux apprennent à reconnaître les similitudes entre des échantillons en alimentant des paires d'images dans le réseau, puis en ajustant les poids pour maximiser

ou minimiser leur similarité ou dissimilarité en fonction de leur appartenance à la même classe. Lors de l'inférence, une image donnée est comparée à cette référence, et le réseau siamois produit une valeur de similitude. Cette approche a également été utilisée par [6][45] pour la classification des caractères. Dey et al [6][44] avaient pour objectif de classer les signatures falsifiées.

Les avantages de l'architecture du modèle siamois sont qu'elle permet d'avoir une seule signature par client au lieu d'avoir besoin de plusieurs échantillons pour former le classificateur. Lors de la formation, le ConvNet extrait les entités d'images données, puis la différence entre ces fonctionnalités est minimisée, si deux images appartiennent à la même classe et la différence est maximisée si les deux images données n'appartiennent pas à la même classe.

## 4.4 Les outils de développement utilisés

### 4.4.1 Outils et technologies

Actuellement, différentes technologies sont disponibles pour mettre en place l'apprentissage machine, cependant elles exigent toutes une ressource matérielle conséquente en fonction des données à traiter et du contexte de la recherche. Dans notre cas, le traitement d'images requiert une quantité importante de ressources matérielles, ce qui signifie qu'il est nécessaire d'utiliser une machine suffisamment performante pour mener à bien les expérimentations.

#### 4.4.1.1 Plateforme d'expérimentation

Après avoir mené diverses expérimentations sur des systèmes locaux, qu'il s'agisse de notre ordinateur personnel doté d'une configuration assez performante (un processeur Core i5-5300U à 2,3 GHz et 8 Go de RAM), ou sur un ordinateur disponible dans notre laboratoire MI200, nous avons finalement opté pour une solution plus avantageuse en matière de ressources et de disponibilité en utilisant l'une des solutions proposées dans le cloud.

Actuellement, plusieurs entreprises proposent des services permettant l'utilisation de serveurs et de postes de travail à distance, avec une allocation de ressources personnalisée en fonction des besoins de l'utilisateur. Ces offres peuvent être payantes ou gratuites, selon les exigences de l'utilisateur.

Ci-dessous, une liste non exhaustive des services d'apprentissage machine :

- Microsoft Azure

- Amazon Web Service (AWS)
- IBM Watson
- Etc.

Dans ce mémoire, nous avons choisi d'utiliser Google Colab, qui est à la fois accessible et flexible, offrant des solutions personnalisables en fonction de nos besoins.

#### 4.4.1.2 Google Colab

Google Colaboratory (ou Google Colab) est un environnement de développement gratuit en ligne pour l'apprentissage machine et l'analyse de données appartenant à Google. Il permet de créer, exécuter et partager des notebooks Jupyter qui contiennent des codes Python et des résultats interactifs. Google Colab est hébergé sur les serveurs cloud de Google, ce qui signifie qu'il fournit un accès à des ressources de calcul puissantes, y compris des GPU et des TPU, qui peuvent être utilisées gratuitement. En outre, il permet de collaborer facilement avec d'autres personnes sur un même notebook, en partageant simplement un lien.

### 4.4.2 Le choix du langage de programmation

Nous avons opté pour Python pour nos expérimentations, en raison de nombreux avantages qu'il présente. En effet, le langage de programmation Python offre une grande flexibilité, ainsi qu'une communauté dynamique qui fournit une abondante documentation. En outre, Python intègre parfaitement un large éventail de bibliothèques de recherche et de traitement d'images, dont nous allons énumérer ci-dessous celles que nous avons utilisées.

### 4.4.3 Framework et Librairies

Il existe actuellement beaucoup de Framework et de librairies dans le domaine du Machine Learning et de la vision par ordinateur. Nous allons présenter ci-dessous quelques-uns de ceux que nous avons utilisés dans notre mémoire.

#### 4.4.3.1 Torchvision

Torchvision est une librairie open source de Python qui fournit des fonctionnalités de traitement d'images pour la plate-forme de calcul scientifique PyTorch. Torchvision fournit des outils pour la manipulation d'images, notamment la transformation, la normalisation, le chargement et l'enregistrement de jeux de données d'images. Il contient également des modèles de réseaux de neurones préentraînés pour la classification d'images, la détection d'objets, la segmentation d'images et d'autres tâches de vision par ordinateur. En utilisant Torchvision, on peut facilement manipuler des images et entraîner des modèles de vision par ordinateur en utilisant PyTorch, ce qui facilite la création de systèmes d'intelligence artificielle pour les applications liées à la vision par ordinateur [46].

### 4.4.3.2 Torch et PyTorch

Torch est une bibliothèque logicielle open source pour le calcul scientifique, avec un accent particulier sur l'apprentissage automatique et le Deep Learning. Il a été développé en C++ avec une interface de programmation en Lua et offre des fonctionnalités pour le traitement de données numériques, le calcul matriciel, la différenciation automatique, l'optimisation et la construction de modèles de réseaux de neurones. Torch est souvent utilisé pour la recherche en intelligence artificielle et l'apprentissage automatique en raison de sa facilité d'utilisation et de sa flexibilité. Il fournit également des outils pour le chargement et la manipulation de données, la visualisation, la parallélisation et la distribution sur des clusters de calcul, ce qui en fait un choix populaire pour la recherche et le développement de solutions en intelligence artificielle [46].

PyTorch est en fait une évolution de Torch, développé en Python plutôt qu'en Lua. PyTorch est construit sur les mêmes principes fondamentaux que Torch et offre des fonctionnalités similaires, avec une interface Python plus familière pour de nombreux utilisateurs.

### 4.4.3.3 Numpy

NumPy est une bibliothèque open source de Python qui fournit des outils pour le calcul numérique. NumPy est largement utilisé en science des données, en apprentissage automatique et dans d'autres domaines de l'informatique pour effectuer des opérations mathématiques et statistiques sur des tableaux de données multidimensionnels. C'est une bibliothèque fondamentale pour de nombreux projets de science de données et de calcul scientifique en Python.

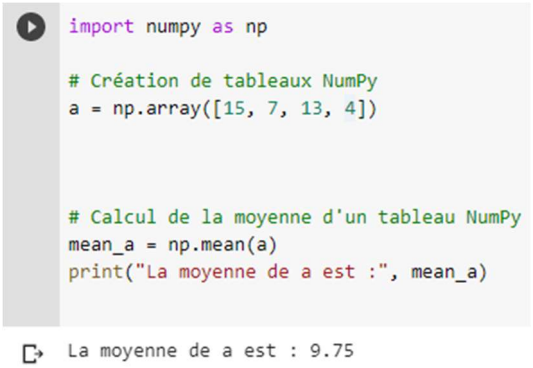


Figure 19 : Utilisation de Numpy pour calculer la moyenne du tableau a.

### 4.4.3.4 Pandas

Pandas est une bibliothèque open source de Python qui permet de faire la manipulation de données en tableaux. Il offre des fonctionnalités pour l'importation et l'exportation de données, la fusion et la jointure de données, la transformation de données,

le filtrage, le tri, la recherche et la sélection de données, ainsi que la visualisation de données. Pandas est cruciale pour faire la manipulation et l'analyse de données structurées.

```
import pandas as pd

# Charger les données à partir d'un fichier CSV
df = pd.read_csv("profs.csv", encoding="ISO-8859-1")

# Affichage les  lignes des données
print(df.head())
```

```
         Nom    Prénom   Département
0  Fathallah   Nouboud  Informatique
1   Ghazzali     Nadia   Statistique
2    Meunier  François  Informatique
3    Mesfoui    Mhamed  Mathématique
```

Figure 20 : Affichage des professeurs avec Pandas

### 4.4.3.5 Matplotlib

Matplotlib est une bibliothèque open source du langage de programmation Python qui permet de créer des graphiques et des visualisations de données. Il est largement utilisé en science des données, en apprentissage automatique, en visualisation de données et dans d'autres domaines de l'informatique. Matplotlib fournit des fonctions pour créer des graphiques de différentes formes et tailles, y compris des graphiques en barres, des graphiques en secteurs, des graphiques en nuage de points, des graphiques en boîtes et des graphiques en surface. Les graphiques peuvent être personnalisés avec des titres, des étiquettes d'axes, des couleurs, des légendes et d'autres options de style. Il est essentiel pour faire la visualisation de données et la communication de résultats d'analyse de données.

```
import matplotlib.pyplot as plt
#import numpy as np

# Créer des données pour tracer
x = np.linspace(0, 10, 100)
y = np.sin(x)

# Tracer la courbe
plt.plot(x, y)
```

```
[<matplotlib.lines.Line2D at 0x7fa56d64deb0>]
```

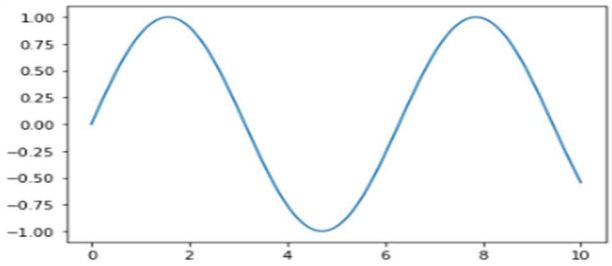


Figure 21 : Exemple de code Matplotlib pour tracer une courbe

## 4.5 Conclusion

Ce chapitre a porté sur les contextes et les technologies que nous avons utilisées dans notre mémoire. Nous avons commencé par présenter les réseaux de neurones convolutifs (CNN). Par la suite, nous avons parlé des réseaux de neurones siamois, qui sont la méthodologie utilisée dans notre mémoire. Enfin, nous avons énuméré les différents outils que nous avons utilisés pour mener à bien nos expérimentations. Dans le prochain chapitre, nous expliquerons en détail comment nous avons mis en œuvre ces technologies et utilisé ces données pour faire nos expérimentations.

# CHAPITRE 5 : Méthodologie

## 5.1 Introduction

Dans ce chapitre, nous allons aborder notre plan de travail, ensuite l'acquisition des signatures et enfin terminer par l'organisation des données.

## 5.2 Plan de travail

L'approche que nous proposons dans ce mémoire est basée sur la dissemblance qui existe à la suite d'une comparaison par paire sur l'ensemble de notre jeu de données (Dataset) en utilisant les réseaux de neurones siamois (voir la figure 22).

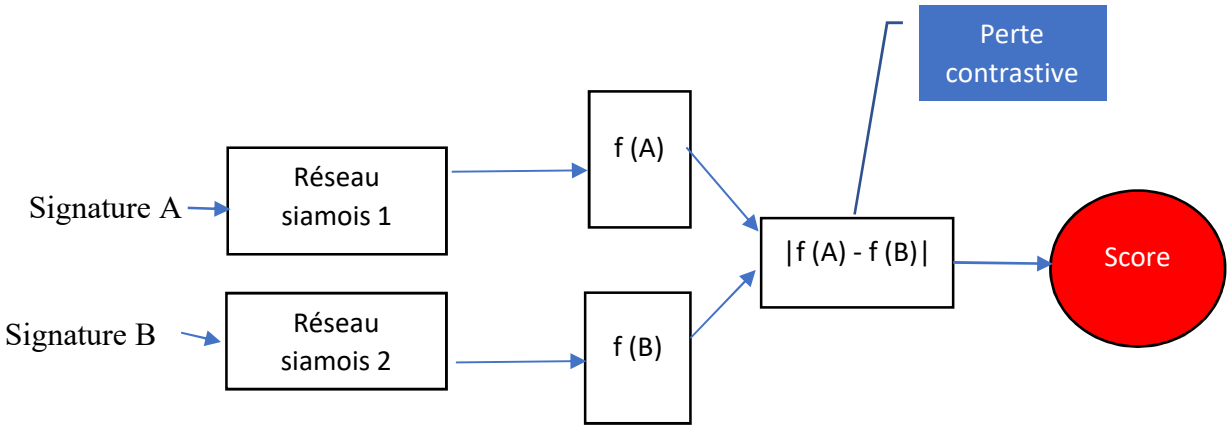


Figure 22 : Calcul de la dissimilarité avec les réseaux de neurones siamois.

Dans ce mémoire, nous avons décidé de différencier notre approche de celle des autres travaux sur les réseaux de neurones siamois en optant pour le calcul de la dissemblance des deux classes, plutôt que celui du score de similarité, qui est plus fréquent. Par exemple, le travail de Abolfazl[6], discuté dans la revue de littérature, utilise une approche de calcul de similarité basée sur une comparaison à N-voies. Pour rappel, les réseaux de neurones siamois ou réseaux jumeaux sont pour la plupart des cas deux CNN qui partagent les mêmes poids et paramètres. Nous pouvons émettre l'hypothèse que si les signatures appartenant à la même classe sont passées aux modèles jumeaux, les caractéristiques extraites devraient être similaires. Cette hypothèse est fondée sur le fait que les Convnets partagent les mêmes paramètres et que les images sont similaires. D'autre part, si les signatures appartiennent à des classes différentes, les caractéristiques produites seront différentes. En considérant l'exemple illustré dans la figure 22, où deux images de signatures A et B sont transmises à deux sœurs jumelles(les deux réseaux siamois), et leurs

caractéristiques respectives, f (A) et f (B), sont extraites automatiquement. Ensuite, chaque entité s'aplatit, puis dans une autre couche, la dissemblance entre elles est calculée. La dissimilarité est calculée comme la distance entre les sorties du modèle pour chaque image d'entrée (A, B).

$$\text{Distance}_i = |f_i(A) - f_i(B)|$$

Cependant, si deux signatures sont visuellement différentes, leur score de dissimilarité calculé par la couche sigmoïde de sortie devrait être supérieur au score de dissimilarité des signatures appartenant à la même classe. Les étiquettes de chaque paire d'images sont également affichées, soit comme « Original » ou « Faux », selon leur véritable étiquette.

## 5.3 Description du jeu de données

Notre jeu de données provient de 55 individus issus de différentes origines, culturelles et professionnelles.

Figure 23:Exemple de signature originale

Figure 24 : Exemple de signature falsifiée

Les signatures sont organisées en catégories : originale et falsifiée. Pour l'apprentissage, on a constitué nos dossiers comme suit :

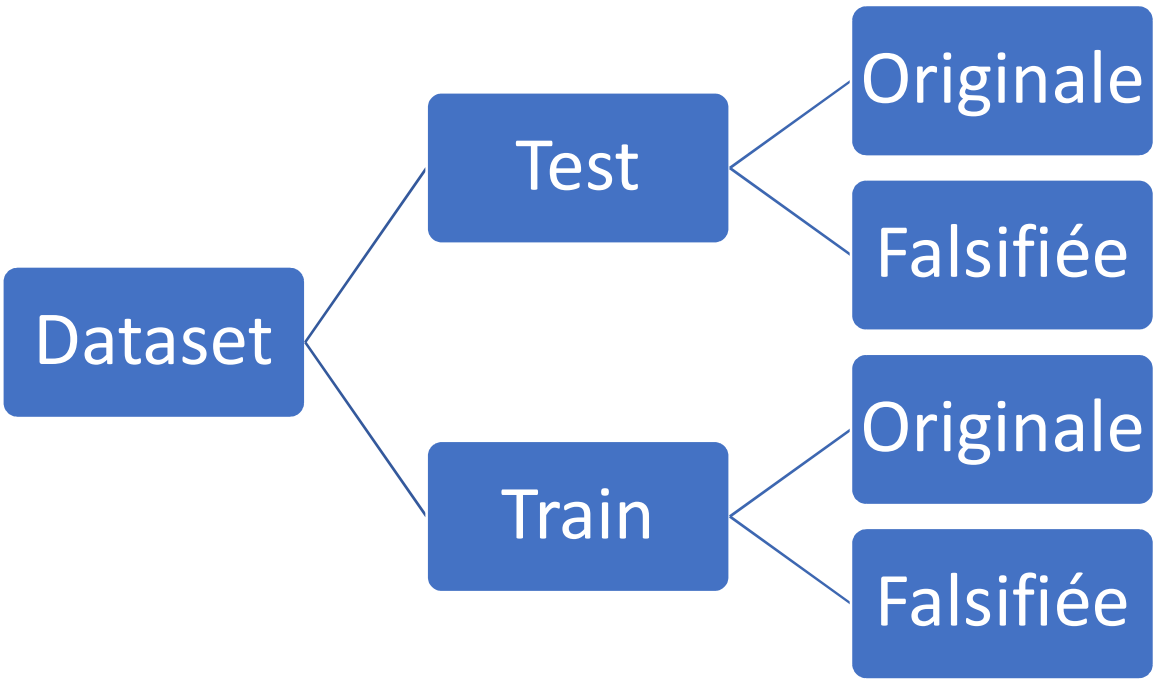


Figure 25 : Organisation des données.

Nous avons veillé à ce que les images de chaque catégorie soient équivalentes pour éviter de biaiser les résultats.

### 5.3.1 Choix des images

Le processus commence par l'extraction et la transformation des images brutes de notre Dataset pour qu'elles puissent être traitées par le modèle. Ensuite, les données de formation sont importées depuis un fichier csv et stockées dans une base de données pandas. À partir de là, le chemin vers les images est extrait et les images correspondantes sont récupérées pour être transformées. Ces images seront converties en images de niveaux de gris, redimensionnées à une taille de 105x105, puis enregistrées sous forme de fichiers .npy (format de fichier standard dans Numpy).

### 5.3.2. L'emplacement des données

Dans le chapitre 4, nous avons parlé de Google Colab, qui représente notre environnement de travail dans ce projet. Ainsi, nous avons jugé approprié de charger les données dans un dossier Drive pour en faciliter l'accès et pouvoir les exploiter aisément. Cependant, pour que le script puisse accéder aux dossiers contenant les données du projet une fois celles-ci chargées sur Google Drive, il est nécessaire d'autoriser l'accès.

Une fois que l'accès est autorisé, nous avons chargé l'ensemble des données à partir de dossiers d'images brutes.

```
[ ]  siamese_dataset = SiameseNetworkDataset(training_csv,training_dir,
                                              transform=transforms.Compose([transforms.Resize((105,105)),
                                                                            transforms.ToTensor()
                                                                            ])
                                             )
```

Ensuite, nous avons affiché des échantillons d'images chargées pour vérifier si nous avons correctement accédé à nos données.

### 5.3.3 Nombre d'images traitées par lot

Bien que nous ayons diminué la taille des images, leur traitement nécessite toujours une quantité considérable de ressources. Pour faire face à ce problème, nous avons adopté une stratégie de traitement par lots, où chaque lot prend en charge 32 images à la fois.

### 5.3.4 Nombre d'itérations

Le terme « epoch » est utilisé pour décrire le nombre de fois où l'algorithme d'apprentissage est appliqué sur l'ensemble des données, chaque epoch produit un résultat et l'évolution de ces résultats constitue l'ensemble de l'apprentissage. Dans notre mémoire, nous avons fixé le nombre d'epochs à 20. Si aucune amélioration n'est observée, les itérations peuvent être interrompues grâce à un script. À la fin des epochs, le modèle d'apprentissage est sauvegardé avec le meilleur résultat obtenu.

### 5.3.5 Le modèle

Notre modèle de réseau de neurones siamois est constitué de 4 couches Conv2d, de 4 couches MaxPool2d avec des décrochages aux couches alternatives et 2 couches entièrement connectées. Nous avons également appliqué la normalisation de la réponse locale et avons utilisé la fonction d'activation ReLU.

Dans notre modèle de réseau siamois pour l'apprentissage de la dissimilarité entre les images, nous avons utilisé la fonction de perte contrastive pour apprendre les caractéristiques discriminatives qui séparent les paires d'images similaires et celles qui ne le sont pas. Cette fonction de perte contrastive calcule la distance entre les caractéristiques des paires d'images et retourne une valeur plus grande si les images sont différentes et une valeur plus petite si les images sont similaires. (Cette distance est calculée dans la section 4.2).

Le « dropout » est une méthode qui aide à prévenir le surapprentissage [47] [48]. Le surapprentissage se définie comme un phénomène indésirable dans le domaine de l'apprentissage automatique. Il se produit lorsque le modèle est capable de fournir des prédictions précises pour les données d'entraînement sur lesquelles il a été formé, mais lorsqu'il est confronté à de nouvelles données, il ne parvient pas à produire des prédictions précises [49]. Nous l'appliquons avant la dernière couche entièrement connectée afin d'améliorer les performances, d'optimiser le temps des itérations et d'éviter le surapprentissage.

Pour la compilation du modèle, nous avons utilisé l'optimisateur « RMSprop », la fonction de perte est « ContrastiveLoss() ». Cette dernière est utilisée en apprentissage par siamois pour entraîner un réseau de neurones à apprendre une représentation de données qui minimise la distance entre des paires d'échantillons similaires et maximise la distance entre des paires d'échantillons dissimilaires.

### 5.3.6 Évaluation des performances

Pour l'évaluation des performances de notre modèle, nous nous sommes basés sur les métriques telles que : la précision, le rappel, le score F1 et la matrice de confusion. Nous

aborderons plus en profondeur ces métriques dans le prochain chapitre consacré à l'analyse des résultats et aux discussions.

## 5.4 Conclusion

Dans le présent chapitre, nous avons expliqué notre approche, la manière dont nous avons organisé les données ainsi que les prétraitements nécessaires pour les rendre exploitables dans le cadre de nos expérimentations. Nous avons également expliqué comment nous avons mis en œuvre des techniques d'apprentissage machine à l'aide des réseaux de neurones siamois sur nos données. Dans le chapitre suivant, nous allons présenter les résultats obtenus en utilisant divers outils de mesure pour évaluer les performances de notre modèle, discuter les résultats obtenus et enfin comparer les résultats avec les travaux d'autres chercheurs.

# Chapitre 6 : Résultats et discussions

## 6.1 Introduction

Dans ce chapitre, nous allons présenter les résultats obtenus au cours de notre recherche, les métriques de performances utilisées et enfin terminer par une discussion de ces résultats.

## 6.2 Structure de notre algorithme et résultats

**Début de l'algorithme:**

1. Initialiser compteur à 0
2. Créer une liste 2D contenant uniquement la valeur 0, appelée liste_0
3. Créer une liste 2D contenant uniquement la valeur 1, appelée liste_1
4. Pour chaque élément de la liste de l'ensemble des données de test(données utilisées pour évaluer les performances de notre modèle) avec un index i commençant à 0, faire les étapes suivantes:

    a. Récupérer les valeurs de x0, x1 et le label de l'élément qui contient les données d'un élément du tenseur T.
    b. Concaténer x0 et x1 en un seul tenseur T.
    c. Passer x0 et x1 dans le modèle pour obtenir output1 et output.
    d. Calculer la distance euclidienne entre output1 et output2.
    e. Si label est égal à liste_0, définir label à Originale. Sinon, définir label à Falsifiée.
    f. Afficher une image créée à partir du tenseur T avec la distance euclidienne calculée et le label correspondant.
    g. Incrémenter compteur de 1
    h. Si compteur atteint la valeur de 20, sortir de la boucle

**Fin de l'algorithme.**

Avec :

Label : pour identifier la catégorie ou la classe à laquelle appartient cet élément. Dans notre algorithme, le label est utilisé pour différencier les données entre les classes "Originale" et "Falsifiée".

x0 et x1 : paire d'images d'entrée

Notre algorithme parcourt l'ensemble des données de test et pour chacun des lots de données, il calcule la dissimilarité d'une paire d'images d'entrée (x0, x1) à l'aide de notre modèle préformé. La dissimilarité est calculée comme la distance euclidienne entre les sorties du modèle pour chaque image d'entrée (output1, output2). Si l'image est considérée comme originale, « label » est défini comme « Originale », sinon « label » est défini comme « Falsifiée ». Il prend en entrée un ensemble de données qui contient des éléments sous la forme de tenseurs. En termes de sortie, l'algorithme génère une représentation visuelle des éléments de données, où chaque image est accompagnée de la distance euclidienne calculée et du label correspondant. Les images de sortie, leurs scores et étiquettes de dissimilarité sont affichés à l'aide de la fonction « imshow ». La figure 26 ci-dessous donne un aperçu des résultats obtenus grâce à notre modèle.

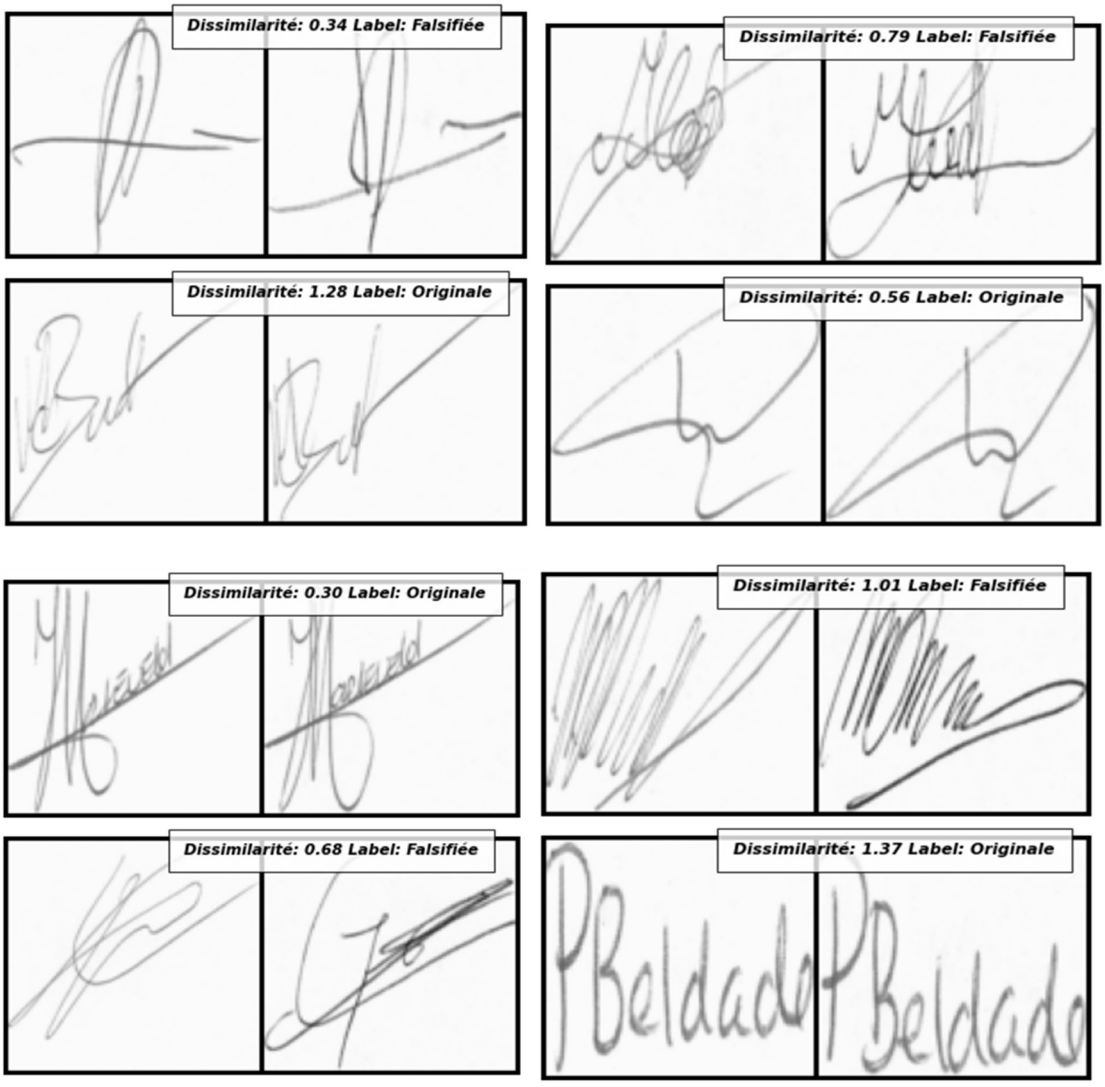


Figure 26:Présentation des résultats.

## 6.3 Les métriques de performances

### 6.3.1 La fonction de perte

Dans le chapitre précédent, nous avons parlé de la fonction de perte « ContrastiveLoss () » utilisée dans notre projet. Cette dernière est utilisée en apprentissage siamois pour entraîner un réseau de neurones à apprendre une représentation de données qui minimise la distance entre des paires d'échantillons similaires et maximise la distance entre des paires d'échantillons dissemblables. Cette fonction de perte diminue progressivement, ce qui indique une amélioration des résultats.

Le résultat que nous avons obtenu à la fin des 20 itérations est très convaincant. L'observation de la diminution de la fonction de perte au fil des epochs est considérée comme un indicateur positif, cela montre que le modèle s'améliore au fil des epochs et se rapproche de son objectif de mieux différencier les signatures authentiques des signatures falsifiées.

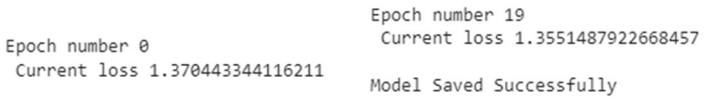


Figure 27: Évolution de la fonction de perte en fonction des itérations.

### 6.3.2 La précision, le rappel et F1-score

- La précision permet de connaitre le nombre de prédictions positifs bien éffectuées. Elle est calculée à l'aide de la formule suivante :

$$\text{Précision} = \frac{\text{Vraie Positive}}{\text{Vraie Positive+Fau Positive}}$$

Le numérateur est le nombre de positifs bien prédit et le dénominateur représente l'ensemble des positifs prédit.

- Le rappel mesure la justesse des prédictions des classes positives parmi toutes les prédictions pertinentes. Contrairement à la précision, il prend en compte les faux négatifs. Sa formule est la suivante.

$$\text{Rappel} = \frac{\text{Vraie Positive}}{\text{Vraie Positive+F Négative}}$$

- Le Fscore permet de mesurer la précision d'un test. Elle est obtenue grâce à la formule suivante :

$$\text{Fscore} = 2 \text{ x } \frac{\text{précision x rappel}}{\text{précision+rappel}}$$

Si le Fscore est élevé, cela signifie que le modèle est efficace dans la détection de tous les échantillons et différenciation précise des autres classes.

Ci-dessous, la matrice de confusion de l'évaluation de notre modèle avec comme nombre d'erreur égal à 68, un vrai positif 359 et un vrai négatif 323.

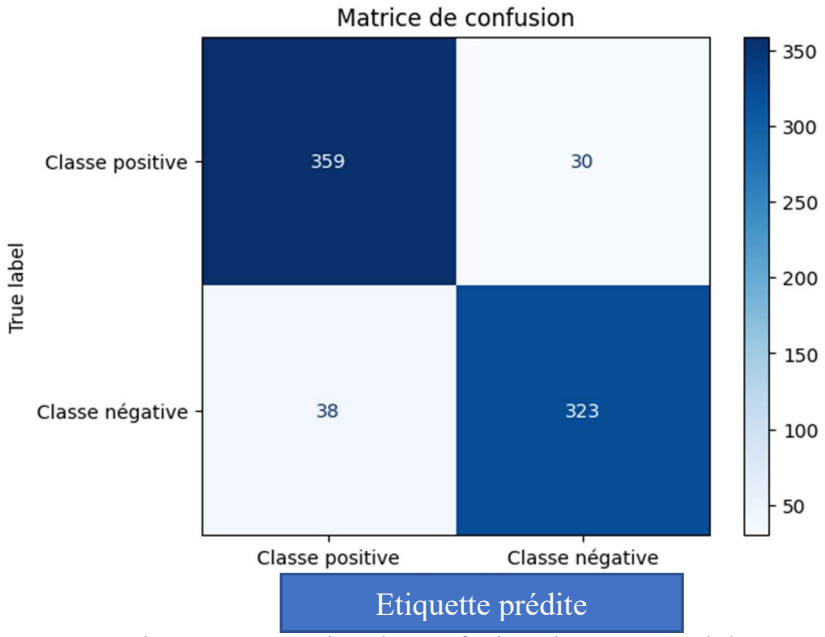


Figure 28: Matrice de confusion de notre modèle.

Vrai positif = 359 : signifie que notre modèle a correctement prédit que 359 signatures sont authentiques.

Vrai négatif= 323 : signifie que notre modèle a correctement prédit que 323 signatures sont fausses.

Faux négatif = 38 : signifie que notre modèle a incorrectement prédit que 38 signatures sont fausses alors qu'elles sont authentiques.

Faux positif = 30 : signifie que notre modèle a incorrectement prédit que 30 signatures sont authentiques alors qu'elles sont fausses.

Cependant, il est important de noter que les vrais positifs et les vrais négatifs sont les résultats souhaitables car ils indiquent que notre modèle a correctement identifié les signatures authentiques et les signatures fausses.

Après l'évaluation des métriques, les résultats obtenus sont donnés dans le tableau suivant :

| Précision | Rappel | Fscore |
|---|---|---|
| 92 % | 90% | 91% |

Tableau 1 : Évaluation des performances de notre modèle.

Ces résultats montrent que notre modèle est efficace en ce qui concerne la détection des signatures originales comme falsifiées. En effet, le modèle s'en sort mieux en matière de rencontre de moins de faux positifs et de classification de chaque nouvelle classe.

## 6.4 Comparaison de notre modèle à d'autres méthodes

### 6.4.1 Comparaison avec les méthodes SVM et K-voisins les plus proches

Maintenant que nous avons mené notre étude en utilisant la méthode des réseaux de neurones siamois, nous allons d'abord procéder à une comparaison en utilisant les résultats de [1] afin de les comparer et de déterminer leurs éventuels points forts et faibles par rapport à notre méthode. L'auteur de l'article [1] a utilisé 75 % des signatures pour l'entraînement, soit 30 signatures par utilisateur, et 25 % pour les tests, soit 10 signatures par utilisateur. Cela donne un total de 1200 signatures pour l'entraînement et 400 pour les tests pour tous les utilisateurs. Le tableau 2 ci-dessous compare nos résultats obtenus avec les réseaux de neurones siamois et ceux de [1] avec les autres méthodes comme SVM et K-voisins les plus proches.

| Méthode | Précision | Rappel | Fscore |
|---|---|---|---|
| Réseaux de neurones siamois | 92% | 90% | 91% |
| Machine à vecteur de support [1] | 87 % | 83 % | 90 % |
| K-voisins les plus proches [1] | 80 % | 80 % | 88 % |

Tableau 2 : Évaluation des performances de notre modèle avec SVM et KNN

Il est évident que la méthode des K-voisins les plus proches n'est pas très appropriée pour notre système de vérification de signature, car elle affiche des performances inférieures à celles des deux autres méthodes, à savoir la machine à vecteur support et notre modèle de réseaux de neurones siamois.

### 6.4.2 Comparaison de notre modèle avec les CNN

Le document [32] présente une méthode proposée par Cozzens et al pour l'analyse de signatures de chèques à l'aide de réseaux de neurones convolutifs (CNN). L'architecture du CNN est simple et utilise la bibliothèque Keras en Python, basée sur TensorFlow, pour améliorer la qualité de l'image de la signature. La méthode utilise un système de comparaison d'images qui repose sur un système de classification d'images. Lorsqu'une signature est soumise au programme, elle est comparée à d'autres caractéristiques de signature portant la même étiquette. La principale contribution de cette méthode est la détection et la réduction de la contrefaçon, en particulier dans le secteur bancaire. Ils ont obtenu un taux de réussite de 83,93%.

Jivesh Poddar et al[33] ont proposé une méthode pour la vérification de signatures. Cette méthode comprend un prétraitement pour faciliter la vérification de signature, l'utilisation d'un réseau de neurones convolutifs (CNN) et d'un modèle basé sur l'algorithme Crest-Trough pour le système de vérification de signature, ainsi qu'un modèle basé sur Harris et Surf pour la détection de la contrefaçon dans la signature. Leur système proposé a atteint une précision de 85-89 % pour la détection de falsification.

| Méthodes | Précision |
|---|---|
| Réseaux de neurones siamois | 92% |
| CNN [32] | 83,93% |
| CNN [33] | 85-89% |

Tableau 3:Évaluation des performances de notre modèle avec les CNN

### 6.4.3 Comparaison de notre modèle avec les réseaux de neurones siamois

Abolfazl [6] a utilisé les réseaux de neurones siamois pour développer un système de classification de signatures statique. Son approche repose sur la comparaison à N voies, où l'image de la requête est comparée avec N autres classes, y compris ses propres échantillons de classes. Au cours de cette comparaison, le modèle produit un score de similarité. Les résultats obtenus ont montré une précision d'environ 90% pour les réseaux de neurones siamois, en utilisant une base de données néerlandaise.

Abolfasl [6] a également utilisé un autre jeu de données composé de signatures chinoises. Pour chacune des 20 classes, il a attribué 6 images pour l'entraînement, 3 pour les tests et 3 images pour la validation. Les résultats ont montré une précision de 84% pour les réseaux de neurones siamois. La comparaison à N voies présente des inconvénients

comme : la sensibilité aux variations intra-classe, le défi de l'évolutivité et le coût de calcul élevé.

| Méthode | Précision |
|---|---|
| Réseaux de neurones siamois | 92% |
| Réseaux de neurones siamois [6] avec une base de données néerlandaise | 90% |
| Réseaux de neurones siamois [6] avec une base de données chinoise | 84% |

Tableau 4: Évaluation des performances de notre modèle avec d'autres réseaux de neurones siamois.

En conclusion, notre modèle présente de bons résultats en termes de détection de signatures falsifiées, surpassant les machines à vecteur de support, les k-voisins les plus proches, les réseaux de neurones convolutifs (CNN) et les travaux précédents sur les réseaux de neurones siamois.

## 6.5 Discussion

Les limitations dans notre étude, comme dans tout problème de classification, sont liées aux données traitées. En effet, l'étape la plus cruciale avant l'entraînement des données est la préparation et le prétraitement des données. Nous jugeons donc qu'il est essentiel de consacrer le temps nécessaire à la phase de préparation et de prétraitement pour construire une base de données de qualité qui soit bien exploitée pour tout problème d'apprentissage.

Des ajustements et des améliorations sont également utiles pour obtenir des performances optimales. L'augmentation des données pendant la phase d'apprentissage est utile sous certaines conditions. Nous avons comparé nos résultats avec les travaux d'autres chercheurs, et on remarque que notre modèle (les réseaux de neurones siamois) donne des résultats très satisfaisants.

## 6.6 Conclusion

Dans ce chapitre, nous avons présenté l'ensemble des résultats obtenus tout au long de notre mémoire. Nous avons également comparé nos résultats avec les travaux d'autres chercheurs afin de comparer les performances. Nous pouvons déduire comme suite aux différentes comparaisons que la vérification de signatures manuscrites utilisant les réseaux de neurones siamois possède des performances importantes.

# Chapitre 7 : Conclusion générale et perspective

Dans ce mémoire, nous avons réalisé un système de vérification de signatures manuscrites statiques grâce à la méthode des réseaux de neurones siamois. D'après nos expériences, les réseaux siamois se révèlent efficaces dans la vérification de signature avec un faible nombre d'échantillons de signatures. Nous avons obtenu une précision de 92 % dans la vérification de signatures en utilisant des réseaux siamois qui se basent sur une comparaison par paire.

Les moyens techniques que nous avons utilisés pour notre mémoire sont à la fois disponibles et accessibles partout où l'on dispose d'une connexion Internet. Les technologies que nous avons utilisées, comme les langages de programmation et les librairies, sont gratuitement accessibles. Bien que le traitement par les réseaux de neurones siamois nécessite une configuration avancée, les services cloud tels que Google Colab, la plateforme que nous avons utilisée, facilitent considérablement la tâche.

La vérification de signature à l'aide d'un réseau siamois présente l'avantage qu'un seul modèle puisse être entraîné pour tous les utilisateurs, ou bien un modèle distinct par utilisateur, tout en offrant un niveau élevé de précision d'environ 88 %. En outre, étant donné que la vérification est déterminée sur la base des distances entre deux signatures (la signature de référence et la signature questionnée), le modèle entraîné à l'aide d'un réseau siamois est évolutif. En d'autres termes, si une nouvelle classe est ajoutée à l'ensemble de données qui n'a pas été entraîné, il est toujours possible d'utiliser le réseau siamois pour calculer l'image questionnée en la comparant avec une image de référence dans la nouvelle classe.

Dans une optique de perspective d'avenir, nous envisageons d'abord d'améliorer nos résultats en effectuant une analyse encore plus approfondie des caractéristiques que nous pourrions ajouter ou supprimer afin de réduire les interférences dans nos caractéristiques sélectionnées et ainsi avoir un impact positif sur nos performances. Ensuite, nous prévoyons développer une application web qui déterminera automatiquement si une signature est authentique ou falsifiée en se basant sur les signatures fournies à l'application. L'interface affichera également le score de prédiction.

# RÉFÉRENCES


[1] Érick ADJE (2018). Étude d'un système de vérification de signature dynamique avec les réseaux de neurones

[2] Jain. A.K, Griess.F.D et Connell.S.D., « On Line Signature Verification », Pattern Recognition letters, vol 35, 2002, p. 2663-2972.

[3] LAHYANE ADIL (2002). Vérification des signatures manuscrites

[4] Benjamin Thiry (Janvier 2012). a signature (pp.81-96) Chapter: Approche graphologique de la signature Publisher: Frederic Bravo Editors: Presses Universitaires de Bordeaux.

[5] https://dekalbmiller.com/forensic-handwriting-analysis/

[6] Abolfazl Vashee (2022). Classification des signatures hors lignes

[7] https://www.stepover.com/us/solutions/ consulté le 17 janvier 2023

[8]https://www.uottawa.ca/archives/sites/www.uottawa.ca.archives/files/digital_signature_-_communication_v2_-_french.pdf consulté le 24 janvier 2023

[9] https://www.nectardunet.com/12739/david-chaum-cryptographe-americain-origine-1er-systeme-cryptographique-appele-ecash/ consulté le 24 janvier 2023

[10]https://www.cakewalk.com/Documentation?product=Music%20Creator%206%20Touch&language=2&help=Notation.41.html

[11] American Heritage Dictionary, 3rd Edition, ver. 3.6 a, (SoftKey Intl. Inc., 1994).

[12] Gubta G. and McCabe A., A Review of Dynamic Handwritten Signature Verification. Department of Computer Science, James Cook University Townsville, Qld 4811, Australia, (1997).

[13] Jain. A.K, Pankanti.S et Bolle.R., « Biometrics », New York, Édition Kluwer, 1999.

[14] R. Plamondon and F. J. Maarse, An evaluation of motor models of handwriting. Systems, Man and Cybernetics, IEEE Transactions on, 19(5) :1060–1072, 1989.

[15] R. A. Huber and A. M. Headrick, Handwriting Identification : Facts and Fundamentals. CRC Press, Boca Roton, 1999.

[16] C. Simon*, E. Levrat*, J. Brémont*, R. Sabourin**. Codage d'images de signatures manuscrites pour la vérification hors ligne

[17] Anatolyevich Kholmatov Alisher, Biometric Identity Verification Using On-Line and Off-Line Signature Verification. MSc Sabanci University, (2003).

[18] Ashwini P. and Shalini B., Handwritten Signature Verification using Neural Network. International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 1– No.2, January 2012 – www.ijais.org

[19] https://www.sicara.fr/parlons-data/machine-learning#:~:text=Les%20d%C3%A9buts%20du%20Machine%20Learning,dames%20au%20fil%20des%20parties.

[20] L'essentiel à savoir pour débuter en machine learning : JAFWIN

[21] https://www.sicara.fr/parlons-data/deep-learning#:~:text=%C3%A0%20la%20r%C3%A9alit%C3%A9.-,L'histoire%20du%20Deep%20Learning,repr%C3%A9sentation%20informatique%20de%20cerveau%20humain.

[22] https://www.ibm.com/topics/deep-learning

[23] Comprendre les Support Vector Machines (SVM) (larevueia.fr) consulté le 6 février 2023.

[24]https://www.ibm.com/cafr/topics/knn#:~:text=L'algorithme%20des%20k%20plus%20proches%20voisins%2C%20%C3%A9galement%20connu%20sous,un%20point%20de%20donn%C3%A9es%20individuel. Consulté le 7 février 2023

[25] Cavalcanti, G. D. D. C., Rodrigo C. Doria, and E. Cde BC Filho. "Feature selection for off-line recognition of different size signatures." Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing. IEEE, 2002.

[26] Oliveira, Luiz S., et al. "The graphology applied to signature verification." 12th conference of the international graphonomics society. 2005.

[27] Zhang, Bailing. "Off-line signature verification and identification by pyramid histogram of oriented gradients." International Journal of Intelligent Computing and Cybernetics (2010).

[28] Harfiya, Latifa Nabila, Agus Wahyu Widodo, and Randy Cahya Wihandika. "Offline signature verification based on pyramid histogram of oriented gradient features." 2017 1st International Conference on Informatics and Computational Sciences (ICICoS). IEEE, 2017.

[29] M. Ashwini. S. Armand. and Muthukkumarasamy, Off-line Signature Verification using the Enhanced Modified Direction Feature and Neural basedClassification. International joint Conference on Neural Networks, 2006

[30] B. Herbst. J. Coetzer. and J. Preez, Online Signature Verification Using the Discrete Radon Transform and a Hidden Markov Model. EURASIP Journal on Applied Signal Processing, vol. 4, pp. 559–571, 2004

[31] T.S. Enturk. E. O zgunduz. and E. Karshgil, Handwritten Signature Verification Using Image Invariants and Dynamic Features. Proceedings of the 13th European Signal Processing Conference EUSIPCO 2005,Antalya Turkey, 4th-8th September, 2005.

[32]Brittany Cozzens et Richard Huanget Maxwell Jayet Felix Zhanet Mark Zhang: OCNN pour d´etecter et r´eduire la falsification des signatures, 2017.

[33] Jivesh Poddar et Vinanti Parikh et Santosh Kumar Bharti: Offline Signature Recognition and Forgery Detection using Deep Learning,2020.

[34] Hubel, D. H., & Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. J Physiol, 148, 574-591. doi:10.1113/jphysiol.1959.sp006308

[35] Fukushima, K. (1980). Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol Cybern, 36(4), 193-202. doi:10.1007/BF00344251

[36] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324. doi:10.1109/5.726791

[37] Paul Blanc-Durand (2018). Réseaux de neurones convolutifs en médecine nucléaire : Applications à la segmentation automatique des tumeurs gliales et à la correction d'atténuation en TEP/IRM.

[38] Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd Edition ed.): O'Reilly Media, Inc. ISBN : 9 781 492 032 649

[39] Bengio, Y., Goodfellow, I., & Courville, A. (2017). Deep learning (Vol. 1) : MIT press Massachusetts, USA :.

[40] Bishop, C. M. (1995). Neural networks for pattern recognition: Oxford university press.

[41] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90. doi:10.1145/3065386

[42] Szegedy, C., Wei, L., Yangqing, J., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going deeper with convolutions. Paper presented at the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[43] BROMLEY J., GUYON I., LECUN Y., SÄCKINGER E. & SHAH R. (1994). Signature verification using a" siamese" time delay neural network. In Advances in Neural Information Processing Systems, p. 737–744.

[44] Dey, Dutta, Toledo (2017). SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification

[45] Koch, Gregory, Richard Zemel, and Ruslan Salakhutdinov. "Siamese neural networks for one-shot image recognition." ICML Deep Learning workshop. Vol. 2. 2015.

[46] https://pytorch.org/vision/stable/index.html

[47] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout : A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Researc, 15, 1929-1958.

[48] Mohamed Amine Bouhedadja (2021). Utilisation de l'apprentissage profond pour la classification des mauvaises herbes dans le bleuet nain

[49]https://aws.amazon.com/fr/whatis/overfitting/#:~:text=Le%20surajustement%20est%20un%20comportement,pas%20pour%20les%20nouvelles%20donn%C3%A9es.