REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 8 Mai 1945 – Guelma
Faculté des Sciences et de la Technologie
Département de Génie Electrotechnique et Automatique

Réf:……/2022

MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER Académique**

Domaine : Sciences et Technologie

Filière : Automatique

Spécialité : Automatique et Informatique industrielle

Par : HAMICI Abderraouf et **MERABTI Mohammed El arbi**

Thème

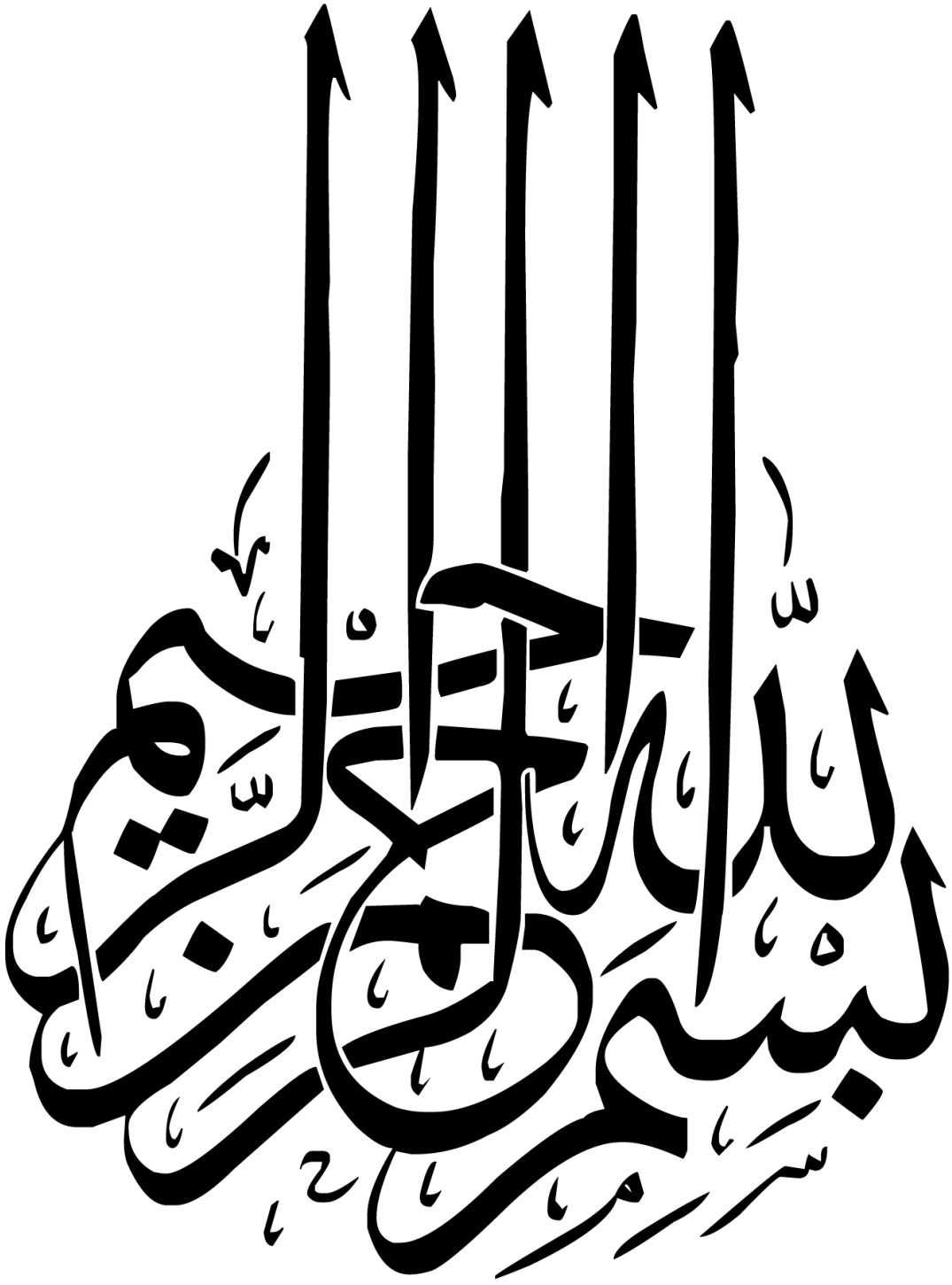
Réalisation d'un système d'authentification des individus par signature manuscrite

Soutenu publiquement, le 19/06 /2022.

Devant le jury composé de :

M^{me}. BOUCERREDJ Leila	MCA	Univ. Guelma	Encadreur
Mr. BOUDJEHEM Badreddine	Pr	Univ. Guelma	Président
Mr. DEBECHE Mehdi	MAA	Univ. Guelma	Examinateur

Année Universitaire : 2021/2022


بسم الله الرحمن الرحيم

REMERCIEMENTS

Nous remercions tout d'abord, ALLAH qui nous a donné la force et le courage, afin d'accomplir ce travail.

Nous tenons à remercier sincèrement notre encadreur Madame Bouccerredj Laila, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.

Nous tenons à remercier les membres du jury pour le temps précieux qu'ils ont consacré à la lecture et à l'évaluation de ce projet.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui ont toujours soutenu et encouragé au cours de la réalisation de ce travail.

Merci à toutes et tous.

Dédicaces

A mon père Noureddine et ma mère Fatiha en témoignage de leur affectation, leurs sacrifices et de leurs précieux conseils qui m'ont conduit à la réussite dans mes études.

A mes chers frères Mehdi, Nassim et Nedjmeddine, avec tous mes souhaits de succès dans leur vie.

A toute ma famille pour leurs soutiens et encouragements.

À tous les professeurs et enseignants qui m'ont suivi durant tout mon cursus scolaire et qui m'ont permis de réussir dans mes études.

A tous mes respectueux collègues.

A mes chers amis
et à tous ceux que j'aime.

Abderraouf

RÈSUMÈ

La signature manuscrite est depuis plusieurs siècles le moyen le plus répandu. Elle estle moyen biométrique d'authentification le plus utilisé et accepté. La signature manuscrite d'un individu représente un bon compromis, tout en étant relativement fiable, elle est facile à acquérir, socialement acceptée comme un mode de reconnaissance. La signature est un moyen utilisé depuis longtemps, pour authentifier des documents, pour responsabiliser les individus face à des engagements (contrats, etc.). La signature est donc reconnue comme mode de validation associé à l'identité d'une personne.

Notre travail de projet de fin d'étude propose un réseau de neurones à apprentissage profond pour apprendre sur un ensemble de données des signatures manuscrite. Nous utilisons le deep learning, qui se base sur une analyse des informations acquise dans la méthode de prétraitement, ensuite l'extraction des caractéristiques biométriques et puis tracer les courbe ROC en utilisant le logiciel Matlab pour visualiser les résultats à vouloir obtenir. Après, Les résultats sont pris et discutés.

Mots-clés : Biométrie, signature manuscrite, processus d'authentification, L'apprentissage en profondeur, réseau de neurone convolutif, Deep Learning, Particle Swarm Optimization et Local Phase Quantization .

ABSTRACT

The handwritten signature has been the most widespread means for several centuries. It is the most widely used and accepted biometric means of authentication. The handwritten signature of an individual represents a good compromise, while being relatively reliable, easy to acquire, socially accepted as a mode of recognition. The signature is a means used for a long time, to authenticate documents, to make individuals responsible for commitments (contracts, etc.). The signature is therefore recognized as a mode of validation associated with the identity of a person.

Our graduation project work proposes a deep learning neural network to learn on a data set of handwritten signatures. We use deep learning, which is based on an analysis of the information acquired in the pre-processing method, then the extraction of biometric characteristics and then plotting the ROC curves using Matlab software to visualize the results to be obtained. Afterwards, the results are taken and discussed.

Keywords : Biometrics, Handwritten Signature, Authentication Process, Deep Learning, Convolutional Neural Network, Deep Learning, Particle Swarm Optimization and Local Phase Quantization.

ملخص

كان التوقيع بخط اليد هو الوسيلة الأكثر انتشارًا لعدة قرون. إنها وسيلة المصادقة البيومترية الأكثر استخدامًا والمقبولة. يمثل التوقيع المكتوب بخط اليد للفرد حلاً وسطًا جيدًا، في حين أنه موثوق نسبيًا، ويسهل الحصول عليه، ومقبول اجتماعيًا كوسيلة للاعتراف. التوقيع هو وسيلة تستخدم لفترة طويلة، لتوثيق المستندات، لجعل الأفراد مسؤولين عن الالتزامات (العقود، وما إلى ذلك). لذلك يتم التعرف على التوقيع على أنه طريقة للتحقق مرتبطة بهوية الشخص.

يقترح عمل مشروع التخرج لدينا شبكة عصبية للتعلم العميق للتعلم على مجموعة بيانات من التوقيعات المكتوبة بخط اليد. نحن نستخدم التعلم العميق، الذي يعتمد على تحليل المعلومات المكتسبة في طريقة المعالجة المسبقة، ثم استخراج الخصائص الحيوية ثم رسم منحنيات ROC) باستخدام برنامج (Matlab) لتصور النتائج التي سيتم الحصول عليها. بعد ذلك، يتم أخذ النتائج ومناقشتها.

كلمات مفتاحية: القياسات الحيوية، التوقيع بخط اليد، عملية المصادقة، التعلم العميق، الشبكة العصبية التلافيفية ، التعلم العميق ، تحسين سرب الجسيمات وتقدير المرحلة المحلية.

Sommaire

Chapitre II : Les Algorithmes Utilisés Pour La Reconnaissance Des Signatures Manuscrite

Chapitre III : Authentification *Par Depp Learning*

Chapitre IV : *Système D'authentification Des Individus Par Signature Manuscrite*

Liste de figures

Chapitre I : Généralité Sur La Biométrie

Chapitre II : Les Algorithmes Utilisés Pour La Reconnaissance Des Signatures Manuscrite

Chapitre III : Authentification Par Depp Learning

Chapitre IV : Système D'authentification Des Individus Par Signature Manuscrite

Liste des tableaux

Abréviations

SVM : Support Vector Machine

CNN : Réseaux neuronaux convolutifs

KNN : k near est Neighbors

LPQ : Quantification de la phase locale

PSO : Particle Swarm Optimization

ReLU : L’unité linéaire redressée

BGD : Batch Gradient Descent

SGD : Descente de gradient stochastique

Introduction générale

Introduction générale

De nos jours, la technologie joue un rôle important dans la vie quotidienne de l'humanité. Les entreprises, le domaine d'administration, public ou privé, tentent de progresser de jours en jours suivant la technologie. Quel que soit le secteur concerné, la technologie apporte son lot d'amélioration.

En parlant d'administration ou d'entreprise, la signature manuscrite est l'un des outils très indispensable pour son fonctionnement. On l'utilise dans les documents, dans les contrats, dans les actes ou d'autres pièces faisant preuves des faits. Elle sert juridiquement d'approbation. Une signature sur une pièce met donc en conséquent le signataire responsable.

La biométrie désigne l'ensemble des technologies de reconnaissance morphologique et comportementales et biologique des individus telles que : les empreintes digitales, l'iris, la signature manuscrite, démarche, l'ADN…etc.

Dans les applications de contrôle d'accès, la biométrie permet d'apporter un niveau de sécurité supérieur en ce qui concerne des accès logiques (ordinateurs, comptes bancaires, etc.) ou des accès physiques (bâtiments sécurisés, aéroports, laboratoires etc.). La biométrie regroupe deux axes principaux : une identification (reconnaissance) et une authentification.

Dans le cas d'authentification, le système biométrique demande une information biométrique et la compare avec chaque information stockée dans la base de données. Alors que pour l'authentification l'utilisateur annonce son identité par une information biométrique, et le système compare les données obtenues à partir de l'information entrée avec la donnée enregistrée.

Il existe plusieurs techniques biométriques qui sont utilisées dans le contrôle d'accès, chaque technique biométrique a ses avantages et inconvénients. En général, si on rencontre ce genre de problème, on fait appel à des experts, car l'analyse d'une signature est un peu difficile. Pour cette raison, la signature manuscrite est un sujet intéressant pour apporter une solution d'usurpation et même solutionner à la sécurité d'accès d'un système. Le présent document est organisé comme suit :

Le premier chapitre : continent des généralités sur la biométrie, dans ce chapitre nous avons introduit l'architecture des systèmes biométriques, leurs modalités et leurs différentes applications.

Le deuxième chapitre : consacré à un aperçu sur la reconnaissance des formes. Tout d'abord, nous présentons les différents algorithmes utilisés pour la reconnaissance des signatures manuscrite. Également, ces algorithmes se devisent en plusieurs catégories, par exemple : les algorithmes de classification.

Dans le troisième chapitre : nous présentons plus en détails les concepts de l'authentification par signature manuscrite avec le Deep learning et les travaux réalisés dans ce domaine.

Le quatrième chapitre : est consacré pour le processus de notre programme d'authentification des signatures manuscrites et les résultats expérimentaux basé sur la méthode de Deep learning et CNN sous le logiciel Matlab.

Finalement, Nous terminons ce mémoire par une conclusion générale.

Chapitre I : Généralité *sur la Biométrie*

I.1.Introduction

Le besoin de reconnaissance de l'individu a toujours existé dans la société et s'est imposé tant dans son interaction quotidienne que dans les services qui la composent. Les personnes sont généralement identifiées les unes aux autres sur la base de caractéristiques physiques (par exemple, le visage, la voix, la posture) ainsi que d'autres informations connexes (par exemple, la nationalité, la langue, les vêtements). Ces caractéristiques constituent son identité. Les services qui fournissent un accès électronique (tels que les fichiers, le courrier, les banques, les services universitaires) ainsi que physique (accès aux pays, aux sites de sécurité étatiques ou non étatiques) nécessitent une gestion des identités plus sécurisée.

Dans ce chapitre nous commençons par présenter la biométrie de manière générale ainsi que les diverses applications qui en découlent.

I.2. Généralités sur la biométrie

I.2.1. Définition de la biométrie

Le terme de biométrie est originaire d'une contraction des deux anciens termes grecs : « bios » qui signifie : la vie « métrique » qui se traduit par : mesure. C'est-`a-dire « mesure du vivant ». La biométrie consiste à vérifier ou déterminer l'identité d'un individu à partir de ses caractéristiques biologiques [1].

I.2.2. Caractéristiques biométriques

Le choix des caractéristiques physiques est important. Il faut qu'elles soient toutes à la fois [2][3] :

- Universelles : existent chez tous les individus.
- Uniques : possibilité de différencier un individu par rapport à un autre.
- Permanentes : stables et invariantes au cours du temps.
- Enregistrables : possibilité d'enregistrer les caractéristiques d'un individu à l'aide d'un capteur approprié qui ne cause aucun dérangement pour l'individu.
- Performance : Signifie que l'authentification doit être précise et rapide.

I.3. Les modalités biométriques

La biométrie est basée sur les caractéristiques biométriques de l'individu, ces caractéristiques peuvent encore être divisé en trois catégories :

- L'analyse morphologique ou physique (empreintes digitales, géométrie de la main, visage...).

- L'analyse comportementale (la signature, dynamique de frappe au clavier, la voix…).

- Les traces biologiques (ADN, odeur, salive...).

La figure 1.2 présente les trois catégories des biométries avec quelques exemples.

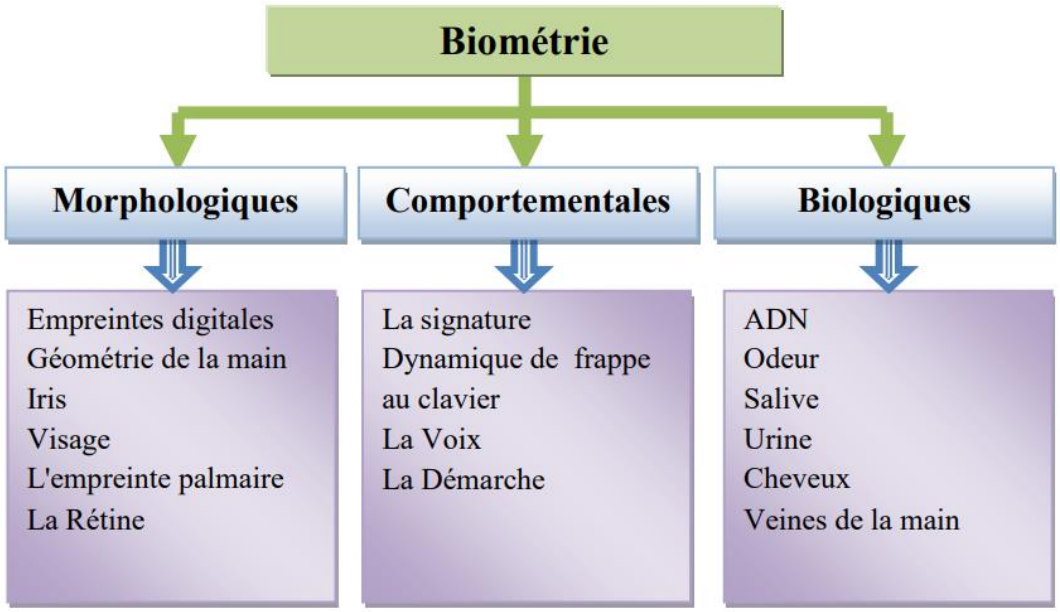


Figure I. 1: Classification d'un certain nombre de modalités biométriques [4].

I.3.1. Modalités morphologiques (physiologiques)

Enfin, nous ferons une brève description des caractéristiques biométriques les plus courantes collectées.

I.3.1.1. Empreinte digitale

La caractéristique biométrique la plus couramment utilisée pour l'identification depuis des centaines d'années, principalement par la police et les services gouvernementaux. L'empreinte digitale peut parfois ne pas être collectée si le doigt présente des pointes, des coupures, des blessures.

Figure I. 2: Système biométrique basé sur les empreintes digitales.

I.3.1.2. Iris

L'iris de l'œil est la partie colorée qui entoure l'extérieur de la sclère (blanc) et l'intérieur de la pupille. Il se stabilise en couleur autour des deux premières années de développement. Sa cohérence complexe contient beaucoup d'informations et est due à des facteurs génétiques.

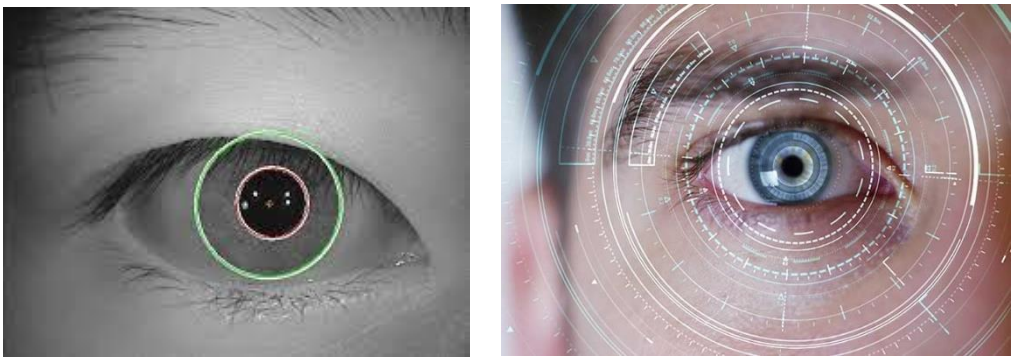


Figure I. 3: Système biométrique basé sur l'Iris.

I.3.1.3. Visage

Le Visage utilisé instinctivement par les gens pour se reconnaître. La plupart des documents qui indiquent l'identité d'une personne comprennent une photographie de celle-ci, dont la prise dépend de restrictions (telles que l'éclairage, l'angle de vue, la couleur). Différents angles de capture d'une même personne, plusieurs fois, la rendent difficile à identifier par les systèmes automatiques. Ces dernières années, des techniques ont été utilisées qui compliquent la collecte d'informations précieuses en examinant la géométrie du visage.

Figure I. 4: Le visage de l'être humain en tant que modalité biométrique.

I.3.1.4. La rétine

La biométrie par la rétine procure un haut niveau en matière de reconnaissance. Cette technologie est bien adaptée pour des applications de haute sécurité (sites militaires, salles de coffres forts, etc..). La disposition des veines de la rétine assure une bonne fiabilité et une haute barrière contre la fraude.[10]

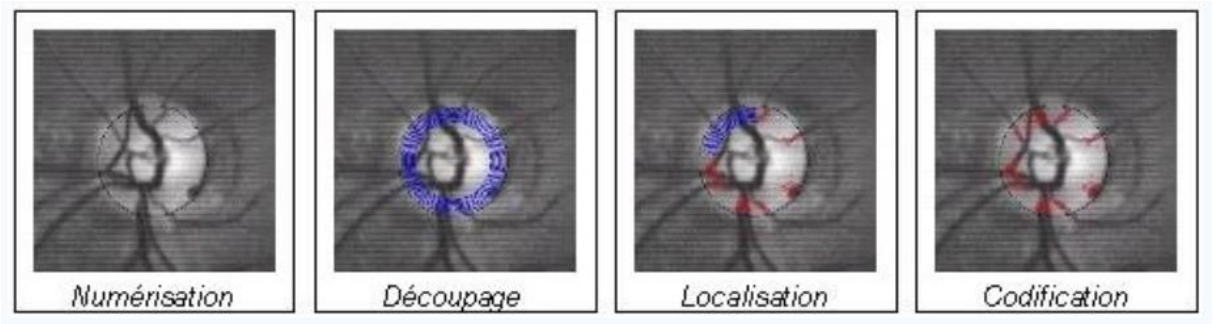


Figure I. 5: Système biométrique basé sur la rétine.

I.3.1.5. Géométrie de la main

Selon cela, la main est examinée du poignet jusqu'aux extrémités et les informations sont extraites en fonction de l'épaisseur des doigts, de la distance entre eux et de la forme de la paume. Elle s'applique dans les cas où les utilisateurs du système sont relativement peu nombreux, tant en termes de volume de données que de possibilité de trouver des échantillons assez similaires les uns aux autres.

Figure I. 6: Système biométrique basé sur la géométrie de la main.

I.3.2. Modalités comportementales

I.3.2.1. La signature

La signature manuscrite fait partie de la modalité comportementale de la biométrie.

Elle appartient à l'heure actuelle au sujet d'un système de reconnaissance biométrique qui ne cesse d'évoluer et utiliser pour la sécurité, le plus souvent d'un Etat ou d'une administration.

Une signature manuscrite peut être considérée comme un tracé graphique fait par un individu sur un support. [5] Elle peut être modifiée avec le temps et est sensible à la fraude.

Figure I. 7: Système biométrique basé sur la signature manuscrite.

I.3.2.2. Démarche (Posture)

La reconnaissance de pose étudie la marche et les mouvements d'une personne afin de la reconnaître. Une autre méthode avec celle du visage qui est également utilisée par les gens instinctivement. Le capteur de prélèvement d'échantillon nécessite un certain temps,

évidemment aussi bien dans la phase d'enrôlement que dans la phase d'identification. Plusieurs facteurs affectent la posture tels que les chaussures, les vêtements, le sol et l'état physique des pieds (blessures, inconfort).

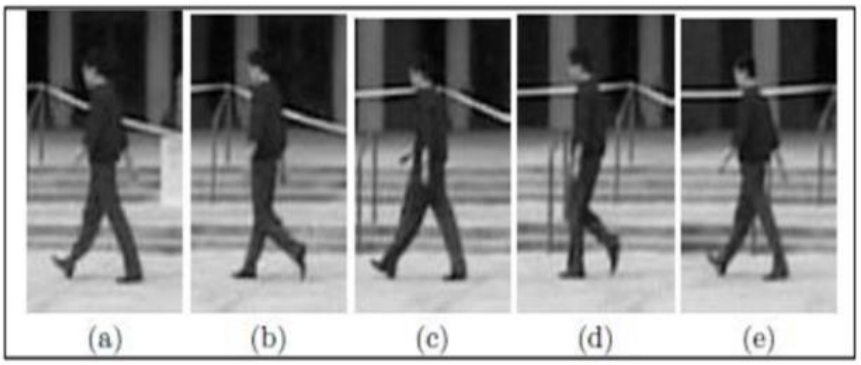


Figure I. 8. Images sur la démarche [6].

I.3.2.3. La voix

La voix est trait unique de chaque individu. En effet, une grande partie de cette caractéristique est d'exterminée par le conduit vocal ainsi que les cavités buccales. La voix n'est pas un attribut permanent. Elle change bien entendu avec l'âge mais peut être aussi affectée temporairement par l´état de santé ou émotionnel du locuteur.

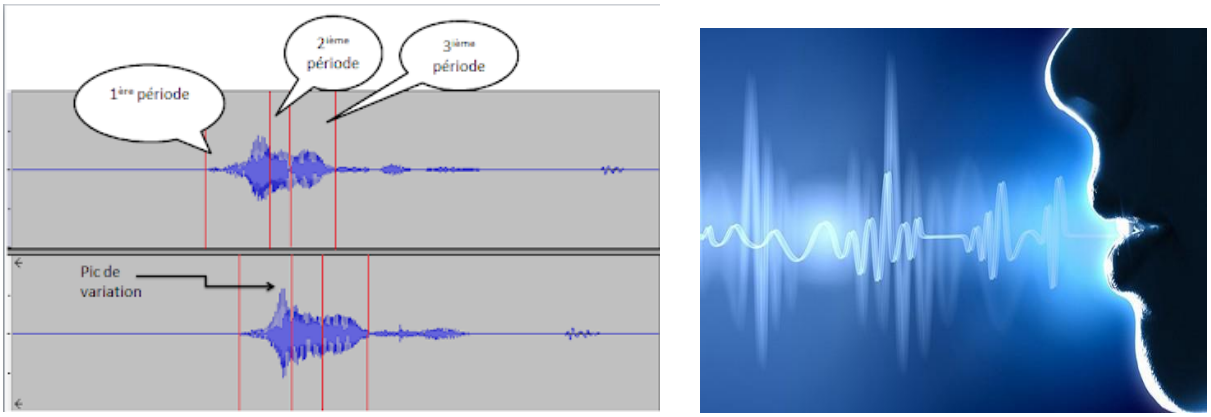


Figure I. 9 : Fonctionnement de reconnaissance vocale [7].

I.3.2.4. Dynamique de frappe au clavier

C'est une modalité biométrique comportementale qui permet d'authentifier des individus selon leur façon de taper au clavier. Un tel système est peu coûteux, car il ne nécessite pas de matériel d'acquisition autre que le clavier de l'ordinateur, et est facilement accepté par l'utilisateur. Nous nous sommes principalement intéressés aux systèmes statiques où le texte saisit par l'utilisateur est connu à l'avance par la machine [6].

Figure I. 10: Images sur le dynamique de frappe au clavier.

I.3.3. Modalités biologiques

I.3.3.1. Veine de la main

Le principe est d'identifier l'unicité du dessin du réseaux veineux. Le capteur biométrique émet une lumière infrarouge. La lumière infrarouge permet de mettre en avant l'hémoglobine contenue dans le sang des veines. En effet, les tissus humains sont relativement transparents à la lumière infrarouge. Cette fréquence de lumière peut donc traverser plusieurs centimètres de tissus.

Figure I. 11: Image de système configuration des veines.

I.3.3.2. ADN

L'ADN est un trait génétique qui est unique (sauf dans le cas des jumeaux monozygotes) et qui permet d'identifier quelqu'un. Trois facteurs rendent son utilisation controversée : (i) il peut être facilement collecté par quelqu'un sans le savoir et en abuser, (ii) son traitement de pointe nécessite un mélange expert et du temps, (iii) il donne accès à des informations sensibles , comme la susceptibilité de l'individu à la maladie. Il est principalement utilisé dans les enquêtes criminelles et dans les cas de reconnaissance de parenté.

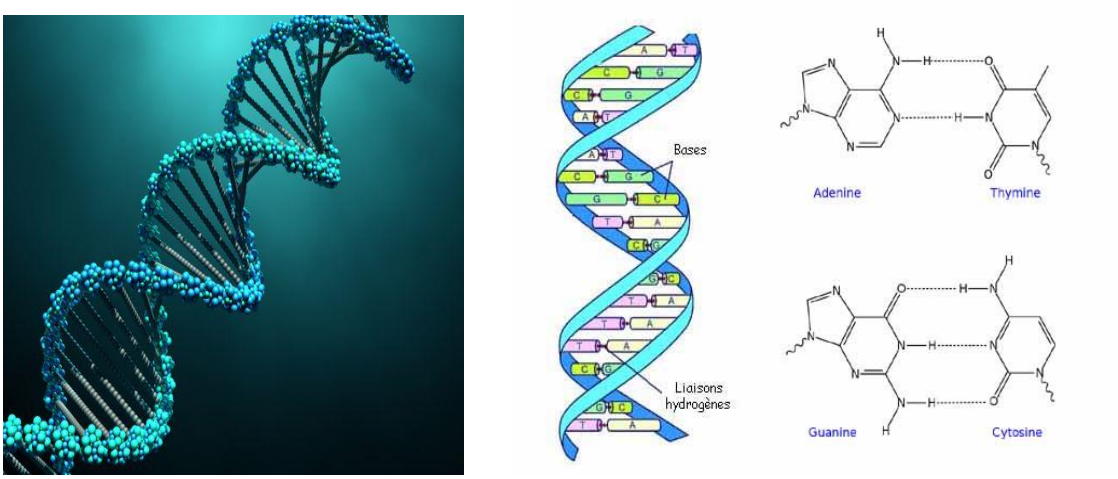


Figure I. 12: Images d'ADN.

I.4. Avantages et inconvénients de la vérification biométrique

I.4.1. Les avantages

- **Confort d'utilisation**

La biométrie offre une bien meilleure expérience utilisateur. C'est beaucoup plus rapide de déverrouiller les appareils, cela prend quelques secondes plutôt que de taper de longs mots de passe avec plusieurs caractères spéciaux.

Cela élimine la frustration d'oublier un mot de passe et de devoir réinitialiser.

Il n'est plus nécessaire de disposer d'un journal de bord pour savoir qui se trouve dans un bâtiment à un moment donné.

Les cartes d'identité ne sont plus nécessaires, il n'y a donc pas de problème à l'avoir oubliée ou à devoir en transporter une.

- **Une plus grande sécurité**

Les mots de passe, les codes PIN et les questions d'identification personnelle telles que le nom de jeune fille d'une mère sont menacés de violation de données, risquant l'accès des fraudeurs qui conservent les réponses. L'authentification biométrique est un obstacle à cela.

La biométrie, comme les modèles de visage, les empreintes digitales, le balayage de l'iris et autres, est beaucoup plus difficile à reproduire avec la technologie actuelle.

Cela élimine le danger des attaques criminelles « par-dessus l'épaule ».

- **Non transférable**

La biométrie n'est pas transférable, ce qui permet aux entreprises de mieux contrôler l'accès. Par exemple, les cartes d'identité donnant accès à l'entrée dans un bâtiment ou un code d'entrée pour un gymnase sont faciles à partager, mais avec l'utilisation de la biométrie, ce n'est pas possible.[8]

I.4.2. Les inconvénients

- **Coûts**

Coût initial considérable pour l'installation et la configuration du système.

Le stockage et la maintenance des données biométriques nécessitent des mesures de sécurité supplémentaires.

L'intégration de la biométrie dans un programme peut être relativement complexe par rapport au déploiement de solutions basées sur des mots de passe.

- **Sécurité des données**

Une sécurité accrue est nécessaire car contrairement à un mot de passe ou à un code PIN, la biométrie ne peut pas être modifiée, donc si quelqu'un obtient ces informations, ils les ont à vie.

Difficile de garder une longueur d'avance sur les progrès de la fraude pour s'assurer qu'aucune violation de données ne peut être commise.

Les erreurs techniques bloquant l'accès rendent la tâche difficile si un mot de passe de sauvegarde n'est pas disponible. Pire encore, des erreurs technologiques permettant l'accès pourraient facilement donner accès à un fraudeur.

C'est plus simple pour les voleurs de forcer l'accès, par exemple en tenant un téléphone devant le visage de quelqu'un pour y accéder plutôt que d'avoir besoin d'un mot de passe écrit.

- **Suivi**

La biométrie risque de laisser un enregistrement numérique permanent qui fait planer la menace d'un suivi potentiel par les autorités gouvernementales. Les données peuvent devenir une étiquette numérique qui peut être utilisée pour identifier et suivre les individus pendant toute leur vie.[8]

I.5 Domaines d'application

Le champ d'application de la biométrie est très vaste. En effet, tous les domaines qui nécessitent de vérifier ou déterminer l'identité d'une personne sont concernés. D'où les applications de la biométrie peuvent être divisées en trois groupes principaux [8].
- **Applications commerciales:** telles que l'ouverture d'un réseau informatique, la sécurité de données électroniques, l'e-commerce, l'accès Internet, les cartes de crédit, le contrôle d'accès physique, le téléphone cellulaire, la gestion des registres médicaux, l'étude à distance, etc.
- **Applications gouvernementales**: telles que la carte d'identité nationale, le permis de conduire, la sécurité sociale, le contrôle des frontières, le contrôle des passeports, etc.
- **Applications légales**: telles que l'identification de corps, la recherche criminelle, l'identification de terroriste, etc..[9].

I.6. Architecture d'un système biométrique

En général, l'architecture d'un système biométrique se résume par deux modules : le module d'apprentissage et le module de comparaison. Le module d'apprentissage sert à créer des modèles qui servent de références pour la vérification ou l'indentification du module de comparaison. A chaque module existe plusieurs étapes indispensables et dépendantes les unes des autres [5].

La figure I.13 représente l'architecture d'un système de reconnaissance biométrique.

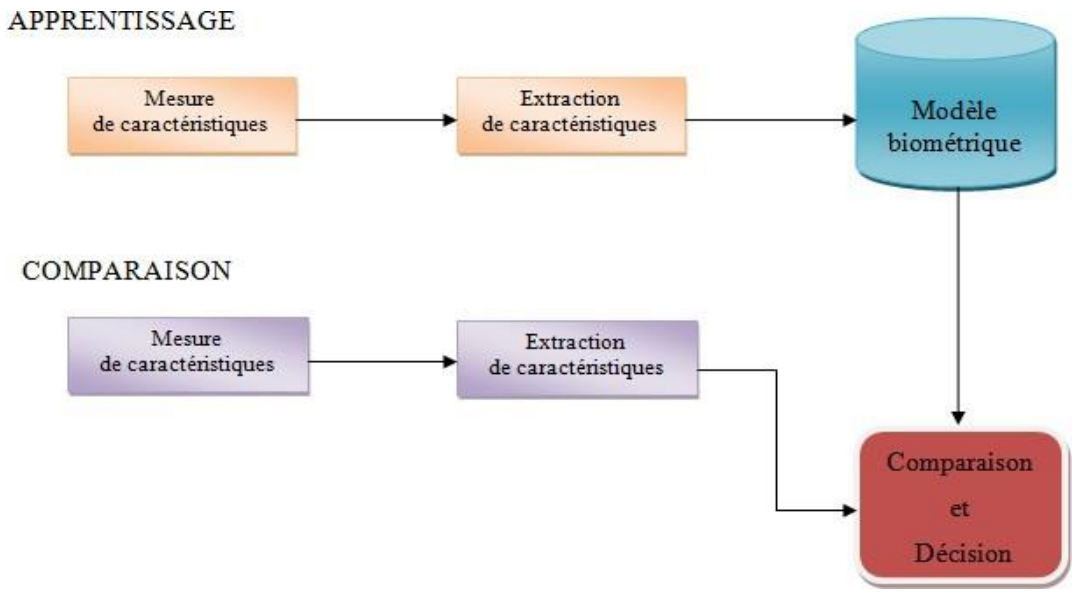


Figure I. 13: Architecture d'un système biométrique [5].

I.6.1. Module d'apprentissage

Le module d'apprentissage est un sous-système qui a pour but de créer un modèle biométrique pour chaque individu. Trois étapes se présentent en général pour cette fin : la mesure des caractéristiques, l'extraction des caractéristiques et la création du modèle biométrique. (Voir Figure I.13).

I.6.2. Mesure des caractéristiques

La première étape débute par la mesure des caractéristiques via un capteur. C'est la phase de capture appelée aussi la phase d'acquisition. Ceci produit un signal contenant les caractéristiques biométriques qui sera sujet à de futures extractions. Pour toutes modalités biométriques des capteurs spécifiques sont conçus. La figure 2.6 présente quelques exemples de capteurs.

I.6.3. Extractions des caractéristiques

Le signal obtenu de l'acquisition contient les informations biométriques mais aussi des informations inutiles. La phase d'extraction sert à extraire de ce signal acquis les informations utiles qui sont les caractéristiques ou les paramètres biométriques pertinents pour la création du modèle.

Le module d'extraction de caractéristiques prend en entrée les données biométriques acquises par le module de capture et extrait seulement l'information pertinente afin de former unenouvelle représentation des données. Généralement, cette nouvelle représentation est censée être unique pour chaque personne et relativement invariante aux variations intra-classes.

Le traitement du signal joue un rôle important dans cette partie, elle représente le meilleur moyen d'extraction.

La figure I.14, montre trois types de caractéristiques exportées pour différentes données biométriques. Ce processus est le sujet principal de ce travail.

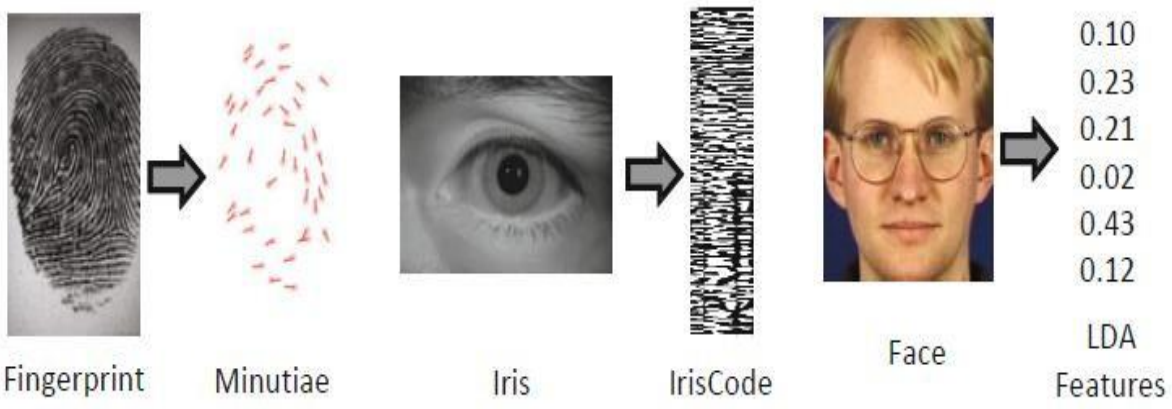


Figure I. 14: Données d'entrée du capteur et données extraites de celles-ci.

La forme de l'entité exportée dépend de la nature de l'entrée biométrique.

I.6.4. Construction des modèles

La phase construction des modèles crée et enregistre un modèle biométrique partir des modalités biométriques d'un individu. Un modèle biométrique (appelé aussi gabarit ou "Template") est l'ensemble des données utilisées pour représenter un utilisateur [5].

I.7. Conclusion

Dans ce chapitre, nous avons présenté quelques notions et définitions de base liées à la biométrie. On a présenté aussi les modalités biométriques. Chaque modalité biométrique à le champ d'application. La signature manuscrite possède des caractéristiques qui sont exploitables pour être sujet à une reconnaissance. Aussi son architecture ne sort pas de celui de la base de la biométrie, c'est le but de notre étude. Le deuxième chapitre est consacré à l'études des méthodes de classification.

Chapitre II : Les algorithmes utilisés pour la reconnaissance des signatures manuscrite

II.1. Introduction

Le système de reconnaissance faciale, comme tous les systèmes biométriques est constitué de trois étapes essentielles : prétraitement, l'extraction des caractéristiques et la classification. Dans la première étape de prétraitement, en utilise des algorithmes qui traitées les images de signatures pour faciliter l'extraction des caractéristiques. Ensuite dans La deuxième étape définir les algorithmes holistiques (linaire /non linaire) sert à traiter l'image de la signature afin d'extraire uniquement les caractéristiques biométriques, sous forme d'un vecteur, qui ensuite peuvent être utilisées pour reconnaitre l'individu. Ces caractéristiques sont uniques à chaque personne et stable. En fin la reconnaissance est faite par la comparaison (classification) du vecteur de caractéristique avec une base de données, c'est ce que nous cherchons à expliquer dans ce chapitre.

II.2. Les algorithmes utilisés pour la reconnaissance des signatures manuscrite

II.2.1. *Support Vector Machine (SVM)*

Le Support Vector Machine consiste à représenter la signature manuscrite comme des traits contenant des coordonnées x et y. Avec ces derniers on applique la recherche de similitude entre la signature à tester et la signature référence. Le SVM présente un résultat moins bon que les autres approches, il est préférable de le combiner avec d'autres méthodes.

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

Pour que le SVM puisse trouver cette frontière, il est nécessaire de lui donner des données d'entraînement. En l'occurrence, on donne au SVM un ensemble de points, dont on sait déjà si ce sont des carrés rouges ou des ronds bleus, comme dans la Figure 1. A partir de ces données, le SVM va estimer l'emplacement le plus plausible de la frontière : c'est la période d'entraînement, nécessaire à tout algorithme d'apprentissage automatique.

Une fois la phase d'entraînement terminée, le SVM a ainsi trouvé, à partir de données d'entraînement, l'emplacement supposé de la frontière. En quelque sorte, il a « appris » l'emplacement de la frontière grâce aux données d'entraînement. Qui plus est, le SVM est maintenant capable de prédire à quelle catégorie appartient une entrée qu'il n'avait jamais vue

avant, et sans intervention humaine (comme c'est le cas avec le triangle noir dans la Figure 1) : c'est là tout l'intérêt de l'apprentissage automatique. [10]

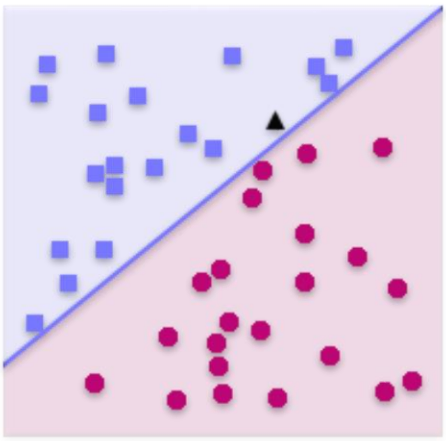


Figure II. 1. Notre SVM, muni des données d'entraînement.

Les carrés bleus et les ronds rouges déjà indiqués comme tels par l'utilisateur, a tranché : le triangle noir est en fait un carré bleu.

Comme vous pouvez le constater dans la figure ci-dessus, pour notre problème le SVM a choisi une ligne droite comme frontière. C'est parce que, comme on l'a dit, le SVM est un classificateur linéaire, la frontière trouvée n'est pas la seule solution possible, et n'est probablement pas optimale non plus.

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

Un des autres avantages des SVM, et qu'il est important de noter, est que ces derniers sont très efficaces quand on ne dispose que de peu de données d'entraînement : alors que d'autres algorithmes n'arriveraient pas à généraliser correctement, on observe que les SVM sont beaucoup plus efficaces. Cependant, quand les données sont trop nombreuses, le SVM a tendance à baisser en performance [10].

II.2.1.1. Les SVM dans les grandes lignes

Bon, nous savons donc que le but, pour un SVM, est d'apprendre à bien placer la frontière entre deux catégories. Mais comment faire ? Quand on a un ensemble de points d'entraînement,

il existe plusieurs lignes droites qui peuvent séparer nos catégories. La plupart du temps, il y en a une infinité… Alors, laquelle choisir ?

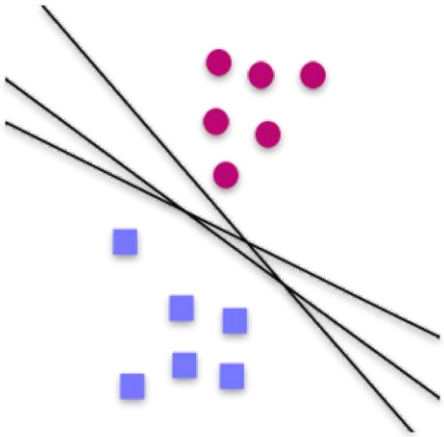


Figure II. 2. Ensemble de données d'entraînement.

Pour un ensemble de données d'entraînement, il existe plusieurs frontières possibles, comme le montre la figure II.2 ci-dessus.

Intuitivement, on se dit que si on nous donne un nouveau point, très proche des ronds rouges, alors ce point a de fortes chances d'être un rond rouge lui aussi. Inversement, plus un point est près des carrés bleus, plus il a de chances d'être lui-même un carré bleu. Pour cette raison, un SVM va placer la frontière aussi loin que possible des carrés bleus, mais également aussi loin que possible des ronds rouges.

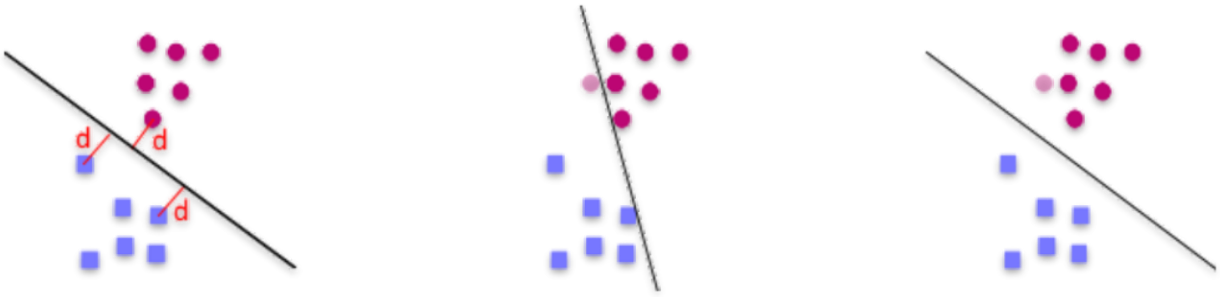


Figure II. 3. Maximisation de la distance 'd'entre la frontière.

A gauche, on maximise la distance 'd'entre la frontière et les points d'entraînement, au centre, une frontière non-optimale, qui passe très près des points d'entraînement, a droite, la frontière optimale (qui maximise 'd') classe bien le rond rouge clair comme un rond rouge.

Comme on le voit dans la figure II. 3, c'est bien la frontière la plus éloignée de tous les points d'entraînement qui est optimale, on dit qu'elle a la meilleure capacité de généralisation. Ainsi,

le but d'un SVM est de trouver cette frontière optimale, en maximisant la distance entre les points d'entraînement et la frontière.

Les points d'entraînement les plus proches de la frontière sont appelés vecteurs support, et c'est d'eux que les SVM tirent leur nom : SVM signifie Support Vector Machine, ou Machines à Vecteur Support en français. Support, parce que ce sont ces points qui « supportent » la frontière. Vecteurs, parce que… on en reparlera plus tard, dans les explications mathématiques.

Ça, c'est pour les principes de base. En théorie, ça suffit, mais en pratique… En effet, les SVM sont conçus de telle manière que la frontière est forcément, dans notre exemple, une ligne droite. Et ça, c'est loin d'être suffisant pour la plupart des cas.

Considérons l'exemple suivant :

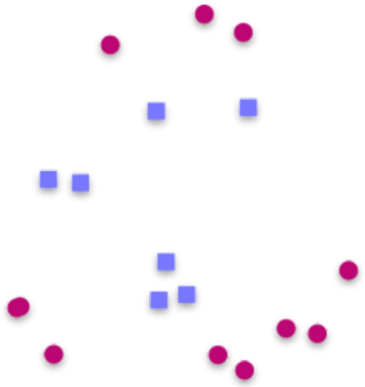


Figure II. 4. Comment séparer les deux catégories avec une ligne droite.

Puisque les carrés sont entourés de ronds de toute part, il est impossible de trouver de ligne droite qui soit une frontière : on dit que les données d'entraînement ne sont pas linéairement séparables. Cependant, imaginez qu'on arrive à trouver une transformation qui fasse en sorte que notre problème ressemble à ça :

Figure II. 5. Les mêmes points d'entraînement après transformation.

A partir de là, il est facile de trouver une séparation linéaire. Il suffit donc de trouver une transformation qui va bien pour pouvoir classer les objets. Cette méthode est appelée kernel trick, ou astuce du noyau en français.

Dans un espace vectoriel de dimension finie n, un hyperplan est un sous-espace vectoriel de dimension $n-1$. Ainsi, dans un espace de dimension 2 un hyperplan sera une droite, dans un espace de dimension 3 un hyperplan sera un plan, etc.

Soit un espace vectoriel E de dimension n. L'équation caractéristique d'un hyperplan est de la forme $\boldsymbol{w1x1 + w2x2 + \cdots + wnxn = 0}$, ou $\boldsymbol{w1}, \ldots, \boldsymbol{wn}$ sont des scalaires. Par définition, toute vecteur x = $\begin{pmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_n} \end{pmatrix}$ ∈ E vérifiant l'équation appartient à l'hyperplan. Par exemple, en dimension 2, $\boldsymbol{ax + by = 0}$ est bien l'équation caractéristique d'une droite vectorielle (qui passe par l'origine) [10].

II.2.1.2. Formalisme des SVM

De façon plus générale que dans les exemples donnés précédemment, les SVM ne se bornent pas à séparer des points dans le plan. Ils peuvent en fait séparer des points dans un espace de dimension quelconque. Par exemple, si on cherche à classer des fleurs par espèce, alors que l'on connaît leur taille, leur nombre de pétales et le diamètre de leur tige, on travaillera en dimension 3.

Un autre exemple est celui de la reconnaissance d'image : une image en noir et blanc de 28*28 pixels contiennent 784 pixels, et est donc un objet de dimension 784. Il est ainsi courant de travailler dans des espaces de plusieurs milliers de dimensions. [10]

II.2.2. Réseaux neuronaux convolutifs (CNN)

La vision par ordinateur évolue rapidement de jour en jour. L'une des raisons est l'apprentissage en profondeur. Lorsque nous parlons de vision par ordinateur, un terme réseau de neurones convolutifs (abrégé en CNN) nous vient à l'esprit car CNN est largement utilisé ici. Des exemples de CNN en vision par ordinateur sont la reconnaissance faciale, la classification d'images, etc. Il est similaire au réseau neuronal de base. CNN a également des paramètres apprenables comme le réseau neuronal, c'est-à-dire les poids, les biais, etc.

Dans cette partie nous allons nous intéresser à l'un des algorithmes les plus performants en deep learning, les Convolutional Neural Networks ou CNNs, ce sont de puissants modèles de

programmation, permettant notamment à chaque image de prendre en entrée une image identifiée, un label correspondant à sa classe de membres.[11]

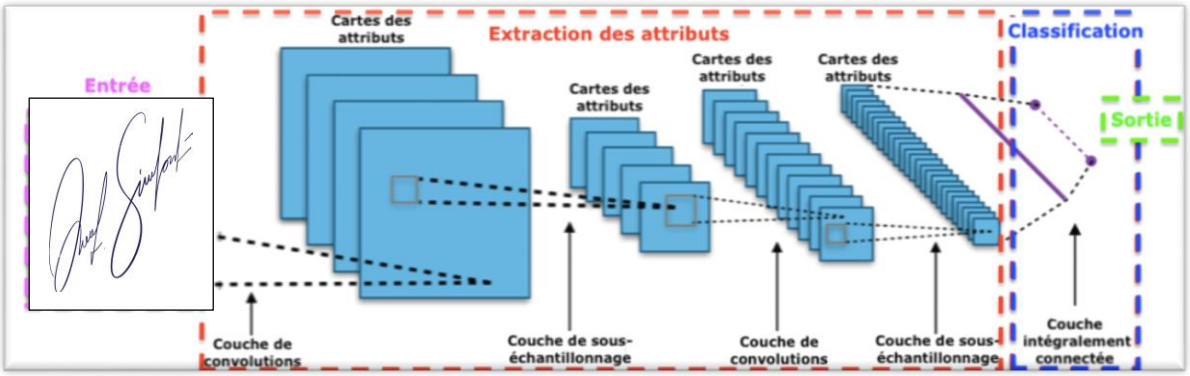


Figure II. 6. Schéma de principe du CNN.

II.2.2.1. Architecture d'un réseau de neurone convolutif

Les neurones des réseaux convergents représentent des tumeurs tridimensionnelles à chaque niveau, contrairement aux neurones des réseaux précédents qui représentaient des unités indépendantes à chaque niveau. Il s'agit de tirer parti du fait que les données représentent des images. Concrètement, chaque niveau d'un CNN est agencé en 3 dimensions (hauteur, largeur, profondeur).

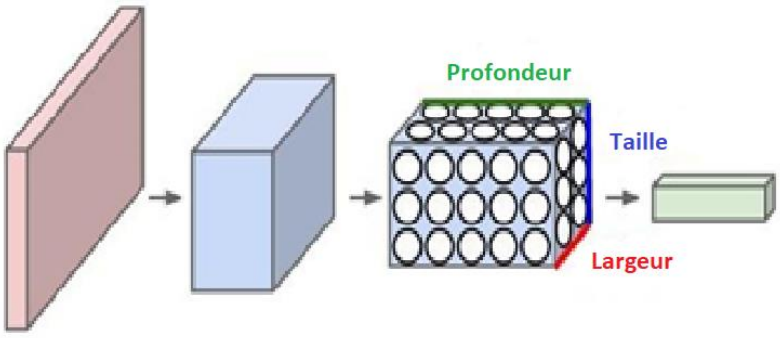


Figure II. 7. CNN dont les niveaux représentent les tumeurs.

La dimension de profondeur est saisie en raison de la troisième dimension de l'image, les 3 canaux de couleur par exemple (la dimension peut être un). De plus, la sortie du réseau sera consistée en un plan de neurones de dimension $1 \times 1 \times K$, où K le nombre de classes.

Exemple : Pour une image 96×96 et un détecteur 8×8 représentants une entité, en commençant par le coin supérieur gauche (1,1) et en balayant l'image vers la droite, retenir les conclusions

pour chaque zone 8 × 8 traversées se terminera à point (89,89) ayant collecté 89 × 89 activations.

La valeur de chaque activation dans le tableau final indique si elle a été détectée la caractéristique spécifique sur l'image.

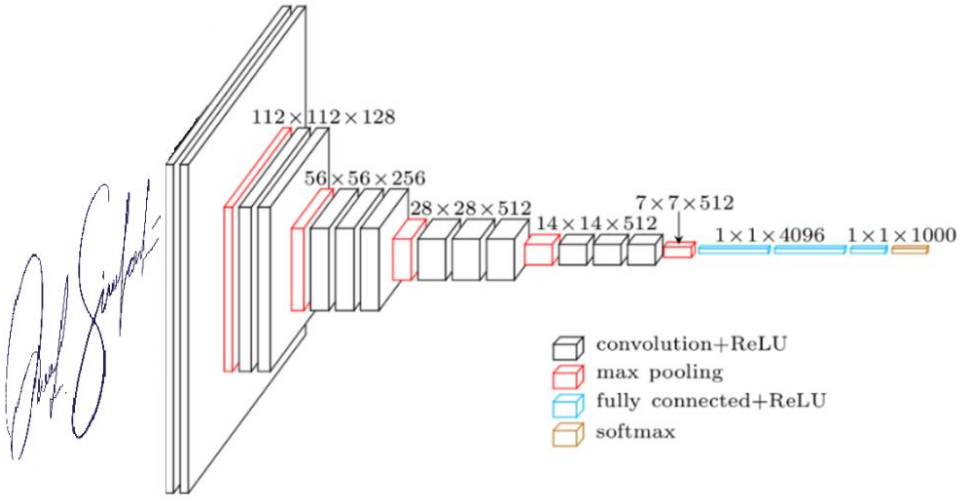


Figure II. 8. Sous-échantillonnage d'architecture CNN [11]

II.2.2.2. Les différentes couches de CNN

Il existe cinq couches différentes dans CNN :

- Couche d'entrée
- Couche Convo (Convo + ReLU)
- Couche de mutualisation
- Couche entièrement connectée (FC)
- Softmax/couche logistique
- Couche de sortie

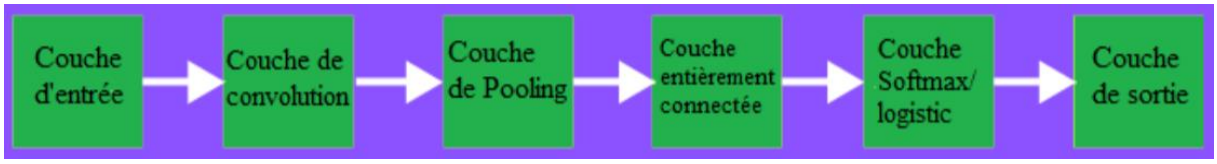


Figure II. 9. Différentes couches de CNN [11]

II.2.3. K-nearest neighbors (KNN)

L'algorithme K-NN (K-nearest neighbors) est une méthode d'apprentissage supervisé. Il peut être utilisé aussi bien pour la régression que pour la classification. Son fonctionnement peut être assimilé à l'analogie suivante *"dis-moi qui sont tes voisins, je te dirais qui tu es…"*.

Pour effectuer une prédiction, l'algorithme K-NN ne va pas calculer un modèle prédictif à partir d'un Training Set comme c'est le cas pour la régression logistique ou la régression linéaire. En effet, K-NN **n'a pas besoin de construire un modèle prédictif.** Ainsi, pour K-NN il n'existe pas de phase d'apprentissage proprement dite. [12]

- K-NN stocke tout le jeu de données pour effectuer une prédiction.
- K-NN ne calcule aucun modèle prédictif et il rentre dans le cadre du Lazy Learning.
- K-NN effectue des prédictions justes à temps (à la volée) en calculant la similarité entre un observation en entrée et les différentes observations du jeu de données.

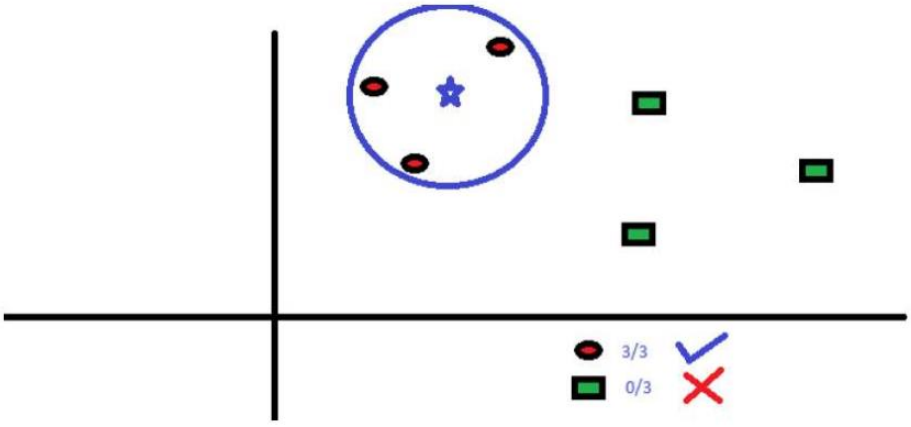


Figure II. 10. Exemple d'application de KNN.

II.2.3.1. Calcul de similarité dans l'algorithme K-NN

Comme on vient de le voir dans notre écriture algorithme, K-NN a besoin d'une fonction de calcul de distance entre deux observations. Plus deux points sont proches l'un de l'autre, plus ils sont similaires et vice versa.

Il existe plusieurs fonctions de calcul de distance, notamment, la distance euclidienne, la distance de Manhattan, la distance de Minkowski, celle de Jaccard, la distance de Hamming…etc. On choisit la fonction de distance en fonction des types de données qu'on manipule. Ainsi pour les données quantitatives (exemple : poids, salaires, taille, montant de panier éléctronique etc…) et du même type, la distance euclidienne est un bon candidat. Quant

à la distance de Manhattan, elle est une bonne mesure à utiliser quand les données (*input variables)* ne sont pas du même type (exemple :âge, sexe, longueur, poids etc…).

Il est inutile de coder, soi-même ces distances, généralement, les librairies de Machine Learning comme Scikit Learn, effectue ces calculs en interne. Il suffit juste d'indiquer la mesure de distance qu'on souhaite utiliser [12].

La distance euclidienne :

- Distance qui calcule la racine carrée de la somme des différences carrées entre les coordonnées de deux points :

$$D_e(x,y) = \sqrt{\sum_{j=1}^{n}(x_j - y_j)^2} \qquad \text{(II.1)}$$

Distance Manhattan :

- La distance de Manhattan: calcule la somme des valeurs absolues des différences entre les coordonnées de deux points :

$$D_m(x,y) = \sum_{i=1}^{k} \mid x_i - y_i \mid \qquad \text{(II.2)}$$

Distance Hamming :

- La distance entre deux points donnés est la différence maximale entre leurs coordonnées sur une dimension.

$$D_h(x,y) = \sum_{i=1}^{k} \mid x_i - y_i \mid \qquad \text{(II.3)}$$

Avec $x = y \implies D = 0$

$x \neq y \implies D = 1$

Notez bien qu'il existe d'autres distances selon le cas d'utilisation de l'algorithme, mais la distance euclidienne reste la plus utilisée.

II.2.3.2. Le choix de la valeur K

Le choix de la valeur K à utiliser pour effectuer une prédiction avec K-NN, varie en fonction du jeu de données. En règle générale, moins on utilisera de voisins (un nombre K petit) plus on sera sujette aux sous apprentissage (underfitting). Par ailleurs, plus on utilise de voisins (un nombre K grand) plus, sera fiable dans notre prédiction. Toutefois, si on utilise K nombre de voisins avec et K = N et N étant le nombre d'observations, on risque d'avoir du overfitting et

par conséquent un modèle qui se généralise mal sur des observations qu'il n'a pas encore vu[12].

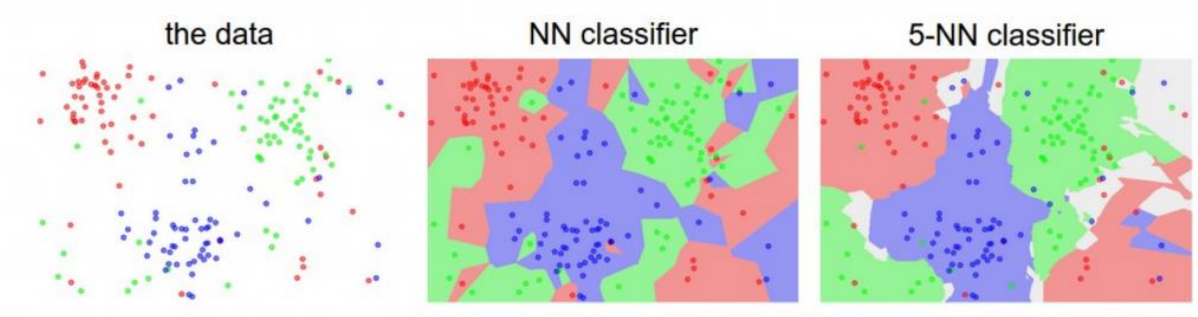


Figure II. 11. Des points trois types d'étiquetages possibles.

L'image ci-dessus à gauche représente des points dans un plan 2D avec trois types d'étiquetages possibles (rouge, vert, bleu).

Pour le 5-NN classifieur, les limites entre chaque région sont assez lisses et régulières. Quant au N-NN Classifier, on remarque que les limites sont "chaotiques" et irrégulières. Cette dernière provient du fait que l'algorithme tente de faire rentrer tous les points bleus dans les régions bleues, les rouges avec les rouges etc… c'est un cas d'overfitting.[12]

Pour cet exemple, on préférera le 5-NN classifier sur le NN-Classifier. Car le 5-NN classifier se généralise mieux que son opposant.

II.2.3.3. Limitations de K-NN

K-NN est un algorithme assez simple à appréhender. Principalement, grâce au fait qu'il n'a pas besoin de modèle pour pouvoir effectuer une prédiction. Le contre coût est qu'il doit garder en mémoire l'ensemble des observations pour pouvoir effectuer sa prédiction. Ainsi il faut faire attention à la taille du jeu d'entrainement.

Également, le choix de la méthode de calcul de la distance ainsi que le nombre de voisins peut ne pas être évident. Il faut essayer plusieurs combinaisons et faire du tuning de l'algorithme pour avoir un résultat satisfaisant. [12]

K-NN est un algorithme assez simple à appréhender. Principalement, grâce au fait qu'il n'a pas besoin de modèle pour pouvoir effectuer une prédiction. Le contre coût est qu'il doit garder en mémoire l'ensemble des observations pour pouvoir effectuer sa prédiction. Ainsi il faut faire attention à la taille du jeu d'entrainement.

II.2.4. Quantification de la phase locale (LPQ)

La quantification de la phase locale ou le descripteur LPQ a été désigné pour la première fois par *Ojansivu et Heikkilä,* pour l'utiliser dans la classification de textures pour les images floues. Il permet d'améliorer la classification de textures pour être robuste aux artéfacts générés par le flou présent dans une image . Le descripteur LPQ est construit de façon à ne retenir dans une image que l'information locale invariante à un certain type de flou. Il est insensible au flou central symétrique, tel que celui causé par le mouvement linéaire et hors du foyer du capteur . Inspiré par cette idée, nous proposons le descripteur LPQ comme une méthode efficace pour résoudre le problème des variations d'expressions dans le système de vérification du visage 3D basé sur les images de profondeur. Dans notre système les images faciales comprennent des mouvements dans différentes régions causées par différentes expressions telles que le rire, le sourire, la colère, la surprise etc…, l'opérateur LPQ est basé sur la transformée de Fourier de la phase. L'extraction de l'information de la phase locale est utilisée par l'application de la transformée de Fourier à court terme (STFT) calculée sur un rectangle Nx de M ∗ M voisins pour chaque pixel x dans l'image faciale f (x) définie par l'équation (I.4), où wu correspond aux vecteurs de base de la décomposition à la fréquence u,fx contient toutes les valeurs de l'image appartenant au voisinage Nx [13].

$$F(u,x) = \sum v\epsilon x \;\; f(x-y)e^{-2j\pi uTy} = wufx \qquad \text{(II.4)}$$

La transformée de Fourier est alors calculée pour seulement 4 fréquences u_i (i = 1, . . . ,4): u_1=[a, 0]^T , u_2 = [0, a]^T, u_3 = [a, a]^T et u_4 = [a, -a]^T, où a représente la fréquence scalaire suffisamment élevée pour H_u >0, alors nous obtenons un vecteurs F_x^c où

$$F_x^c = [F(u1,x), F(u2,x), F(u3,x), F(u4,x)] \qquad \textbf{(II.5)}$$

Par la suite, un quantificateur scalaire simple est utilisé pour l'extraction des informations de la phase dans chaque coefficient de Fourier en observant les signes des parties réelles (Re) et imaginaires (Im). Le quantificateur scalaire est donné par l'équation suivante :

$$g_j(x) = \begin{cases} 1 & g_j(x) \geq 0 \\ 0 & ailleurs \end{cases} \qquad \text{(II.6)}$$

Où $g_j(x)$ représente la n^{ième} $_{i=1}$ composante du vecteur G_x = $[Re\{F_x\}, Im\{F_x\}]$. Les huit coefficients binaires obtenus $q_j(x)$ sont représentés comme des valeurs entières entre 0 et 255 en utilisant un codage binaire simple pour obtenir les étiquettes de LPQ, F_{LPQ} qui est définie Par :

$$F_{LPQ}(X) = \sum_{i=1}^{8} g_j\ (x)\ 2^{j-1} \qquad \textbf{(II.7)}$$

En conséquence, nous obtenons l'étiquette d'image F_{LPQ}, dont les valeurs sont invariantes pour le flou (barbouillage). La figure II.12 présente l'organigramme de l'ensemble des étapes nécessaires à la construction du descripteur LPQ pour une image faciale.

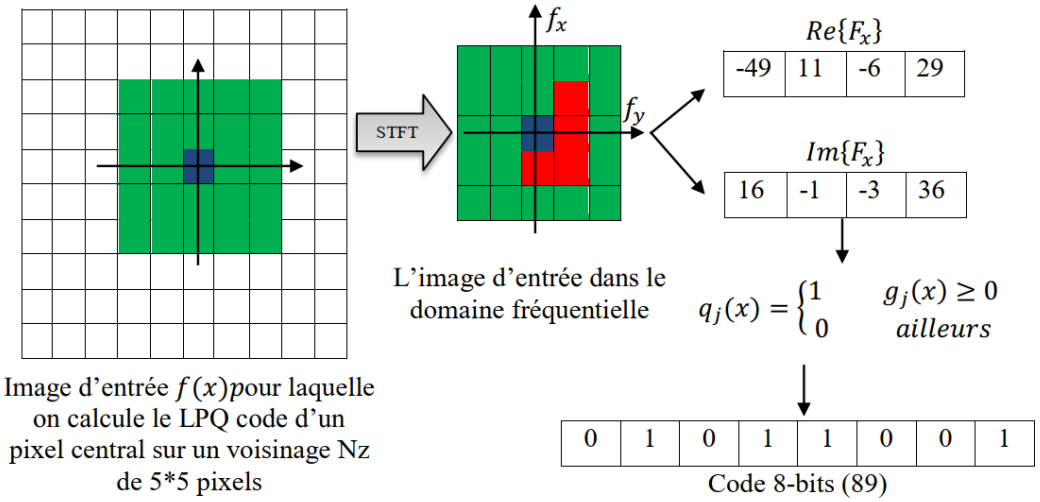


Figure II. 12. Organigramme de l'ensemble des étapes nécessaires du descripteur LPQ.

La figure II.13 illustre une représentation de l'image de profondeur 3D par le descripteur LPQ en voisinage de pixels Nz = 3, 5, 7 et 9.

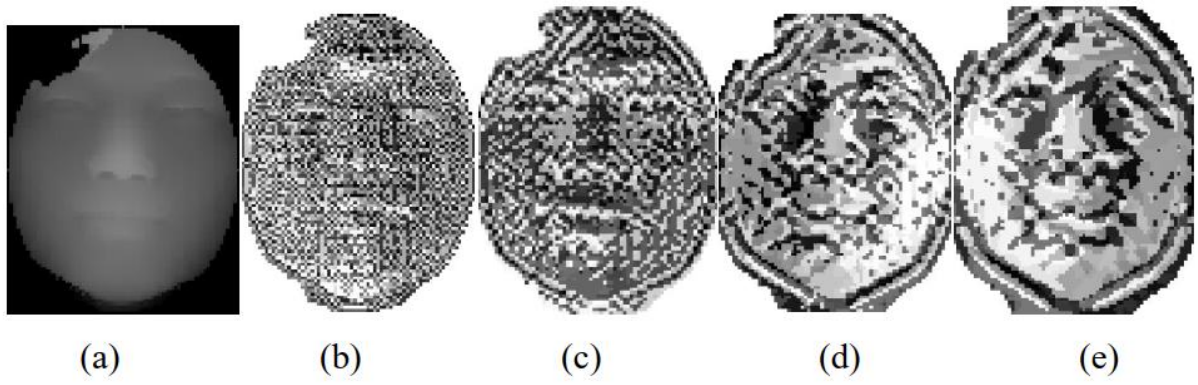


Figure II. 13. Représentation d'une image de profondeur avec le descripteur LPQ sous différents

Voisinage de pixel Nz;

(a) image d'entrée, (b) code LPQ (Nz=3), (c) code LPQ (Nz=5), (d) code LPQ (Nz=7), (e) code LPQ (Nz=9) [13].

II.2.5 Particle Swarm Optimization (PSO)

L'optimisation des essaims de particules (PSO) a été appliquée avec succès dans de nombreux domaines de recherche et d'application. Pour ma part, j'ai beaucoup apprécié l'application de cet algorithme dans l'article de G. Sermpinis sur la prévision des taux de change.

Il est démontré que la PSO peut avoir de meilleurs résultats de manière plus rapide et moins chère par rapport aux autres méthodes. Il peut aussi être parallélisé. De plus, il n'utilise pas le gradient du problème à optimiser. En d'autres termes, contrairement aux méthodes d'optimisation traditionnelles, PSO n'exige pas que le problème soit différentiable.

Enfin, il y a très peu d'hyperparamètres. Ces paramètres sont très simples à appréhender et ne nécessitent pas de notions avancées. Pour les mêmes hyperparamètres, PSO va travailler sur une très grande variété de tâches, ce qui en fait un algorithme très puissant et flexible.

Tout au long de cet article, je détaillerai les mécanismes derrière l'algorithme Particle Swarm Optimization en prenant comme métaphore un groupe d'oiseaux [14].

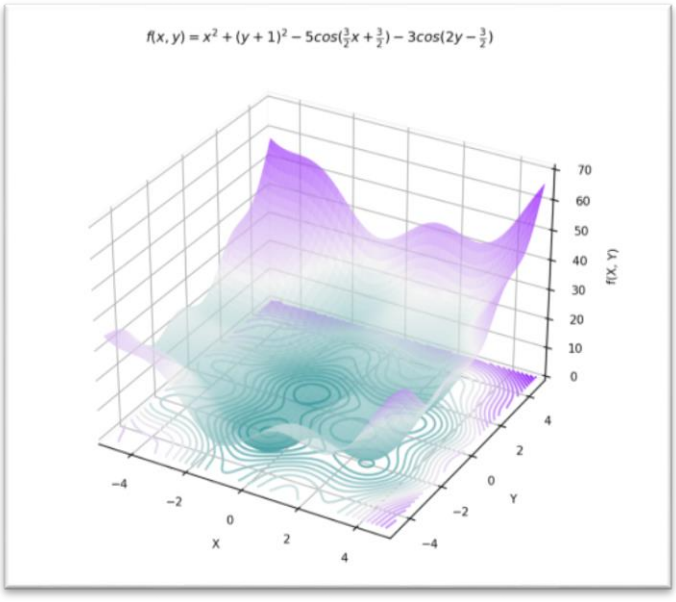


Figure II. 14. Fonction pour minimiser.

L'objectif de cet article sera de minimiser la fonction que vous voyez ci-dessus. La fonction en question est clairement définie, ne comprend que 2 variables et est différentiable. Mais gardez

à l'esprit que cette fonction pourrait être une fonction en escalier non différentiable, ou une fonction définie par les poids d'un réseau de neurones dont on ne connaît pas le minimum global. A des fins pédagogiques, nous considérerons la fonction :

$$f(x,y) = x^2 + (y+1)^2 - 5cos(1.5x + 1.5) - 3cos(2x - 1.5) \qquad \text{(II.8)}$$

Qui nous permet une visualisation 2D et 3D. Ainsi l'objectif de cet article sera d'optimiser la fonction *f* son minimum global étant donné *x* et *y* .

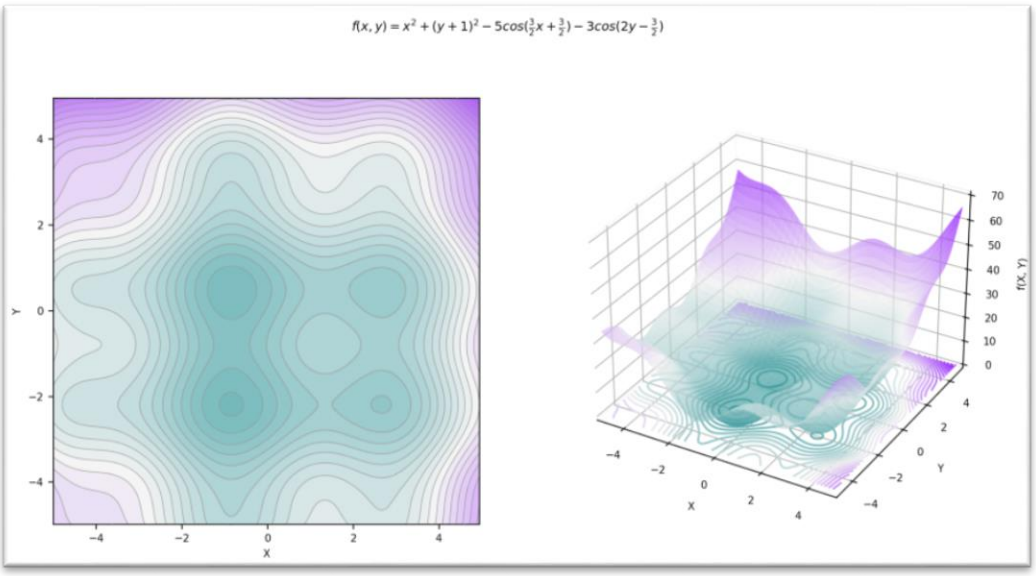


Figure II. 15. Fonction de minimisation — Vue 2D et 3D [14].

Avant de nous plonger dans notre cas d'application simple, sautons dans le passé. L'optimisation des essaims de particules est une technique d'optimisation stochastique basée sur la population développée par le Dr Eberhart et le Dr Kennedy en 1995 inspirée par le comportement social des oiseaux ou des bancs de poissons.

Histoire du coucher : un groupe d'oiseaux cherche de la nourriture dans une vaste vallée. Il n'y a de la nourriture qu'à un seul endroit dans cette vallée. Aucun des oiseaux ne sait où se trouve la nourriture, mais tous les oiseaux ont une idée de la distance qui les sépare de la nourriture.

PSO traduction : un groupe de particules (solutions potentielles) du minimum global dans un espace de recherche. Il n'y a qu'un minimum global dans cet espace de recherche. Aucune des particules ne sait où se situe le minimum global, mais toutes les particules ont des valeurs de fitness évaluées par la fonction de fitness à optimiser.

$$P_i^t = [x_{0,i}^t, x_{1,i}^t, x_{2,i}^t, x_{3,i}^t, \dots, x_{n,i}^t] \quad \text{(II.9)}$$

Avant d'aller plus loin dans l'explication de l'algorithme PSO, concentrons-nous un instant sur nos particules, chacune de ces particules est une solution potentielle de la fonction à minimiser. Ils sont définis par leurs coordonnées dans l'espace de recherche.

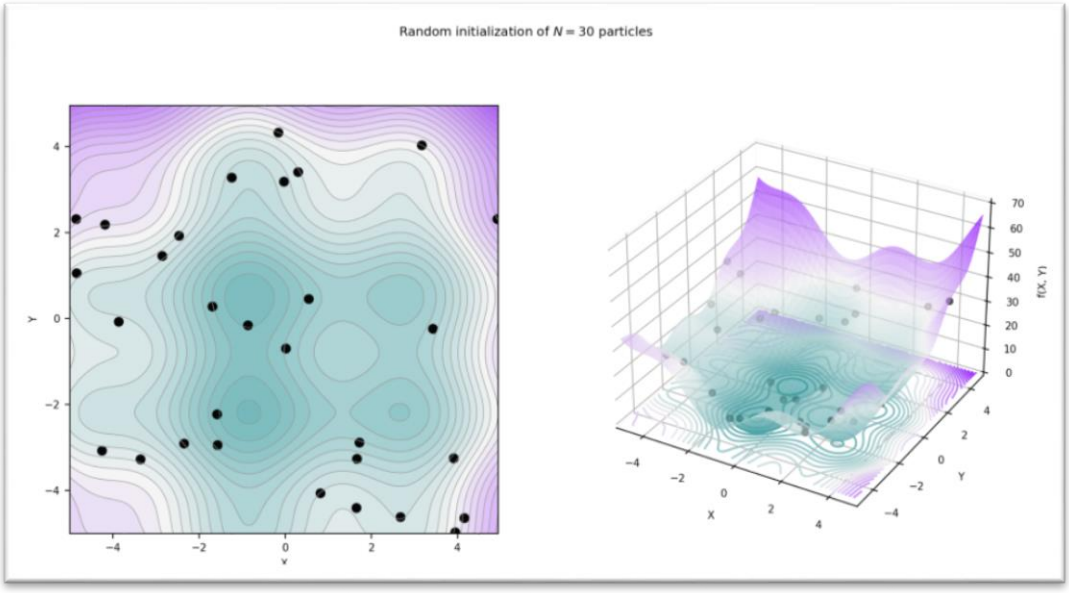


Figure II. 16. Initialisation aléatoire de la position des particules

On peut alors définir aléatoirement des particules dans l'espace de recherche comme dans l'image ci-dessus. Mais ces particules doivent être en mouvement pour trouver la fonction optimale.

Histoire du coucher : chacun de ces oiseaux se déplace avec une certaine vitesse de vol à travers la vallée pour trouver de la nourriture.

PSO traduction : chacune de ces particules est en mouvement avec une vitesse leur permettant de mettre à jour leur position au fil des itérations pour trouver le minimum global.

$$V_i^t = [v_{0,i}^t, v_{1,i}^t, v_{2,i}^t, v_{3,i}^t, \dots, v_{n,i}^t] \quad \text{(II.10)}$$

Les particules ont déjà été réparties aléatoirement dans l'espace de recherche. Leur vitesse doit alors être initialisée. Défini par sa vitesse dans chaque direction, le vecteur vitesse sera à nouveau randomisé. Pour cette raison, on parle d'algorithmes stochastiques.

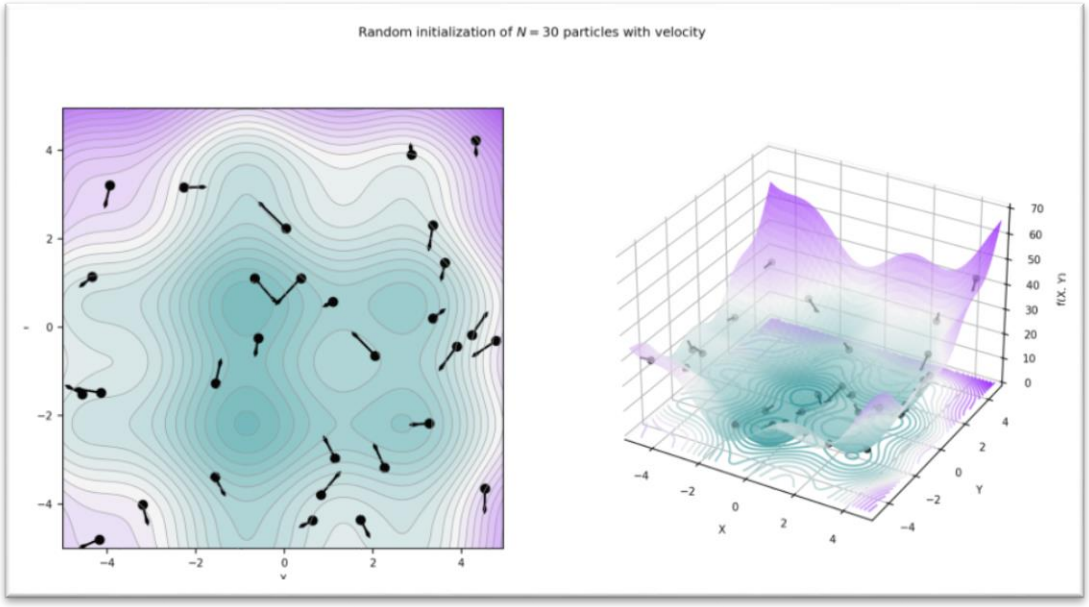


Figure II. 17. Initialisation aléatoire de la position et de la vitesse des particules

II.3. Conclusion

Dans ce chapitre nous avons étudié les différentes méthodes de classification des données, fait classer les caractéristiques semblables d'un ou plusieurs individus à la même classe, cette étape est appliquée par des algorithmes comme SVM, CNN, KNN, LPQ et PSO.

Chapitre III : **Authentification** *par Depp Learning*

III.1. Introduction

L'apprentissage en profondeur (deep learning) est un domaine de recherche sur l'apprentissage automatique basé sur un type particulier de mécanisme d'apprentissage. Il est caractérisé par l'effort de créer un modèle d'apprentissage à plusieurs niveaux, dans lequel les niveaux les plus profonds prennent en compte les résultats des niveaux précédents, les transformant et en faisant toujours plus d'abstraction. Cet aperçu des niveaux d'apprentissage est inspiré par la façon dont le cerveau traite l'information et apprend en réagissant aux stimuli externes. Chaque niveau d'apprentissage correspond, par hypothèse, à l'une des différentes zones qui composent le cortex cérébral. [15]

Il existe de nombreuses méthodes d'apprentissage automatique, comme nous l'avons vu dans le chapitre précédent, mais notre approche passera par l'apprentissage en profondeur.

III.2. Définition

L'apprentissage profond ou apprentissage en profondeur (en anglais : deep learning, deep structured learning, hierarchical learning) est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires. Ces techniques ont permis des progrès importants et rapides dans les domaines de l'analyse du signal sonore ou visuel et notamment de la reconnaissance faciale, de la reconnaissance vocale, de la vision par ordinateur, du traitement automatisé du langage.[16]

III.3. Fonctionnement du deep Learning

Le deep Learning s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain.

Ce réseau est composé de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente. Le système apprendra par exemple à reconnaître les lettres avant de s'attaquer aux mots dans un texte, ou détermine s'il y a un visage sur une photo avant de découvrir de quelle personne il s'agit. [17]

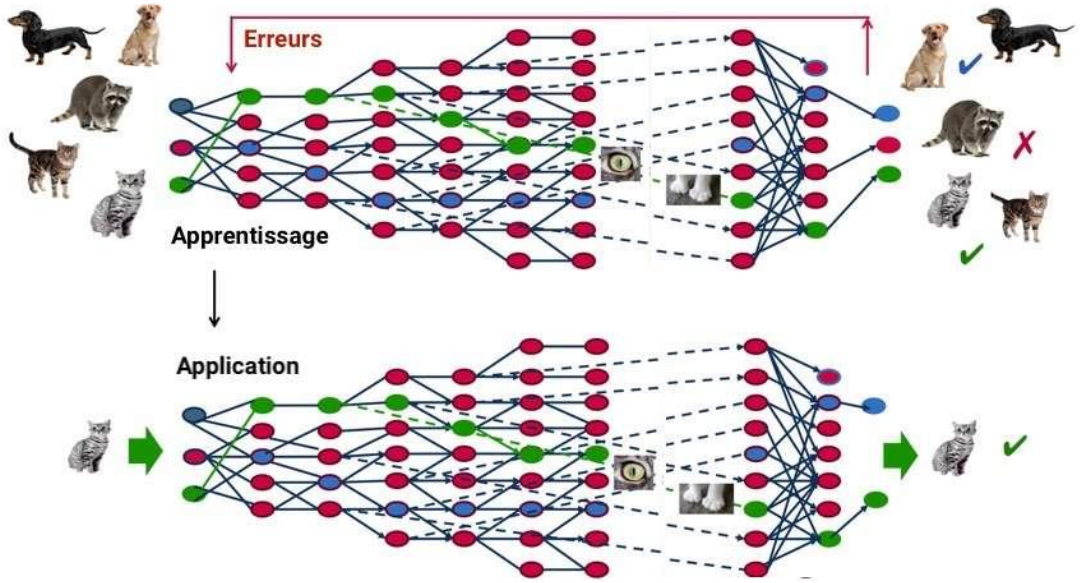


Figure III. 1. Un processus d'autoapprentissage.

À travers un processus d'autoapprentissage, le deep Learning est capable d'identifier un chat sur une photo. À chaque couche du réseau neuronal correspond un aspect particulier de l'image.[17]

III.3.1. Réseaux de neurones artificiels

Leur nom est inspiré des neurones biologiques qu'ils essaient d'imiter. Il s'agit essentiellement d'un ensemble d'unités, appelées neurones, qui sont localement connectées les unes aux autres, formant un réseau.

Chaque neurone reçoit un signal à son entrée et à travers une fonction d'activation qu'il contient, une sortie est générée.

La taille d'un réseau de neurones est déterminée par ses niveaux. Un niveau est constitué d'un ensemble de neurones à caractéristiques communes qui reçoivent le même signal à leur entrée et alimentent simultanément les suivants. Selon le nombre de couches, l'architecture du réseau est appelée réseau de neurones peu profond ou réseau de neurones profond. Les couches intermédiaires sont appelées couches cachées.

Chaque neurone peut avoir des synapses avec plusieurs neurones en même temps, que ce soit en termes d'entrée ou de sortie. Le signal à l'entrée de chaque neurone est transmis par ces synapses, et leur but est de l'amplifier ou de l'atténuer. Par conséquent, nous voyons que chaque connexion a un facteur de poids différent et en conclusion le but de chaque réseau de neurones est de déterminer ces poids qui conduiront aux valeurs prédites.

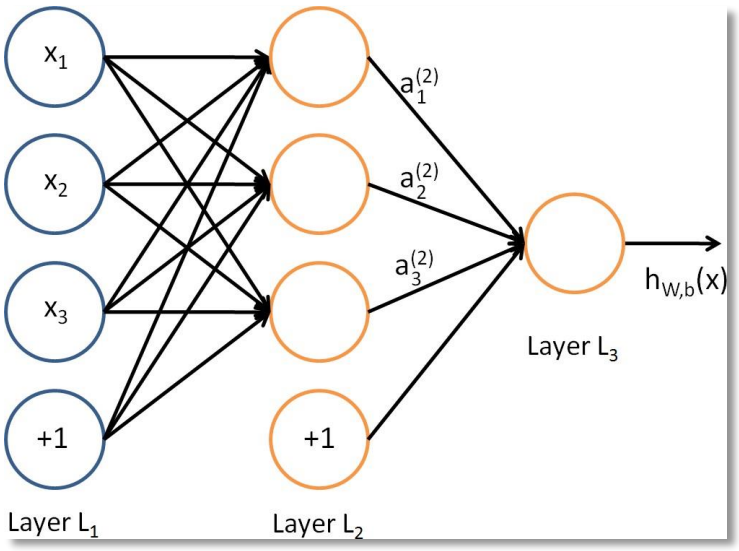


Figure III. 2. Vue simplifiée d'un réseau artificiel de neurones [18]

III.3.2. Fonction d'activation

La fonction d'activation (ou fonction de seuillage, ou encore fonction de transfert) sert à introduire une non-linéarité dans le fonctionnement du neurone.

Les fonctions de seuillage présentent généralement trois intervalles :

En dessous du seuil, le neurone est non-actif (souvent dans ce cas, sa sortie vaut 0 ou -1) ;

Aux alentours du seuil, une phase de transition ;

Au-dessus du seuil, le neurone est actif (souvent dans ce cas, sa sortie vaut 1). [18]

Dans sa forme la plus simple, un neurone reçoit une valeur x à son entrée et via une fonction d'activation $f(\cdot)$ produit une sortie $f(x)$. Les fonctions d'activation ciblent une sortie entre [0,1] ou [−1,1] selon le problème. Les fonctions d'activation les plus courantes sont :

$$f(x) = \frac{1}{1+e^{-x}} \quad \text{(III.1)}$$

$$f(x) = \frac{e^{x}+e^{-x}}{e^{x}+e^{-x}} \quad \text{(III.2)}$$

Ainsi que l'unité linéaire redressée (ReLU) :

$$f(x) = max(0, x) \quad \text{(III.3)}$$

Ce dernier est principalement utilisé dans les réseaux de neurones profonds et sa différence avec le reste est qu'il n'est pas différentiable.

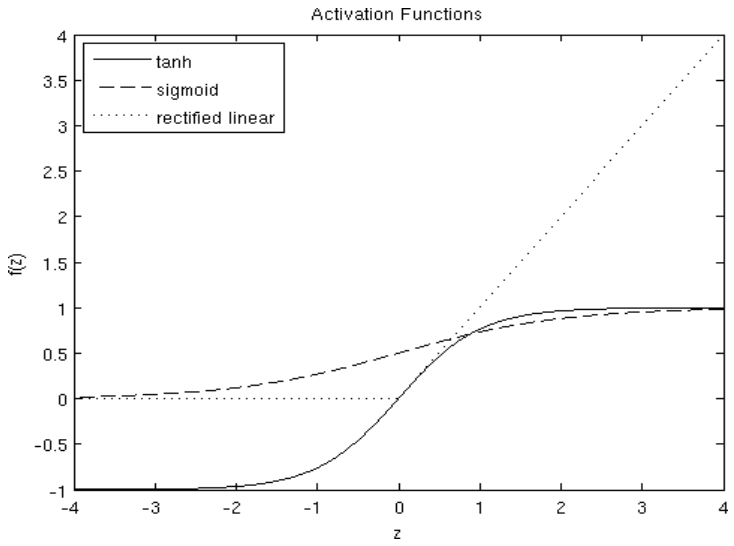


Figure III. 3. Les trois fonctions d'activation.

Mais supposons maintenant que le neurone est à sa connexion la plus générale et accepte N valeurs à son entrée : x_1, x_2,…, x_N. Chacune de ces valeurs vει atteint par une contraction de poids Wi, résultant en W_1, W_2,…, W_N les poids du neurone. La sortie du neurone est :

$$h_{W,b}(x) = f(W^T x) = f\left(\sum_{i=1}^{N} W_i\, x_i + b\right)$$

$$Ou\ f : \mathbb{R} \rightarrow \mathbb{R} \qquad \textbf{(III.4)}$$

$h_{W,b}(x)$ Est appelée la fonction d'hypothèse et le terme b terme de biais.

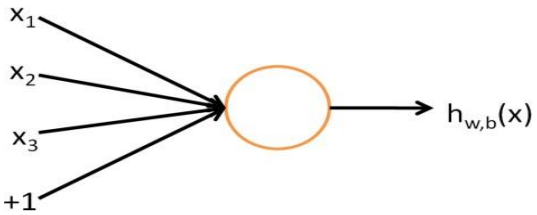


Figure III. 4 Neurone simple à 3 entrées.

Terme de biais : Il s'agit d'un terme fixe dont l'utilisation est très importante car lorsqu'il entre en WTx il permet à la fonction d'activation de se déplacer vers la gauche ou vers la droite lui donnant la possibilité de s'approcher encore mieux de la forme idéale.

Par contre, on peut dire que la somme $\sum_{i=1}^{N} W_i\, x_i + b$, sert à une fonction $y = ax + b$ donc sans introduction de biais, elle passera toujours du début (0,0) donnant peut-être de mauvais résultats.

III.3.3. Modèle de réseau neuronal entièrement connecté

Sur la figure III.4, nous observons un réseau de neurones simple à trois niveaux où le premier niveau correspond à l'entrée de données dans le réseau, le second est le niveau intermédiaire, tandis que le troisième est constitué d'un seul neurone et sa sortie représente la sortie de l'ensemble réseau.

Nous notons n le nombre de niveaux du réseau, donc n = 3. Nous nommons les niveaux L_l, donc L_1 le niveau d'entrée et L_3 la sortie du réseau.

Les paramètres du réseau sont $(W, b) = (W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)})$ où $W_{ij}^{(l)}$ le poids de la connexion entre le neurone j au niveau l et du neurone i au niveau $l + 1$. Dans l'exemple, $W^{(1)} \epsilon \mathbb{R}^{3\times 3}$ car il s'agit d'un tableau 3 × 3 où chaque ligne correspond aux poids de chacun des trois neurones et de chaque colonne dans les trois synapses de chaque neurone au niveau précédent, tandis que $W^{(2)} \in \mathbb{R}^{1\times 3}$ car nous avons j = 3 neurones connectés à i = 1 neurone du niveau suivant. (Notez que les termes de biais ne sont pas inclus car leur sortie est 1).

L'activation ou la sortie de chaque neurone i du niveau l est notée al. Pour les paramètres déjà définis, les sorties sont calculées sur la base de la relation précédente comme :

$$a_1^{(2)} = f(W_{11}^{(1)} x_1 + W_1^{(1)} x_2 + W_{13}^{(1)} x_3 + b_1^{(1)}) \quad \text{(III.5)}$$

$$a_2^{(2)} = f(W_{21}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{23}^{(1)} x_3 + b_2^{(1)}) \quad \text{(III.6)}$$

$$a_3^{(2)} = f(W_{31}^{(1)} x_1 + W_{32}^{(1)} x_2 + W_{33}^{(1)} x_3 + b_3^{(1)}) \quad \text{(III.7)}$$

Résultant en la fonction d'hypothèse produisant :

$$h_{W,b}(x) = a_1^{(3)} = f(W_{11}^{(2)} a_1^{(2)} + W_{12}^{(2)} a_2^{(2)} + W_{13}^{(2)} a_3^{(2)} + b_1^{(2)}) \quad \text{(III.8)}$$

Ensuite, en incluant les termes biais, nous désignons par z (l) la somme pondérée des entrées du neurone i au niveau l . Activation sous forme de vecteurs de sorte que f ([z1, z2, z3]) = [f (z1), f (z2), f (z3)], on se retrouve avec une forme plus compacte de ce qui précède :

$$z^{(2)} = W^{(1)}x + b^{(1)} \quad \text{(III.9)}$$

$$a^{(2)} = f(z^{(2)}) \quad \text{(III.10)}$$

$$z^{(3)} = W^{(2)}a^{(2)} + b^{(2)} \quad \text{(III.11)}$$

$$h_{W,b}(x) = a^{(3)} = f(z^{(3)}) \quad \text{(III.12)}$$

C'est ce qu'on appelle la propagation vers l'avant. En organisant nos paramètres dans des tableaux, profitant des propriétés de l'algèbre linéaire, nous accélérons les calculs.

L'architecture que nous avons utilisée dans l'exemple précédent est dite entièrement connectée car elle consiste à connecter chaque neurone à tous les niveaux précédant et suivant. Cette façon de connecter les neurones n'est pas la seule mais la plus courante, le nombre de neurones dans la sortie dépend également du problème respectif.

III.3.4. Modèles d'apprentissage supervisé

Les problèmes les plus courants modélisés par les réseaux de neurones sont les problèmes de régression linéaire, logistique et soft max. L'idée de base est la même et les différences se trouvent dans la fonction de coût et le nombre de neurones dans le dernier niveau.

- Régression linéaire

Le but d'un tel problème est de prédire une valeur de sortie pour un vecteur de données importées dans le réseau. Chaque xi qui correspond à un échantillon est appelé une caractéristique ou une caractéristique et le nombre d'entre eux définit ses dimensions. Il est nécessaire que chaque échantillon comporte le même nombre de caractéristiques.

La valeur de la sortie du réseau est le résultat d'une fonction $y = h_\theta$ (x) (où θ les paramètres respectifs) donc en pratique nous avons $y^{(i)} \approx h_\theta\ h\theta$ ($x^{(i)}$) pour chaque échantillon. Nous définissons comme fonction de coût une fonction dont la valeur est une mesure de la distance entre h_θ (·) et la demande :

$$J(\theta) = \frac{1}{2} \quad \Sigma_i (h_\theta(x^{(i)}) - y^{(i)})^2 \quad \text{(III.13)}$$

Qui est fonction des paramètres θ puisque, comme mentionné ci-dessus, h_θ (·) est généralement fonction des poids et du biais. Nous essayons de trouver les bons paramètres qui minimisent la fonction de coût.

Régression comptable

Dans de nombreuses applications, la valeur τιμή des échantillons n'est pas nécessairement une valeur continue, mais une valeur discrète. C'est-à-dire qu'il existe des problèmes qui obligent le réseau à classer les données entre deux classes qui peuvent représenter deux catégories quelconques (par exemple " oui / non ", " blanc / noir ", ' chien / chat ' etc.).

Dans les problèmes de tri, l'étiquette obtient des valeurs binaires ($y^{(i)} \in \{0,1\}$). Par conséquent, le concept de fonction hypothétique acquiert un autre sens, en particulier il nous donne la probabilité que l'échantillon appartienne à la classe "1" versus la probabilité qu'il appartienne à la classe "0".

Notamment : où σ la fonction sigmoïde dont l'ensemble de valeurs est [0,1]. Le but est :

$$P(y = 1 \mid x) = h_\theta(x) = \sigma(W^T x + b) \quad \textbf{(III.14)}$$

$$P(y = 0 \mid x) = 1 - P(y = 1 \mid x) = 1 - h_\theta(x) \quad \textbf{(III.15)}$$

$P(y = 1 \mid x)$ prend de grandes valeurs lorsque x appartient à la classe "1" et de petites valeurs lorsque x appartient à la classe "0". Pour un ensemble d'échantillons (étiquetés en binaire) $\{(x(i,), y(i)) : i = 1, \ldots, m\}$ la fonction de coût est la suivante :

$$J(\theta) = -\sum_i (y^{(i)} \log(h_\theta(x^{(i)})) + (1 - (y^{(i)})) \log(1 - h_\theta(x^{(i)}))) \quad \textbf{(III.16)}$$

Si l'on observe, on verra facilement qu'un seul des deux termes n'est pas nul dans la somme pour chaque échantillon, donc en le minimisant quand $y^{(i)} = 1$ on doit augmenter beaucoup την $h_\theta x^{(i)}$, tandis que respectivement lorsque $y^{(i)} = 0$, nous devons augmenter considérablement $1 - h_\theta(x^{(i)})$.

- Régression Soft max

La régression Soft max est une généralisation de la régression comptable dans le cas où nous voulons gérer plus de deux classes. Dans ce Approximativement, les étiquettes étiquetées dans nos données sont de la forme $y^{(i)} \in \{1, \ldots, K\}$ Pour K nombre de classes.

Étant donné un échantillon x, nous voulons utiliser $h_\theta(x)$ pour estimer la probabilité

$P\ (y = k \mid x)$ pour chaque classe. Ainsi, l'hypothétique $h_\theta\ (x)$ exportera un vecteur de dimension K (dont la somme des éléments donne 1, puisque nous parlons de probabilités) de la forme :

$$h_{W,b}(x) = \begin{bmatrix} P(y = 1|x; \theta) \\ P(y = 2|x; \theta) \\ \vdots \\ P(y = K|x; \theta) \end{bmatrix} = \sum_{j=1}^{K} \frac{1}{e^{(\theta(j)Tx)}} \begin{bmatrix} exp(\theta^{(1)^T} x) \\ exp(\theta^{(2)^T} x) \\ \vdots \\ exp(\theta^{(K)^T} x) \end{bmatrix} \quad \text{(III.17)}$$

Si chaque échantillon a n caractéristiques, il est logique que $\theta\ ^{(1)}$, $\theta\ (2)$, …, $\theta\ ^{(K)}$ $\mathbb{R}^n$.

La somme exponentielle au dénominateur normalise le tableau de sorte que la somme de ses éléments fasse 1, c'est-à-dire notre exigence initiale.

Le paramètre θ au format vectoriel est décrit par un tableau $n \times K$.

$$\theta = \begin{bmatrix} \theta_{11} & \theta_{12} & \cdots & \theta_{1K} \\ \theta_{21} & \theta_{22} & \cdots & \theta_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{n1} & \theta_{n2} & \cdots & \theta_{nK} \end{bmatrix} \quad \text{(III.18)}$$

Pour décrire la fonction de coût, nous aurons besoin d'une fonction de contrôle 1, $y\ (i) = k$ pour lequel ce qui suit s'applique :

$$\delta = \begin{cases} 1,\ y^{(i)} = k \\ 0,\ y^{(i)} \neq k \end{cases} \quad \text{(III.19)}$$

III.4. L'algorithme de rétropropagation

Comme mentionné ci-dessus, le but d'un Neural Net est de donner à ses poids des valeurs telles que pour chaque donnée qu'il reçoit en entrée, il produit la sortie souhaitée. Pour y parvenir, il doit passer par une étape de formation ou au contraire d'apprentissage, c'est-à-dire redéfinir plusieurs fois ses valeurs pour arriver au résultat souhaité.

Fonction de coût : Supposons donc que nous ayons un ensemble d'échantillons d'entraînement $\{(x^{(1)}, y^{(1)}, \ldots, (x^{(m)}, y^{(m)})$. Nous avons défini la fonction de coût comme une fonction dont la valeur représente la performance du réseau dans la prévision des coûts souhaités sous la forme la plus générale :

$$J(W, b;\, x, y) \;=\; \frac{1}{2}\; \|hW, b(x) \;-\; y\|^{2} \qquad \text{(III.20)}$$

Décroissance du poids : Nous remarquons que dans la fonction de coût, en plus de l'erreur quadratique moyenne, nous avons ajouté un terme supplémentaire appelé décroissance du poids et il essaie de réduire de manière exponentielle les poids à zéro. Le but est d'éviter de sur-adapter le réseau. Nous avons donc introduit une gaussienne afin de configurer la fonction de coût. En pratique, il « punit » les lourdes charges en restreignant la liberté du système.

Descente de gradient : Puisque la performance du réseau sur le problème est représentée par la fonction de coût, notre objectif est de la minimiser. La fonction de coût a les poids et le biais comme paramètres indépendants et comme non concave, elle présente un minimum, donc un algorithme d'optimisation est la descente de gradient qui obtient ses valeurs les plus basses proches des minima locaux.

Premièrement, les poids et les biais avec de petites valeurs autour de zéro doivent être randomisés. L'initialisation aléatoire sert à créer un asymétrie aux sorties des neurones. Si nous fixons la même valeur sur tous les poids, tous les neurones auraient la même sortie.

À chaque itération, en fonction de la descente du gradient, les pondérations et le biais changent comme suit :

$$W_{ij}^{(l)} :=\; W_{ij}^{(l)} - a^{1}\frac{\partial}{\partial W_{ij}^{(l)}} j(W, b) \qquad \text{(III.21)}$$

$$b_{i}^{(l)} :=\; b_{i}^{(l)} - a\frac{\partial}{\partial b_{ij}^{(l)}} j(W, b) \qquad \text{(III.22)}$$

Nous observons qu'une dérivée positive de la fonction de coût indique que la pente est positive, ce qui signifie qu'à mesure que l'on s'éloigne d'un minimum local, les poids prennent une valeur inférieure à celle qu'ils avaient à l'itération précédente. Au contraire, pour une pente négative, on s'approche du minimum local, donc les poids prennent des valeurs plus élevées que les précédents, en essayant de "l'atteindre".

La forme de la fonction de coût dans l'espace dépend du nombre de charges du problème. Les dérivées partielles de la fonction de coût dans le cas de la descente de gradient stochastique, c'est-à-dire lorsqu'elle est calculée à chaque itération sur un échantillon, se calculent comme suit :

$$\frac{\partial \mathrm{J(W,b)}}{\partial W_{ij}^{(l)}} = \left[\frac{1}{m}\sum_{i=1}^{m}\frac{\partial}{\partial W_{ij}^{(l)}} j(W,b,x^{(i)},y^{(i)}\right] + \lambda \partial W_{ij}^{(l)} \qquad \text{(III.23)}$$

$$\frac{\partial \mathrm{J(W,b)}}{\partial W_{ij}^{(l)}} = \left[\frac{1}{m}\sum_{i=1}^{m}\frac{\partial}{\partial W_{ij}^{(l)}} j(W,b,x^{(i)},y^{(i)}\right] \qquad \text{(III.24)}$$

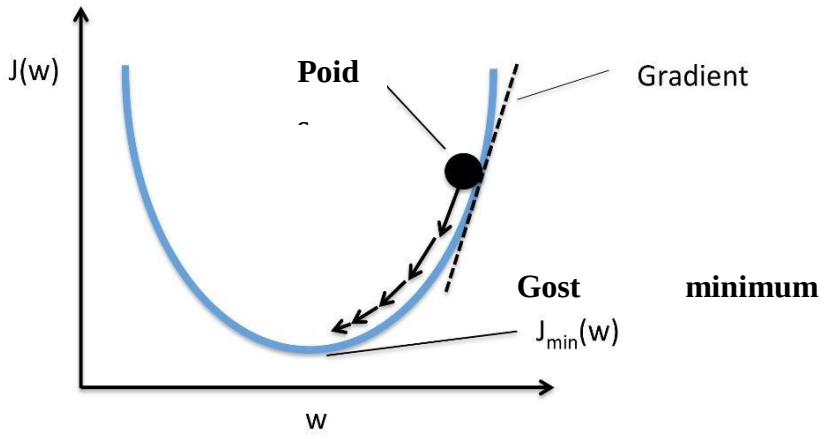


Figure III. 5. Problème unidimensionnel avec une pente positive dans la fonction de coût.

Il est facile de comprendre que le terme de perte de poids ne dépend pas d'un terme de biais. La fonction de coût du neurone de sortie est facilement calculée en comparant la valeur prédite avec la sortie du neurone, c'est-à-dire $h_{W,b}(x)$. Mais qu'en est-il de moi les niveaux intermédiaires ? Nous devons définir un terme $\delta^{(l)}$ qui représente la différence entre sa sortie et la valeur attendue (de ce neurone particulier),

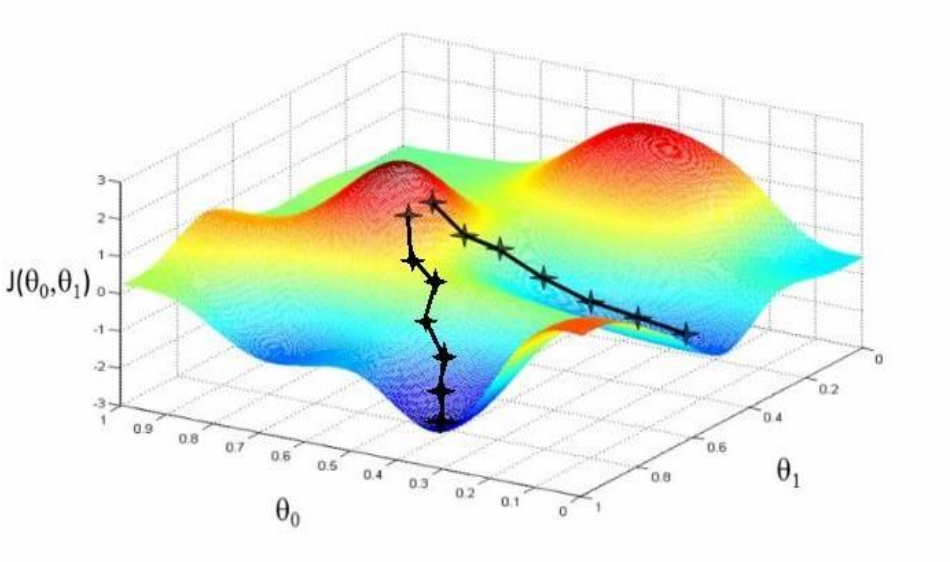


Figure III. 6. Problème bidimensionnel avec deux paramètres de poids indépendants θ_0 et θ_1.

Plus elle est grande, plus elle est responsable de l'écart de la production totale par rapport à la valeur attendue lorsque cette erreur se propage aux niveaux suivants. L'algorithme de rétropropagation est donné dans les quatre étapes suivantes :

1. lors du renvoi, les activations de chaque niveau jusqu'à la sortie finale sont calculées (L_2, L_3, …, L_{nl})

2. pour chaque neurone de sortie, la dérivée partielle de la fonction de sortie, selon ce qui précède, est calculée comme suit :

$$\delta_i^{(nl)} = \frac{\delta}{\delta_{Zi}^{(nl)}} \frac{1}{2} \left\| y - h_{W,b}(x) \right\| = -(y_i - \alpha_i^{(nl)}) \, . \, f\,'\,(z_i^{(nl)}) \qquad \text{(III.25)}$$

3. puis le δ des neurones intermédiaires est calculé à partir des derniers niveaux vers l'arrière (d'où le terme "propagation inverse"). Donc pour chaque nœud i à chaque niveau l :

$$\delta_i^{(l)} = \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)} \, \delta_i^{(l+1)} \right) . \, f\,'\,(z_i^{(l)}) \qquad \text{(III.26)}$$

4. et certaines de leurs dérivées sont données :

$$\frac{\partial}{\partial W_{ij}^{(l)}} j(W, b, x, y) = a_j^{(l)} \delta_i^{(l+l)} \qquad \text{(III.27)}$$

$$\frac{\partial}{\partial b_i^{(l)}} j(W, b, x, y) = \delta_i^{(l+l)} \qquad \text{(III.28)}$$

5. la dernière étape consiste à mettre à jour les pondérations et le biais :

$$\Delta W_{ij}^{(l)} = -a \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) \quad \text{(III.29)}$$

$$\Delta W_{ij}^{(l)} = -a \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) \quad \text{(III.30)}$$

Où $W_{ij}^{(l)}$ la différence de la nouvelle valeur des poids avec ij précédent (respectivement pour le biais) [7-9].

III.4.1. Formation réseau

Avant de continuer, nous devons clarifier certains concepts. Le réseau est dit formateur tant qu'il minimise sa fonction de coût. Chaque fois que la descente de gradient est appliquée, les poids changent légèrement vers le minimum local (ou total). Par conséquent, l'algorithme de rétropropagation doit être exécuté plusieurs fois. Cela nous amène à l'introduction des termes :

Saison : Une saison dure aussi longtemps qu'une passe avant et une passe arrière se déroulent tous aux longues formations organisées via le réseau.

Taille du lot : il s'agit du nom du nombre d'échantillons d'apprentissage dans lesquels une passe avant et une passe arrière se produisent simultanément. En d'autres termes, ils sont un sous-ensemble de l'ensemble d'apprentissage.

Itération : à chaque répétition, un lot a subi des passes avant et arrière.

Par exemple, dans un ensemble d'apprentissage contenant 500 données et divisé en une taille de lot de 250, il faudra deux répétitions pour terminer une session. Lorsque la fonction de coût n'est plus minimisée, on dit que l'apprentissage du réseau est terminé.

III.4.2. Optimisation par l'algorithme de gradient de descente

L'algorithme de gradient de descente est l'un des algorithmes les plus courants pour l'optimisation, ainsi que le moyen le plus courant d'améliorer un réseau de neurones. La raison de l'optimisation est d'accélérer des informations et les calculs.

Ses variantes les plus basiques sont : Batch Gradient Descent :

Calcule les dérivées du paramètre de la fonction de coût θ pour l'ensemble de l'échantillon d'apprentissage et est mis à jour à chaque instant :

$$\theta = \theta - a \nabla_{\theta} J(\theta) \quad \text{(III.31)}$$

Si le calcul est effectué dans l'ensemble, les calculs peuvent être très lents ou difficiles pour un très grand échantillon de données, mais si la surface de la fonction de coût est convexe, elle convergera vers le minimum total ou le minimum local s'il ne l'est pas convexe.

✓ **Algorithme de Gradient de descente stochastique (**Stochastic Gradient Descent SGD**) :**

Contrairement à BGD, les paramètres sont calculés et mis à jour pour chaque échantillon qui entre dans le réseau (x (i), y (i)) :

$$\theta = \theta - a\nabla_{\theta} J(\theta; x^{(i)}, y^{(i)}) \quad \text{(III.32)}$$

Les calculs sont effectués très rapidement car ils portent sur un échantillon à chaque itération mais certains d'entre eux sont redondants. L'ensemble du processus est très lent.

✓ **Algorithme de Gradient de descente en mini-lot :**

Dans ce cas, nous obtenons une combinaison de BGD et SGD car les paramètres sont calculés sur un mini-lot de données, donc à chaque saison, plus le mini-lot est grand, plus il y a de mises à jour.

$$\theta = \theta - a\nabla_{\theta} J(\theta; x^{(i:i+n)}, y^{(i:i+n)}) \quad \text{(III.33)}$$

L'estimation adaptative du moment est une méthode qui ajuste le taux d'apprentissage pour chaque paramètre. Stocke un milieu d'amortissement exponentiel des dérivées précédentes mt ainsi que des dérivées au carré ut similaires à la quantité de mouvement :

$$m_t = \beta_1 \, \mathrm{m}_{t-1} + (1 - \beta_1) g_1 \quad \textbf{(III.34)}$$

$$m_t = \beta_1 \, \mathrm{m}_{t-1} + (1 - \beta_1) g_1 \quad \textbf{(III.35)}$$

m_t et u_t sont respectivement les estimations du premier moment (moyenne) et du second moment (variance non orientée) des dérivées. Parce qu'ils sont initialisés comme des vecteurs nuls, les créateurs de la méthode ont observé qu'ils sont biaisés vers zéro, en particulier dans les premiers instants et lorsque les termes de réduction sont très petits (c'est-à-dire lorsque $\beta_{1,2} \cong 1$).

Ils réagissent à ce comportement en corrigeant les termes d'appréciation du premier et du second instant du temps :

$$m_t = \frac{\mathrm{m_t}}{1-\beta_1^t} \quad \text{(III.36)}$$

$$v_t = \frac{\mathrm{v_t}}{1-\mathrm{v}_2^t} \quad \text{(III.37)}$$

Ensuite, ils mettent à jour les paramètres tels que définis par la méthode :

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t} + \varepsilon} m_t \quad \text{(III.38)}$$

Les créateurs proposent comme valeurs par défaut : β_1= 0.9, β_2= 0.999 et $\varepsilon = 10^{-8}$. Il a été démontré que l'algorithme fonctionne bien par rapport à d'autres algorithmes d'apprentissage adaptatif.[25]

En plus des variations de l'algorithme, nous pouvons modifier d'autres paramètres qui conduisent à l'optimisation d'un réseau. Ceux-ci sont appelés hyperparamètres et nous ne pouvons les modifier qu'en modifiant leur valeur. L'un d'eux est le facteur que nous rencontrons dans la perte de poids (décroissance du poids). D'autres sont :

- **Taux d'apprentissage** : Le taux d'apprentissage ou Learning rate (Lr) est le facteur qui entre devant les dérivées des paramètres et détermine la taille du pas au minimum. Choisir le bon est très important car des prix très élevés entraînent de grandes étapes qui empêchent la convergence (dépassement) ainsi que de très petits prix entraînent de très petites étapes qui retardent considérablement la formation. Une solution moyenne est le taux d'apprentissage variable. En choisissant un rythme d'apprentissage qui diminue progressivement au fil des saisons, cela conduit à une convergence avec de très petites fluctuations au cours de l'entraînement. Une autre façon consiste à choisir un ordonnanceur qui planifie avant l'entraînement comment il va changer, ou pendant l'entraînement lorsque la fonction de coût dépasse une valeur seuil.

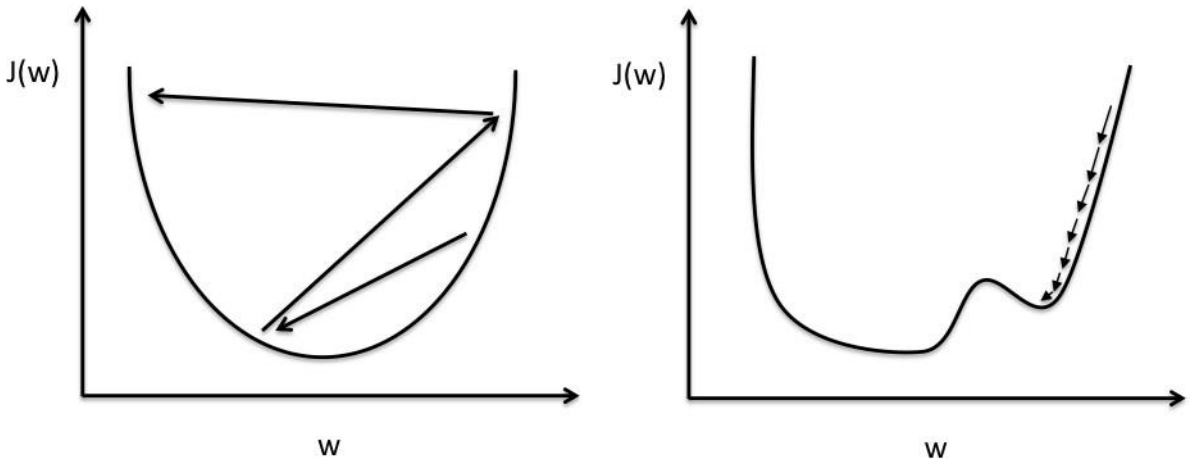


Figure III. 7. Taux d'apprentissage élevé et très faible.

- **Moment** : SGD rencontre des problèmes lorsqu'il traverse des zones abruptes de la fonction de coût (points où la pente est plus raide dans une dimension que dans l'autre) qui sont plus fréquentes autour d'un extrême. Alors, il se met à osciller sur les pentes, hésitant vers le minimum. Nous pouvons accélérer ce chemin dans la direction souhaitée en utilisant un paramètre appelé moment.

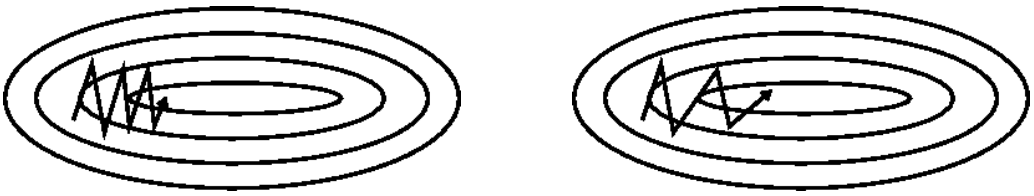


Figure III. 8. Gauche : sans moment. A droite : avec moment.

Dans le vecteur vitesse u_t, qui a les mêmes dimensions que θ, nous entrons un paramètre $\gamma \in [0,1]$.

$\nabla_\theta J\ (\theta)$ qui représente la pente, donc le taux d'apprentissage devrait être faible. Finalement, la quantité de mouvement confère une vitesse, forçant la fonction à faire des pas plus grands pour ne pas être piégée dans un minimum local et continuer à rechercher le tout. Idéalement, sa valeur devrait être constante à une valeur intermédiaire tout au long de la formation et augmenter à mesure que la convergence commence à se stabiliser [9, 10].

$$ut \ = \gamma u_{t-1} - 1 \ + \alpha \nabla_\theta J(\theta) \quad \text{(III.39)}$$

$$\theta = \theta - u_t \quad \text{(III.40)}$$

III.5. Conclusion

Dans ce chapitre on a présenté les notions importantes qui sont en relation avec l'apprentissage profond (définition, Architectures. . .etc.). Aussi qu'une vision générale sur l'apprentissage profond, toute en donnant en détail la méthode choisie dans notre travail de recherche qui est le deep learning. Le prochain chapitre, nous allons voir la vérification de la signature avec deep learning.

Chapitre IV : Système d'authentification des individus par signature manuscrite

IV.1. Introduction

La signature manuscrite est depuis plusieurs siècles le moyen le plus répandu. Elle est le moyen biométrique d'authentification le plus utilisé et accepté. La signature manuscrite d'un individu représente un bon compromis : tout en étant relativement fiable, elle est facile à acquérir, socialement acceptée comme un mode de reconnaissance. La signature est un moyen utilisé depuis longtemps, pour authentifier des documents, pour responsabiliser les individus face à des engagements (contrats, etc.). La signature est donc reconnue comme mode de validation associée à l'identité d'une personne [19].

Dans ce chapitre nous allons expliquer le fonctionnement et le processus de notre système d'authentification des signatures manuscrites d'une manière générale et abrégé, plusieurs programmes et méthodes sont mises en œuvre pour l'authentification des signatures, en utilisant la simulation sur Matlab pour visualiser les résultats à vouloir obtenir.

IV.2. Différences entre signature en ligne ou hors ligne

IV.2.1. La vérification de la signature En ligne (Online)

La vérification de la signature est l'opération de l'authentification d'une personne en se basant sur sa signature. Etant donné un utilisateur U et une signature S, le système de vérification doit déterminer si la signature S est produite par l'utilisateur U. Si la réponse est oui, l'utilisateur est accepté comme client du système, sinon, il est considéré comme imposteur (intrus).

Dans un système de vérification de signatures en ligne, les utilisateurs sont d'abord introduits dans le système par l'enregistrement de quelques échantillons de leurs signatures qui servent de références. Plus tard, quand un utilisateur, qui prétend être un client particulier du système, présente sa signature pour la vérification, elle est comparée avec les signatures de référence de l'individu proclamé. A l'issue de cette comparaison un score mesurant la similarité (ou la dissimilarité) entre les deux signatures est fourni. Si le score de similarité est supérieur (score de dissimilarité inferieur, respectivement) `a un seuil, fixé à l´étape d'entraînement du système, l'utilisateur est accepté, sinon il est rejeté.

Dans le cas d'un système en ligne, la signature est effectuée sur une tablette graphique ou tout autre support muni d'un stylet électronique. La signature est donc représentée par une suite de points définis par au moins 3 valeurs : x, y, t. Nous avons remarqué, lors de nos

expérimentations, que les dispositifs actuels d'acquisition de l´écriture manuscrite en ligne sont loin d'offrir une ergonomie suffisante pour que les usagers les utilisent sans stress.

En effet, la gêne occasionnée entraîne des efforts supplémentaires. Beaucoup de personnes adaptent ou modifient leur manière d'´écrire et de signer lors du passage sur un support numérique. Cela est critique lorsqu'il s'agit de signer car on ne signe pas de la même manière sur papier ou avec un stylet et un temps d'adaptation au support numérique est donc n´nécessaire avant d'obtenir une stabilité suffisante de la signature [20][21].

IV.2.2. La vérification de la signature Hors ligne (Offline)

Dans un système hors ligne, la signature est effectuée sur un support papier puis scannée. La signature est donc assimilée à une image en niveaux de gris. C'est le cas notamment pour les systèmes de vérification de chèques. En hors ligne, on ne dispose pas de la dynamique de façon directe mais d'autres informations sont disponibles comme l'épaisseur du trait ou la variation d'intensité du niveau de gris constituant la signature. Au contraire, lors d'une acquisition en ligne, le trait n'a pas d'épaisseur et est représenté avec la même intensité sur les systèmes ne permettant pas l'acquisition de la pression. Hormis pour l'´étude de la forme, les techniques appliquées en hors-ligne ne peuvent donc pas, en général, être adaptées aux techniques en ligne puisqu'elles sont basées la plupart du temps sur l'étude des niveaux de gris de l'image.

Les problèmes liés à l'acquisition sont différents dans le cadre du en ligne et dans celui du hors ligne.

En effet, en hors ligne, le papier utilisé pour signer peut être de différentes textures, le stylo a aussi une grande influence et enfin l'acquisition via le scanner peut donner des résultats différents suivant la résolution choisie. C'est aussi le cas pour les systèmes d'acquisition en ligne pour lesquels la résolution ou la fréquence d'acquisition ne sont pas fixées [20][21].

IV.3. Fonctionnent d'un système biométrique

Les systèmes biométriques fonctionnent selon trois modes que sont l'enrôlement, la vérification d'identité et l'identification [22] :

- Enrôlement : L'enrôlement est la première phase de tout système biométrique. Il s'agit de l'étape pendant laquelle un utilisateur est enregistré dans le système pour la première fois. Elle est commune à la vérification et l'identification. Pendant l'enrôlement, la caractéristique biométrique est mesurée en utilisant un capteur biométrique afin d'extraire une représentation numérique. Cette représentation est ensuite réduite, en utilisant un algorithme d'extraction bien défini, afin de réduire la quantité de données à

stocker pour ainsi faciliter la vérification et l'identification. Le module d'enrôlement correspond à l'enregistrement biométrique des individus dans la base de données du système.

- **Vérification** : La vérification d'identité consiste à contrôler si l'individu utilisant le système est bien la personne qu'il prétend être. Le système compare l'information biométrique acquise avec le modèle biométrique correspondant stocké dans la base dedonnées, on parle de test 1 : N. Dans ce cas, le système renvoie uniquement une décision binaire (oui ou non) pouvant être pondérée.

- **Identification** : En mode identification, le système biométrique détermine l'identité d'un individu inconnu à partir d'une base de données d'identités, on parle de test 1 : N. Dansce cas, le système peut alors soit attribuer à l'individu inconnu l'identité correspondant au profil le plus proche retrouvé dans la base (ou une liste des profils proches), soit rejeter l'individu.

L'identification est un problème de recherche du plus proche voisin parmi un ensemble de possibilités alors que la vérification est un problème de discrimination à deux classes, acceptation ou rejet. Par conséquent, les approches utilisées ne sont pas les mêmes pour ces deux problèmes. Alors que tous les modèles sont disponibles pour un problème d'identification, la difficulté de la vérification est accrue car on ne dispose que du modèle d'une personne à chaque fois pour prendre la bonne décision**.**

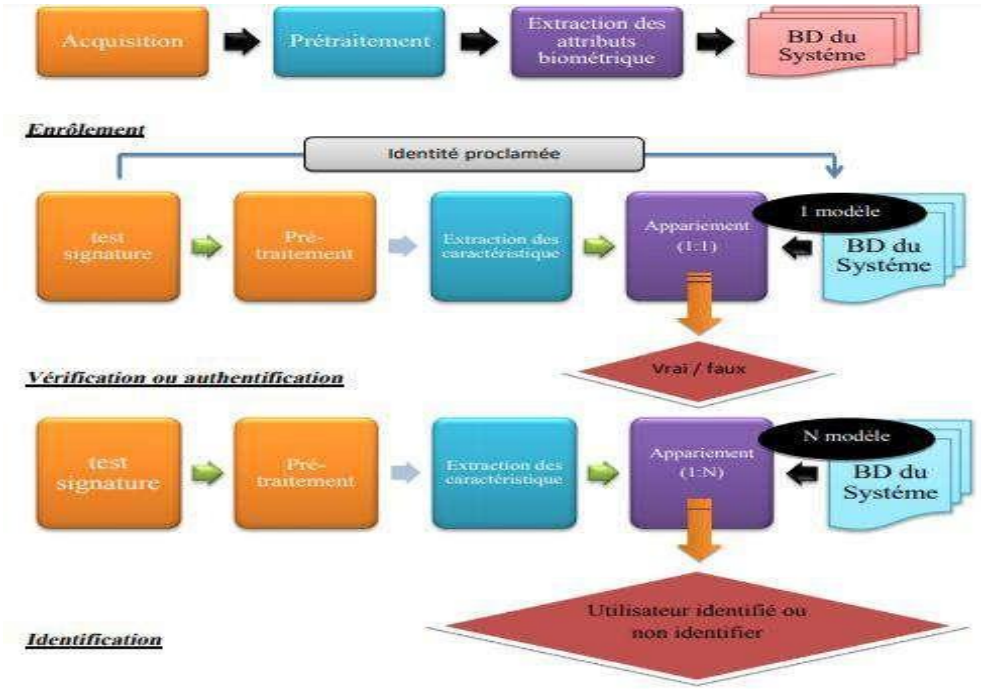


Figure IV. 1. Schéma de fonctionnement d'un système biométrique, Diagrammes des processus d'enroulement, de vérification et d'identification.

Les schémas d'un système de vérification et d'un système d'identification sont illustrés dans la figure (IV.1) [22]. Pendant la phase d'enrôlement, la caractéristique biométrique d'un individu est capturée par un lecteur biométrique. Un contrôle de qualité est généralement effectué pour s'assurer que la prise de l'échantillon est effectuée de manière fiable et pour garantir une bonne qualité de l'acquisition.

IV.4. Processus de vérification de signature hors ligne

Comme dans tout système de reconnaissance biométrique, la reconnaissance de signature manuscrite hors ligne passe principalement par quatre étapes : les prétraitements, l'extraction des caractéristiques, classification et l'appariement de caractéristiques. Dans ce qui suit nous détaillons chacune de ces étapes qu'on a appliquées à la base de données GPDS 100.

IV.4.1. Prétraitements

La plupart des systèmes de reconnaissance comportent une étape de prétraitement après que l'acquisition est faite, son but est amélioré les résultats et les performances du module de reconnaissance. Dans notre système en a passé par les opérations suivantes :

- Réduction de bruit **:** cette étape vise à nettoyer l'image de l'entrée, éliminer les points redondants car ces points-là vont causer les confusions pour le classificateur.
- Normalisation : les tailles des images de caractères sont variées. Ce phénomène peut perturber le système de reconnaissance des formes. On a besoin de normaliser les images obtenues l'hors de la lecture de la base GPDS 100 on a choisi de normaliser toute l'image a une taille de 255. Le classificateur va effectuer plus efficacement sur les images homogènes. La figure ci-dessous donne quelques résultats de la normalisation de la taille.

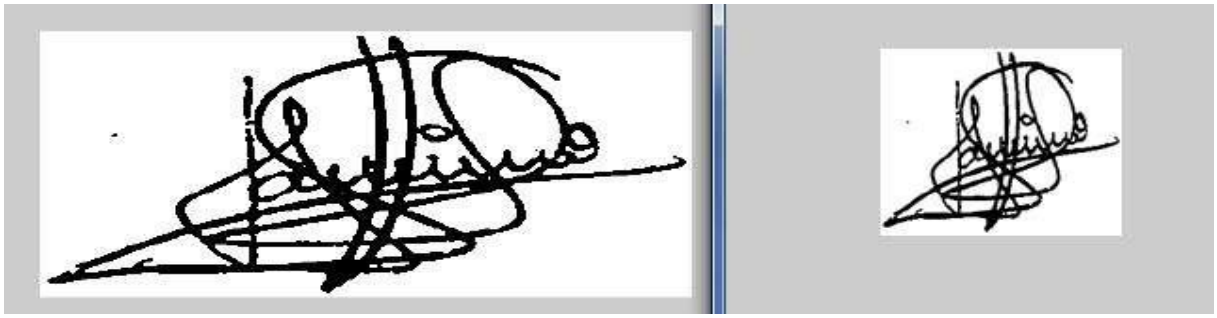


Figure IV. 2. Un échantillon de signatures avant (à gauche) et après (à droite) normalisationde la taille.

- **Squelettisation** : dans la plupart des cas, la forme à reconnaitre ne dépend pas

géométriquement de l'épaisseur du tracé de l'objet, la squelettisation est une procédure qui a pour but de réduire l'épaisseur du tracé d'un caractère à un pixel seulement. L'amincissement jusqu'à ce que l'épaisseur reste un seul point peut constituer une procédure très utile.

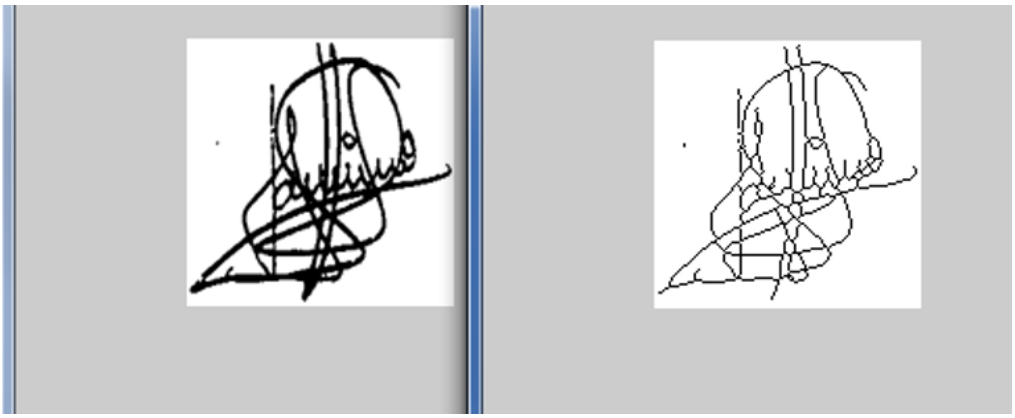


Figure IV. 3. La squelettisation d'un échantillon de signature.

IV.4.2. Extraction des caractéristiques

Cette étape représente le cœur du système de reconnaissance, on extrait de l'image les informations qui seront sauvegardées en mémoire pour être utilisées plus tard dans la phase de décision. L'analyse est appelée indexation, représentation, modélisation ou extraction de caractéristiques. L'efficacité de cette étape a une influence directe sur la performance du système de reconnaissance de signature.

IV.4.3. Classification et décision

La classification est l'élaboration d'une réglé de décision qui transforme les attributs caractérisant les formes en appartenance à une classe (passage de l'espace de codage vers l'espace de décision). Comme tout système biométrique, avant qu'un modèle de décision ne soit intégré dans un système de reconnaissance de signature, il faut avoir procédé auparavant à deux étapes : l'étape d'apprentissage et l'étape de test.

IV.4.3.1 Phase d'apprentissage

L'étape d'apprentissage consiste à caractériser les classes de formes de maniéré à bien distinguer les familles homogènes de formes. L'apprentissage consiste à mémoriser les représentations calculées dans la phase analyse pour les individus connus.

IV.4.3.2 Phase de test

Permet d'évaluer les performances du classificateur pour un apprentissage donné. Elle consiste à modéliser les paramètres extraits d'une signature ou d'un ensemble de signaturesd'un individu en se basant sur leurs caractéristiques communes.

L'apprentissage consiste donc à mémoriser les représentations calculées dans la phase analyse

pour les individus connus. Généralement les deux étapes d'analyse et d'apprentissage sont confondues et regroupées en une seule étape.

- **La décision** : C'est l'étape qui fait la différence entre un système d'identification d'individus et un système de vérification. Dans cette étape, un système d'identification consiste à trouver le modèle qui correspond le mieux à la signature prise en entrée à partir de ceux stockés dans la base de données, il est caractérisé par son taux de reconnaissance. Par contre, dans un système de vérification il s'agit de décider si la signature en entrée est bien celui de l'individu (modèle) proclamé ou il s'agit d'un imposteur. Pour estimer la différence entre deux images, il faut introduire une mesure de similarité. On définit ainsi plusieurs facteurs de performances du système tels que :
- Le taux de reconnaissance : qui présente le pourcentage des caractères reconnus parmi les caractères présentés.
- Taux d'erreurs : qui représente le pourcentage des caractères acceptés par le système mais classés de façon incorrecte.
- Le taux de rejet : qui représente le pourcentage des caractères rejetés parmi les caractères présentés.
- Le taux d'ambiguïté : qui représente le pourcentage des caractères ambigus parmi les caractères présentés.

IV.5. Extraction des caractéristiques

L'étape d'extraction des paramètres réduit les dimensions des images de signatures originales

tout en préservant et en extrayant les informations importantes codées dans l'image. Un ensemble soigneusement sélectionné de caractéristiques transformera les images afin qu'il devienne plus facile de distinguer entre les classes authentiques et falsifiées, nous présentons dans ce chapitre les techniques que nous avons utilisées dans le but d'extraire des informations biométriques texturées.

IV.6. Base des donnés

Nous avons utilisé la base des données GPDS 100. Elle est numérisée à 600 dpi, ce qui garantit une représentation suffisante de la texture grise. Dans la base de données GPDS 100, tous les utilisateurs ont signé avec leurs propres stylos sur différentes surfaces. Le corpus de signature GPDS-100 contient 24 signatures authentiques et 30 contrefaçons de 100 individus. Donc, il y a 100 x 24 donnantes 2400 signatures authentiques et 100 x 30 donnantes 3000 contrefaçons [35]. Les signataires ont utilisé leur propre stylo sur du papier blanc A4, après que les formulaires de signature ont été recueillis, chacun a été numérisé sur 256 niveaux de gris à une résolution de 600 dpi.

IV.7. Méthodologie

L'identification est un problème de recherche du plus proche voisin parmi un ensemble de possibilités alors que la vérification est un problème de discrimination à deux classes, acceptation ou rejet.

Notre système biométrique [23], nécessite deux phases opérationnelles. La première est unephase d'apprentissage : elle consiste à enregistrer les caractéristiques de signature hors ligne dechaque individu afin de créer son propre modèle biométrique ; puis a été enregistré dans la base de données. La deuxième phase est la phase de test qui consiste à enregistrer les mêmes caractéristiques et à les comparer aux modèles biométriques stockés dans la base de données si les données enregistrées correspondent à un modèle biométrique de la base de données. Le schéma général est représenté sur la figure (IV.9).

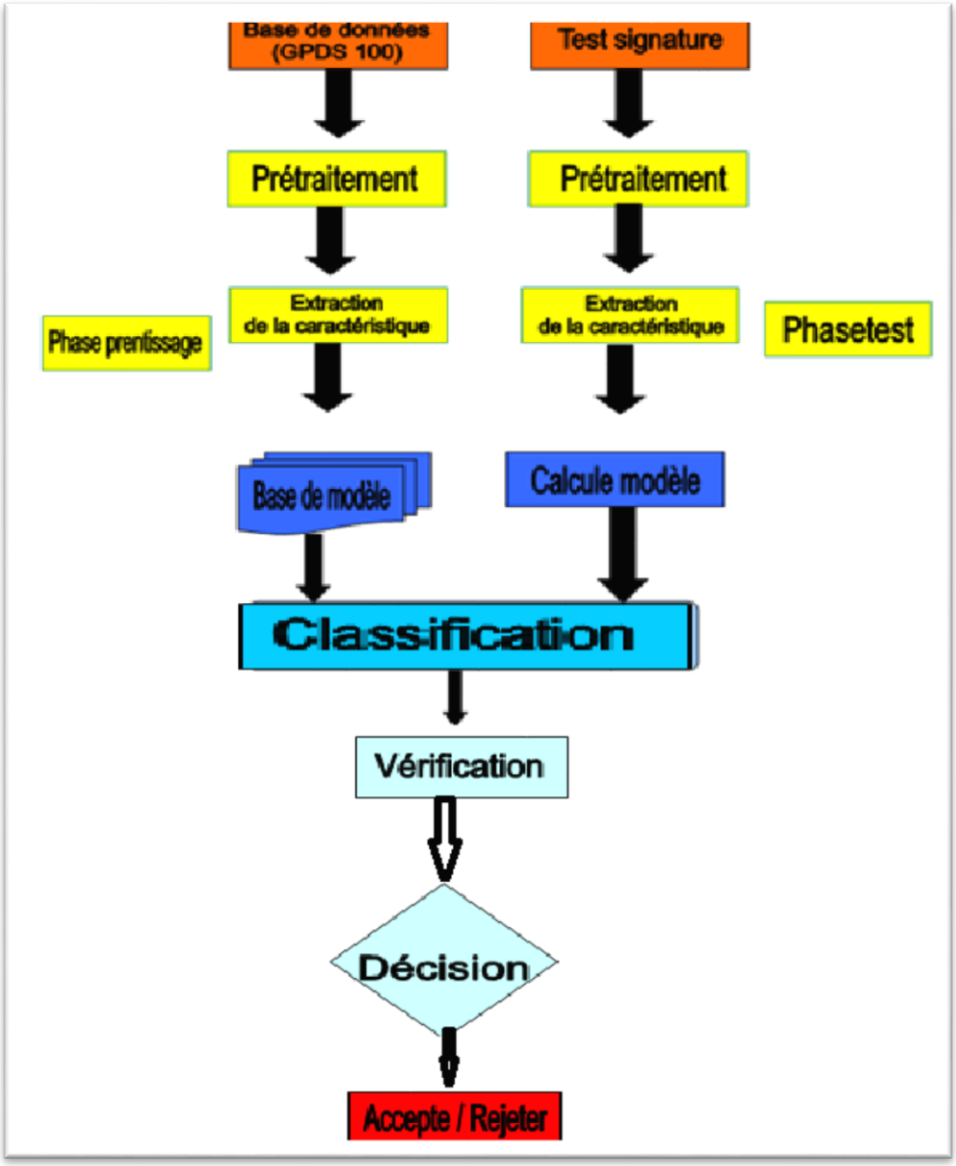


Figure IV. 4. Schéma synoptique de notre système d'authentification de signature hors ligne proposé.

IV.7.1. Système de reconnaissance automatique de signature basé sur le Deep Learning

Comme nous l'avons précisé dans le chapitre 3 ; la structure de CNN contient des couches Convolutional, Pooling, Rectified Linear Unit (ReLU) et Fully Connected [24]

- ✓ **Première étape pour la conception du CNN** :

Avant tout, nous devons définir l'architecture du réseau CNN utilisé en construisant les différentes couches de traitement :

1. Couche de convolution (CONV) qui traite les données d'un champ réceptif ; La couche de convolution est le bloc de construction de base d'un CNN. Le détail de son fonctionnement est précisé dans le chapitre précédent.
2. Couche de pooling (POOL), qui permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par sous-échantillonnage).
3. Couche de correction (ReLU), souvent appelée par abus « ReLU » en référence à la fonction d'activation (Unité de rectification linéaire)
4. Couche « entièrement connectée » (FC), qui est une couche de type perceptron ; FC est considéré comme la dernière couche de pool alimentant les fonctionnalités d'un classificateur qui utilise la fonction d'activation Softmax. La somme des probabilités de sortie de la couche entièrement connectée est 1. Ceci est assuré en utilisant le Softmax comme fonction d'activation. La fonction Softmax prend un vecteur de scores réels arbitraires et l'écrase sur un vecteur de valeurs entre zéro et un qui somme à un.

- ✓ **Deuxième étape pour la conception du CNN :**

Dans cette étape, nous procédons au paramétrage des couches utilisées : trois hyper paramètres permettent de dimensionner le volume de la couche de convolution (aussi appelé volume de sortie) : la profondeur, le pas et la marge.

- Profondeur de la couche : nombre de noyaux de convolution (ou nombre de neurones associés à un même champ réceptif).
- Le pas contrôle le chevauchement des champs réceptifs. Plus le pas est petit, plus les champs réceptifs se chevauchent et plus le volume de sortie sera grand.
- La marge (à 0) ou zero padding : parfois, il est commode de mettre des zéros à la frontière du volume d'entrée.

La section suivante décrit l'algorithme proposé et les résultats expérimentaux obtenus par le système d'authentification.

IV.7.2. L'algorithme utilisé

L'algorithme est principalement réalisé en trois étapes comme ci-dessous [24]:

1) Redimensionner les images d'entrée.
2) Construire une structure CNN : filtre convolutif, batch normalisation, ReLu et de regroupement maximum (Max Pooling).
3) Après avoir extrait toutes les fonctionnalités, utiliser le classificateur Softmax pour la classification.

La structure du réseau de neurones convolutif, utilisé ici pour extraire les caractéristiques se compose de 11 niveaux de couches de convolution et le regroupement à la sortie basé sur la fonction softmax.

IV.8. Résultats expérimentaux et discussion

IV.8.1. Bases de données

Nous avons utilisé la base des données GPDS 100. Elle est numérisée à 600 dpi, ce qui garantit une représentation suffisante de la texture grise. Dans la base de données GPDS 100, tous les utilisateurs ont signé avec leurs propres stylos sur différentes surfaces. Le corpus de signature GPDS-100 contient 24 signatures authentiques et 30 contrefaçons de 100 individus. Donc, il y a 100 x 24 donnantes 2400 signatures authentiques et 100 x 30 donnantes 3000 contrefaçons [35]. Les signataires ont utilisé leur propre stylo sur du papier blanc A4, après que les formulaires de signature ont été recueillis, chacun a été numérisé sur 256 niveaux de gris à une résolution de 600 dpi.

IV.8.2. Résultats d'authentification de signature

Cette section décrit les expériences réalisées et les résultats expérimentaux du système de reconnaissance. Dans ce travail, nous avons testé les résultats pour mettre en œuvre l'algorithme CNN et vérifier sa fonctionnalité et efficacité sur la base de données GPDS-100 pour la tâche d'authentification de signature manuscrite hors ligne.

Pour montrer à quel point le CNN s'est bien comporté pendant la formation, nous extrayons les courbes ROC, le taux de fausse acceptation (FAR), le taux de faux rejet (FRR) et à partir de ceux-ci, nous calculons le taux d'erreur égal (ERR).

Les courbes ROC sont utiles pour organiser les classificateurs (deux classes) et pour afficher leurs performances. Pour avoir une idée de leur fonctionnement, nous commençons par la matrice de confusion d'un classifieur qui a une classe positive et une classe négative.

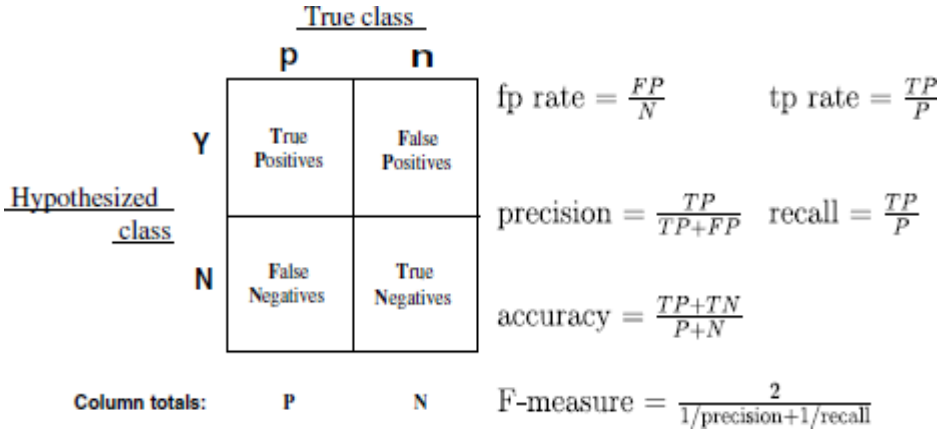


Figure IV. 5. La matrice de confusion et les grandeurs usuelles mesurées

Donner un échantillon au classificateur a quatre versions. L'échantillon est positif et est classé comme positif (TP) ou négatif (FN). L'échantillon est négatif et est classé comme négatif (TN) ou positif (FP).

Dans le cas des signatures, la taille FP est la taille FAR et exprime ce que son nom implique, le pourcentage d'échantillons acceptés dans le système (considérés comme positifs alors que non). Respectivement, le FRR est la taille FN (pourcentage de signatures rejetées alors qu'elles étaient authentiques), et peut autrement être calculé comme $FRR = 1 - TP$.

Dans l'image les éléments de la diagonale principale représentent les bonnes décisions du classificateur et de l'autre diagonale la confusion.

Par conséquent, les courbes ROC sont des graphiques bidimensionnels dans lesquels le FAR est sur l'axe x et le TP (vrais positifs) sur l'axe y. Il représente donc une relation entre l'avantage (vrais positifs) et le coût (faux positifs). Il est évident que les points en haut à gauche de la courbe sont les plus souhaitables.

Les points sur la diagonale principale $y = x$ expriment la pire performance possible du classificateur car le pourcentage d'échantillons positifs mal appariés est égal au pourcentage de positifs correctement triés, c'est-à-dire que nous avons un caractère aléatoire absolu.

Le classificateur binaire pour décider à quelle classe appartient un échantillon produit un score, lorsque cette valeur est supérieure à une valeur seuil, l'échantillon correspond à la classe positive sinon à la négative. Pour chaque valeur de seuil différente, nous avons un point différent dans l'espace ROC. Une courbe continue est idéalement produite à partir d'échantillons infinis. Parce qu'en pratique les échantillons sont limités, une fonction en escalier est produite.

Nous avons mentionné que les courbes ROC reflètent les performances du classifieur. Mais pour comparer les performances des classificateurs entre eux, un gradient est plus utile qu'une fonction. La méthode la plus courante consiste à calculer l'aire sous la courbe AUC (aire sous la courbe) dont la valeur varie de 0 à 1,04. Pour le classifieur dont les prédictions sont complètement aléatoires (ligne droite) l'aire sous la courbe est de 0,5.[16]

Les résultats sont divisés en deux catégories :

• global : pour les contrefaçons aléatoires et les contrefaçons de compétences

• falsifications de compétences : uniquement pour les falsifications de compétences

Ci-dessous la courbe ROC du système d'authentification, ainsi que le tableau et les figures respectivement avec les valeurs AUC, FC calculées, sont présentées dans les tableaux (IV.1) et (IV.2).

Tableaux IV. 1

AUC	1	2	3	4	5	6	7	8	9	10
Global	0.99	0.97	0.87	0.96	0.94	0.90	0.85	0.74	0.89	1
FC	1,04	1,03	0.94	1,02	0.01	0.98	0.97	0.8	0.99	1

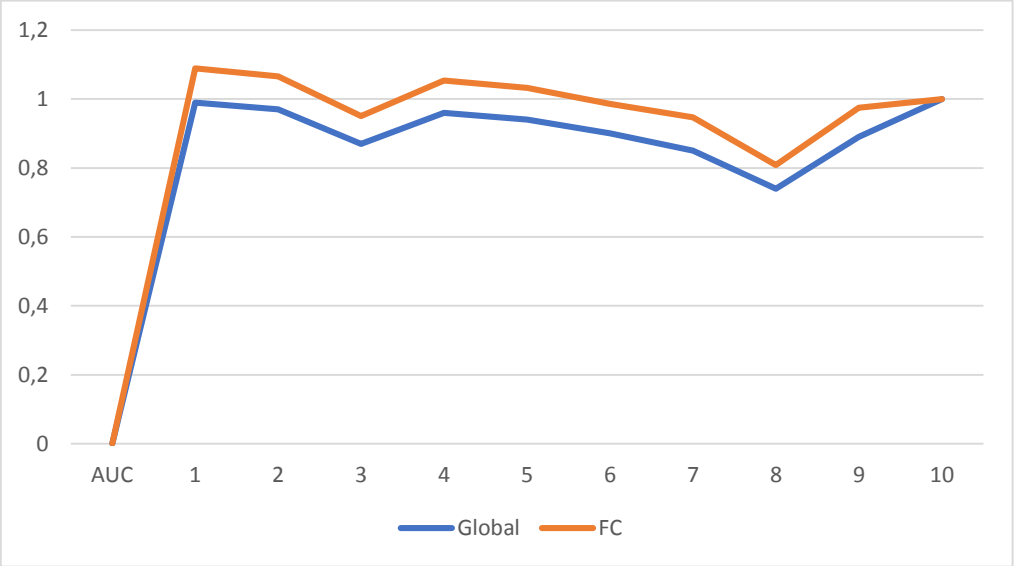


Figure IV. 6. Courbes AUC, FC.

L'EER (taux d'erreur égal) pour lequel $EER = FAR$ = FRR, est également mesuré.

Tableaux IV. 2

ERR	1	2	3	4	5	6	7	8	9	10
FC	0.04	0.05	0.20	0.12	0.09	0.19	0.22	0.28	0.16	0

L'EER moyen pour tous les écrivains est $EERmean$ = 0,13

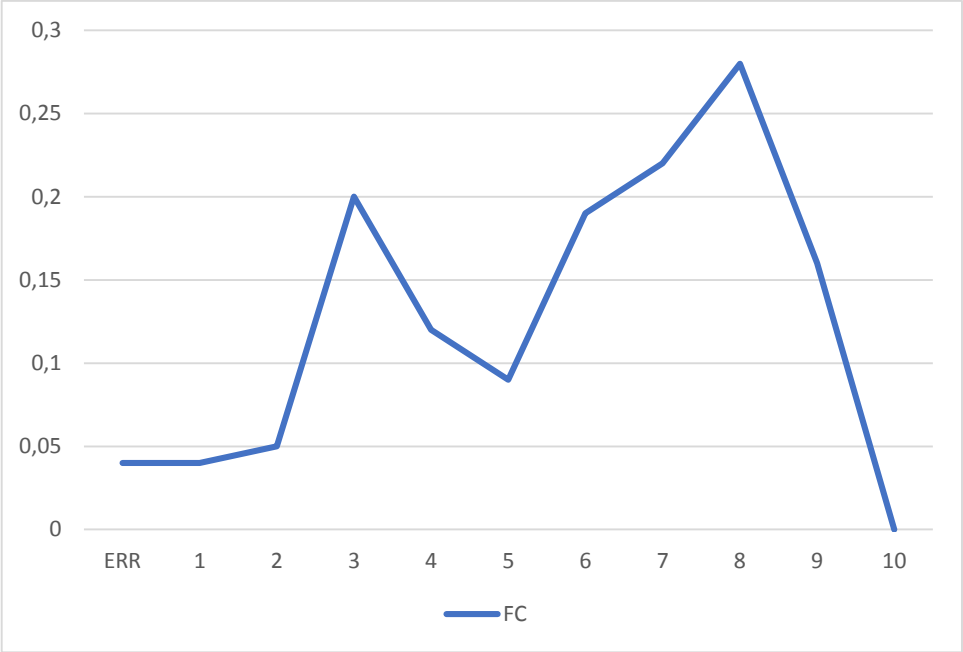


Figure IV. 7. Courbes FC, ERR.

Les résultats obtenus des tableaux (IV.1 et IV.2) présentent simultanément la performance de CNN pour l'authentification de la base GPDS100.

La courbe caractéristique (ROC), qui est un terrain de FRR contre FAR est représenté par la figure ci-dessous :

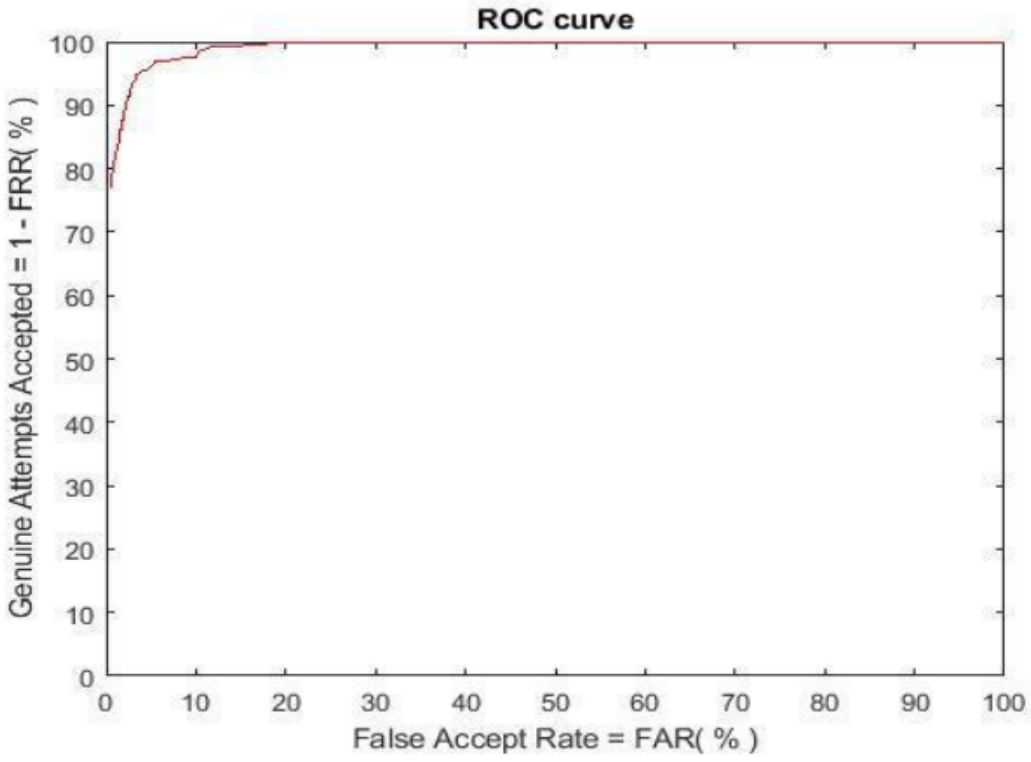


Figure IV. 8. Courbes de performance ROC.

Cette courbe des caractéristiques (figure 4.8), représente les performances d'un système biométrique par un graphique nommé courbe ROC (Receiver Operating Characteristic). Qui représente les valeurs de FRR en termes de FAR. Ceci est obtenu en calculant le couple (FAR, FRR).

La précision du modèle sur l'ensemble de données d'entraînement et de test pour chaque période d'entraînement est représenté par la figure d'apprentissage suivant :

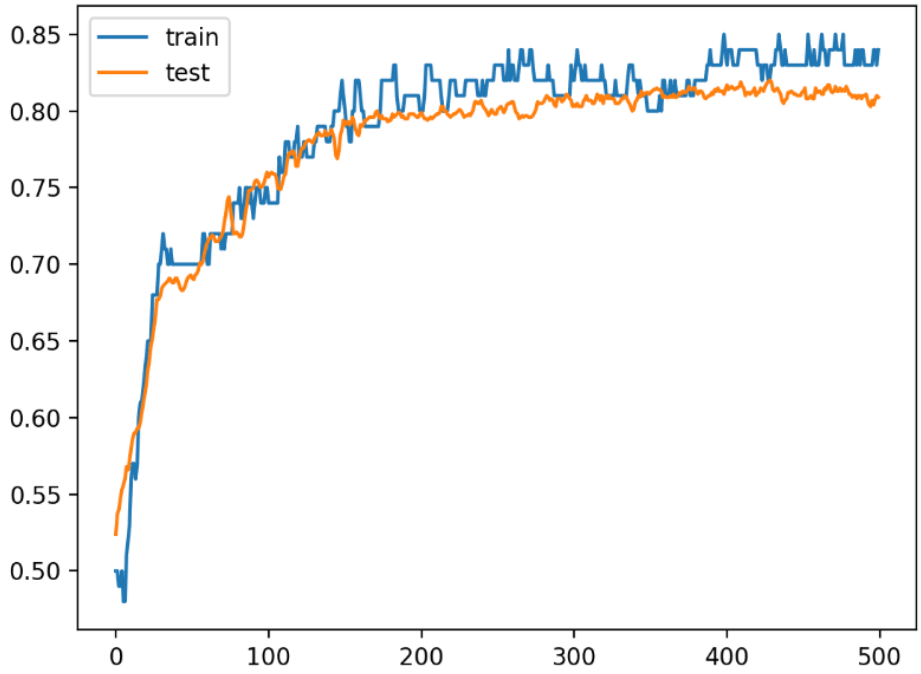


Figure IV. 9. Courbes d'apprentissage de la précision du modèle sur l'ensemble de données d'entraînement.

Discussion : Dans cette expérience les performances obtenues montrent le CNN est très efficace pour l'authentification des signatures manuscrite, nous concluons que le système d'authentification biométrique des personnes est fiable vu que les résultats obtenus sont satisfaisants. L'étude de l'ensemble des tests effectués ont permis de conclure, qu'avec l'utilisation du Deep Learning nous avons apporté une amélioration considérable du EER et des deux taux d'erreurs FAR et FRR. Dans le cas de la base de données traité, la courbe ROC pour les différentes itérations montrent que le système donne une bonne précision.

IV.9. Conclusion

Dans ce chapitre, nous présentons la conception d'un système de reconnaissance automatique de signature basée sur le Deep Learning utilisant CNN (Convolutional Neural Network). Le principe général des Réseaux de Neurones Artificiels (RNA) est à l'origine inspiré de certaines fonctions de base des neurones naturels du cerveau. Un réseau de neurones artificiel est généralement organisé en plusieurs couches : une couche d'entrée, une couche de sortie, des couches intermédiaires appelées couches cachées. Le réseau de neurones réalise un travail de classement lors de la reconnaissance, les grands avantages des réseaux de neurones résident dans leur capacité d'apprentissage automatique, ce qui permet de résoudre des problèmes sans nécessiter l'écriture de règles complexes. Dans ce chapitre, nous présentons donc la conception d'un système de d'authentification automatique de signature basé sur CNN (Convolutional Neural Network),

Conclusion générale

Conclusion générale

La biométrie est un domaine en expansion dont le nombre de recherches est en croissance continue dont le but est d'aboutir à un moyen efficace, fiable et rapide pour identifier les personnes. Elle utilise, des outils mathématiques souvent très développés, pour identifier et reconnaître des individus.

Malgré le développement considérable des différentes modalités biométriques, les signatures sont restées le mécanisme d'authentification le plus largement accepte dans les documents juridiques et les transactions financières.

La signature manuscrite est depuis plusieurs siècles le moyen le plus répandu. Elle est le moyen biométrique d'authentification le plus utilisé et accepté. La signature manuscrite d'un individu représente un bon compromis, tout en étant relativement fiable, elle est facile à acquérir, socialement acceptée comme un mode de reconnaissance. La signature est un moyen utilisé depuis longtemps, pour authentifier des documents, pour responsabiliser les individus face à des engagements (contrats, etc.). La signature est donc reconnue comme mode de validation associé à l'identité d'une personne.

C'est dans ce cadre que s'inscrit notre travail, qui a pour objectif de proposer un système d'authentification de signature. Le système de vérification de signature hors ligne est développé pour distinguer les signatures authentiques ou falsifiées.

Dans ce travail, nous présentons la conception d'un système de reconnaissance automatique de signature basée sur le Deep Learning utilisant CNN (Convolutional Neural Network). Le principe général des Réseaux de Neurones Artificiels (RNA) est à l'origine inspiré de certaines fonctions de base des neurones naturels du cerveau. Un réseau de neurones artificiel est généralement organisé en plusieurs couches : une couche d'entrée, une couche de sortie, des couches intermédiaires appelées couches cachées. Le réseau de neurones réalise un travail de classement lors de la reconnaissance, les grands avantages des réseaux de neurones résident dans leur capacité d'apprentissage automatique, ce qui permet de résoudre des problèmes sans nécessiter l'écriture de règles complexes.

Aussi nous présentons donc la conception d'un système de d'authentification automatique de signature basé sur CNN (Convolutional Neural Network), ce choix est justifié par la simplicité et l'efficacité de la méthode. Après, les résultats sont pris et discutés.

Références bibliographiques

Références bibliographiques

[1] Ibtissam BENCHENNANE. Etude et mise au point d'un procède biométrique multimodale pour la reconnaissance des individus. PhD thesis, Université sciences et technologie-Oran, 2015.

[2] S. AKROUF, "Une Approche Multimodale pour l'Identification du Locuteur", thèse de doctorat, Université Ferhat Abbas- Sétif , 2011.

[3] L. ALLANO, 'La Biométrie multimodale : stratégies de fusion de scores etmesures de dépendance appliquées aux bases de personnes virtuelles', thèse de doctorat, Université D'Avery Val D'Essonne, 2009.

[4] ZITOUNI Sif Eddine SACI Abdelmoumen, "Authentification et Identification biométrique des personnes par les empreintes palmaires ", Université Kasdi Merbah-Ouargla, 2016.

[5] RAKOTONIRINA Bodoarivola, "ALGORITHME DE RECONNAISSANCE DE SIGNATURESMANUSCRITES EN LIGNE", Université d'Antananarivo-Madagascar, 2019.

[6] ZITOUNI Sif Eddine et SACI Abdelmoumen, Authentification et Identification biométrique des personnes par les empreintes palmaires, Mémoire MASTER ACADEMIQUE, UNIVERSITE KASDI MERBAH OUARGLA, 2016.

[7] https://sites.google.com/site/tpelabiometrie/home/biometrie-par-reconnaissance-vocale

[8]https://www.staffgroup.fr/avantages-et-inconvenients-de-la-verification-biometrique/#:~:text=La%20biom%C3%A9trie%20n'est%20pas,ce%20n'est%20pas%20possible

[9] Doufa Ismahane et Tadjine Abdelkhaleq, "Identification biométrique des personnes par signature manuscrite"Université 8Mai 1945 – Guelma, 2022.

[10] BENCHADI Djafer Yahia Messaoud, "Etude Comparative de Différents Descripteurs Locaux Dans La Vérification Faciale", Université Mohamed Khider-Biskra, 2020.

[11] https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529

[12] https://mrmint.fr/introduction-k-nearest-neighbors

[13] CHOUCHANE Ammar, " Analyse d'images d'expressions faciales et orientation de la tête basée sur la profondeur", thèse de doctorat, Université Mohamed Khider-Biskra,2016.

[14] https://towardsdatascience.com/particle-swarm-optimization-visually-explained-46289eeb2e14

[15] ZACCONE Giancarlo, MD REZAUL Karim, MENSHAWY Ahmed, Deep learning with tensorflow. 2017.

[16] Voir Wikipédia https://fr.wikipedia.org/wiki/Apprentissage_profond à partir de cette dernière mise à jour en février 2022.

[17] https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/

[18] Voir Wikipédia https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_artificiels à partir de cette dernière mise à jour en 9 mai 2022.

[19] YANG, Ming-Hsuan. Kernel Eigen faces vs. Kernel Fisher faces : Face Recognition Using Kernel Méthodes. In : Fgr. 2002. p. 215.

[20] R. Sabourin et G. Genest et F. J. Prêteux: Off-Line Signature Verification by Local Granulometric Size Distribution, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19,no. 9, pp. 976-988, 1997.

[21] C. Santos et E.J.R. Justino et F. Bortolozzi et R. Sabourin : An Off-Line Signature Verification Method based on the Questioned Document Expert's Approach and a Neural Network Classifier, International Workshop On Frontiers in Handwriting Recognition (IWFHR), Tokyo (Japon), pp. 498-502., 2004. M. Wirotius, A. Seropian et N. Vincent, "Writer Identification from Gray Level Distribution", Proceedings of the International Conference on Document Analysis and Recognition (ICDAR'03), Edinburgh (Ecosse). pp. 1168-1172, 2003.

[22] THÈSE Présentée en vue de l'obtention du diplôme en Electronique Doctorat 3ème Cycle en LMD, Présentée par : Hedjaz HEZIL. THÈSE dirigée par : Rafik DJEMILI Professeur des Universités Skikda.

[23] L. BOUCERREDJ.al. ''Etude de la fiabilité d'un système biométrique dédie à la reconnaissance de signatures manuscrites''. Séminaire international sur l'industrie et la technologie en ligne (webinaire),12 et 13 Mars 2021, Oran, Algérie.

[24] Majister T. Hocine. Deep learning , Université Mohamed Khider Biskra.

[25] Ruder, S., Single-Layer Neural Networks and Gradient Descent. 2017.