UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

VÉRIFICATION DE SIGNATURES MANUSCRITES BASÉE SUR LE DEEP LEARNING

MÉMOIRE (OU ESSAI) PRÉSENTÉ
COMME EXIGENCE PARTIELLE DE LA

MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE APPLIQUÉES

PAR
OMAR BEN SLIMENE

NOVEMBRE 2024

Résumé

Dans le contexte de sécurité accrue et de lutte contre la fraude, la vérification des signatures manuscrites est primordiale. En effet, ces dernières sont couramment utilisées pour authentifier des documents officiels, et leur falsification peut avoir de graves conséquences. Le présent mémoire vise à développer un système de vérification de signatures manuscrites hors ligne avec apprentissage profond, intégrant deux modèles d'apprentissage automatique et une interface graphique

Le premier modèle, un réseau Siamois, est conçu pour évaluer la similarité entre une signature de référence et une à vérifier. Le second, basé sur l'architecture VGG16, est utilisé pour l'authentification des signatures en extrayant les caractéristiques distinctives des images de signatures et en les classant pour vérifier leur authenticité, renforçant ainsi le processus de vérification. Les deux modèles sont intégrés à une interface graphique qui permet de soumettre des vérifications et de visualiser les résultats de manière claire.

La méthodologie de cette mémoire comprend la collecte de données de signatures authentiques et falsifiées, leur prétraitement pour améliorer la qualité et la pertinence, et le développement de modèles d'apprentissage automatique. Une méthode de perte atypique, l'entropie croisée binaire (BCE), est utilisée pour optimiser la comparaison des signatures dans le cadre de similarité Siamois.

De plus, cette approche a été validée en utilisant des ensembles de données provenant de bases populaires et reconnues dans le domaine de la vérification des signatures, démontrant sa fiabilité par rapport aux variations des données et produisant des résultats satisfaisants.

Mot clés : Siamois, VGG16, BCE, Vérification

Abstract

In the context of heightened security and the fight against fraud, the verification of handwritten signatures is of prime importance. In fact, they are commonly used to authenticate official documents, and their falsification can have serious consequences. The aim of this memoir is to develop an offline handwritten signature verification system, integrating two machine learning models and a graphical user interface.

The first model, a Siamese network, is designed to evaluate the similarity between a reference signature and one to be verified. The second, based on the VGG16 architecture, is used for signature authentication by extracting distinctive features from images and classifying them to verify their authenticity, thus reinforcing the verification process. The graphical interface developed makes it possible to submit verifications and visualize the results in a clear manner.

The methodology of this dissertation includes the collection of authentic and forged signature data, their pre-processing to improve data quality and relevance, and the development of machine learning models. An untypical loss method, binary cross-entropy (BCE), is used to optimize signature comparison in the Siamese similarity framework.

In addition, this approach has been validated using datasets from popular and recognized signature verification databases, demonstrating its reliability against data variations and producing satisfying results.

Key words: Siamese, VGG16, BCE, Verification

REMERCIEMENTS

Je tiens à remercier toutes les personnes qui ont contribué à la réussite de mon projet et m'ont aidé à rédiger ce mémoire.

Je remercie tout particulièrement mes directeurs de recherche, Ismaïl Biskri et Fathallah Nouboud, pour leur soutien appréciable et leurs contributions à la réalisation de ce mémoire.

Je tiens également à remercier mes collègues pour leur aide précieuse dans la programmation et la mise en œuvre des méthodes développées dans le cadre de cette maîtrise.

Mes remerciements les plus sincères vont à ma famille : ma mère Leila, mon père Mohsen, mes frères Nejmeddine, Mohamed et son épouse Emna, pour leurs encouragements et leur soutien financier.

Enfin, je tiens à remercier tous mes amis qui, d'une manière ou d'une autre, ont participé à la réalisation, en particulier mon frère Mohamed et mon ami Amine Arfaoui pour leur soutien.

Table des matières

Liste des figures

Liste des tableaux

1 Introduction

L'identification d'une personne est un processus essentiel dans de nombreux contextes, notamment pour des raisons de sécurité, administratives, commerciales ou juridiques. Elle consiste à vérifier et à authentifier l'identité d'une personne à travers des informations ou des caractéristiques spécifiques. Elle permet de s'assurer que la personne qui se présente comme une certaine entité est bien celle qu'elle prétend être.

Au cours des années, les méthodes d'identification ont évolué, passant de techniques traditionnelles telles que les documents d'identité et les signatures manuscrites à des approches plus technologiques telles que la biométrie et les systèmes de reconnaissance faciale.

Bien que la technologie propose constamment de nouvelles alternatives plus efficaces, la signature manuscrite reste le moyen d'identification le plus utilisé dans le monde actuel. Elle demeure le moyen le plus simple et le plus rapide pour les individus de s'identifier, néanmoins, un grand problème de sécurité et de falsification persiste.

L'intelligence artificielle est en mesure de résoudre efficacement toutes ces difficultés. Un système intelligent automatisé est une solution méthodique rapide pour réduire le risque de falsification des signatures. De nos jours, des milliers de chercheurs se concentrent sur ce sujet, car, en tant qu'êtres humains, il est impossible de survivre sans sécurité.

Les systèmes de vérification automatique se divisent en deux grandes approches, selon le type de données et d'images à traiter. Les systèmes en ligne, également connus sous le nom de signatures électroniques, sont des outils numériques permettant de signer des documents électroniquement, sans recourir à la signature manuscrite traditionnelle, bien qu’important, ces systèmes ne seront pas développés dans ce mémoire. Nous nous concentrons sur les systèmes hors ligne, où les signatures sont numérisées à partir d'un support physique tel qu'un spécimen de chèque ou tout autre type de papier.

La vérification hors ligne des signatures manuscrites présente des défis plus complexes que la méthode en ligne. L'approche hors ligne nécessite l'analyse de signatures numérisées à partir de documents physiques, ce qui implique de tenir compte des variations de la qualité de l'image, de l'usure du papier et d'autres artefacts visuels. Contrairement à la méthode en ligne, où des paramètres dynamiques tels que la pression, la vitesse et le rythme de la signature sont disponibles, l'approche hors ligne doit s'appuyer uniquement sur les caractéristiques visuelles statiques de la signature, ce qui rend l'authentification plus difficile et plus exigeante en termes de traitement d'images et de reconnaissance des formes.

Dans la littérature, il existe quatre approches principales pour vérifier les signatures manuscrites : l'approche statistique, la correspondance des modèles, l'apprentissage automatique classique et profond. Les approches statistiques, comme les chaînes de Markov cachées (HMM), examinent les caractéristiques statistiques des signatures pour détecter les similarités et les différences [9]. Les techniques de correspondance de modèles, comme le Dynamic Time Warping (DTW), comparent les nouvelles signatures aux modèles de référence [3]. L'apprentissage automatique classique, avec des méthodes comme les forêts aléatoires et les SVM, a montré une efficacité notable [6]. L'apprentissage profond utilise des réseaux neuronaux convolutifs et siamois pour capturer des caractéristiques complexes et améliorer la précision de la vérification [17][18]. Ces techniques, combinées à d'autres méthodes permettent d'atteindre des taux de précision élevés.

Dans le domaine de la vérification des signatures manuscrites, plusieurs défis subsistent malgré les avancées technologiques. Parmi ces défis, on peut citer la variabilité naturelle des signatures, due à des facteurs tels que la fatigue, l'âge ou l'état émotionnel de l'auteur. De plus, la présence de signatures falsifiées et frauduleuses rend la tâche de vérification encore plus complexe.

L'objectif de notre recherche est de développer un système automatisé efficace capable de vérifier l'authenticité des signatures manuscrites. Pour ce faire, nous utilisons des méthodes d'analyse d'images pour extraire les caractéristiques des signatures et les intégrer dans un processus de vérification automatisé. En particulier, le système exploitera des techniques d'apprentissage profond pour améliorer la précision et

l'efficacité de la vérification en capturant des caractéristiques complexes des signatures.

L'hypothèse principale est que les signatures manuscrites authentiques présentent des caractéristiques uniques qui peuvent être détectées et exploitées par des algorithmes d'apprentissage automatique et ainsi les distinguer des contrefaçons. En outre, l'optimisation du temps d'exécution de ces algorithmes est essentielle pour qu'ils puissent être appliqués dans des environnements à forte demande, tels que la vérification de signatures en masse pour les institutions financières.

Notre méthodologie comporte quatre étapes visant à améliorer la précision de la vérification des signatures manuscrites hors ligne. Tout d'abord, la préparation des données, qui implique l'assemblage d'un grand nombre de signatures authentiques et falsifiées, en utilisant à la fois des bases de données publiques et des échantillons spécifiques. Ces signatures sont ensuite numérisées et nettoyées de tout bruit.

Une fois les signatures préparées, l'étape 2 consiste à mesurer la similarité entre les signatures en utilisant un modèle siamois, qui emploie une fonction de perte particulière, l'entropie croisée binaire (BCE). Cette approche facilite la comparaison des signatures en termes de performance et de temps d'exécution. Lors de la troisième étape, le modèle VGG16 est utilisé pour extraire les caractéristiques des signatures afin de vérifier leur authenticité. Ces deux modèles siamoise et VGG16 sont intégrés de manière cohérente dans une interface utilisateur graphique afin de créer un système pour la vérification des signatures.

Pour finir, les modèles sont validés et évalués sur les ensembles de validation et de test à l'aide de plusieurs métriques de performance. Les résultats sont ensuite analysés en comparant les performances obtenues avec différents ensembles d'entraînement et en examinant les erreurs pour identifier les possibilités d'amélioration.

Notre mémoire est organisée en six chapitres. Outre le premier chapitre consacré à l'introduction, le deuxième chapitre aborde les différentes approches proposées dans la littérature pour la vérification des signatures manuscrites. Il comportera également une description des méthodes utilisées par les développeurs lors de l'authentification de la signature manuscrite.

Le troisième chapitre, intitulé « Méthodologie » présente notre système pour l'acquisition et le prétraitement des images numérisées, ainsi que les différentes techniques développées lors de la phase de recherche pour parvenir à la solution qui sera finalement retenue.

Le quatrième chapitre, présente l'implémentation des modèles de réseaux de neurones convolutifs (VGG16) pour la vérification de l'authenticité des signatures et des réseaux siamois pour l'évaluation de leur similarité, en expliquant le rôle de chaque modèle.

Le cinquième chapitre, dédié à l'expérimentation, présente les résultats des tests effectués sur les bases de données de signatures. Cette étape validera les performances du système de vérification et d'authentification proposé en le comparant aux méthodes existantes. L'analyse des résultats permettra d'identifier les forces et les limites des modèles développés, ainsi que de proposer des pistes de recherche pour l'avenir.

Le sixième chapitre présente les conclusions et les perspectives de recherche.

Chapitre 1 État de l'art

Cette section présente une revue de la littérature des différentes recherches effectuées dans le domaine de l'authentification et de la reconnaissance automatique des signatures manuscrites. Dans ces travaux, on trouve deux concepts différents : la vérification de signature, qui consiste à déterminer si une signature est authentique ou fausse, et la reconnaissance de signature, qui consiste à comparer une nouvelle signature à une signature de référence.

1.1 Approches de reconnaissance des signatures hors ligne

Les technologies de reconnaissance des signatures manuscrites peuvent être classées en fonction du mode d'acquisition des données : vérification en ligne et hors ligne. La vérification en ligne traite des images dynamiques incorporant des caractéristiques telles que la pression de l'encre et la vitesse d'écriture. Ces images sont obtenues à partir de dispositifs tels que les écrans tactiles et les tablettes. Cette étude se concentre uniquement sur la reconnaissance hors ligne, où l'on traite des signatures papier numérisées à l'aide d'appareils photo ou de scanners.

Plusieurs projets ont été réalisés pour développer des systèmes de vérification de signature hors ligne, principalement à l'aide de techniques de vision par ordinateur et de logiciels informatiques.

La vision par ordinateur consiste à automatiser et intégrer les processus de perception visuelle en produisant des modèles 3D à partir des données d'images. Les techniques de calcul flou visent à traiter l'imprécision, l'incertitude et l'approximation afin d'atteindre une haute performance, une meilleure précision et efficacité en vérification de signatures.

1.2 Techniques de vérification des signatures

Les techniques de vérification des signatures peuvent être classées en quatre catégories: les approches statistiques, la correspondance de modèles, les modèles classiques d'apprentissage automatique et les modèles d'apprentissage profond. Dans cette section, les différentes méthodes de vérification proposées dans la littérature sont présentées et discutées par catégorie.

1.2.1 Approches statistiques

Les approches statistiques sont utilisées pour l'analyse des données obtenues à partir des caractéristiques statistiques des images de signatures. Ces techniques examinent des similarités et les différences entre les signatures. Les chaines de Markov cachées sont les modèles statistiques les plus utilisés dans la littérature.

L'article [9] propose un système de vérification de signatures des chèques basé sur le modèle de Markov caché (HMM) avec un nombre fixe d'état : projection horizontale, projection verticale, enveloppe droite, enveloppe gauche, enveloppe supérieure, enveloppe inférieure, variations de taille horizontales et les variations de taille verticales. Ces caractéristiques sont utilisées comme des observations pour calculer les probabilités de transition et d'émission du model HMM. Ainsi la probabilité globale de la séquence d'observations est calculée pour mesurer la similarité entre la signature observée et celle authentique. Six signatures authentiques de 60 individus ont été utilisées pour entraîner le HMM. Des contrefaçons expertes de type amateur ont été utilisées pour tester le taux de fausse acceptation du système. Les résultats montrent un taux de vérification de 85 %.

Le modèle de Markov a été déployé avec le transformé discret de Radon dans l'article [10] pour identifier hors ligne les signatures de la base de données de Dolfing qui comparent 924 signatures. Le système a réalisé un taux d'erreur de 18% et de 4,5% sur les contrefaçons expertes et simples.

Justino et Yacoubi [11] ont extrait des caractéristiques graphométriques comme la densité de pixels et la pente axiale pour les utiliser dans la classification avec le HMM. Ils ont effectué une segmentation horizontale de 16 pixels avec 25 cellules pour calculer les densités de pixels et la pente axiale à l'aide de la grille. Ces vecteurs sont ensuite convertis en séquence en utilisant la quantification vectorielle. L'entrainement du modèle a été effectuée sur 60 signatures et l'évaluation a montré un taux d'erreur de moins que 0.5%.

L'analyse en composantes principales (ACP) a été déployée par Arunalatha et al [12] pour vérifier les signatures en abordant les variabilités intra-personnelles et interpersonnelles. Le système représente les images en extractant la densité de pixels et à distance du centre de gravité avant d'appliquer l'ACP pour analyser les caractéristiques des signatures. Les variances sont représentées par l'inertie des composantes principales, déterminée par leurs valeurs propres. Ainsi, si une composante principale a plus d'inertie, elle appartient au même groupe. Cette technique a réalisé une bonne performance mesurée par un taux d'erreur de 17,06% sur la base de données GPDS.

Selon [2] parmi les méthodes statistiques les plus simples et efficaces est la classification basée sur la distance, vue que cette approche ne nécessite pas l'entrainement de modèle. Plusieurs techniques sont utilisées dans la littérature comme la distance euclidienne, de bloc de ville, Chi-carré, de Manhattan et de Hamming.

Prakash et Guru [13] ont développé une méthode basée sur la distance euclidienne entre les centroïdes des signatures qui forment un vecteur caractéristique à valeurs intervalles. Ainsi, les signatures similaires sont regroupées dans un même cluster. Le compte d'acceptation est incrémenté par 1 lorsque la valeur de caractéristique tombe dans l'intervalle de référence. La décision est ainsi prise en comparant le compte d'acceptation a un seul.

L'article [14] propose une approche qui utilise la distance de Hamming pour classer les images après l'extraction des caractéristiques de contexte de forme à partir de contours prétraités. Le système a réalisé une précision de 84,15% sur la base de données de signature de DS-I et DS-II.

1.2.2 Correspondance de modèles

La correspondance de modèles est le processus de comparaison des figures qui est couramment utilisées dans la vérification des signatures en comparant les nouvelles signatures aux modelés de référence (ou tempêtes) déjà stockes dans la base de données [2]. La technique de Dynamic Time Warping (DTW) est la plus utilises dans cette approche.

L'étude [3] a présenté une méthode efficace pour vérifier les signatures qui consiste à extraire des données de position de stylo a partir des signature réelles. Une fois les données sont prétraitées et normalisées, la distance de déformation est calculée en utilisant DTW pour mesurer la déformation des nouvelles signatures par rapport a la signature de référence. Ainsi, un seuil est calculé pour chaque échantillon authentique pendant l'entraînement, et la décision d'accepter ou de rejeter est prise par rapport à la moyenne de ce score. Cette méthode a été testée sur une base de données qui comparent 10 utilisateurs qui ont fourni 10 signature authentique et 10 falsifiées chacun. Le système a réalisé une précision moyenne de 90,4%.

L'article [5] a présenté un system de vérification de signature hors ligne basée sur une version modifiée de DTW qui tiens en compte la stabilité des composantes de la signature. L'approche pose sur l'hypothèse que les composantes très stables de la signature sont les plus difficile à falsifier qu'un composant variable. Ainsi la stabilité est utilisée pour pondérer les différentes caractéristiques statistiques utilisée dans le calcul de distance afin de la rendre plus robuste a la contrefaçon des caractéristique clés de la signature. Le système a été évalué sur une base de données de signatures de 100 individus. Les résultats indiquent que l'algorithme DTW modifié, intégrant des mesures de stabilité, présente un taux d'erreur égal de seulement 2 %, comparé à 29 % pour l'algorithme DTW de base.

Parziale et al. [4] ont mené une étude similaire sur les bases de données MCYT-100 et BiosecureID-SONOF en utilisant le DTW pour calculer les distances entre des vecteur de caractéristiques statistique de signature comme la position de stylo et la pression. Les taux d'erreur moyen enregistrées sur la falsification aléatoire et experte sont respectivement 1,23% et 4,55%.

D’autres chercheur comme [6], [7] et [8] ont vérifié les signatures sur des bases de données Open-Source comme CEDAR en utilisant le DTW avec extraction des caractéristiques statistiques en réalisant d’excellente précision de classification.

1.2.3 Méthodes d’apprentissage automatique classique

Forets aléatoires:

Plusieurs études ont été menées dans le cadre de la vérification des signatures hors ligne. La méthode foret aléatoire (Random Forest en anglais) a été parmi les techniques qui a pu gagner dans une comparaison avec d’autres méthodes de l’apprentissage machine tels que le classificateur naïf Bayésien (MNBC) et le classificateur de régression logistique (LRC). Sa précision a été la meilleur pour la même base de test et qui vaut 0,67, comparé à 0,4 et 0,53 pour les autres modèles [6]. Les expérimentations ont prouvé que le foret aléatoire est capable de supporter une grande variation de signatures. Plus que le nombre est élevés pour les échantillons lors de l’apprentissage, plus le modèle est plus précis. Néanmoins, les auteurs ont souligné la difficulté de construire un tel système a cause de la variation des signatures a le fil du temps, et aussi sa forte corrélation avec ka condition physique et émotionnelle de la personne qui signe.

Une autre étude qui tourne autour de la forêt aléatoire a été guidé par Shah et Al. Qui met en valeur la performance de cette technique comparée a la méthode de perceptron multicouche [47]. La base de données utilisée est constituée de 2250 signatures y compris 15 signatures authentiques et 15 signatures forgées, collectées de 75 individus. La comparaison consiste à appliquer le perceptron multicouche sur les images prétraitées, d’un côté, puis de répéter la même expérimentation mais en utilisant plutôt la classification de la forêt aléatoire. Le résultat du modèle montre une efficacité remarquable de la forêt aléatoire avec une précision de 0,77 contrairement a une performance de 0,57 de précision pour la deuxième méthode.

SVM

Les machines à vecteurs de support (Support Vector Machine en anglais), est un classificateur discriminant statistique, basé principalement sur la classification sur un hyperplan qui se manifeste par une ligne. Ce dernier permet de diviser l'espace bidimensionnel en deux. Dans ce cas, on parle d'une classification binaire ou chaque échantillon est étiquetté d'un côté ou de l'autre de l'hyperplan. Il s'agit d'une technique reconnue pour sa performance de précision dans l'apprentissage supervise. Parmi les travaux reconnus dans ce cadre, est celui de Shekal et al ont extrait les caractéristiques en se basant sur la méthode de Taylor Series Expansion (TSE). Ils ont eu recours à la base de données CEDAR [22] et ont établi le classificateur SVM pour la reconnaissance de signature[48]. La précision qu'ils ont eu vaut 95.25% ce qui est une valeur remarquable dans ce domaine de reconnaissance.

Une autre étude qui met en relief la technique SVM est celle de Kiani et al. Qui ont étudié la signature hors ligne en se référant sur la transformation de Radon, aussi nommée comme projection de Radon, en collaboration avec les machines a vecteurs de support. Dans la modélisation de la solution, ils ont effectué la binarisation et la suppression des marges dans le pré-traitement des images, à l'aide de l'algorithme Otsu. Par la suite, ils ont inverse les images à l'aide de la transformation de Radon. Ils ont segmenté à l'aide de cette technique, les images en fenêtres locales à travers en calculant la somme de différentes projections de l'intensité de l'image par rapport à l'angle et la ligne [41]. Ils ont par la suite confirmé l'existence du segment de ligne à travers la comparaison de la valeur maximale atteinte et le seuil. En outres, ils ont extrait les vecteurs de caractéristiques à travers un vecteur ayant une largeur de ligne de 6 pixels. Ces vecteurs ont été normalise par la division de chaque composante par la valeur maximale pour avoir des valeurs entre 0 et 1. Finalement, ils ont recours au classificateur SVM des images d'entrée. Concernant expérimentations, Kiani et al Ont utilisé une base de données de signatures persanes qui comporte 30 signatures pour 20 individus. Cette méthode a rapporté une précision de 0,96 et un taux de fausse rejection de 0,04.

La régression PLS:

Pushpalatha et al. Ont établi la vérification des signatures hors ligne avec une description de caractéristiques polaire. La première étape consiste générer le premier paramètre appelé score de régression à partir de la régression PLS. Dun autre cote, ils ont calculé le Logarithme de vraisemblance des signatures à l'aide du modèle de Markov cache [40]. La comparaison entre les deux paramètres va décider la classification de la signature. Si l'écart entre eux est inferieur a 0.05 alors la classification est confirmée correcte. Pour valider ces résultats de classifications, les auteurs ont recours à la technique SVM pour la validation de l'approche de régression PLS sur les signatures hors ligne. Une précision de 0.98 a été prouvée.

KNN

Pal et al. Ont étudié les performances du modèle des k plus proches voisions, aussi nommée KNN dans la vérification des signatures hors ligne. Cette méthode calcule la similarité pour la vérification des signatures. La base de données utilisée est nommée BHSig260 qui comportent des signatures pour 100 personnes de Bengali et des signatures de 160 personnes indien. Cette approche a réalisé un taux d'erreur égal de 0.24 et 0.33 respectivement pour la section hindi et de bengali des signatures [39].

1.2.4 Approches d'apprentissage profond

L'apprentissage en profondeur est une sous-catégorie de l'apprentissage artificiel qui se concentre sur l'utilisation de réseaux neuronaux artificiels profonds pour résoudre des problèmes complexes de perception comme la reconnaissance des formes et le traitement du langage naturel. Le terme profond ou « Deep » en anglais fait référence à la présence de nombreuses couches de neurones dans les réseaux neuronaux, ce qui permet de capturer des caractéristiques complexes et arborescentes à partir des données d'entrée [15].

Cette approche est largement utilisée dans le système de vérification de signature hors ligne grâce à sa puissance, la facilite d'entrainement et la capacite à apprendre et reconnaitre les formes [16]. Dans cette section nous citons quelques travaux réaliser dans cette catégorie de modèles.

L'étude de [17] présente un system basé sur le réseau de neurone convolutif (CNN) hiérarchique de classe unique qui apprend les formes des signatures authentiques et

permet de découvrir les contrefaçons. Cette méthode nécessite seulement des signatures réelles pour l'entrainement ce qui représente un avantage significative vue que les contrefaçons ne sont pas disponibles pour chaque utilisateur inscrit dans un scénario d'application réel. Ce système a été évalué sur quatre bases de donnes, notamment PHBC, UTSig, MCYT-75 et CEDAR et on enregistre d'excellente performance en termes de précision de classification.

Une autre forme de CNN appelée InceptionSVGNet a été utilisée dans le travail de [18] pour vérifier les signatures en s'inspirant de l'architecture InceptionV1 connu sous le nom de GoogleNet. Pour préparer les données, les image sont transformées en niveau gris et filtrées par d'un flou gaussien afin de réduire le bruit. Elles sont ensuite binarisées par la méthode Otsu. Une fois les image sont normalisées et redimensionnées, elles sont alimentées au CNN qui serait plus large plutôt que profond pour l'entrainement. Les résultats montrent que le modèle InceptionSVGNet a généralement surpassé les autres méthodes dans la plupart des ensembles de données, avec une amélioration significative de la précision dans les ensembles de données Bengali et Hindi avec une précision de 97.77% et de 95.4%. Cette approche a eu des performances comparables à d'autres méthodes classiques de la littérature sur l'ensemble de données CEDAR et UTSig en termes de précision, de taux de faut rejet et de fausse acceptation.

Wei et al [20] ont présenté le modelé de réseaux discriminants inverse (IDN) qui comprend quatre flux de réseau incluant deux paires d'échantillons de signature. La première paire inclus les signatures de référence et de test centrée sur les caractéristique convolution elles des images. L'autre paire comprend l'échantillon de référence en niveau gris et celui de test qui sert à extraire les traits de signature. Ce système a été évalué sur une grande base de données chinois de 29000, CEDAR, BHSig-B et BHSig-H et a réalisé des précisions de 90,17 %, 95,98 %, 95,32 % et 93,04 %, respectivement.

Le potentiel d'apprentissage par transfert a été évalué dans le travail de [21] pour vérifier les signatures hors ligne. Les auteurs ont passé en revue des méthodes classiques basées sur le CNN afin de comparer leurs performances en termes de précision de classification. Ainsi, six modèles pré-entraînés ont été déployés et évalues sur deux ensembles de données de signatures de référence : GPDS Synthétique et

MCYT75 pour les signatures latines, ainsi que deux ensembles de données persanes : UTSig et FUM-PHSD.Les modèles les plus utilisés dans les taches génériques de vision par ordinateur VGG16, VGG19, ResNet50 et InceptionV3 ainsi que les modèles pré-entraînés SigNet et SigNetF qui sont spécifiques aux tâches de traitement des signatures sont également évalués. Les résultats expérimentaux obtenus confirment l'efficacité des modèles VGG16 et SigNet avec une supériorité du VGG16 pour la tache de reconnaissance des signatures avec une précision moyenne de 99.62% sur les quatre bases de données utilisées.

La technique développée dans [22] propose la modélisation de la tache de vérification hors ligne avec un réseau Siamois convolutionnel utilisant des réseaux jumeaux entrainées pour apprendre un espace de caractéristique. Cela est réalisé en exposant le réseau à une paire d'observations similaires et dissemblables et en minimisant la distance euclidienne entre les paires similaires tout en la maximisant entre les paires dissemblables. Le réseau proposé a surpassé les techniques de pointes sur la plupart des ensembles de données utilisées notamment le GPDS Synthetic, GPDS300, Hindi, Bengali et CEDAR en réalisant des précisions de 77,76, 76,83, 84,64, 86,81 et 100% respectivement. Les expériences sur des ensembles de données inter-domaines soulignent la capacité de l'architecture à modéliser les fraudes de différents styles d'écriture de signataires et de faussaires avec des scripts et des arrière-plans divers.

L'article [23] a proposé un système utilisant une architecture de réseau neuronal siamois entièrement connecté pour une détection efficace des falsifications de signatures. Trois réseaux différents de CNN ont été évaluées avec une architecture Siamoise: CNN1 comparent 4 couches dans le même ordre, augmentant les canaux à chaque couche de convolution avec utilisation de la normalisation locale des réponses et du max pooling pour réduire la dimension. Le 2eme réseau utilise la normalisation par lots pour une meilleure optimisation avec un taille de noyau fixée à 2 pour toutes les couches de max pooling. La 3eme architecture est la même que la 2eme mais avec une augmentation du nombre de canaux. Les résultats montrent que la 3eme architecture a eu la meilleure performance sur une ensemble de 2149 signature en termes de precision, recall et le score F1 en réalisant respectivement 0,73, 0,92 et 0,81.

Dans cette étude [42], les auteurs ont démontré l'efficacité d'une approche basée sur le réseau siamois et l'apprentissage en une seule itération pour la vérification de signature

hors ligne. Les résultats obtenus ont montré une capacité précise à distinguer entre les signatures réelles et fausses sur plusieurs ensembles de données. Sur l'ensemble de données 4NSigComp2010, l'exactitude atteint 89,99%, avec un taux de fausse acceptation (FAR) de 10,22% et un taux de faux rejet (FRR) de 8,33%. De manière similaire, sur l'ensemble de données SigComp2011, l'exactitude est de 90,11%, avec un FAR de 7,89% et un FRR de 6,42%. Ces performances se confirment également sur d'autres ensembles de données, tels que 4NSigComp2012, BHsig260-Bengali et BHsig260-Hindi, où les taux d'exactitude varient entre 91,17% et 93,23%. Ces résultats témoignent de l'efficacité du modèle de vérification de signature proposé, mettant en évidence sa capacité à distinguer efficacement entre les vraies signatures et les signatures falsifiées.

1.3 Comparaison des approches de modélisation

Après l'étude de ces travaux, on trouve qu'ils reposent sur deux concepts, la vérification de l'authenticité et la similarité de signature, aussi, il est clair que les approches basées sur les méthodes correspondance de modèles, les approches statistiques et méthodes d'apprentissage automatique classique reposent principalement sur la comparaison des signatures. Les caractéristiques visuelles de la signature soumise sont comparées à celles d'une signature de référence pour déterminer si elles correspondent. Ces techniques traditionnelles, telles que la comparaison de formes, de taille et de structure des caractères, sont souvent utilisées. Cependant, si le système ne dispose pas d'une signature similaire d'une personne, la prédiction sera très probablement erronée.

Ces techniques utilisent généralement des algorithmes spécifiques au problème, tels que la régression linéaire, les arbres de décision et les machines à vecteurs de support (SVM). Les performances de ces méthodes sont souvent limitées par la qualité des caractéristiques extraites et la complexité des données. Ces techniques sont généralement moins efficaces que celles basées sur l'apprentissage profond qui se concentre sur l'utilisation de réseaux neuronaux et sur le comportement de la signature.

L'apprentissage profond a transformé la façon dont on aborde les problèmes d'apprentissage automatique en permettant aux modèles d'apprendre des caractéristiques à partir de données brutes, ce qui améliore considérablement les

performances sur les données complexes. Il a démontré de meilleurs résultats sur ces données, telles que des images et du texte, dans notre cas. Cependant, les modèles d'apprentissage profond nécessitent généralement de grandes quantités de données pour s'entraîner efficacement et d'importantes ressources informatiques, des GPU par exemple, pour l'entraînement.

En effet, le grand problème de la classification des signatures est le manque et la difficulté de collecter les images. On trouve les techniques de transfert d'apprentissage, qui consistent à utiliser des modèles pré-entraînés comme point de départ, réduisant ainsi la nécessité de calculs intensifs et de grandes quantités de données pour l'entraînement.

1.4 Etapes de reconnaissance des signatures hors ligne

Le système de reconnaissance de signatures vise à détecter les signatures contrefaites ou falsifiées en comparant attentivement les caractéristiques des signatures, Ce processus commence par l'acquisition des données, soit en collectant de nouvelles signatures soit en utilisant un ensemble de données public. Ensuite, les images de signatures subissent un prétraitement pour améliorer leur qualité et les rendre aptes à l'analyse. L'étape suivante consiste en l'extraction des caractéristiques pertinentes des signatures, telles que les traits, les courbes et les angles, qui serviront de base à la comparaison. Enfin, la classification des signatures est effectuée pour déterminer si une signature est authentique. Ce qui suit, la démarche à suivre pour développer ces systèmes.

1.4.1 Acquisition des données

Les bases de données publiques proposent une variété d'échantillons de signatures manuscrites. Ces ressources incluent des plateformes telles que Kaggle, le UCI Machine Learning Repository (l'une des plus anciennes), l'outil de recherche de jeux de données de Microsoft (msropendata), ainsi que GitHub. De plus, diverses organisations académiques et de recherche mettent à disposition des jeux de données pour soutenir les études et la recherche.

Les institutions financières et les banques recueillent fréquemment des échantillons de signatures de leurs clients afin de vérifier les transactions. De même, les organisations

professionnelles collectent souvent des échantillons de signatures de leurs membres pour des raisons administratives diverses, sauf que, Il est essentiel d'obtenir le consentement éclairé des personnes avant de recueillir leur signature. Cela inclut également la protection des données personnelles associées aux signatures, afin d'éviter toute utilisation non autorisée ou violation de la vie privée, aussi, les échantillons de signatures doivent être stockés de manière sécurisée pour prévenir tout accès non autorisé ou vol de données.

1.4.2 Prétraitement d'image

Le prétraitement des images englobe toutes les opérations effectuées sur les images avant qu'elles ne soient analysées ou traitées à l'aide d'algorithmes. Dans la collecte et l'authentification des signatures manuscrites, le prétraitement des images est essentiel pour améliorer la qualité, la cohérence et la fiabilité des données.

Il existe de nombreuses méthodes de prétraitement d'images connues qui sont couramment utilisées dans divers domaines, Voici quelques-unes de ces techniques :

Binarisation : La binarisation transforme une image en niveaux de gris en une image binaire en fonction d'un seuil. Elle permet de simplifier le traitement en convertissant les pixels en noir ou en blanc [43].

Filtrage médian : Le filtrage médian est efficace pour éliminer le bruit d'impulsion (ou bruit de type "sel et poivre") en remplaçant la valeur d'un pixel par la valeur médiane des pixels environnants.

Correction gamma : La correction gamma ajuste la luminosité des pixels pour mieux représenter les niveaux de gris d'une image, ce qui peut être utile pour améliorer la lisibilité [44].

1.4.3 Entraînement du modèle :

Lors de la construction d'un modèle d'authentification de signature manuscrite, il faut disposer d'un ensemble de données suffisamment riche et varié pour que le modèle puisse apprendre et se généraliser correctement. Cependant, il est également nécessaire de vérifier que le modèle ne se limite pas à l'apprentissage des données par cœur (sur-apprentissage) et qu'il peut également obtenir des résultats sur de nouvelles données qu'il n'a jamais traitées. Le principe fonctionne de la manière suivante :

Division des données : La division des données se fait généralement en trois parties principales : l'ensemble d'entraînement, l'ensemble de validation et l'ensemble de test. Une règle empirique courante est de répartir les données avec 70-80 % pour l'entraînement, 10-20 % pour la validation, et le reste pour l'ensemble de test.

Entraînement du modèle : Pendant la phase d'entraînement, celui-ci ajuste ses paramètres et ses filtres pour minimiser la perte en comparant ses prédictions avec les étiquettes des données d'apprentissage fournies.

Validation du modèle : L'ensemble de validation est utilisé pour évaluer les performances du modèle à chaque itération de leur entraînement.

Le système évalue le degré de confiance de la signature testée en fonction de la similitude de ses caractéristiques avec celles des signatures référence. Un score élevé indique une forte probabilité d'authenticité, tandis qu'un score faible peut suggérer la possibilité d'une falsification.

Le système peut être régulièrement mis à jour en intégrant de nouvelles données d'entraînement afin d'améliorer sa précision et sa capacité à détecter des méthodes de contrefaçon plus complexes.

1.5 Explication des notions techniques

Dans cette section, tous les concepts utilisés dans la méthodologie ont été passés en revue dans la littérature. Nous détaillerons tout d'abord le réseau neuronal convolutif (CNN), avec ses différentes couches, le modèle siamois et la manière dont il peut être obtenu à partir de ce qui précède. Enfin, le transfert d'apprentissage VGG16 et les différentes approches et étapes nécessaires pour le mettre en œuvre seront détaillés.

1.5.1 Réseaux neuronaux convolutifs (CNN)

La computer vision est une filière de l'informatique qui permet aux ordinateurs de reconnaître le monde visuel. Plusieurs algorithmes basés sur l'apprentissage automatique et l'apprentissage profond qui aident à développer des modèles pour faire des prédictions sur les images

Dans le contexte de l'apprentissage profond, les réseaux neuronaux convolutifs (CNN) peuvent être considérés comme l'une de ces sous-catégories, offrant une solution technique pour les tâches de traitement d'images dans les domaines de la classification et de la vérification. Ce qui suit, on explique les différentes étapes de la création d'un réseau de neurones convolutifs

-Canaux d'image

Pour adapter une image à l'algorithme CNN (Convolutional Neural Network), la première étape consiste à la représenter numériquement. Pour une image en niveaux de gris (noir et blanc), elle est représentée par une matrice 2D de dimensions m par n, où chaque élément de la matrice contient la valeur du pixel correspondant, généralement comprise entre 0 (noir) et 255 (blanc).

En revanche, pour une image couleur de même taille, on utilise une matrice 3D. Chaque pixel est alors caractérisé par ses valeurs dans trois canaux distincts : rouge, vert et bleu (RGB). Chaque canal a une valeur comprise entre 0 et 255, déterminant l'intensité de la couleur respective à ce canal pour ce pixel spécifique. Ainsi, une image couleur est représentée par une matrice de dimensions m par n par 3, où chaque canal contient les valeurs des pixels pour la couleur correspondante.

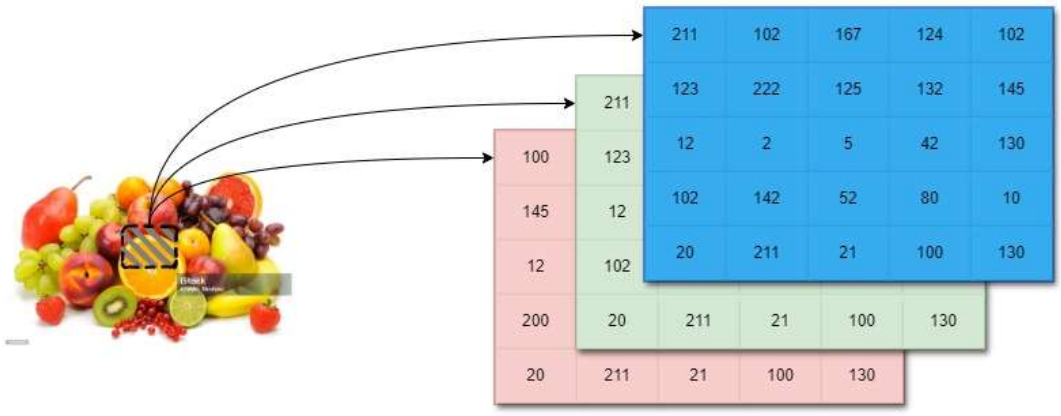

Figure 1.1 Canaux d'image

La Figure 2.1 illustre la manière dont les couleurs sont transformées en représentations numériques sous la forme d'un tableau tridimensionnel.

- Couche de convolution

Une fois que l'image a été exprimée sous forme de valeurs numériques, la prochaine étape est d'identifier les aspects de l'image. Cette tâche est accomplie à l'aide d'une technique connue sous le nom de convolution. Un processus où une fonction altère la forme d'une autre. Dans le contexte des images, on utilise un filtre ou Kernal. Un tableau qui représente la caractéristique à extraire. Ce filtre est déplacé sur le tableau d'entrée, et la convolution résultante est un tableau bidimensionnel contenant la corrélation de l'image par rapport au filtre appliqué. La figure 2.2 montre le tableau de la carte des caractéristiques [46].

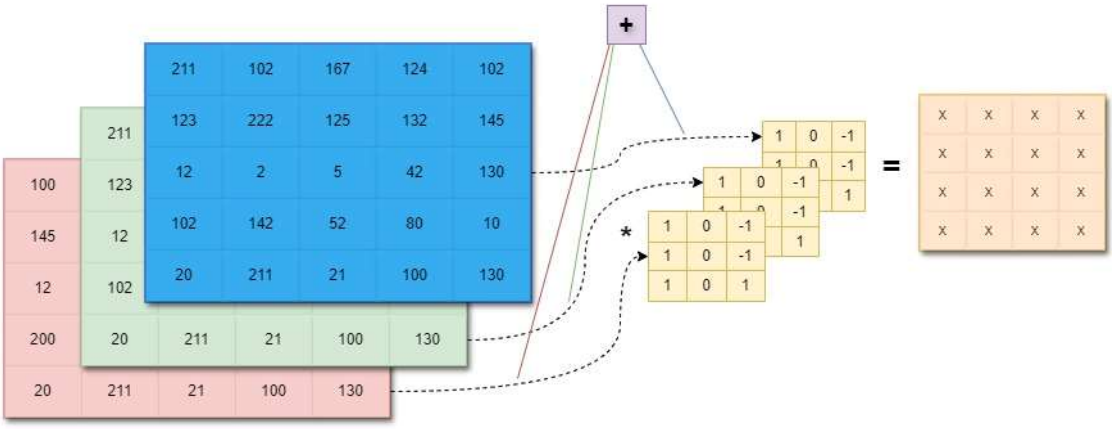

Figure 1.2 Couche de convolution

-Couche de sous-échantillonnage [45]

Après la convolution, pour réduire encore la taille de la carte des caractéristiques, une étape de mise en commun est utilisée avant la poursuite du traitement. Cette étape permet de comprimer davantage les dimensions de la carte des caractéristiques. C'est pourquoi le regroupement est également appelé pooling ou sous-échantillonnage.

Il consiste à résumer les caractéristiques d'un groupe de cellules. Ce résumé des cellules peut être obtenu en prenant le maximum, le minimum ou la moyenne dans un groupe de cellules. Chacune de ces méthodes est appelée respectivement pooling maximum, pooling minimum ou pooling moyen.

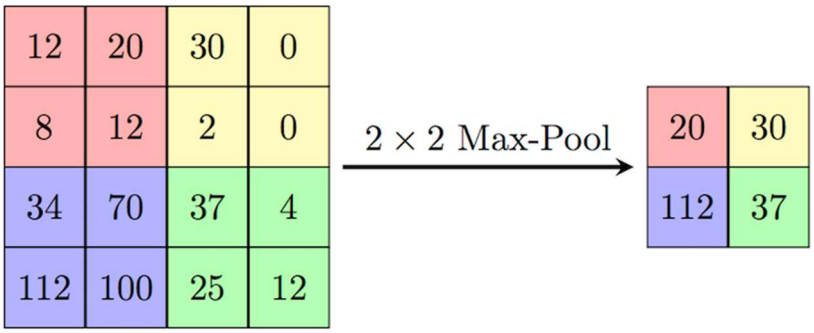

Figure 1.3 Opération de sous-échantillonnage

La Figure 2.3 illustre l'application du pooling max à un filtre de taille 2x2. Cela implique que pour chaque groupe de cellules 2x2 dans la carte des caractéristiques, la valeur maximale de cette région est extraite dans la cellule de sortie.

- Aplatissement

Après avoir suivi une séquence d'opérations visant à capturer les caractéristiques de l'image, notamment le sous-échantillonnage et la convolution, la dernière étape avant d'appliquer un réseau de neurone convolutionnel consiste à assurer sa compatibilité avec le format d'entrée de celui-ci. Cela est réalisé grâce au "flattening", qui transforme le tableau multidimensionnel résultant des opérations précédentes du CNN en un vecteur unidimensionnel. La figure 2.4 montre un aplatissement appliqué à une seule carte de caractéristiques.

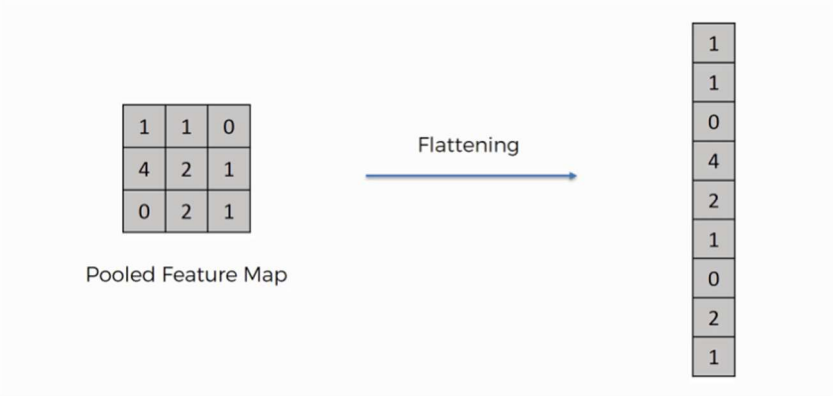

Figure 1.4 Opération d'aplatissement

La figure 2.5 montre toutes les étapes qui résument un réseau neuronal conventionnel de manière plus détaillée, en commençant par l'image d'entrée, en passant par les couches choisies pour le traitement, jusqu'à la sortie montrant la prédiction de la classification.

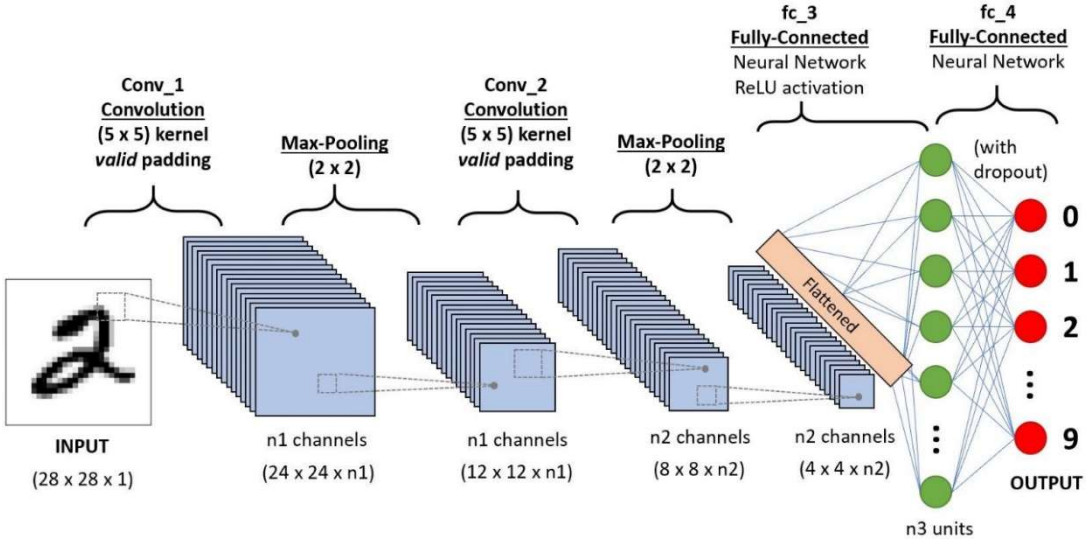

Figure 1.5 Architecture de réseau neuronal conventionnel

1.5.2 Réseaux siamois

Les réseaux siamois tirent leur nom des chats siamois, dont l'apparence est souvent très similaire. Tout comme ces chats ont des traits physiques presque identiques, dans

un réseau siamois, deux réseaux neuronaux semblables sont utilisés en parallèle. Ces réseaux sont souvent appelés "frères" car ils partagent la même architecture et les mêmes poids [34].

Essentiellement, ces réseaux se composent de deux sous-réseaux (généralement des réseaux neuronaux convolutifs, CNN) partageant la même structure, des paramètres et des poids similaires. Comme leur nom l'indique, la principale caractéristique de ces réseaux est leur capacité à comparer deux entrées. Cela signifie que les données sont introduites dans les deux branches du réseau et que chacune produit une représentation des données de sortie. Ensuite, une mesure de similarité est calculée entre ces représentations pour évaluer la similarité des deux entrées.

Ces réseaux sont largement utilisés dans les domaines de la vérification, en particulier ceux où les ensembles de données sont déséquilibrés et peu nombreux.

-Fonction de perte :

Également appelées fonctions d'erreur ou de coût, mesurent l'écart entre les valeurs prédites par un modèle et les valeurs réelles attendues. Elles jouent un rôle crucial dans l'entraînement des modèles d'apprentissage automatique car elles guident l'optimisation des paramètres du modèle pour minimiser cette erreur.

Plusieurs fonctions de perte peuvent être utilisées pour entraîner le modèle en fonction de la tâche spécifique et des caractéristiques des données. Voici quelques-unes des fonctions de perte les plus couramment utilisées :

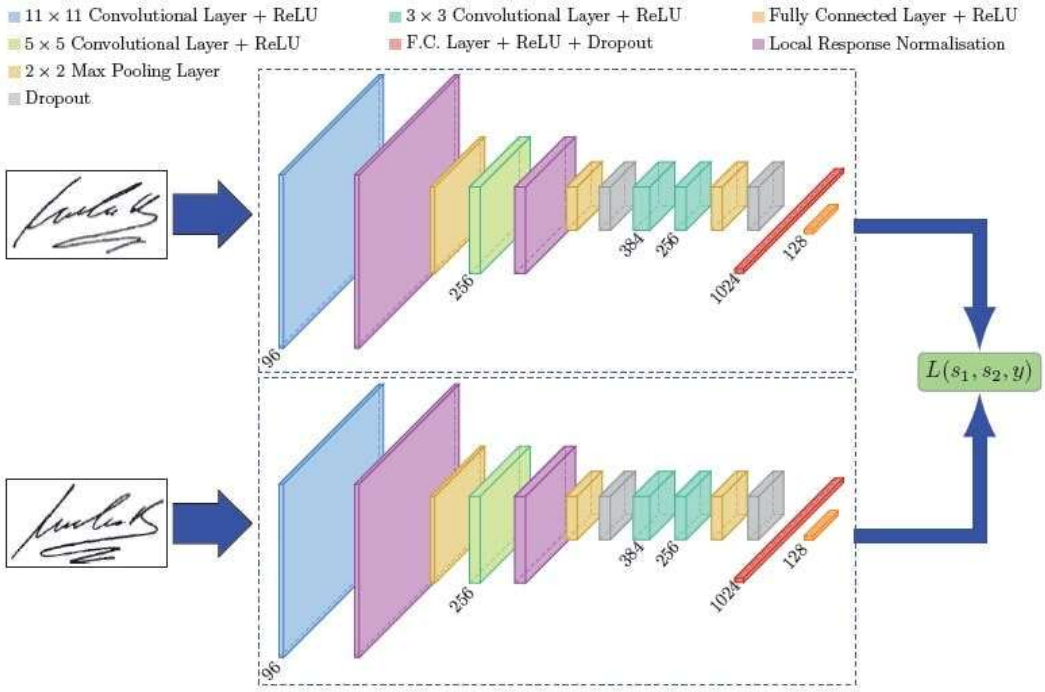

Figure 1.6 Architecture de SigNet [22]

Dans la figure 2.6 Deya, S., Dutta, A. [22] utilisent deux sous-réseaux CNN qui extraient des caractéristiques (s1 et s2) et sont liés par une fonction de **perte de contraste** L « Contrastive Loss », qui compare la similarité entre les caractéristiques s_1 et s_2. y Est une autre variable qui détermine si les deux signatures données correspondent ou non au même signataire.

$$\boldsymbol{L(s_1, s_2, y) = \alpha(1-y)D_w^2 + \beta y \max(0, m - D_w)^2} \quad \textbf{(1.1)}$$

Dans l'équation (2.1) [22], β et α sont des constantes de pondération, m est une marge fixée et $D_w$est la distance euclidienne entre les deux échantillons dans l'espace de caractéristiques appris par le réseau.

$(1-y)D_w^2$: Cette partie de la perte s'applique lorsque les deux échantillons appartiennent à des classes différentes (y=0). Elle pénalise les paires dissemblables en fonction de leur distance euclidienne D_w. L'objectif est de maximiser cette distance pour les paires dissemblables.[36]

$y\max(0, m - D_w)^2$: Ceci est le cas lorsque les deux exemples correspondent à la même classe. (y = 1). Elle considère que les paires similaires sont affectées par la différence entre la marge m et la distance D_w. L'objectif est de minimiser cette distance pour les paires similaires, tout en s'assurant qu'elle reste en dessous de la marge m.[36]

1.5.3 Modèle VGG16

VGG16 est une architecture de réseau neuronal convolutionnel profond développée par Karen Simonyan et Andrew Zisserman en 2014 dans un article publié sous le titre "Very Deep Convolutional Networks for Large-Scale Image Recognition" [33] (Réseaux convolutionnels très profonds pour la reconnaissance d'images à grande échelle). Il a été élaboré pour la classification d'images et a obtenu des résultats exceptionnels dans divers congrès de reconnaissance d'images. La figure 2.7 présente l'architecture du modèle, suivie d'une explication de ses caractéristiques et le processus de Transfert d'Apprentissage.

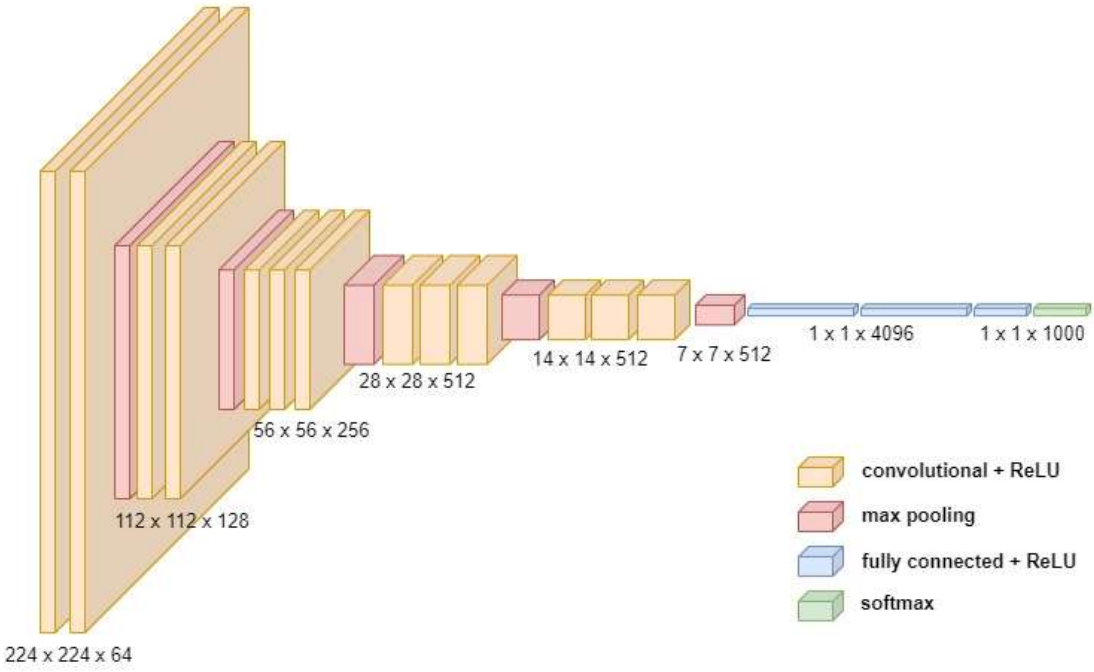

Figure 1.7 Architecture de VGG16

-Caractéristiques Clés de VGG16

Performance : entraîné sur l'ensemble de données ImageNet, qui contient plus de 14 millions d'images et 1 000 classes.

Taille des Images :il prend des images d'entrée de taille fixe de 224x224 pixels. Cette taille standardisée permet au modèle de traiter les images de manière uniforme et de simplifier les étapes de prétraitement.

Nombre de couches : composé de 16 couches équilibrées, ce qui explique son nom. Celles-ci comprennent 13 couches convolutives et 3 couches entièrement connectées.

Structure des Couches Les couches convolutives utilisent des filtres 3x3 avec un pas de 1 pixel, et les couches de pooling utilisent des filtres 2x2 avec un pas de 2 pixels. Cette configuration permet de capturer des caractéristiques de plus en plus complexes à mesure que l'on progresse dans le réseau.

Uniformité Le VGG16 se distingue par son architecture uniforme. Toutes les couches convolutives utilisent des filtres 3x3 et les couches de regroupement des données (pooling) utilisent des filtres 2x2, ce qui facilite la conception du modèle et le rend plus cohérent.

-Processus de Transfert d'Apprentissage :

L'apprentissage par transfert commence par le chargement du modèle pré-entraîné, qui a déjà appris à extraire des caractéristiques visuelles générales, telles que les formes, les textures et les objets partiels. Ensuite, afin de conserver ces caractéristiques et les connaissances acquises, les couches convolutives du modèle sont gelées. Cela signifie que leurs poids ne seront pas modifiés pendant l'entraînement sur le nouvel ensemble de données, pour obtenir de bonnes performances avec une quantité relativement faible des images.

Les couches supérieures du modèle sont ensuite ajustées ou réentraînées sur le nouvel ensemble de données spécifique, permettant au modèle de s'adapter aux caractéristiques plus subtiles de la nouvelle tâche. Ce processus, connu sous le nom d'ajustement fin (fine-tuning), est essentiel pour améliorer les performances du modèle

sur la nouvelle tâche, tout en réduisant le besoin en données d'entraînement et le temps d'apprentissage.

1.6 Métriques

Les métriques d'évaluation fournissent des mesures quantitatives qui permettent de juger de la précision et de l'efficacité des modèles par rapport aux données qu'ils manipulent. Parmi les métriques les plus couramment utilisées, on trouve l'exactitude (accuracy), la précision (precision), le rappel (recall), le score F1 (F1 score),

Où :

VP (Vrai Positif) : Nombre d'éléments positifs correctement classifiés comme positifs.

VN (Vrai Négatif) : Nombre d'éléments négatifs correctement classifiés comme négatifs.

m : Nombre total d'éléments dans le jeu de données.

FN (Faux Négatif) : Nombre d'éléments positifs incorrectement classifiés comme négatifs.

FP (Faux Positif) : Nombre d'éléments négatifs incorrectement classifiés comme positifs.

Définition 2.2. L'exactitude (accuracy) est la proportion d'éléments correctement classifiés.

$$\textbf{Exactitude} = \frac{\textbf{VP} + \textbf{VN}}{\boldsymbol{m}} \qquad 1.2$$

Définition 2.3. La précision (precision) indique la proportion d'exemples prédits positifs qui sont réellement positifs.

$$\textbf{Précision} = \frac{\boldsymbol{VP}}{\boldsymbol{VP} + \boldsymbol{FP}} \qquad 1.3$$

-Définition 2.4. Le rappel (recall) est la mesure dans laquelle les éléments positifs sont correctement classés.

$$\textbf{Rappel} = \frac{\textbf{VP}}{\textbf{VP} + \textbf{FN}} \qquad 1.4$$

- Définition 2.5 le Score F1 (F1 Score) : Représente une moyenne harmonique entre la précision et le rappel, offrant une mesure combinée de la performance du modèle.

$$\textbf{F1-score} = \textbf{2} \cdot \frac{\textbf{Precision} \cdot \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}} \qquad 1.5$$

Chapitre 2 Méthodologie

La méthodologie proposée utilise deux modèles distincts pour la vérification des signatures. Tout d'abord, on utilise un modèle siamois qui vise à rendre les signatures d'un même individu aussi cohérentes que possible, en identifiant les caractéristiques uniques qui les définissent. En revanche, il cherche à identifier les différences notables entre les signatures de différents individus ou celles falsifiées d'un même signataire, afin de pouvoir les distinguer de manière fiable.

Ensuite, le modèle VGG16 est utilisé pour vérifier la signature de manière indépendante. Il analyse les caractéristiques visuelles de la signature et extrait les informations nécessaires pour en vérifier l'authenticité.

En combinant ces deux modèles, on obtient une vérification robuste des signatures, qui tient compte à la fois de la similarité entre les signatures et de l'authenticité des caractéristiques visuelles. Cette combinaison permet d'obtenir des performances élevées en matière de vérification.

Cette méthodologie se décompose en quatre étapes : la préparation des données, les mesures de similarité des signatures, la vérification de l'authenticité des signatures et, enfin, l'évaluation des modèles à l'aide des métriques.

La figure 3.1 présente l'architecture des deux modèles utilisés, en mettant l'accent sur la similarité et l'authenticité. Les champs "Similarité" et "Authenticité" sont respectivement utilisés pour évaluer la similarité entre les signatures et pour vérifier l'authenticité de la signature à vérifier.

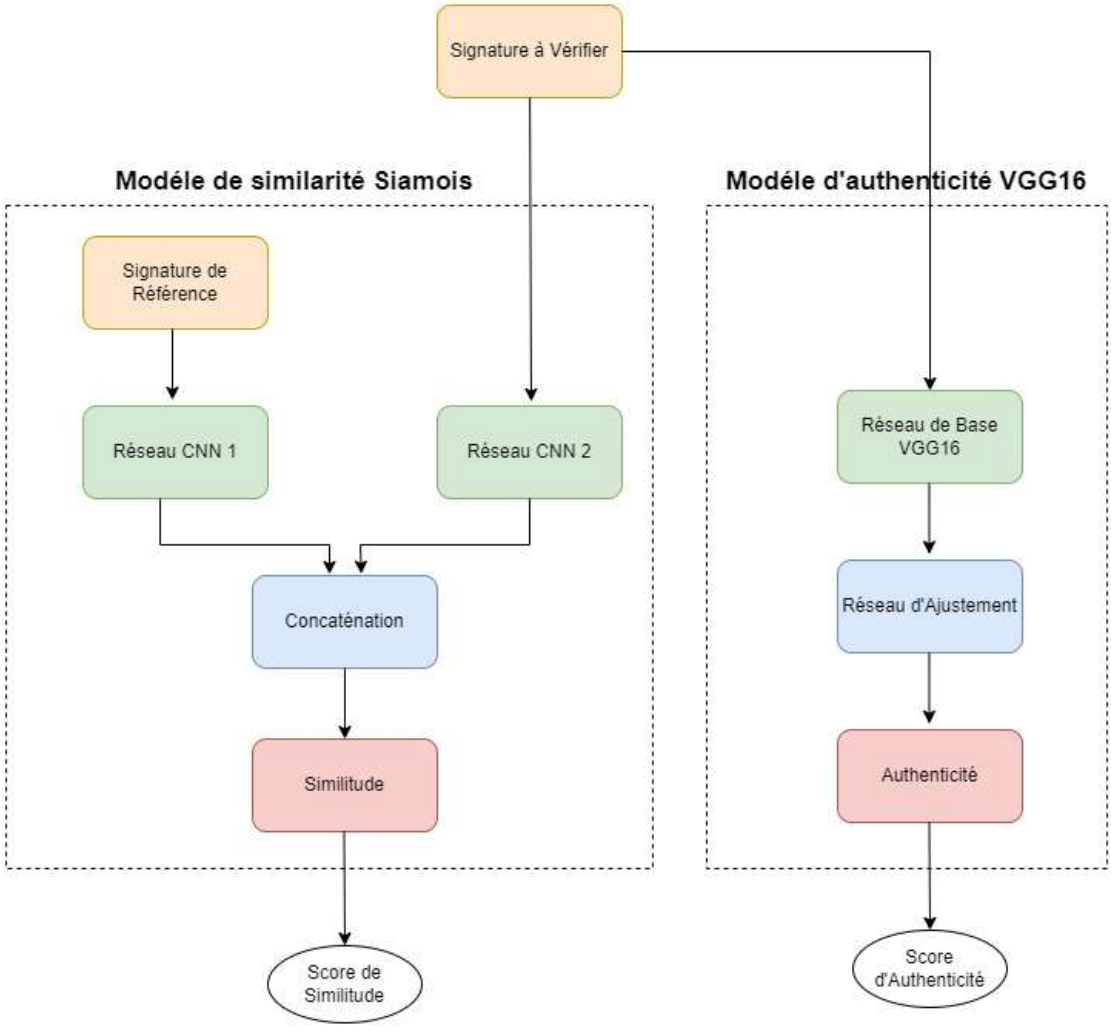

Figure 2.1 Architecture de Similarité et d'Authenticité pour la Vérification des Signatures

2.1 Etape 1 : Préparation des données

La première étape consiste à construire et préparer la base de données qu'on va utiliser pour entrainer et évaluer les modèles. Ainsi, on commence par collecter les signatures pertinentes pour la tâche d'apprentissage. Une fois les données recueillies, elles sont nettoyées et prétraitées pour les rendre exploitables par les modèles. Cela inclut notamment le redimensionnement, la normalisation et la conversion dans un format approprié. Ensuite, on divise l'ensemble des données en ensembles d'entraînement, de validation et de test, afin de pouvoir choisir une architecture de modèle adaptée à la tâche.

Pour préparer les données nécessaires à l'apprentissage et à l'évaluation des modèles, plusieurs étapes clés doivent être suivies :

Collecte des données : Cette étape consiste à rechercher et rassembler des signatures pour la tâche d'apprentissage. Cela peut inclure la collecte de signatures manuscrites provenant de diverses sources, telles que des bases de données en ligne, des archives ou d'autres documents officiels.

Nettoyage des données : une fois les données collectées, elles doivent être nettoyées pour éliminer le bruit ou les incohérences. Il s'agit notamment de filtrer les signatures illisibles, les données incomplètes ou corrompues afin de garantir la cohérence.

Prétraitement des données : les fichiers doivent être nettoyés et prétraités pour les rendre compatibles avec les modèles. Il s'agit de redimensionner les images pour qu'elles correspondent à la taille d'entrée, de les normaliser pour adapter les valeurs dimensionnelles appropriées.

Construction de l'ensemble de données : Une fois les images préparées, elles sont divisées en trois ensembles : L'ensemble d'entraînement sert à entraîner le modèle, tandis que la validation est utilisée pour définir les hyperparamètres et de contrôlé l'entrainement. Troisième partie est utilisée pour tester le comportement du modèle avec des données qu'il n'a jamais vues précédemment.

2.2 Etape 2 : Mesure de similarité des signatures du modèle siamois

La deuxième étape de la méthodologie consiste à entraîner le modèle siamois en utilisant des paires de signatures. Le réseau se compose de deux branches, chacune recevant une signature différente à comparer. Ces deux composantes sont identiques en termes d'architecture et de poids, partageant les mêmes paramètres. L'objectif est d'apprendre des représentations caractéristiques des signatures de manière que les signatures similaires soient rapprochées dans l'espace des caractéristiques, tandis que les signatures différentes sont éloignées.

2.2.1 Construction du réseau Siamois

Le réseau Siamois utilisé comprend deux branches parallèles. Chaque branche du réseau reçoit une signature en entrée et en extrait les caractéristiques. Celles-ci sont ensuite utilisées pour calculer la similarité entre les deux signatures. En d'autres termes, le même ensemble d'opérations est appliqué à chaque signature d'entrée.

Il est nécessaire de déterminer les types de couches à utiliser dans chaque branche du réseau. Les choix possibles comprennent les couches de convolution, qui sont utilisées pour extraire les caractéristiques visuelles importantes, les couches de mise en commun pour réduire la taille des cartes de caractéristiques et préserver les informations essentielles, et les couches entièrement connectées pour effectuer des classifications. Une fois les caractéristiques extraites par chaque branche, une couche de fusion, pour mesurer la similarité entre les deux signatures.

Le réseau Siamois est construit par plusieurs couches de convolution pour extraire des caractéristiques visuelles des signatures. Les premières couches emploient des filtres pour détecter des motifs simples, tandis que les couches ultérieures utilisent des filtres plus nombreux et plus petits pour capturer des détails plus complexes.

La normalisation par lots est appliquée après chaque convolution pour stabiliser l'apprentissage en ajustant les activations, tandis que l'activation Leaky ReLU introduit une petite pente pour les valeurs négatives, évitant ainsi l'inactivité des neurones.

Le Pooling réduit la dimensionnalité des caractéristiques tout en préservant les détails importants. Les sorties des deux branches du réseau sont ensuite concaténées et passées par des couches entièrement connectées pour produire un vecteur de haute résolution, toutes ces couches seront plus détaillées dans la section consacrée à l’implémentation. Enfin, une couche de sortie avec une fonction sigmoïde calcule le score de similarité entre 0 et 1.

2.2.2 Calcul de la similarité entre les paires d'images

Une fois le réseau est construit, le calcul de similarité est effectué en calculant la distance euclidienne entre les caractéristiques. Plus les représentations sont similaires, plus la distance entre elles est petite.

Supposant que $f1(x)$ et $f2(x)$ les fonctions représentant les deux branches du réseau chacune prenant une signature $x1$, $x2$ en entrée. Ces fonctions produisent les vecteurs de représentation E_1 et E_2 dans un espace caractéristique [27].

L'objectif de la fonction de perte est de minimiser la distance entre les représentations pour les paires de signatures similaires. Cela peut être formulé mathématiquement comme suit :

$$Perte\ de\ similarité = \| \mathrm{E}_1(x_1) - E_2\ (x_2)\| \quad (3.1)$$

2.2.3 Prédiction de la similarité

Dans cette méthodologie, pour l'entraînement du réseau pour la vérification de la similarité, les paires de signatures similaires (appartenant à la même personne) sont associées à une valeur cible de 1, tandis que les paires de signatures non similaires (des personnes différentes ou falsifiées de la même personne) sont également associées à une valeur cible de 0. L'objectif est d'apprendre au réseau à différencier efficacement les signatures similaires des signatures non similaires, y compris les signatures falsifiées.

La méthode traditionnelle de fonction de perte pour les réseaux siamois, souvent utilisée dans les tâches de vérification de la similarité, est la perte de contraste (contrastive loss). Elle est utilisée pour apprendre des représentations où les paires d'entrées similaires sont rapprochées et les paires non similaires sont éloignées. Dans cette étude, on va utiliser une autre fonction de perte appelée l'entropie croisée binaire (BCE).

La (BCE) [37] est utilisée pour déterminer directement si deux entrées sont similaires ou non. Elle mesure la probabilité de similarité entre les paires indépendamment de la distance dans l'espace des caractéristiques. Cette fonction est calculée comme suit :

$$\mathbf{BCE}\ (\boldsymbol{y}, \hat{\boldsymbol{y}}) = -(\boldsymbol{y}\mathbf{log}\ (\hat{\boldsymbol{y}}) + (\mathbf{1} - \boldsymbol{y})\mathbf{log}\ (\mathbf{1} - \hat{\boldsymbol{y}}))2.\mathbf{2})$$

y : Étiquette (1 pour des signatures similaires, 0 pour des signatures non similaires).

$\hat{y}$: Probabilité prédite par le modèle que les signatures soient similaires, Si le modèle prédit correctement une probabilité élevée (proche de 1), la perte est faible, si non élevée.

Ensuite, dans la dernière couche du modèle, une fonction d'activation sigmoïdale est appliquée à la perte binaire croisée (BCE). Cette fonction permet de produire des sorties dans l'intervalle. que l'on peut alors interpréter comme des probabilités ou des scores de similarité, ce qui est souvent crucial dans les problèmes de classification ou de régression. De plus, en introduisant une non-linéarité, la fonction sigmoïdale aide le modèle à capturer des relations complexes entre les entrées et les sorties.

La fonction d'activation sigmoïdale est définie mathématiquement comme suit :

$$6(x) = \frac{1}{1 + 6^{-x}}2.3)$$

Où x est une sortie de la dernière couche du réseau avant l'application de sigmoïde, et 6 est la base de l'exponentielle naturelle, approximativement égale à 2,71828. Cette fonction transforme une valeur d'entrée x en une valeur comprise entre 0 et 1, ce qui est particulièrement utile pour modéliser des probabilités [26].

2.3 Vérification de l'authenticité de la signature avec le modèle VGG16

Après avoir évalué la similarité des signatures en entrée avec une référence, l'attention se porte sur la vérification de leur authenticité. L'objectif est de déterminer si la signature appartient effectivement à son signataire légitime ou si elle est potentiellement falsifiée. Même si le modèle Siamois indique une forte similarité entre les deux signatures, il est essentiel de quantifier cette similarité de manière indépendante pour chaque signature. Pour ce faire, le modèle VGG16 est utilisé pour analyser les caractéristiques de la signature et renforcer les résultats de la vérification.

Ce processus suit une approche méthodique visant à garantir une vérification efficace des signatures.

Le réseau VGG16 est d'abord alimenté avec ses poids pré-entraînés et ses couches supérieures sont adaptées avec un classificateur spécifique à notre tâche. Ce modèle de vérification de signatures produit en sortie une distribution de probabilités sur deux classes : la première classe représente la probabilité que les signatures soient authentiques (cible 1), tandis que la seconde classe représente la probabilité que les signatures soient falsifiées (cible 0). Par exemple, si la sortie est [0,8, 0,2], cela signifie que le modèle prédit avec une probabilité de 80 % que la signature est réelle, et avec une probabilité de 20 % qu'elle est falsifiée.

Après avoir définie l'architecture du réseau, on commence l'entraînement du modèle. Cela consiste à optimiser les poids du modèle afin de minimiser une fonction de perte. Après l'entraînement, il est nécessaire d'évaluer le modèle sur l'ensemble de validation pour ajuster les hyperparamètres (comme le taux d'apprentissage, la taille du lot, le nombre d'époques, etc.) et ainsi améliorer ses performances.

2.4 Résultats d'analyse

Une fois les modèles entraînés, on évalue leur efficacité en vérification et en reconnaissance sur l'ensemble de test afin d'obtenir une estimation finale de leurs performances.

Les résultats obtenus seront analysés pour évaluer la performance du modèle. Cette analyse inclura la vérification de divers scores d'évaluation tels que l'exactitude, le rappel et le score F1. Une attention particulière sera portée aux faux positifs et aux faux négatifs pour identifier les domaines nécessitant des améliorations.

Lorsque le modèle est affiné, il est crucial de le soumettre à des ensembles de données variés pour évaluer sa généralisation et sa robustesse. Cette étape permet de déterminer si le modèle est capable de traiter des nouvelles données efficacement ou s'il est trop spécifique aux données d'entraînement.

Chapitre 3 Implémentation

3.1 Introduction

Ce chapitre présente la mise en œuvre de notre approche de vérification de signatures développée dans le cadre de cette recherche. Après avoir exposé les étapes de la méthodologie proposée, ce chapitre devient un élément clé, car les méthodes théoriques se traduisent en résultats concrets. L'objectif est de décrire le développement et l'entraînement, en y intégrant les informations et les outils nécessaires à l'intégration de la méthodologie.

Cette partie débute par une explication des langages de programmation et des bibliothèques utilisés pour la mise en œuvre. Ensuite, on décrira le flux de travail suivi lors de cette phase, en mettant en évidence les principales étapes. Enfin, ce chapitre offre un aperçu des interfaces graphiques que l'on a développées pour rendre ces modèles simples d'utilisation, en détaillant les aspects techniques et les fonctionnalités qu'ont amélioré l'expérience utilisateur.

3.2 Langage utilisé, bibliothèques et environnement

La mise en œuvre de la méthodologie requiert une sélection minutieuse des langages de programmation et des bibliothèques d'apprentissage automatique, ainsi qu'un environnement de développement qui facilite l'exploration et la visualisation des résultats.

3.2.1 Langage

Le langage principal choisi pour ce projet est Python, et ce choix repose sur plusieurs considérations importantes :

-Richesse des bibliothèques : Python dispose d'un large écosystème de bibliothèques pour l'apprentissage automatique et le traitement des données, y compris, mais sans s'y limiter, TensorFlow, Keras, Scikit-learn, et OpenCV. Ces bibliothèques fournissent des outils puissants et performants pour construire, entraîner et évaluer des modèles d'apprentissage automatique.

-Lisibilité et simplicité : Python se distingue par une syntaxe claire et précise, ce qui facilite la lecture et la maintenance du code. Ce langage est particulièrement adapté aux projets de recherche, qui reposent sur la coopération et une documentation rigoureuse.

-Communauté active : Python dispose d'une communauté de développeurs très active. Cela implique la disponibilité de nombreuses ressources et tutoriels, ainsi qu'une excellente assistance en ligne. Cela permet d'accélérer le développement de solutions, évitant ainsi de devoir chercher des réponses dans un vide d'informations.

3.2.2 Bibliothèques

Divers outils et bibliothèques ont été déployés au cours de cette phase d'implémentation :

-TensorFlow/Keras :

Ces deux outils sont des produits open-source développés par Google. Les fonctionnalités de TensorFlow comprennent l'apprentissage automatique et multimodal, ainsi que l'exécution de tâches d'apprentissage profond. Keras est une API de réseau neuronal intégrée à TensorFlow ; c'est une bibliothèque qui permet de construire et d'entraîner des modèles de réseaux neuronaux, tout en offrant une configuration simplifiée.[28]

-OpenCV :

OpenCV (Open Source Computer Vision Library) est une bibliothèque open-source dédiée au traitement des images et à la vision par ordinateur. Elle est utilisée pour effectuer des opérations telles que le redimensionnement, la normalisation et l’augmentation des données. Étant donné que l’entraînement des modèles est une tâche de big data, ces étapes sont essentielles pour préparer les données en masse dans un état optimal [29].

-Scikit-learn :

Dans ce projet, la bibliothèque d'apprentissage automatique open-source Scikit-learn sera utilisée. Ce choix est justifié par le fait qu'elle offre des outils simples et efficaces pour les tâches d'analyse et de fouille de données. Scikit-learn est employée pour des tâches complémentaires telles que le calcul des métriques de performance, la validation croisée et d'autres analyses statistiques des résultats des modèles. Parmi ces tâches, on trouve le calcul de l'exactitude, du score F1, de la précision et du rappel [30].

3.2.3 Environnement

Afin d'accélérer l'exécution du modèle, Google Colab a été utilisé comme plateforme de développement pour le projet. Ce service de Google Research est conçu pour permettre l'écriture et l'exécution de code Python directement dans le navigateur. Il s'agit d'un bloc-notes basé sur le cloud qui fournit des outils de calcul gratuits et payants, y compris des GPU et des TPU.

Les avantages de Google Colab incluent l'accélération matérielle, l'accès aux GPU et TPU pour les tâches de calcul intensif, ainsi qu'une interface ne nécessitant aucune configuration matérielle ou logicielle. De plus, il facilite le partage des notebooks avec les collègues et la collaboration en temps réel. Les blocs-notes sont sauvegardés sur Google Drive, ce qui permet un accès facile depuis n'importe quel appareil. Cependant, pour les projets à grande échelle, les environnements de production ou les exigences spécifiques en matière de ressources et de sécurité, il est important de bien évaluer les limites de Google Colab. Dans ce cas, il pourrait être nécessaire d'opter pour des solutions de cloud computing plus robustes, configurables et payantes, telles qu'Amazon AWS, Microsoft Azure ou Google Cloud Platform, ou de créer un environnement local adapté aux besoins du projet.

La Figure 4.1, qui suit, représente l'interface du service Google Colab utilisé durant toute la période de recherche.

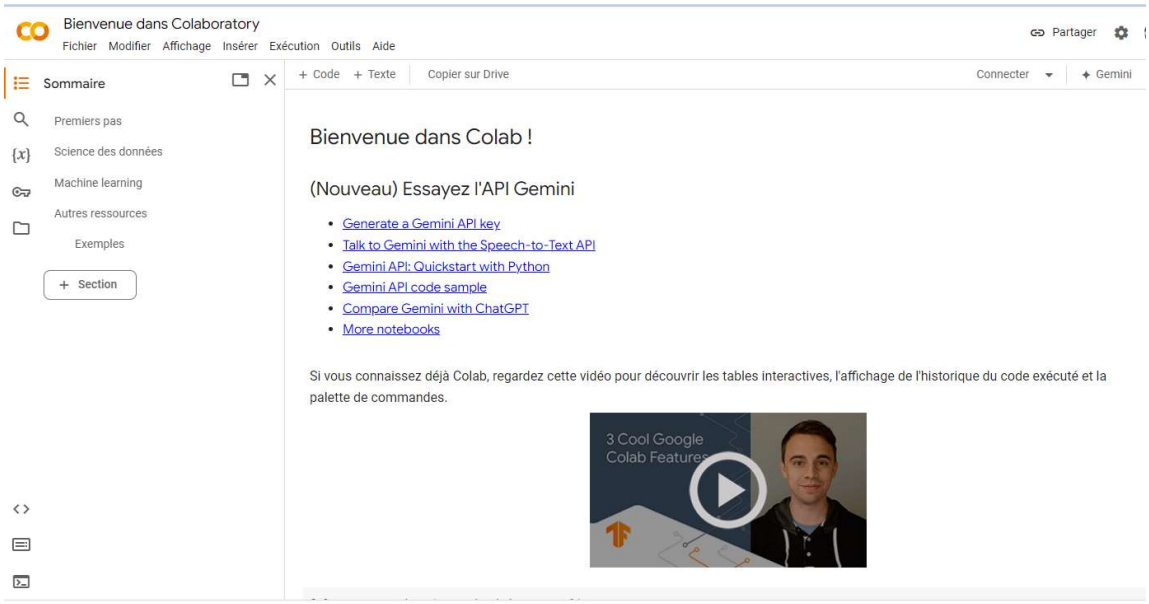

Figure 3.1 plateforme de Google Colab [31]

3.3 Implémentation de workflow

Dans cette section, on détaille le processus de mise en œuvre de la procédure de vérification des signatures, conçue pour être robuste et fiable. Le processus est basé sur trois étapes principales : L'implémentation de deux modèles pour la similarité et l'authenticité des signatures Ainsi que la mise en œuvre de l'interface graphique pour une visualisation claire des résultats.

Pour le modèle Siamois, l'architecture de ce modèle y compris les couches et les fonctions d'activation seront détaillés. Ainsi, on mentionnera les techniques de prétraitement de normalisation des données comme l'opération de mise à l'échelle de l'image de la signature utilisés dans le processus d'apprentissage.

La deuxième étape consiste à vérifier l'authenticité des signatures. Pour ce faire, le modèle VGG16, un réseau neuronal convolutionnel pré-entraîné sur une vaste base de données d'images, est adapté. Dans la suite de cette étude, les modifications apportées au modèle de base ainsi que le processus d'adaptation du modèle à la base de données

seront expliquées. Enfin, l'interface graphique développée pour visualiser les résultats de la vérification sera présentée.

3.3.1 Modèle de similarité des signatures

Dans cette étape, à l'aide du modèle siamois, on compare les signatures. Ce réseau est conçu pour recevoir une paire de signatures en entrée et déterminer le degré de similitude entre les deux. Tout d'abord, on prépare l'ensemble de données en attribuant des étiquettes à chaque paire de signatures. Ensuite, les données passent par le réseau entraîné à distinguer les signatures similaires de celles qui ne le sont pas.

-Prétraitement des données

Des signatures ont été collectées à partir de diverses ressources en quantité suffisante pour permettre au modèle un apprentissage adéquat. Ensuite, les signatures en double ou mal capturées ont été éliminées. Une matrice de données à trois colonnes a été construite : X1 contenant les signatures de référence réelles, X2 contenant les signatures authentiques de la même personne et les signatures falsifiées. La falsification peut prendre plusieurs formes : elle peut être réalisée par la même personne avec diverses variations, par une autre personne, ou bien les signatures peuvent être authentiques mais appartenir à des signataires différents, de manière aléatoire. Chaque ligne de la matrice est associée à une valeur Y indiquant si les signatures X1 et X2 sont similaires (1) ou non (0).

Pour une meilleure compréhension de la méthode, le tableau 4.1 présente un exemple avec deux signataires différents, chacun ayant six signatures (trois réelles (R) et trois falsifiées (F)). Le jeu de données associé est illustré dans le tableau 4.2.

Signataire 1	Signataire 2	Type de signature
S1_R1	S2_R1	Réel
S1_R2	S2_R2	
S1_R3	S2_R3	
S1_F1	S2_F1	Falsifiée
S1_F2	S2_F2	
S1_F3	S2_F3	

Tableau 3.1 Exemple de deux signataires déférents

CLASSEMENT	X1	X2	Y
Paires de signatures de même personne réel	S1_R1	S1_R2	1
	S1_R1	S1_R3	1
	S1_R2	S1_R3	1
	S1_R2	S1_R2	1
	S1_R3	S1_R2	1
	S1_R3	S1_R1	1
	S2_R1	S2_R2	1
	S2_R1	S2_R3	1
	S2_R2	S2_R3	1
	S2_R2	S2_R2	1
	S2_R3	S2_R2	1
	S2_R3	S2_R1	1

Paires des signatures de même personne falsifiée	S1_R1	S1_F1	0
	S1_R1	S1_F2	0
	S1_R1	S1_F3	0
	S1_R2	S1_F1	0
	S1_R2	S1_F2	0
	S1_R2	S1_F3	0
	S1_R3	S1_F1	0
	S1_R3	S1_F2	0
	S1_R3	S1_F3	0
Paires des signatures de personne défèrent aléatoirement	S1_R1	S2_R3	0
	S1_R1	S2_F2	0
	S1_R1	S2_R3	0
	S1_R2	S2_F1	0
	S1_R2	S2_R2	0
	S1_R2	S2_F3	0
	S1_R3	S2_R1	0
	S1_R3	S2_F2	0
	S1_R3	S2_R3	0

Tableau 0.1 le jeu de données associé à l'exemple.

Une fois la préparation du jeu de données avec les paires et les étiquettes terminée, on le mélange de manière aléatoire avec une graine fixe pour garantir la reproductibilité des opérations aléatoires en utilisant la fonction random de NumPy. Ensuite, la matrice est divisée en trois parties distinctes pour l'entraînement, la validation et le test : 60 %, 20 % et 20 % respectivement.

-Architecture du modèle Siamois

Le modèle siamois se compose de deux sous-réseaux identiques, partageant les mêmes poids pour l'extraction des caractéristiques de la signature. La figure 4.2 ci-dessous donne un aperçu de l'architecture, suivie d'une explication du modèle couche par couche :

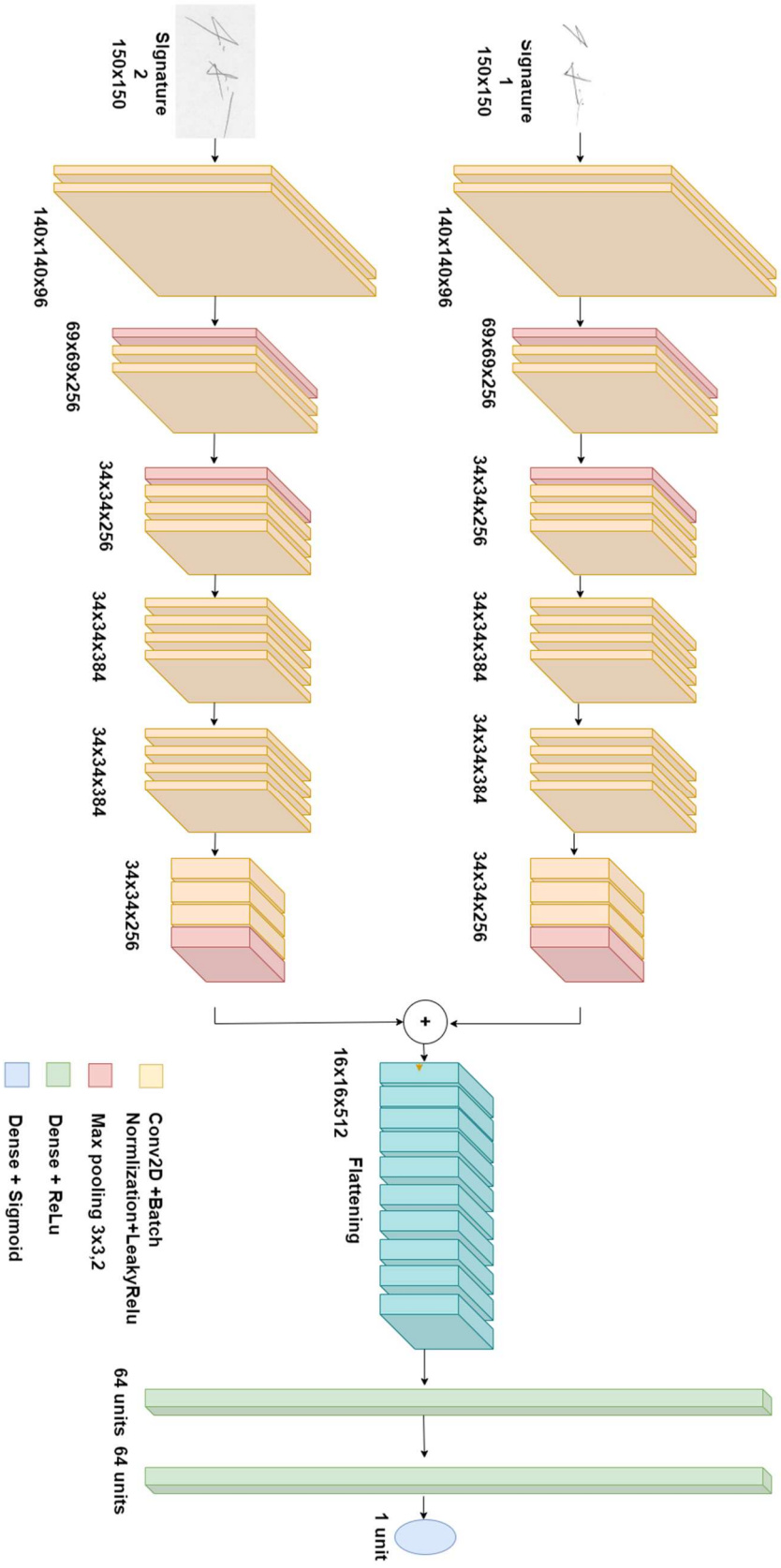

Figure 4.2 Architecture du modèle Siamois.

- Couches de Convolution :

L'image passe à travers plusieurs couches de convolution pour extraire des caractéristiques visuelles locales. Les premières couches utilisent 96 filtres de taille 11x11 avec un stride de 1, ce qui signifie que le filtre se déplace d'un pixel à la fois lors de la convolution pour détecter des motifs spécifiques dans les signatures d'entrée. Les couches suivantes utilisent 256 filtres, puis 384 filtres deux fois finnisant avec une autre couche convolutive de 256 filtres pour capturer des caractéristiques plus complexes.

- Normalisation par lots et Leaky ReLU:

À la fin de chaque couche de convolution, la normalisation par lots est effectuée pour stabiliser et accélérer l'entraînement du réseau en normalisant les activations des neurones. Cela permet d'assurer que les valeurs passent à travers le réseau à une vitesse améliorée et constante, ce qui permet de réduire le problème de décalage des co-variables et facilite l'apprentissage de modèles profonds.

La normalisation par lots désigne une technique qui vise à stabiliser et à accélérer le processus d'apprentissage d'un réseau de neurones en ajustant les activations produites à chaque couche. Elle est exécutée en soustrayant la moyenne du lot et en divisant le résultat par l'écart type du lot. Cette technique corrige les problèmes de convergence posés par des activations très grandes ou très petites, rendant ainsi l'apprentissage plus rapide et plus stable. Généralement, la normalisation par lots est réalisée après une opération de convolution et d'opérations de densité dans une couche d'un réseau de neurones.

Ensuite, l'activation Leaky ReLU est appliquée. Contrairement à ReLU traditionnelle qui met à zéro tous les neurones de valeurs négatives, elle permet aux valeurs de passer avec une petite pente (on a utilisé 0,2). Cette activation ajoute de la non-linéarité dans le modèle, ce qui aide à capturer les complexités des données d'entrée, elle empêche également les neurones de devenir inactifs. De cette façon, même les valeurs négatives ont un gradient qui les traverse et qui ne se bloque pas à zéro permettant au modèle d'apprendre en conséquence. La valeur de la pente est également fixée à une valeur faible fixée généralement à 0,2, mais peut être ajusté selon les besoins.

- Couches de sous-échantillonnage :

Le pooling, ou sous-échantillonnage, comme mentionné dans l'état de l'art, est effectué après la couche de convolution pour réduire la dimensionnalité des caractéristiques tout en conservant les détails nécessaires. Cette étape utilise un max pooling 3×3 avec un pas de 2.

- Concaténation :

Chaque sous-réseau produit un vecteur de caractéristiques pour son image de signature respective. La concaténation consiste à combiner les sorties des deux sous-réseaux pour former un seul vecteur.

- Couches entièrement connectées :

Les caractéristiques concaténées passent par des couches entièrement connectées pour produire un vecteur de représentation à haute résolution adapté à l'analyse comparative. Deux couches entièrement connectées avec 64 neurones chacune et des activations ReLU sont utilisées.

- Couche de sortie :

Enfin, un calibreur de sortie avec une fonction d'activation sigmoïde produit le score de similarité compris entre 0 et 1.

-Compilation du modèle

La méthode *compile()* de Keras est utilisée pour définir la fonction de perte, l'optimiseur et les métriques de suivi de l'apprentissage du modèle avant de commencer l'entraînement. Pour ce problème de classification binaire, la fonction de perte binary_crossentropy est choisie. L'optimiseur Adam est utilisé avec un taux d'apprentissage faible de 10^{-8} afin de garantir des mises à jour de poids fines et précises. Ensuite, plusieurs mesures telles que l'exactitude, la précision, le rappel et le score F1 sont spécifiées à ce stade pour évaluer les performances du modèle.

-Entraînement du modèle :

Après avoir compilé le modèle, on l'entraîne à l'aide de la fonction Keras fit() avec les paramètres suivants :

- Génération de l'entrainement et la validation : Pendant cette procédure, un générateur de données alimente le modèle Siamois avec des paires d'images de signatures et leurs étiquettes respectives. Celui-ci charge les données en mémoire par petits lots et continue à les produire tout au long des cycles d'entrainement. Il permet de gérer de grandes quantités d'images sans surcharger la mémoire du système.
- Nombre d'époques (epochs) : Spécifié à 120, indiquant que le modèle parcourra l'ensemble des données d'entraînement 120 fois.
- Taille du lot (batch size) : Fixée à quatre, ce qui signifie que les mises à jour des poids sont effectuées après le traitement de chaque lot de 4 paires d'images.

3.4 Modèle d'authenticité des signatures

Dans cette étape, l'authenticité des signatures est vérifiée en utilisant le modèle VGG16. Chaque signature de l'ensemble de données est étiquetée. Ensuite, les signatures sont incorporées dans l'architecture choisie. Pour ce faire, le modèle est adapté en ajoutant des couches supplémentaires et en formant un nouveau classificateur.

-Prétraitement des Données

En utilisant la même base de données, un nombre suffisant de signatures a été rassemblé pour garantir un apprentissage efficace du modèle. Par la suite, un jeu de données a été construit avec deux colonnes : X représente les signatures d'images, et Y contient les étiquettes correspondantes, avec 1 pour les signatures authentiques et 0 pour les signatures falsifiées.

-Architecture du Modèle VGG16 :

La structure du modèle est divisée en deux parties : La base pré-entraînée, à laquelle des modifications seront appliquées, et la partie complémentaire pour l'adaptation du modèle aux besoins. La figure 4.3 présente l'architecture proposée.

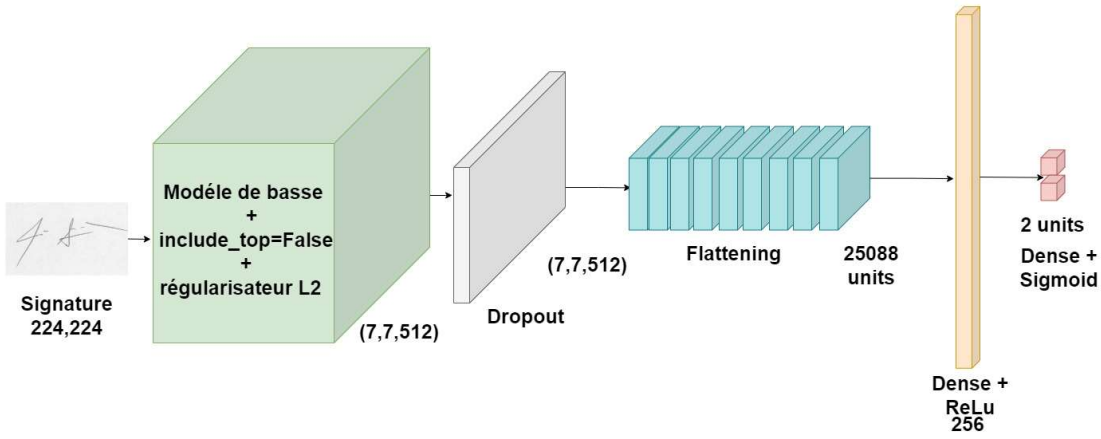

Figure 4.3 architecture de modèle VGG16 adapté

Modifications apportées au modèle de base : Ce modèle à 16 couches développé sur Keras et pré-entraîné sur l'ensemble de données ImageNet. Les couches entièrement connectées du modèle original sont exclues (include_top=False), ce qui permet d'utiliser uniquement des couches convolutives pour l'extraction des caractéristiques.

La régularisation L2 est ajouté à chaque couche convolutive pour réduire les poids élevés et éviter le sur-apprentissage.

Couches supplémentaires ajoutées : Pour améliorer la généralisation et la robustesse du modèle, plusieurs couches supplémentaires sont intégrées avec la base. Il s'agit des couches de décroissance, d'aplanissement et de densité.

- Couche de décroissance (Dropout) **:**

Un taux de décroissance de 0,2 est appliqué. Il aide à réduire le surapprentissage en désactivant aléatoirement 20% des neurones pendant l'apprentissage. Cela permet

d'entraîner des réseaux plus simples, rendant ainsi le modèle plus robuste et améliorant sa capacité de généralisation.

- Couche d’aplatissement (Flatten) :

Cette couche convertit les cartes de caractéristiques 2D issues des couches convolutionnelles en un vecteur 1D. Cette transformation permet de relier les couches convolutionnelles aux couches entièrement connectées.

- Dense Layer (256 unités, activation ReLU) :

Cette couche apprend des combinaisons complexes de caractéristiques grâce à des extractions effectuées par les couches de convolutions.

- Couche entièrement connectée (256 unités, activation ReLU) :

Cette couche apprend des combinaisons complexes de caractéristiques grâce aux extractions effectuées par les couches convolutionnelles.

- Couche entièrement connectée (2 unités, activation Sigmoid) :

La dernière couche entièrement connectée génère les probabilités des classes (authentique ou falsifiée) en utilisant une activation sigmoïde. La sortie est un vecteur de taille 2, chaque élément représentant la probabilité d'appartenance à l'une des classes.

-Entraînement du modèle :

Le modèle VGG16 est compilé avec la fonction de perte binary_crossentropy. Il utilise l'optimiseur SGD (Stochastic Gradient Descent), qui ajuste les poids du modèle de manière itérative pour minimiser la fonction de perte. Cela se fait en calculant les gradients sur un petit sous-ensemble aléatoire des données d'entraînement à chaque étape, avec un taux d'apprentissage de 0,0001. Les métriques suivantes sont également utilisées : exactitude, précision, rappel et score F1. On a utilisé la configuration suivante pour l’entrainement du modèle :

- Nombre d'époques : 300 est le nombre de fois où le modèle parcourt l'ensemble complet des données d'entraînement et de validation.
- Taille du lot (Batch Size) : Le nombre d'exemples d'entraînement utilisés pour estimer le gradient lors d'une seule mise à jour des poids du modèle. Dans la configuration que l'on a choisie, la taille du lot est fixée à 128,

3.5 Interface graphique

L'interface graphique a été développée pour être simple à utiliser et à manipuler. Elle permet de charger deux signatures et de visualiser les résultats de la comparaison selon les modèles de similarité et de vérification. Les principales étapes de la procédure sont détaillées dans cette section.

L'interface graphique est créée à l'aide de la bibliothèque Tkinter , TensorFlow/Keras sont utilisées pour charger et exécuter les modèles, Siamois et VGG16 sont préformés et sauvegardés sur le disque. Ces modèles seront chargés dans le script afin d'être utilisés pour les prédictions.

Les images de signatures doivent être prétraitées de la même manière que lors de l'entraînement des modèles. Cela inclut des étapes comme le redimensionnement des images à la taille appropriée et la normalisation des valeurs des pixels.

Une fois les images chargées et prétraitées, elles sont passées à travers les modèles Siamois et VGG16 pour obtenir les prédictions. Les résultats des prédictions sont ensuite affichés sur l'interface graphique.

Les éléments de l'interface comprennent : deux boutons pour importer des images de signatures, deux panneaux pour afficher les images chargées, un bouton pour déclencher les prédictions, et deux zones de texte pour visualiser les résultats prédits par les deux modèles.

Ce qui suit des exemples de capture d’écran de l’interface graphique, pour une meilleure compréhension,

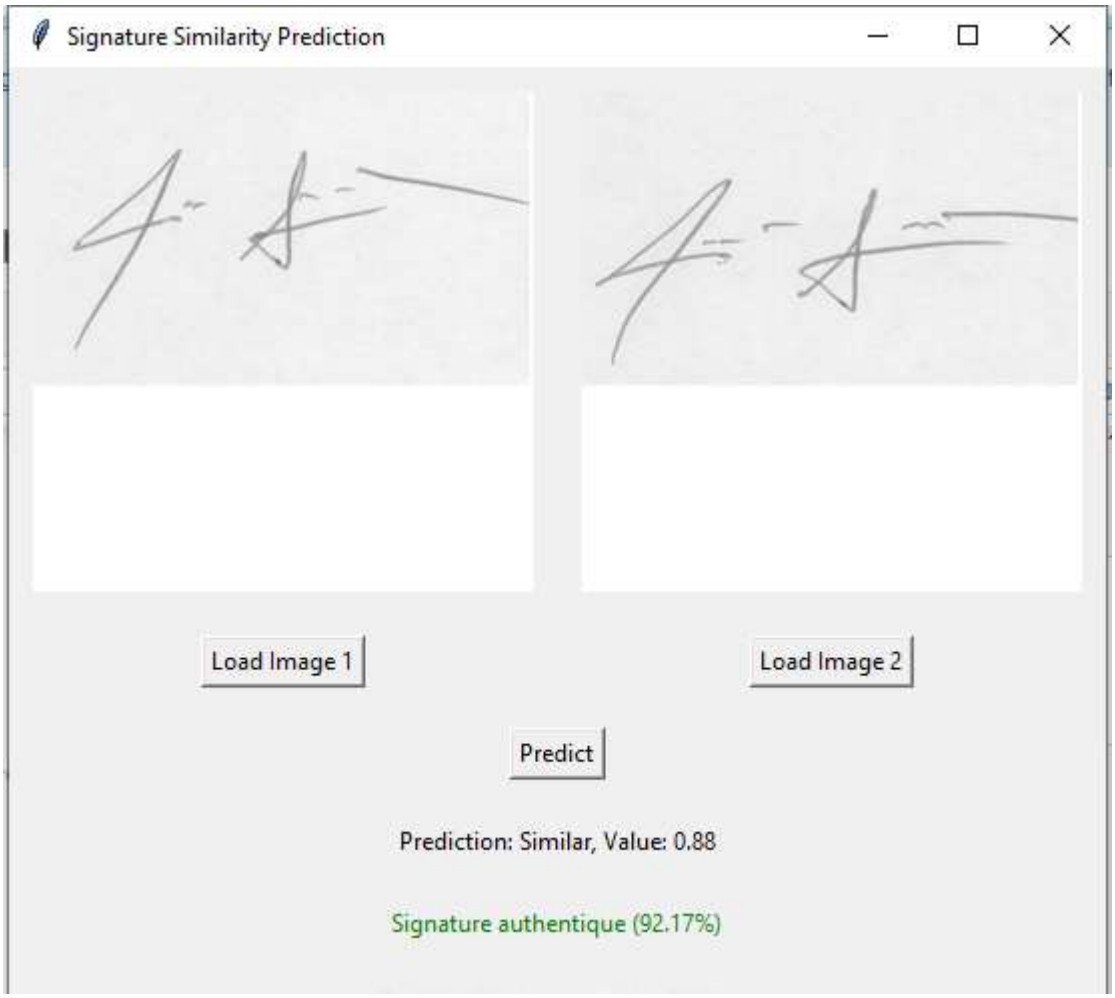

Figure 4.4 Exemple 1

Dans la figure 4.4, deux signatures réelles de la même personne ont été chargées. Lorsque l'on clique sur "prédire", deux résultats s'affichent : le premier indique une similarité de 0,88 entre les deux signatures, basée sur le modèle Siamois. La seconde valeur, calculée à l'aide du modèle VGG16, indique une probabilité de 92,17 % que la signature soit authentique.

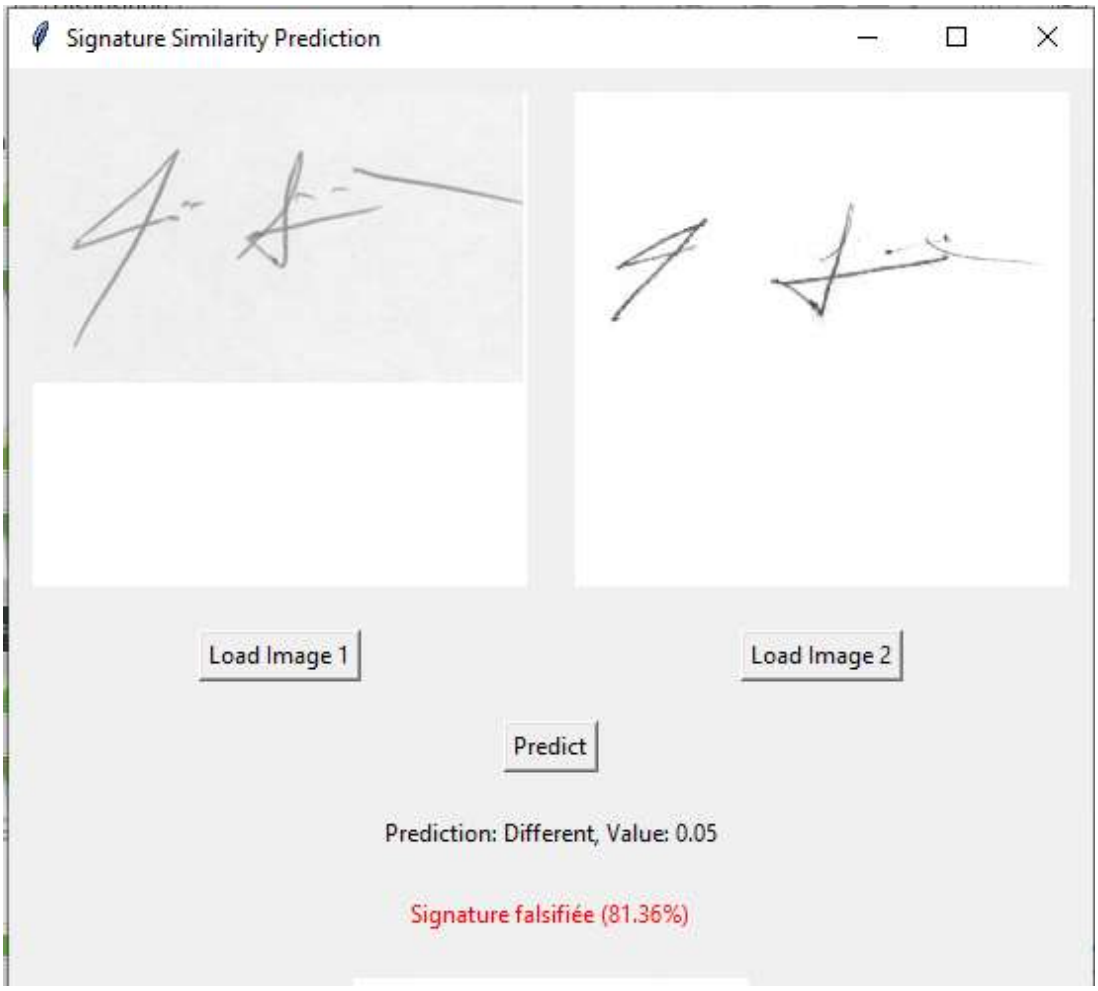

Figure 4.5 Exemple 2

La figure 4.5, représente deux signatures de la même personne, la première est réelle et la deuxième est falsifiée, la première valeur de similarité indique 0,05, tandis que le second donne un pourcentage de 81,36 % de signature falsifiée. Cela indique que la signature est à la fois falsifiée et non authentique.

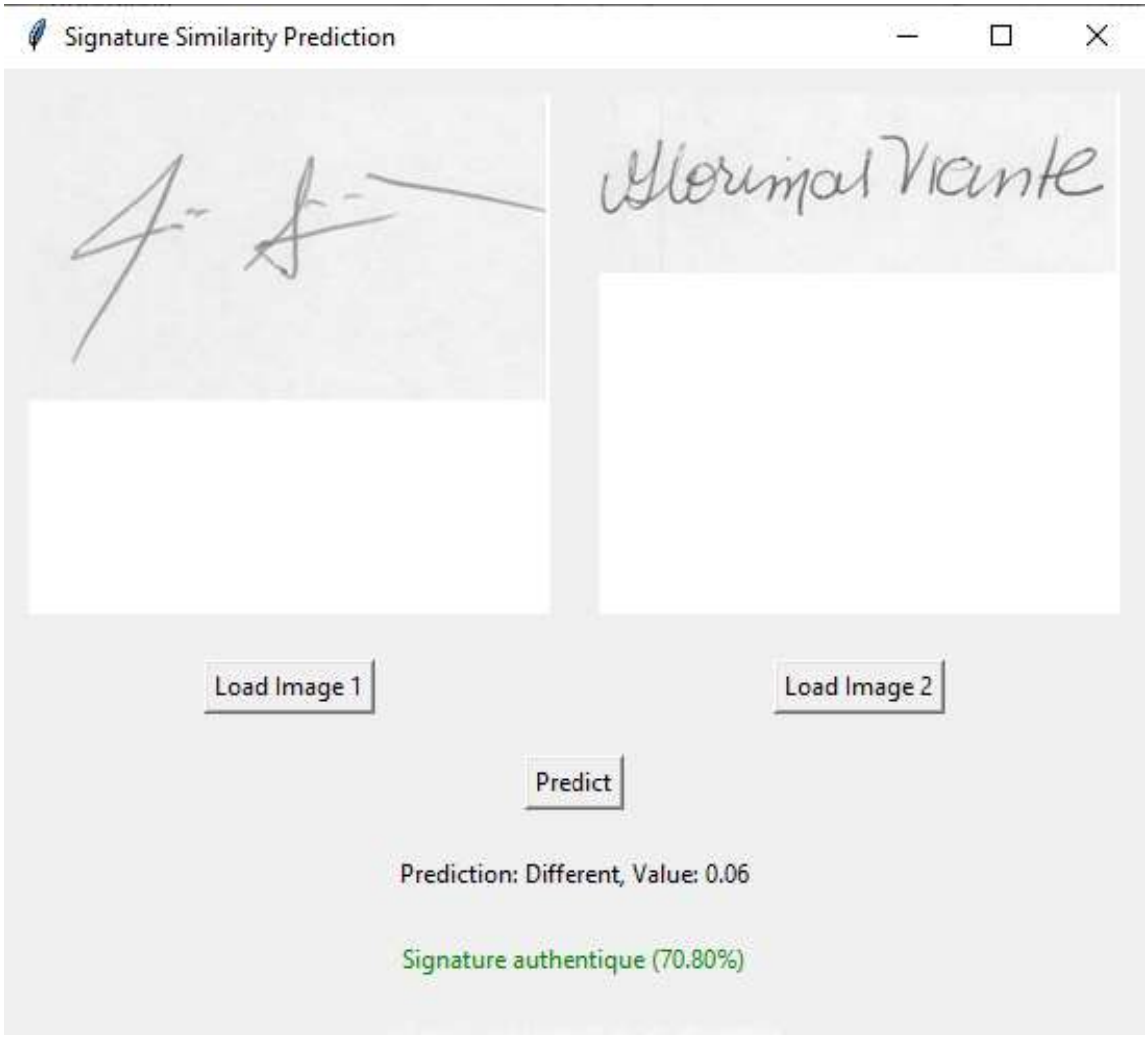

Figure 4.6 Exemple 3

Dans la Figure 4.6, la valeur de similarité est de 0,06, tandis que le pourcentage d'authenticité de la signature est de 70,80 %. En d'autres termes, bien que la signature provienne d'une personne différente, elle est considérée comme authentique.

Chapitre 4 Expérimentation

Dans ce chapitre, l'objectif est de présenter les performances des deux modèles développés (modèle Siamois et modèle VGG16) pour la vérification des signatures. On évaluera chaque modèle en termes de précision, de rappel, de F1-score et d'autres mesures pertinentes, afin de démontrer leur efficacité dans la détection des signatures non-authentiques et falsifiées. Une analyse détaillée des résultats obtenus permettra de mettre en évidence la capacité de chaque modèle à accomplir cette tâche.

4.1 Données

La base de données utilisée dans l'expérimentation est un mélange de plusieurs bases de données bien connues dans le domaine de la recherche sur la vérification de signatures. Ce mixte de bases de données nous permet d'avoir une diversité et une quantité suffisante de signatures pour entraîner et évaluer les modèles de manière efficace.

La première base de données, fournie par M. Nouboud, mon premier superviseur, comprend 1020 images de 30 signataires différents. Elle est constituée de 50 % de signatures authentiques et de 50 % de signatures falsifiées. Cette base est inspirée de la base de données ICDAR 2011 (International Conference on Document Analysis and Recognition) [32], qui contient des échantillons de signatures manuscrites. Elle est couramment utilisée dans les compétitions organisées par l'ICDAR pour évaluer et comparer les performances des diverses méthodes et approches développées par les chercheurs.

Une autre base de données, CEDAR, est utilisée dans le cadre de ce projet, développée par le Centre d'excellence pour l'analyse et la reconnaissance des documents de l'université de Buffalo, aux États-Unis, entre les années 1990 et 2014. Cette base fournie un ensemble de signatures manuscrites pour le développement et l'évaluation de systèmes de vérification et de reconnaissance de signatures. Elle contient à la fois des signatures authentiques et des signatures falsifiées souvent réalisées par d'autres personnes essayant d'imiter les signatures des signataires originaux.

Cet ensemble de données contient en totale de 2640 signatures de 55 personnes. Chaque personne a 24 signatures authentiques et 24 signatures falsifiées.

4.2 Résultats et analyse

Dans cette étape, l'objectif est de présenter et d'analyser les performances des deux modèles que l'on a développés. Seules les expériences les plus importantes sont présentées, étant donné que des dizaines d'expériences ont été réalisées au cours de la période de recherche. Cette analyse détaillée démontrera l'efficacité des modèles dans la détection des signatures authentiques et falsifiées. Chaque modèle est évalué en termes d'exactitude, de précision, de rappel et de score F1.

4.2.1 Résultats de modèle Siamois

Cette section présente également les expériences pertinentes qui ont conduit au modèle Siamois final. Tout d'abord, les résultats de l'approche adoptée avec la fonction de perte BCE sont présentés, suivis d'une comparaison avec la seconde approche qui utilise la fonction de perte de contraste traditionnelle en termes de métriques et de temps d'exécution sur l'ensemble de test, puis, une comparaison avec d'autres études existantes.

-Résultat du modèle Siamois avec la perte BCE :

La figure 5.1 illustre l'exactitude de l'apprentissage sur l'ensemble d'entrainement de 58,57 % et atteint progressivement 95,18 %. Cette augmentation régulière indique une amélioration continue du modèle au fil des itérations. De même, la validation commence de 65,26 %, progresse jusqu'à 94,24 %, ce qui suggère une capacité de généralisation du modèle sur les données.

Au cours des dernières époques, l'apprentissage de l'entrainement s'est stabilisé autour de 95,12 % à 95,18 %, tandis que la validation oscille entre 94,20 % et 94,24 %. Cette stabilité indique que le modèle a atteint un point où il n'y a plus d'améliorations

significatives à chaque itération. Cela suggère que le modèle a appris de manière optimale les caractéristiques des données sans surapprentissage.

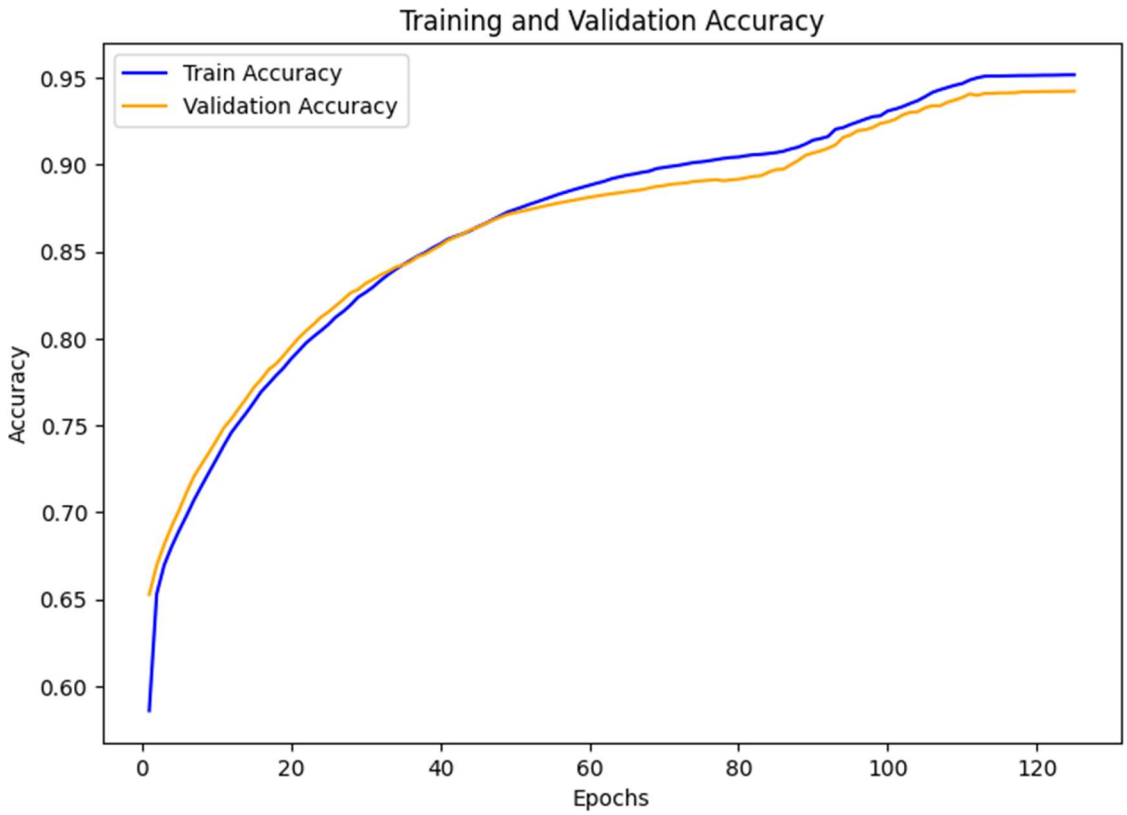

Figure 4.1 Courbe d'exactitude sur l'ensemble d'entrainement et de validation

Les résultats présentés dans le tableau d'évaluation (Tableau 5.1) offrent une vue d'ensemble complète des performances du modèle sur l'ensemble de validation. L'exactitude du modèle est de 94,11%, ce qui signifie qu'il a correctement classé 94,11% des échantillons de validation. La précision, quant à elle, est de 94,15%. Elle indique que parmi toutes les instances prédites positives, 94,15% étaient effectivement positives, montrant ainsi que le modèle est précis et commet peu de faux positifs. Le rappel du modèle est de 94,81%, indiquant que parmi toutes les instances réellement positives, le modèle a réussi à en identifier ce pourcentage. Le score F1, qui combine la précision et le rappel, est de 94,43%.

Exactitude	Précision	Rappel	Score F1
94,11 %	94,15 %	94,81 %	94,43 %

Tableau 4.1 Résultats d'évaluation du modèle proposé sur l'ensemble de validation

Comparaison de la fonction de perte BCE proposée avec la fonction contractive traditionnelle :

Le tableau 5.2 montrant les résultats comparatifs des fonctions de perte révèle des différences notables dans leurs performances et leur efficacité sur l'ensemble de test. La fonction de perte Binaire Cross-Entropy (BCE) affiche une exactitude de 92,11%, avec une précision de 92.15% et un rappel de 91,81%. Le score F1 associé est de 92.02%, ce qui témoigne d'un équilibre entre la précision et le rappel. En termes de temps d'exécution, la BCE se distingue par sa rapidité, nécessitant seulement 0,6 seconde pour le traitement. En revanche, la fonction de perte de contraste obtient des résultats supérieurs, avec une exactitude de 94,31%, une précision de 94,32% et un rappel de 93,84%. Son score F1 de 94,08% reflète une performance encore plus équilibrée et précise. Toutefois, cette fonction de perte est plus coûteuse en termes de temps, avec un temps d'exécution de 1,1 seconde. Ainsi, bien que la fonction de perte de contraste offre des performances accrues, elle nécessite un compromis en termes de vitesse.

La fonction de perte de contraste offre légèrement une meilleure performance en termes d'exactitude, de précision, de rappel et de score F1, mais nécessite un compromis en termes de vitesse d'exécution. La BCE, malgré des résultats légèrement inférieurs, se caractérise par sa rapidité, qui peut être un facteur décisif dans des contextes où le temps d'exécution est essentiel.

Fonction de perte	Exactitude	Précision	Rappel	Score F1	Temps d'exécution
Approche proposer BCE	92,23%	92,12%	91,71%	91,93%	0,6s
Perte de contraste	94,31%	94,32%	93,84%	94,08%	1,1s

Tableau 4.2 Résultats de prédiction des fonctions de perte BCE et de contraste sur ensemble de test

Figure 4.2 Exemple de paire de signatures réelles

Cependant, il est important de noter les différences en termes de temps d'exécution. Depuis le début du chargement du modèle, le prétraitement de l'image et la prédiction, la figure 5.2 illustre un exemple de paire qui a donné une similarité de 0,88 entre les deux images avec la perte BCE proposée et de 0,91 pour la perte de contraste, sauf que le temps d'exécution était de 0,52 seconde pour le modèle BCE et de 0,89 seconde pour le modèle utilisant la perte de contraste. Cela souligne le fait que, bien que ce modèle offre une meilleure précision, il nécessite également plus de temps pour faire des prédictions.

Dans le cas de la vérification de 10 000 paires de signatures pour une application bancaire, le modèle BCE nécessiterait environ 5 200 secondes (ou 1 heure et 26 minutes) pour traiter toutes les paires, tandis que le modèle utilisant la perte de contraste nécessiterait environ 8 900 secondes (ou 2 heures et 28 minutes). Ainsi, bien que la perte de contraste offre une meilleure précision, le modèle BCE est

41,5% plus rapide et, par conséquent, moins couteux tout en gardant une performance comparable.

4.2.2 Résultats de modèle VGG16

Cette section présente les résultats finaux du modèle VGG16. Elle débute par la présentation des métriques obtenues, suivie de l'examen du rôle des méthodes de régularisation dans l'amélioration des performances du modèle, notamment en réduisant le surapprentissage et en augmentant la généralisation. Ensuite, une analyse comparative avec d'autres méthodes de transfert d'apprentissage est effectuée, avant de conclure par une comparaison avec quelques études de la littérature.

-Résultats de l'entrainement :

Dans la figure 5.3, la précision initiale du modèle était de 0,4815, ce qui signifie que le modèle faisait des prédictions quasiment aléatoires.

D'après les valeurs de l'entrainement, on observe que la précision montre une tendance à la hausse continue, passant à 0,6135 à l'époque 15, 0,8692 à l'époque 100, et atteignant finalement 0,9523 à l'époque 300 pour l'ensemble de validation.

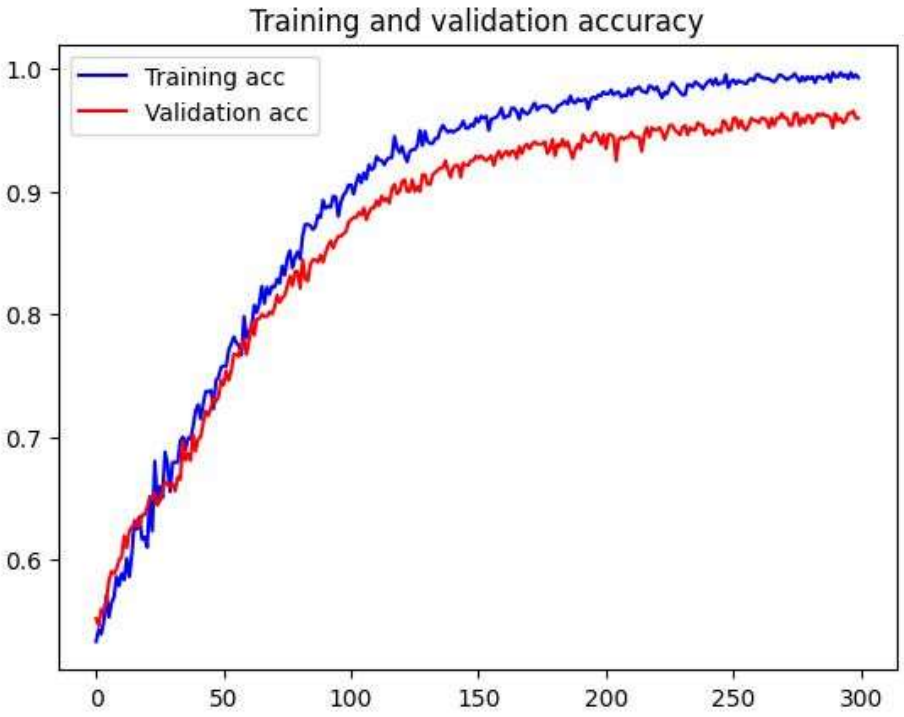

Figure 4.3 Courbe de l'Exactitude de l'entrainement et validation

Afin de d'optimiser les performances du modèle, on a laissé celui-ci s'entraîner pendant 400 époques, comme illustré dans la figure 5,4 On observe que la précision de l'entraînement (ligne bleue) et la précision de la validation (ligne rouge) restent globalement stables et n'affichent pas de gains significatifs après un certain nombre d'époques. Cela suggère que le modèle a atteint son plafond de performance, et que continuer l'entraînement au-delà de ce point ne contribue pas à une amélioration substantielle des résultats.

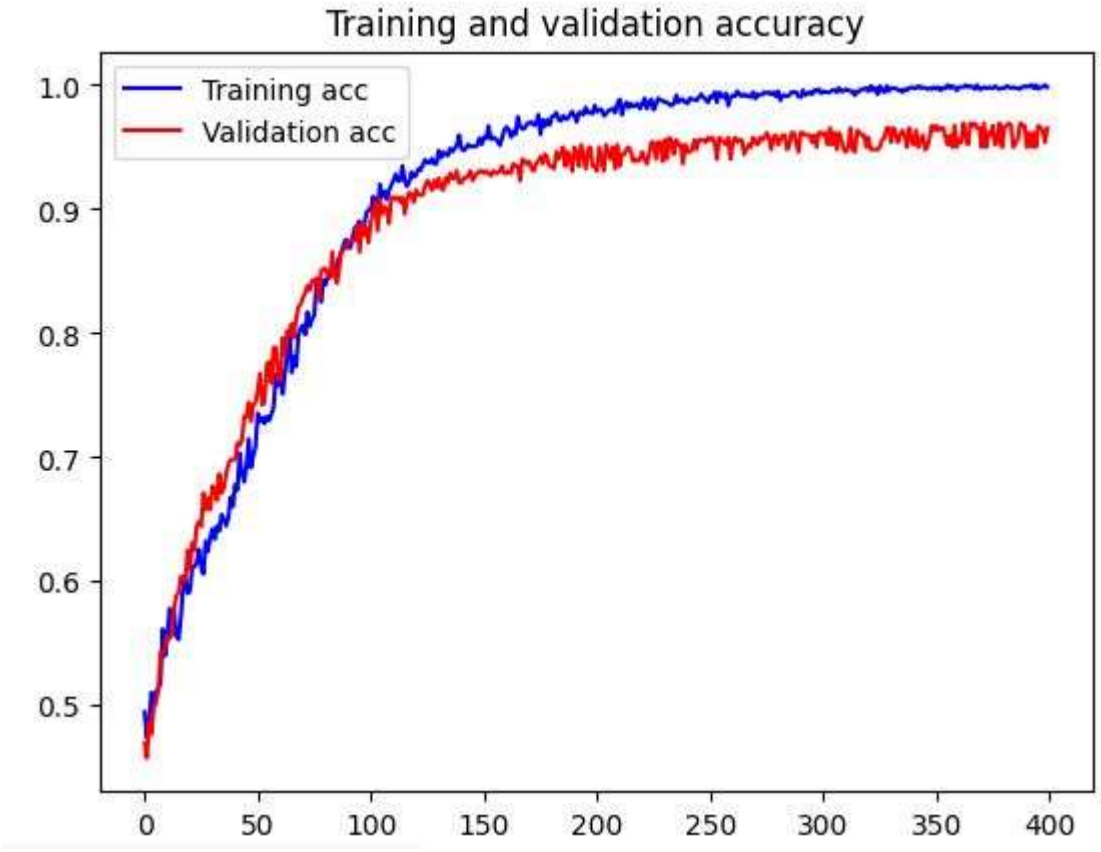

Figure 4.4 Courbe de l'Exactitude de l'entrainement et validation sur 400 époques

Ces résultats indiquent que le modèle a bien appris les caractéristiques des données au cours de l'entraînement. La diminution continue de la perte et l'augmentation de la précision, tant sur les données d'entraînement que de validation, indiquent que le modèle n'est pas en surapprentissage (overfitting) et qu'il généralise bien aux nouvelles données.

Ensemble de donnée	Exactitude	Précision	Rappel	Score F1
Entrainement	0,945	0,937	0,928	0,931
Validation	0,931	0,9273	0,921	0,923
Test	0,92	0,915	0,911	0,912

Tableau 4.3 Résultats d'évaluation sur l'ensemble de donnée

Le tableau 5.3 présente les résultats d'évaluation du modèle sur les ensembles d'entraînement, de validation, et de test.

Ensemble d'entraînement : Avec une exactitude de 94,5%, une précision de 93,7%, un rappel de 92,8% et un score F1 de 93,1%, le modèle performe très bien sur les données d'entraînement, indiquant qu'il a bien appris à identifier les patterns des données.

Ensemble de validation : Des valeurs légèrement inférieures à celles de l'entraînement sont observées : exactitude de 93,1 %, précision de 92,73 %, rappel de 92,1 % et score F1 de 92,3 %. Ces résultats, proches de ceux obtenus lors de l'entraînement, suggèrent que le modèle n'est pas surajusté.

Ensemble de test : Les performances restent élevées avec une exactitude de 92%, une précision de 91,5%, un rappel de 91,1% et un score F1 de 91,2%. Cela confirme que le modèle généralise bien sur des données qu'il n'a jamais vues ,voir tableau 5.3.

Ces résultats confirment que le modèle VGG16 est bien entraîné et capable de fournir des prédictions précises et fiables sur des données non vues.

	Exactitude	Précision	Rappel	Score F1
VGG16 sans régularisation	0,873	0,866	0,861	0,897

Tableau 4.4 Résultats d'évaluation du modèle VGG16 sans régularisation sur l'ensemble de test

Sans régularisation, le modèle VGG16 obtient des résultats inférieurs dans toutes les mesures avec une exactitude de 87,3 % et un score F1 de 89,7 %. Cela indique une performance réduite. L'absence de régularisation peut entraîner un surapprentissage où le modèle s'ajuste trop aux données d'entraînement, compromettant sa capacité à généraliser sur des données nouvelles. Deux techniques de régularisation améliorent les performances : L2 (weight decay) est appliqué sur toutes les couches Conv2D du modèle de base pour pénaliser les poids de grande amplitude, ce qui encourage l'apprentissage de poids plus petits et généralisables. De plus, des couches Dropout supplémentaires sont ajoutées après le modèle de base. Ces couches désactivent aléatoirement un certain pourcentage de neurones pendant l'entraînement pour prévenir le surapprentissage.

-Analyse comparative avec d'autres méthodes de transfert d'apprentissage :

Le tableau 5.5 montre les performances des modèles VGG16, VGG19, InceptionV3 et MobileNet sur les ensembles de données. Le modèle VGG16 obtient les meilleurs résultats, avec une précision de 93,1 %, une exactitude de 92,73 %, un rappel de 92,1 % et un score F1 de 92,3 %. Ces chiffres indiquent que le modèle VGG16 est le plus performant parmi ceux évalués.

Le modèle VGG19 le suit de près, avec une exactitude de 92,3 %, une précision de 92,1 %, un rappel de 91,9 % et un score F1 de 92,0 %. Bien qu'il soit légèrement inférieur au VGG16, il demeure un modèle très compétent avec des performances globales comparables.

Le modèle InceptionV3 obtient des résultats plus modestes, avec une exactitude de 90,7 %, une précision de 90,6 %, un rappel de 91,5 % et un score F1 de 91,0 %. Bien que ses performances soient inférieures à celles des modèles VGG, il reste performant.

Enfin, MobileNet a enregistré les performances les plus faibles parmi les modèles testés, avec une exactitude de 88,6 %, une précision de 88,2 %, un rappel de 89,1 % et un score F1 de 88,6 %. Malgré ses performances globalement inférieures, MobileNet peut être utilisé dans des situations nécessitant une solution plus légère et moins gourmande en ressources.

Modèle	**Exactitude**	**Précision**	**Rappel**	**Score F1**
VGG16	0,931	0,9273	0,921	0,923
VGG19	0,923	0,921	0,919	0,920
InspectionV3	0,907	0,906	0,915	0,910
MobileNet	0,886	0,882	0,891	0,886

Tableau 4.5 Résultats de plusieurs modèles de base

4.3 Analyse comparative :

Afin de comparer le modèle VGG16 proposé avec les résultats des études existantes dans la littérature, le tableau 5.6 présente un résumé des performances de plusieurs modèles de référence. Cette comparaison permet de situer le modèle VGG16 dans le contexte des approches actuelles et d'évaluer son efficacité relative par rapport aux méthodes établies

Architecture	**Base de données**	**Performance**	**Reference**
Siamoise + perte de contraste	ICDAR 2011	Précision : 0,73 Rappel : 0,92 Score F1 : 0,81	Jain, S [23]
Siamoise + perte de contraste	CEDAR	Exactitude :100%	Dey, S. [22]
Siamoise+ perte de contraste	ICDAR 2011	Exactitude :90,11%	Arısoy, M [42]
Réseaux discriminants inverse (IDN)	CEDAR	Exactitude :90,17 %	Wei P, Li H, Hu P (2019) [20]
Siamoise+ perte d'entropie croisée binaire	ICDAR 2011	Exactitude :0,93 Précision : 0,925 Rappel : 0,921 Score F1 : 0,923	Approche proposée Siamoise
Siamoise + perte d'entropie croisée binaire	CEDAR	Exactitude :0,920 Précision : 0,920 Rappel : 0,916 Score F1 : 0,919	
		Exactitude : 0,911 Précision : 0,903	

VGG16	ICDAR 2011	Rappel : 0,901 Score F1 : 0,902	Approche proposée VGG16
VGG16	CEDAR	Exactitude :0,938 Précision : 0,935 Rappel : 0,928 Score F1 : 0,931	

Tableau 5.6 Comparaison avec les études de la littérature

Les résultats des études comparées dans le tableau 5.6 montrent que l'approche proposée utilisant les architectures siamoise et VGG16 pour la vérification des signatures manuscrites est compétitive et souvent supérieure aux autres méthodes.

Jain, S. [23] a appliqué une architecture siamoise associée à une perte de contraste sur la base de données ICDAR 2011, obtenant un score F1 de 0,81, inférieur à celui du modèle proposé.

Dey, S. [22] a également utilisé une architecture siamoise avec une perte de contraste, mais sur la base de données CEDAR, rapportant une exactitude de 100 %. Cependant, l'absence d'information sur le temps d'exécution limite l'évaluation de l'efficacité de cette méthode en termes de rapidité.

L'architecture siamoise associée à une perte de contraste sur la base de données ICDAR 2011, comme présenté dans l'étude [42], a obtenu une exactitude de 90,11 %, ce qui est inférieur à la performance de l'approche proposée avec VGG16 ou de la méthode siamoise utilisant la perte d'entropie croisée binaire.

Dans l'article [20], des réseaux discriminants inverses (IDN) ont été déployés sur la base de données CEDAR. Cette approche a atteint une exactitude de 90,17%, inférieure à l'approche proposée avec VGG16 ou la méthode siamoise.

L'approche siamoise proposée, utilisant une architecture associée à une perte d'entropie croisée binaire, a été testée sur les bases de données ICDAR 2011 et CEDAR. Cette méthode a montré des performances remarquables, obtenant des scores

F1 de 0,923 pour ICDAR 2011 et 0,919 pour CEDAR. Ces résultats surpassent ceux de la plupart des autres études, à l'exception notable de celle de Dey, S. [22].

De son côté, l'approche basée sur VGG16 a également montré une performance robuste. Les scores F1 obtenus sont de 0,902 sur ICDAR 2011 et de 0,931 sur CEDAR. Ces résultats permettent à l'approche VGG16 de se démarquer parmi les différentes méthodes analysées, surpassant plusieurs d'entre elles dans la comparaison.

En conclusion, les approches proposées avec les modèles Siamois et VGG16 démontrent leur efficacité dans la vérification des signatures manuscrites, en offrant des performances élevées et une généralisation efficace par rapport aux autres méthodes existantes.

Chapitre 5 Conclusion

Ce mémoire a exploré différentes approches pour la vérification des signatures manuscrites hors ligne, en se concentrant principalement sur l'application et l'évaluation de modèles de transfert d'apprentissage, notamment les architectures VGG16 pour l'authenticité et les réseaux siamois pour la similarité. L'objectif principal était d'améliorer la précision et la fiabilité des systèmes de vérification des signatures tout en tenant compte de l'efficacité temporelle.

Les résultats montrent que le modèle VGG16, avec une régularisation appropriée, a atteint une exactitude de 92,0 % sur l'ensemble de données combiné ICDAR et CEDAR. Comparativement, sans régularisation, les performances du modèle VGG16 étaient nettement inférieures, soulignant l'importance de la régularisation pour éviter le surapprentissage et améliorer la généralisation du modèle.

Concernant les réseaux siamois, les résultats ont montré que, bien que le modèle utilisant la perte de contraste offre une précision légèrement supérieure 94,31% que la méthode de perte BCE proposée 92 ,33%, ils nécessitent plus de temps pour effectuer les prédictions. En comparant les temps d'exécution, on constate que le modèle utilisant la perte d'entropie croisée binaire (BCE) proposée est de loin le plus rapide (seconde par prédiction) par rapport au modèle utilisant la perte de contraste (0,52 ,0,89 seconde par prédiction respectivement.

Une interface graphique sophistiquée a été conçue pour optimiser l'interaction entre l'utilisateur et le système. Cette interface assure non seulement une utilisation intuitive, mais aussi une visualisation détaillée et claire des résultats de la vérification.

L'exploitation des modèles peut être grandement améliorée en intégrant des fonctionnalités avancées, telles que des outils interactifs pour l'analyse des données et des options de personnalisation pour répondre aux besoins spécifiques des utilisateurs.

Dans de futurs travaux, l'utilisation de bases de données plus diversifiées et plus grandes, comprenant des signatures de différentes cultures et langues, ainsi que l'intégration de l'apprentissage semi-supervisé et non supervisé, pourrait exploiter des données non étiquetées, améliorant ainsi la performance du modèle en présence de

données limitées. En affinant les architectures actuelles (VGG16 et le réseau Siamois), il sera possible de réduire les temps d'exécution sans sacrifier la précision.

7 Références

[1] Wang, Z., Muhammat, M., Yadikar, N., Aysa, A., & Ubul, K. (2023). Advances in Offline Handwritten Signature Recognition Research: A Review. IEEE Access, 11, 120222-120236.

[2] Bibi, K., Naz, S., & Rehman, A. (2020). Biometric signature authentication using machine learning techniques: Current trends, challenges and opportunities. *Multimedia Tools and Applications*, *79*(1), 289-340.

[3] Patil, B. V., & Patil, P. R. (2018). An efficient DTW algorithm for online signature verification. In *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)* (pp. 1-5). IEEE.

[4] Parziale, A., Diaz ,M., Ferrer ,MA., Marcelli A (2019) Sm-dtw: stability modulated dynamic time warping for signature verification. Pattern Recogn Lett 121:113–122

[5] Shanker, A. P., & Rajagopalan, A. N. (2007). Off-line signature verification using DTW. *Pattern recognition letters*, *28*(12), 1407-1414.

[6] Kumar,MM, Puhan,N (2014) Offline signature verification using the trace transform. In: 2014 IEEE international advance computing conference (IACC), pp 1066–1070. IEEE

[7] Shanker,AP, Rajagopalan A (2007) Off-line signature verification using dtw. Pattern Recogn Lett 28(12):1407–1414

[8] Kennard, DJ., Barrett, WA., Sederberg, TW (2012) Offline signature verification and forgery detection using a 2-d geometric warping approach. In: 2012 21st international conference on pattern recognition (ICPR), pp 3733–3736. IEEE

[9] Jayadevan, R., Subbaraman, S., & Patil, P. M. (2007). Variance based extraction and hidden Markov model based verification of signatures present on bank cheques. In International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007) (Vol. 2, pp. 451-455). IEEE.

[10] Coetzer ,J., Herbst BM, du Preez JA (2004) Offline signature verification using the discrete radon transform and a hidden markov model. EURASIP Journal on applied signal processing 2004:559–571

[11] Justino , EJ., El Yacoubi, A,. Bortolozzi, F., Sabourin ,R (2000) An off-line signature verification system using hmm and graphometric features. In: Proc. of the 4th international workshop on document analysis systems, pp 211–222. Citeseer

[12] Arunalatha, J. S., Prashanth, C. R., Tejaswi, V., Shaila, K., Raja, K. B., Anvekar, D., ... & Pawan, K. S. (2015). OSPCV: Off-line Signature Verification using Principal Component Variances. *IOSR Journal of Computer Engineering (IOSR-JCE)*, *17*(2015), 08-23.

[13] Prakash, H. N., & Guru, D. S. (2009). Geometric centroids and their relative distances for off-line signature verification. In *2009 10th International Conference on Document Analysis and Recognition* (pp. 121-125). IEEE.

[14] Du ,X., Abdalmageed , W,. Doermann, D (2013) Large-scale signature matching using multi-stage hashing. In: 2013 12th international conference on document analysis and recognition (ICDAR), pp 976–980. IEEE

[15] Chai, J., Zeng, H., Li, A., & Ngai, E. W. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. Machine Learning with Applications, 6, 100134.

[16] Kaur, H., & Kumar, M. (2023). Signature identification and verification techniques: state-of-the-art work. Journal of Ambient Intelligence and Humanized Computing, 14(2), 1027-1045.

[17] Shariatmadari, S., Emadi, S., & Akbari, Y. (2019). Patch-based offline signature verification using one-class hierarchical deep learning. International Journal on Document Analysis and Recognition (IJDAR), 22(4), 375-385.

[18] Mohapatra, R. K., Shaswat, K., & Kedia, S. (2019). Offline handwritten signature verification using CNN inspired by inception V1 architecture. In 2019 Fifth International Conference on Image Information Processing (ICIIP) (pp. 263-267). IEEE.

[20] Wei, P., Li ,H., Hu, P (2019) Inverse discriminative networks for handwritten signature verifcation. In: Proceedings of IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 5757–5765

[21] Foroozandeh, A., Hemmat, A. A., & Rabbani, H. (2020). Offline handwritten signature verification and recognition based on deep transfer learning. In 2020 International Conference on Machine Vision and Image Processing (MVIP) (pp. 1-7). IEEE.

[22] Dey, S., Dutta, A., Toledo, J. I., Ghosh, S. K., Lladós, J., & Pal, U. (2017). Signet: Convolutional siamese network for writer independent offline signature verification. arXiv preprint arXiv:1707.02131.

[23] Jain, S., Khanna, M., & Singh, A. (2021). Comparison among different cnn architectures for signature forgery detection using siamese neural network. In 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 481-486). IEEE.

[26] Haykin, S. (2009). Neural Networks and Learning Machines (3rd Edition). Prentice Hall, 14

[27] Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06) (Vol. 2, pp. 1735-1742). IEEE.

[28] TensorFlow Team. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Google. Retrieved from https://www.tensorflow.org/.

[29] Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools. Retrieved from https://opencv.org/.

[30] Pedregosa, F. (2011). Scikit-learn: Machine learning in python Fabian. Journal of machine learning research, 12, 2825.

[31] Google. (n.d.). Google Colaboratory. Retrieved from https://colab.research.google.com/.

[32] García-Salicetti, S., Dorizzi, B., (2011). ICDAR 2011 Signature Verification Competition (SigComp2011). In Proceedings of the 2011 International Conference on Document Analysis and Recognition (ICDAR), pp. 1473-1477. DOI: 10.1109/ICDAR.2011.299.

[33] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. https://arxiv.org/pdf/1409.1556.pdf

[34] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1993). Signature verification using a" siamese" time delay neural network. Advances in neural information processing systems, 6.

[35] Chopra, S., Hadsell, R., & LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 539-546). IEEE.

[36] Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06) (Vol. 2, pp. 1735-1742). IEEE.

[37] Liu, Z., Peng, J., Guo, X., Chen, S., & Liu, L. (2024). Breast cancer classification method based on improved VGG16 using mammography images. Journal of Radiation Research and Applied Sciences.

[38] Thenuwara, M., & Nagahamulla, H. R. (2017). Offline handwritten signature verification system using random forest classifier. In 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer) (pp. 1-6). IEEE.

[39] Pal, S., Alaei, A., Pal ,U., Blumenstein M (2016) Performance of an of-line signature verifcation method based on texture features on a large Indic-script signature dataset. In: Proceedings of 12th IAPR workshop on document analysis systems (DAS), pp 72–77.

[40] Pushpalatha, K. N., Gautham, A. K., Shashikumar, D. R., ShivaKumar, K. B., & Das, R. (2013). Offline signature verification with random and skilled forgery

detection using polar domain features and multi stage classification-regression model. International Journal of Advanced Science and Technology, 59, 27-40.

[41] Kiani, V., Pourreza, R., & Pourreza, H. R. (2010). Offline signature verification using local radon transform and support vector machines. International journal of Image Processing (IJIP), 3(5).

[42] Arısoy, M. V. (2021). Signature verification using siamese neural network one-shot learning. International Journal of Engineering and Innovative Research, 3(3), 248-260.

[43] Sauvola, J., & Pietikäinen, M. (2000). Adaptive document image binarization. Pattern recognition, 33(2), 225-236.

[44] Kanjanasurat, I., Domepananakorn, N., Archevapanich, T., & Purahong, B. (2022). Comparison of image enhancement techniques and CNN models for COVID-19 classification using chest x-rays images. In 2022 8th International Conference on Engineering, Applied Sciences, and Technology (ICEAST) (pp. 6-9). IEEE.

[45] Gholamalinezhad, H., & Khosravi, H. (2020). Pooling methods in deep neural networks, a review. arXiv preprint arXiv:2009.07485.

[46] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

[47] Shah, AS., Shah, M., Fayaz, M., Wahid, F., Khan, HK., Shah, A (2017) Forensic analysis of offline signatures using multilayer perceptron and random forest. nternational Journal of Database Theory and Application (10) 139–148

[48] Shekar, B. H., Pilar, B., & Sunil Kumar, D. S. (2018). Offline signature verification based on partial sum of second-order taylor series expansion. In Data Analytics and Learning: Proceedings of DAL 2018 (pp. 359-367). Singapore: Springer Singapore.