
MEMOIRE DE FIN D'ETUDE

La reconnaissance automatique de l'écriture manuscrite à l'aide l'intelligence artificielle

Réalisé par : ZINE Mohcine

Encadrant pédagogique : M. BENDRAOU Reda

Juillet 2019

Table des matières

1	Remerciements	5
2	Introduction	7
3	Etat de l'art	11
3.1	Historique IA Education	
3.1.1	Présentation de certaines applications de l'IA sur l'éducation	
3.1.2	Historique	12
3.2	Intelligence artificielle	15
3.2.1	Définition	
3.2.2	L'apprentissage automatique	16
3.2.3	Type d'apprentissage automatique	17
3.3	Processus de la reconnaissance de l'écriture manuscrite	22
3.3.1	Prétraitement	
3.4	Modèle d'entraînement	25
3.4.1	Le modèle caché de Markov	
3.4.2	Les réseaux de neurones récurrents :	27
3.4.3	Long Short-Term Memory Units :	28
3.4.4	Réseaux de neurones convolutifs :	29
3.5	Entraînement de modèle	31
3.5.1	Méthode des moindres carrés :	32
3.5.2	Méthode de l'entropie croisée :	
3.6	Prédiction	33
4	Reconnaissance de l'écriture manuscrite : Cas d'étude sur la base de données IAM-DB	35
4.1	Introduction	
4.2	Préparation des données	36

4.2.1	Description de la base de données IAM-DB	
4.2.2	Préparation du data frame	37
4.2.3	Définition du nombre de classes	
4.2.4	Données d'apprentissage vs données de tests	38
4.2.5	Preprocessing sur les données	
4.3	Processus d'apprentissage	39
4.3.1	Phase de pré-entraînement : Auto-encodeur	
4.3.2	Phase d'entraînement pour la Classification	43
4.4	Conclusion	45
5	Conclusion et Discussion	47

Liste des figures

1	Tableau de l'historique de l'évolution de l'IA	14
2	Représentation de la relation entre l'IA , machine learning et deep learning .	16
3	Historisation de L'IA, machine learning et deep learning	18
4	Caractéristiques des différents types d'apprentissage	
5	Représentation des types de machine learning	21
6	Exemple de binarisation à l'aide de Matlab	23
7	Exemple d'amincissement	24
8	Exemple d'image avec bruit	25
9	Représentation du modèle HMM	26
10	Représentation du modèle RNN	27
11	Comparaison entre RNN et LSTM	28
12	Exemple de réseau neuronal convultif	29
13	Tableau comparatif des modèles LSTM CNN HMM	31
14	Explication des différents types de résultat	33
15	Représentation du processus d'apprentissage automatique	36
16	Exemple de tableau image/label	37
17	Exemple d'image après l'étape de pré-processing	39
18	Distribution de notre jeux de données	
19	Descriptif de l'auto-encodeur	40
20	Architecture du modèle d'apprentissage	41
21	Résultat de la fonction objectif	42
22	Training and validation loss	43
23	Architecture de notre algorithme de classification	44
24	Résultat de la fonction objectif	
25	Exemple	45
26	Présentation de l'accuracy en fonction de la fréquence	46
27	comparaison de notre algorithme et RESNET-34	47

1 Remerciements

Dans un premier temps, je tiens à remercier toute l'équipe pédagogique de l'Université Paris Nanterre et les intervenants professionnels responsable de la formation du master MIAGE agilité des systèmes d'information et e-business, pour avoir assuré la partie théorique de celle-ci.

Je remercie également Mr. BENDRAOU Reda pour son aide précieuse et les conseils qu'il m'a apporté durant la réalisation du mémoire.

Enfin je tiens à remercier toutes les personnes qui ont contribué, par leur disponibilité, leurs conseils avisés et leur attention, à la réalisation de ce document.

2 Introduction

L'intelligence artificielle est une notion apparue dans les années 1950, cette technique permet aux machines d'imiter une forme d'intelligence humaine en s'inspirant du fonctionnement du cerveau humain. Durant la dernière décennie l'IA a connu un tournant, grâce à l'explosion de la puissance de calcul se retrouvant ainsi implémentée dans un nombre grandissant de domaines d'application et passe ainsi d'un classique de science-fiction à une réalité devenue un enjeu scientifique et industriel majeurs.

Malgré l'expansion remarquable de l'utilisation de l'intelligence artificielle, cette dernière a dû faire face aux problèmes de la fiabilité de ses résultats. Cette réticence du public est liée à la complexité des sciences et aux algorithmes qui la définissent. Les personnes n'ayant pas ce bagage technique perçoit l'IA comme une boîte noire et ont du mal à comprendre son fonctionnement et par conséquent à lui faire confiance. De point de vue industriel, on retrouve aussi d'autres contraintes liées au coût élevé requis pour l'implémentation de cette nouvelle technologie et le manque d'expertise, ce qui a fait de l'IA une solution difficile à intégrer dans les entreprises.

Néanmoins ces contraintes ne sont pas insurmontables, avec l'augmentation des puissances de calcul, la facilité de leurs accès grâce aux solutions Cloud, les différentes spécialités proposées par les universités et la montée des événements IA, cette technologie devient de plus en plus accessible.

De nos jours, ils existent plusieurs applications de l'IA qu'on peut classer en quatre domaines phares : Traitement automatique du langage, Traitement des images, Traitement des séquences et l'analyse prédictive des données massives. Dans le traitement d'image, on retrouve par exemple l'amélioration de la qualité de l'image ou encore la transformation des images noir et blanc en couleurs. Mais aussi le modèle très connu de Facebook, DeepFace [1], qui permet la vérification des visages à partir des photos postées sur le réseau social

. Quant au Traitement des séquences, on peut citer la restitution des séquences audio en fichier texte (L'api Google SpeechToText). De plus une chercheuse du MIT [2] a conçu un modèle se basant sur l'analyse des émotions à partir d'une séquence audio permettant ainsi de détecter les personnes dépressives. Finalement, les analyses prédictives et statistiques d'importants volumes de données ont donné lieu à de nombreuses applications, notamment dans la crypto-finance [3] où l'IA est utilisée pour optimiser la gestion du portfolio de la monnaie virtuelle.

A l'ère du 21ème siècle les technologies ne connaissent plus de frontières, les smartphones, ordinateurs portable et tablette ont envahi notre quotidien. Au cours de cette phase, le système éducatif évolue dans un souci d'amélioration, en proposant de nouvelle solution tel que les cours en ligne qui propose des formations spécifiques, des manuels numériques mais aussi des animations qui permettent aux étudiant de comprendre des sujets relativement difficiles grâce à une représentation plus simplifiée [4]. Tous les outils numériques modernes peuvent aussi aider à automatiser facilement les processus internes, améliorer la communication entre les élèves, les enseignants et les parents ou même permettre de personnaliser le parcours étudiant. Cependant la digitalisation de l'éducation doit faire face à certaines contraintes tel que l'élimination des documents papiers, l'automatisation des processus ou encore l'intégration de nouvelles plateformes mobile.

De nos jours l'intelligence artificielle et l'apprentissage automatique sont considérés comme les principaux acteurs à la pointe de la révolution numérique. De nombreux secteurs tels que le système éducatif, les transports, les soins de santé et le marketing ont commencé à adopter la technologie émergente pour améliorer leurs plans d'action. L'enseignement peut bénéficier de cette technologie qui ne cesse d'accroître ses domaines d'application, en améliorer le ciblage des cours, la personnalisation du suivi des étudiants mais aussi faciliter le passage des documents physiques au format numérique.

Il existe actuellement plusieurs systèmes permettant la numérisation de texte manuscrit, ces systèmes s'inscrivent dans le domaine de la reconnaissance optique de caractères (ROC). En effet, ces méthodes ont déjà fait leurs preuves dans le secteur bancaire à travers la lecture des chèques ou des adresses dans le domaine postal mais restent méconnues voire inutilisées dans le système éducatif. De ce fait, ce mémoire propose une étude comparative entre différentes solutions et méthodes de ROC, plus précisément celles basées sur l'apprentissage automatique, appliquées à la numérisation des notes des étudiants et l'aide à la correction des examens. La conclusion de notre étude comparative nous permettra donc de proposer la méthode la plus adaptée au traitement des documents éducatifs en tenant en compte de leurs spécificités, comme par exemple le fait de contenir une combinaison plus variée de caractère et de formules scientifiques.

3 Etat de l'art

3.1 Historique IA Education

3.1.1 Présentation de certaines applications de l'IA sur l'éducation

À ce jour, l'intelligence artificielle est progressivement devenue omniprésente dans de nombreux aspects de notre vie quotidienne. Des capteurs intelligents qui nous aident à réaliser des photos d'une grande qualité, des fonctions de stationnement automatique dans les voitures, ou encore par les assistants personnels dans les smartphones, l'intelligence artificielle s'impose ainsi dans tous les domaines. Par conséquent, l'éducation est l'un des secteurs où l'intelligence artificielle est sur le point d'apporter de grands changements.

L'intelligence artificielle vise à améliorer l'apprentissage des élèves et à fournir un meilleur soutien aux enseignants et aux apprenants. Plusieurs solutions répondent à cet objectif, comme les plateformes d'éducation ludique tel que « kidaptive » ou « osmo » dédiées à l'éducation à travers de jeux théoriques et pratiques. D'autres solutions, comme « Cousrera », « Udacity » et « edX », utilisent l'intelligence artificielle pour mieux cibler les sujets, les cours et les services d'assistance selon chaque profil d'étudiant.

De nombreux projets utilisent l'intelligence informatique pour aider les étudiants et les enseignants à tirer davantage parti de l'expérience éducative. Voici quelques exemples d'applications permettant de définir l'expérience éducative du futur :

- **Prédire la performance des étudiants :** Un des principaux avantages de l'apprentissage automatique est sa capacité à prédire les performances. En comprenant le profil et les habitudes de chaque élève, la technologie peut identifier les faiblesses et suggérer des moyens d'amélioration adaptés, tels que des tests pratiques supplémentaires
- **Étudiants test :** L'apprentissage automatique peut aider à s'éloigner des tests standardisés selon Rose Luckin [5]. L'évaluation basée sur l'intelligence artificielle pourra

fournir aux enseignants, aux élèves et aux parents un retour constant sur la manière dont les élèves apprennent, des moyens de soutien et des indicateurs précis sur leurs progrès dans la réalisation de leurs objectifs d'apprentissage

- **Améliorer le système de notation :** L'apprentissage automatique peut également attribuer une note équitable aux élèves en supprimant les préjugés humains. Bien que l'IA ait déjà commencé à noter les examens à choix multiples, nous commençons à remarquer que l'apprentissage automatique peut également analyser et traiter l'écriture manuscrite et pourra ainsi évaluer les feuilles d'examens manuscrites des différents étudiant
- **Fournir un apprentissage personnalisé :** L'apprentissage automatique permet également de personnaliser l'apprentissage pour chaque élève de la classe. Les enseignants pourront utiliser les données pour identifier les élèves ayant besoin d'une assistance supplémentaire. La technologie peut également suggérer des outils d'apprentissage adaptés au profil de chaque élève.

3.1.2 Historique

Au fur et à mesure que nous découvrons des moyens d'utiliser l'apprentissage automatique en classe, nous trouverons des solution pour améliorer l'éducation. Parmi ces nombreux débouchés on retrouve la reconnaissance de l'écriture qui est apparu en 1929 par Tauschek. [6], une solution sous forme de mécanisme équipé d'une source lumineuse et de filtres des caractères. C'est un mécanisme qui permet la reconnaissance de l'écriture dactylographiée. Cependant, ce système est difficilement adaptable à la variabilité de l'écriture manuscrite. Avec l'apparition des premiers ordinateurs en 1950, les premiers systèmes de reconnaissance ont permis le transfert de la solution mécanique vers une solution électronique mais qui se base sur le même système de filtres pour l'identification des caractères ce qui reste inadapté pour l'écriture manuscrite.

Entre 1950 et 1986, de nouveaux modèles statistiques sont introduits pour remplacer ces filtres. Ces modèles statistiques sont plus adaptés à l'écriture manuscrite mais ils nécessitent une étape d'extraction de caractéristiques afin de construire des vecteurs de caractérisation de l'écriture. De plus, ces systèmes nécessitent la détection de début et fin d'un caractère ce qui complique la reconnaissance des caractères manuscrit.

La période entre 1986 et 2006 a vu l'apparition des modèles de markov cachés [7] qui a connu des résultats satisfaisants dans le domaine de la reconnaissance de la parole. Ces modèles ont été appliqués avec succès à la reconnaissance de l'écriture puisqu'ils possèdent plusieurs avantages tel que la capacité à segmenter les images, ce qui les catégorisent en tant que premier classifieur dynamique. Mais ces modèles sont génératifs avec une faible capacité de discrimination, ce qui sera, par la suite, compensé par la combinaison avec les modèles discriminants. De ce fait, de nouveaux modèles hybrides ont été proposés et sont potentiellement déclinable avec différents classifieurs tels que les réseaux profonds et les réseaux de neurones.

Période	Méthode générale	Remarque
1950	Le mot a été modélisé comme étant composé de blocs définis par les coordonnées de leurs sommets et par la spécification d'informations de bord.	La qualité était fortement dépendante de l'image d'origine.
1960	Systèmes de segmentation et d'interprétation intégrés.	Cette époque a vu l'amélioration rapide de l'acquisition d'images avec le développement d'équipements de qualité.
1970	Développement de calcul, algorithmes et niveau d'implémentation.	La recherche a principalement consisté en la recherche de contours, la croissance et la segmentation de régions et des processus de niveau supérieur tels que la reconnaissance de forme et le raisonnement.
1980	Une nouvelle direction de la vision par ordinateur a émergé sous la forme d'une vision active. La perception visuelle a été traitée comme un processus actif car le système de vision s'adapte en permanence à un environnement en mutation. Par exemple : explorer et rechercher des informations.	La théorie de la décision est le cadre de la fusion d'informations et du contrôle de différents capteurs.
1990	Des algorithmes plus efficaces : correspondance de programmation dynamique, modèles de markov cachés (HMM), réseaux de neurones (NN).	Un regain d'intérêt s'est manifesté avec la montée en puissance des applications bancaires et postales, des ordinateurs portables, avec de nouveaux systèmes d'acquisition plus adaptés, tels que des scanners, des blocs-notes, des papiers électroniques, etc.
2000	La combinaison ou la coopération de plusieurs identifiants indépendants, l'utilisation de lexiques ou dictionnaires et de modèles de langage.	À cette époque, le post-traitement a été suggéré pour améliorer l'efficacité globale du système.

Figure 1: Tableau de l'historique de l'évolution de l'IA

3.2 Intelligence artificielle

3.2.1 Définition

La perspective de créer des ordinateurs intelligents fascine beaucoup de gens depuis la création du premier ordinateur. Les premières indications en matière d'Intelligence Artificielle datent bien avant cela. Mais qu'entendons-nous par intelligence artificielle si même le terme intelligence demeure difficile à définir ? La définition et la signification précises de l'intelligence artificielle, font l'objet de nombreuses discussions et ont créé beaucoup de confusions. Le dictionnaire « The New International Webster's Comprehensive Dictionary of the English Language », par exemple, donne quatre définitions de l'intelligence artificielle :

- Un domaine d'étude dans le domaine de l'informatique. L'intelligence artificielle s'intéresse au développement d'ordinateurs capables de s'engager dans des processus de pensée semblables à ceux de l'homme, tels que l'apprentissage, le raisonnement et l'autocorrection.
- Le concept selon lequel les machines peuvent être améliorées pour prendre en compte certaines capacités normalement considérées comme l'intelligence humaine, telles que l'apprentissage, l'adaptation, l'autocorrection, etc.
- L'extension de l'intelligence humaine grâce à l'utilisation des ordinateurs, tout comme jadis le pouvoir physique s'est étendu à l'utilisation d'outils mécaniques.
- Dans un sens restreint, l'étude des techniques permettant d'utiliser les ordinateurs plus efficacement grâce à des techniques de programmation améliorées.

Les définitions ont également changé au fil du temps, en raison des développements rapides. Les définitions les plus récentes parlent d'« imitation du comportement humain intelligent », ce qui est déjà une définition beaucoup plus forte.

Depuis un certain temps déjà, la communauté de l'intelligence artificielle tente d'imiter un comportement intelligent avec des programmes informatiques. Ce n'est pas une tâche fa-

cile, car un programme informatique doit être capable de faire beaucoup de choses différentes pour être appelé intelligent.

Au lieu de regarder une définition générale de l'intelligence artificielle, on peut aussi se limiter à la définition de systèmes artificiellement intelligents. Il existe de nombreuses définitions, mais la plupart d'entre elles peuvent être classées dans les quatre catégories suivantes : des systèmes qui pensent comme des humains - des systèmes agissant comme des humains - des systèmes qui pensent rationnellement - des systèmes qui agissent de manière rationnelle.

3.2.2 L'apprentissage automatique

L'apprentissage automatique est un sous-ensemble de l'IA. Autrement dit, tout apprentissage automatique compte comme une intelligence artificielle, mais pas toutes les IA représentent un apprentissage automatique.

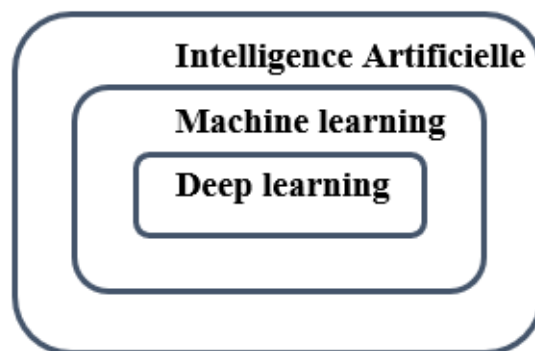


Figure 2: Représentation de la relation entre l'IA , machine learning et deep learning

Un aspect qui distingue l'apprentissage automatique des graphes de connaissances et des systèmes experts est sa capacité à évoluer lorsqu'il est exposé à davantage de données ; c'est-à-dire que l'apprentissage automatique est généralisable, dynamique et ne nécessite aucune intervention humaine pour effectuer certains changements. Cela le rend moins rigide et moins tributaire d'experts humains.

Définition selon Tom Mitchell [8] de l'apprentissage automatique : Un programme informatique sensé apprendre de l'expérience E en ce qui concerne certaines classes de tâches T et de mesurer la performance P si sa performance aux tâches en T , mesurée par P , s'améliore avec l'expérience E .

En 1959, Arthur Samuel [9], l'un des pionniers de l'apprentissage automatique, définissait l'apprentissage automatique comme un domaine d'étude permettant aux ordinateurs d'apprendre sans être explicitement programmé. Les programmes d'apprentissage automatique, en un sens, s'adaptent en fonction des données auxquelles ils sont exposés.

3.2.3 Type d'apprentissage automatique

La partie « apprentissage » de l'apprentissage automatique signifie que les algorithmes essaient d'optimiser sur une certaine dimension ; c'est-à-dire qu'ils essaient généralement de minimiser les erreurs ou d'optimiser la probabilité que leurs prédictions soient vraies. Ce qui définit les fonctions suivantes : une fonction d'erreur, une fonction de coût ou fonction objectif. Quand quelqu'un dit qu'il travaille avec un algorithme d'apprentissage automatique, vous pouvez accéder à l'essentiel de sa valeur à travers sa fonction objective.

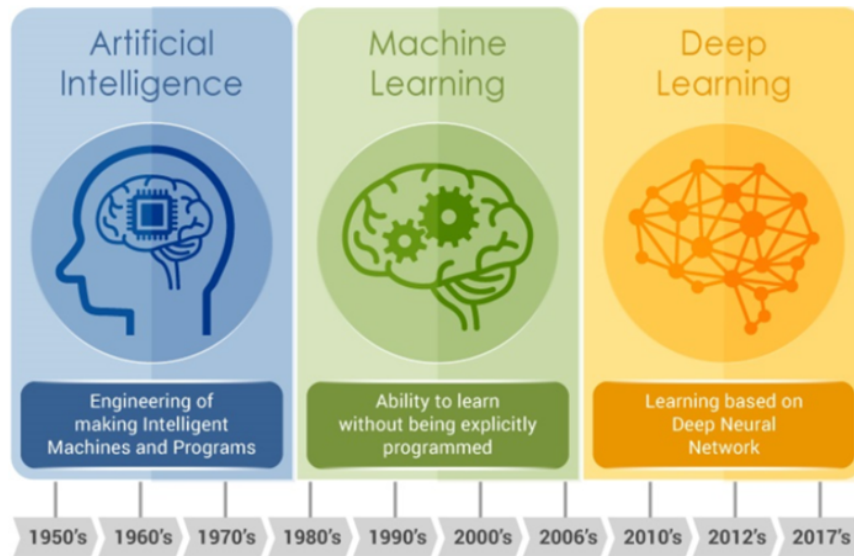


Figure 3: Historisation de L'IA, machine learning et deep learning

Dans l'apprentissage automatique, il existe différents modèles qui appartiennent généralement à 3 catégories différentes : apprentissage supervisé, apprentissage non supervisé et apprentissage par renforcement.

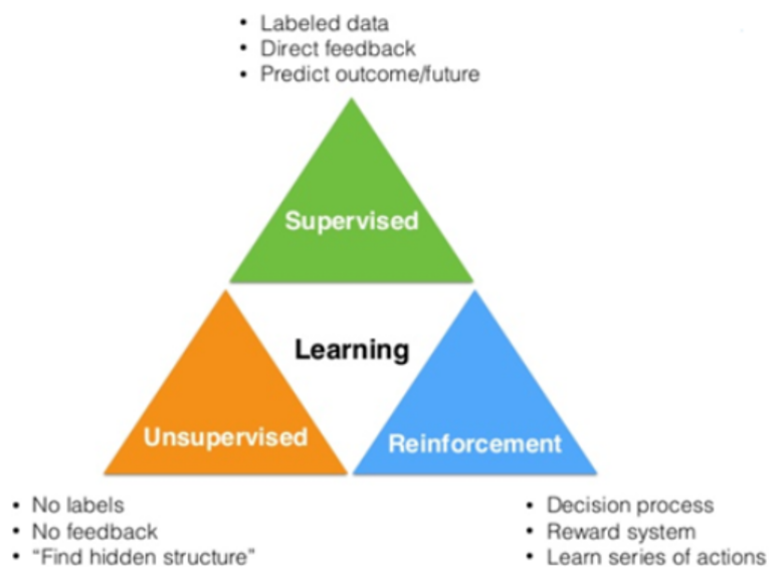


Figure 4: Caractéristiques des différents types d'apprentissage

Apprentissage supervisé : Il implique une étiquette de sortie associée à chaque instance du jeu de données d'entrée. Cette sortie peut être discrète / catégorique (vert, oiseau, spam...), dans ce cas on parle de classification, ou continue / réelle quand il s'agit d'un problème de régression. Actuellement, la forme d'apprentissage automatisé la plus utilisée est l'apprentissage supervisé. Les données qu'on cherche à modéliser ont un sous échantillon ayant des étiquettes connues en sortie. Cela implique un superviseur plus compétent que le réseau de neurones lui-même. Par exemple, le superviseur fournit des exemples de données dont il connaît déjà les réponses. Le superviseur guide le système en balisant la sortie. Par exemple, un système d'apprentissage automatique supervisé peut savoir quels courriels sont des "spam" et lesquels ne sont pas des "spam". L'algorithme serait d'abord entraîné avec l'ensemble de données d'entrée disponibles (de l'ordre million d'e-mails) déjà étiquetés avec cette classification binaire afin de paramétrer un système d'apprentissage automatique capable d'identifier les caractéristiques ou les paramètres du courrier électronique "spam" et à le distinguer de ceux de "non spam". Des modèles tels que les régressions linéaires ou logistiques et la classification des arbres de décision entrent dans cette catégorie d'apprentissage.

Régression : Il s'agit d'un type de problème pour lequel nous devons prévoir valeurs de réponse continue. Quelques exemples sont quel est le prix de la maison dans une ville spécifique avec 3 chambres et plus de 2000 mètres carrés ? Prédire les résultats financiers, les cours des actions ou le nombre total de courses pouvant être possibles dans une partie de cricket. Vous disposez d'un ensemble de données et de sorties (apprentissage supervisé) et votre algorithme prédit le résultat en fonction d'une fonction d'adaptation.

Classification : Vous devez classer une observation donnée dans une catégorie définie. Quelques exemples : Prédire si un email donné est un spam ou non ? Une particule détectée est-elle un boson de Higgs ou une particule sous-atomique normale ? Assigner un article d'actualité à un groupe comme le sport, la météo ou la science. Est-ce qu'un chat apparaît dans cette photo ou non ? Détecter la fraude ou évaluer le risque de fraude ou d'assurance

par écrit.

Apprentissage non supervisé : Il s'agit d'un apprentissage de type « sans aide » lorsque vos données ne comportent généralement pas d'étiquettes de sortie connues ni de boucle de retour. Ceci est utile lorsqu'il n'y a pas d'exemple de jeu de données avec des réponses connues et que vous recherchez un modèle caché. Dans ce cas, le regroupement, c'est-à-dire la division d'un ensemble d'éléments en groupes selon un modèle inconnu, est effectué sur la base des ensembles de données existants. Le système doit se comprendre à partir du jeu de données que nous fournissons. En général, l'apprentissage non supervisé est un peu difficile à mettre en œuvre et n'est donc pas utilisé aussi largement que l'apprentissage supervisé. Les types les plus populaires sont le clustering et l'association comme ci-dessous.

Clustering : Il s'agit d'un type de problème d'apprentissage non supervisé dans lequel nous regroupons des choses similaires. En voici quelques exemples : À partir d'articles de presse ou de livres, regroupez-les en différents types de thèmes. Étant donné un ensemble de citation, regroupez-les en fonction du contenu de citation. Peut aussi être utilisé pour la politique, la santé, les achats, l'immobilier etc.

Association : une règle d'association vous permet de découvrir les règles exactes décrivant les grandes parties de vos données. Nous pouvons le rencontrer lorsque nous recevons une recommandation de livre ou de film basée sur des achats ou des recherches antérieurs. Ces algorithmes sont également utilisés pour l'analyse du panier de marché à l'aide de nos données d'achat en ligne ou hors ligne de détaillants (points de vente).

Apprentissage par renforcement (RL) : Maintenant, si on identifie l'apprentissage automatique à l'apprentissage d'un petit enfant pour identifier les objets, au lieu de dire à l'enfant quel jouet placer dans quelle boîte, vous donnez à l'enfant une récompense quand il fait le bon choix ou vous faites une pénalisation quand il fait la mauvaise action. De façon

rapide, après quelques itérations, l'enfant apprend quels jouets doivent être placés dans quelle boîte - c'est ce que l'on appelle l'apprentissage par renforcement. Les systèmes sont formés en recevant des récompenses ou des punitions virtuelles, essentiellement par apprentissage par essais et erreurs.

Cette stratégie repose sur l'observation, les essais et les erreurs pour atteindre les objectifs ou maximiser les récompenses. L'agent prend une décision en observant son environnement. Si l'observation est négative, l'algorithme ajuste ses poids pour pouvoir prendre une décision différente requise la prochaine fois. On peut compter l'apprentissage par renforcement dans l'apprentissage en profondeur (deep learning) sur la base du nombre de nœuds cachés et de la complexité des algorithmes. Les algorithmes d'apprentissage par renforcement essaient de trouver les meilleurs moyens de gagner la plus grande récompense. Les récompenses peuvent être gagner une partie, gagner plus d'argent ou vaincre d'autres adversaires. Ils présentent des résultats de pointe sur des tâches très humaines.

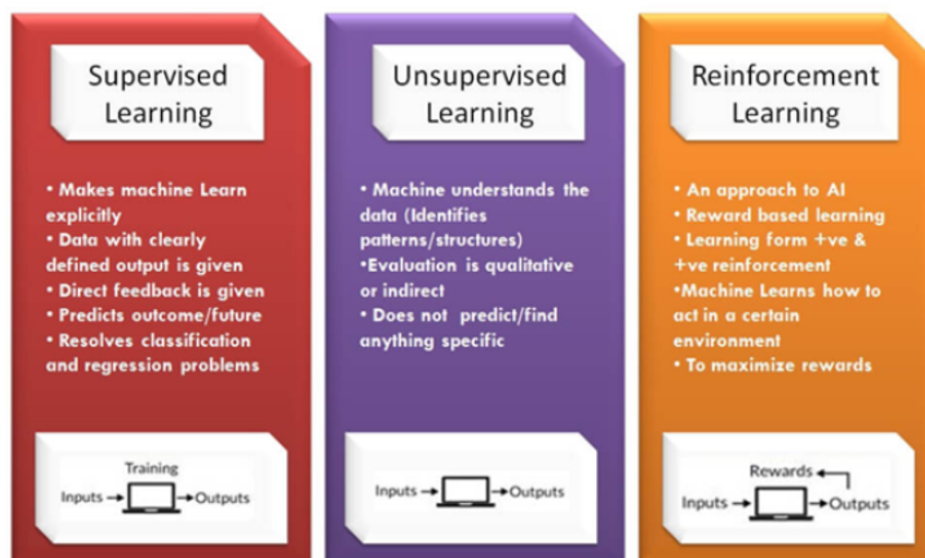


Figure 5: Représentation des types de machine learning

3.3 Processus de la reconnaissance de l'écriture manuscrite

Ce mémoire s'inscrit sous le thème de la recherche exploratoire et descriptive. Le but de cette recherche est d'identifier les informations et les caractéristiques utiles à la compréhension de notre problématique, mais aussi l'explication et la documentation sur le sujet pour ainsi tester la faisabilité de la solution.

Afin de mieux répondre à notre problématique, il est nécessaire de bien comprendre le processus de la reconnaissance de l'écriture manuscrite à travers l'apprentissage automatique. Une problématique qu'on peut définir comme étant un problème de classification des images afin d'identifier le mot présent dans l'image et ainsi permettre la numérisation du texte manuscrit.

Dans cette optique, après avoir défini quelques notions de base, nous allons présenter notre modèle proposé pour la reconnaissance de l'écriture manuscrite.

3.3.1 Prétraitement

Le processus d'amélioration de l'image, pour préparer la phase d'apprentissage du modèle, est appelé prétraitement. Le prétraitement représente une étape majeure dans le système de reconnaissance de l'écriture manuscrite, car elle permettra une convergence plus rapide et précise du modèle de classification.

Le bruit dans une image de document est dû à des pages mal photocopiées. Les images initiales que nous avons tendance à utiliser contiennent des informations qui ne sont pas nécessaires pour l'extraction des informations (e.g. les entêtes, les logos...). Il est à noter aussi que les images numérisées peuvent également contenir des bruits.

Donc, les bruits et les informations indésirables doivent être supprimés de l'image. Le prétraitement peut être divisé en une liste de traitements qui sont :

- **Binarisation** : C'est la conversion de l'image en niveaux de gris en noir et blanc. Tout d'abord, l'image RVB (Rouge, vert, bleu) d'origine doit être convertie en niveaux de gris, puis l'image est convertie en image noir et blanc. La plupart des packages de reconnaissance de l'écriture manuscrite fonctionnent sur les images binarisées qui sont également appelées bi-niveau ou à deux niveaux. La conversion est possible car les valeurs supérieures ou égales à un seuil prédéfini sont blanches et les valeurs inférieures à ce seuil sont noires. La méthode d'Otsu [10] par exemple est utilisée pour réaliser un seuil basé sur un cluster, c'est-à-dire d'une image à niveaux de gris à une image binaire.

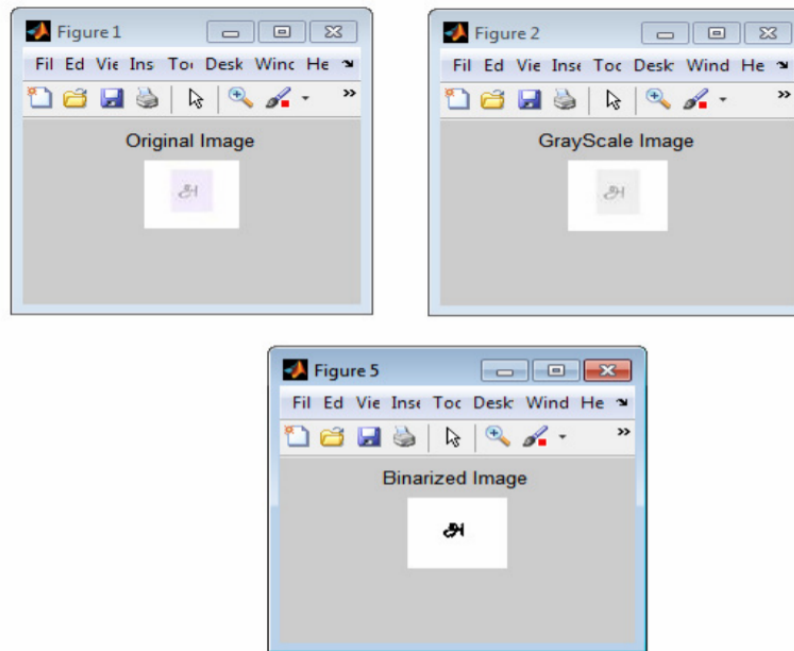


Figure 6: Exemple de binarisation à l'aide de Matlab

- **Normalisation** : Le processus de modification de la valeur d'intensité du pixel dans la plage de $[0,1]$ est appelé normalisation en traitement d'image. Mais aussi la conversion d'images de différentes dimensions en dimensions fixes s'appelle également normalisation ce qui permet la comparaison entre des images de même taille.
- **Amincissement** : C'est un prétraitement qui permet d'obtenir une image d'une

largeur de pixel unique permettant de reconnaître facilement le caractère manuscrit. Il est appliqué de manière répétée permettant ainsi des représentations linéaires à l'échelle du pixel des caractères de l'image.



Figure 7: Exemple d'amincissement

- **Denoising :** Les images numériques sont sujettes à divers types de bruit. Le bruit peut être introduit de plusieurs manières dans une image, lors du scanne ou la numérisation de l'image ou encore lors de la transmission de cette dernière. Le « denoising » [11] est le processus permettant l'élimination de ces perturbations pour ne garder que les informations nécessaires à la reconnaissance du texte.

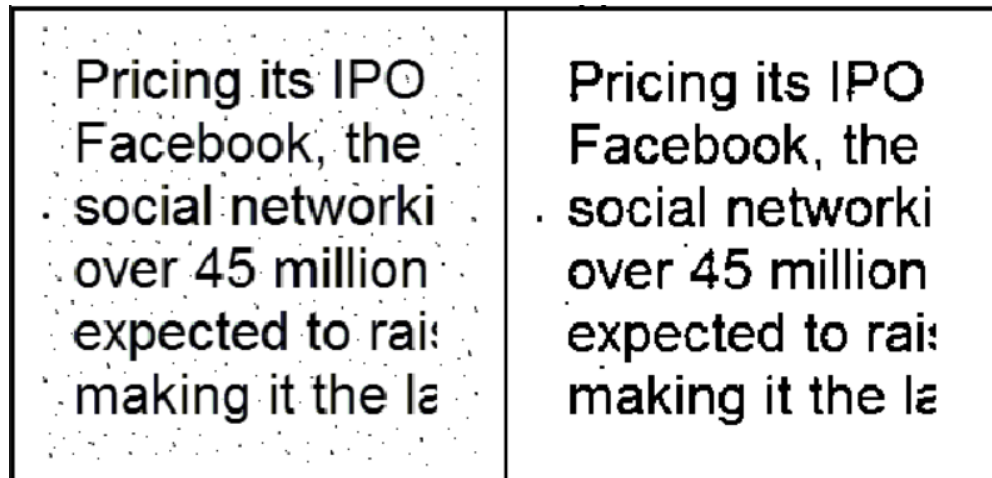


Figure 8: Exemple d'image avec bruit

3.4 Modèle d'entraînement

L'apprentissage automatique est étroitement lié aux statistiques mathématiques, pour la reconnaissance de l'écriture manuscrite ils existent plusieurs algorithmes définissant différents modèles. Nous nous intéressons dans notre mémoire aux modèles d'entraînements utiles à la classification de nos images de mots, et parmi les plus utilisés on retrouve :

3.4.1 Le modèle caché de Markov

: Le problème de la reconnaissance manuscrite des lignes de texte est maintenant largement abordé avec des modèles de Markov cachés (Plötz et Fink, 2009). Ces modèles sont conçus pour traiter des données séquentielles, avec des états cachés émettant des observations. Il existe des techniques bien connues pour calculer la probabilité d'une séquence d'observation à partir d'un modèle, trouver la séquence d'états la plus susceptible d'avoir produit une séquence observée, et trouver les paramètres du modèle pour maximiser la probabilité d'observer une séquence. Le troisième problème est l'apprentissage réel de ces modèles, tandis que le second permet de décoder une séquence d'observation. Dans ce paradigme, les personnages sont chacun représentés par un modèle de Markov caché. Une

simple concaténation de ceux-ci produit des modèles de mots. Les avantages sont doubles :

- à partir de quelques modèles de caractères (environ une centaine), nous pouvons construire des modèles de mots pour, potentiellement n'importe quel mot du langage. Ainsi, cette méthode est beaucoup plus évolutive face aux problèmes de vocabulaire volumineux que les systèmes essayant de modéliser chaque mot séparément
- la reconnaissance d'un mot à partir de modèles de caractères ne nécessite pas de segmentation préalable du mot image en caractères. Comme le mot modèle est un modèle de Markov caché, la segmentation en caractères est un sous-produit de la procédure de décodage, qui consiste à rechercher la séquence d'états la plus probable
- le deuxième point concerne le problème de la reconnaissance de ligne; il n'est pas nécessaire de scinder la ligne de texte en mots pour reconnaître les séquences de mots
- la reconnaissance automatique de la parole est un exemple célèbre d'application réussie des HMM. Ce domaine possède une expérience de plus de 25 ans de ces systèmes et de nombreux progrès peuvent être transférés à la reconnaissance de l'écriture manuscrite

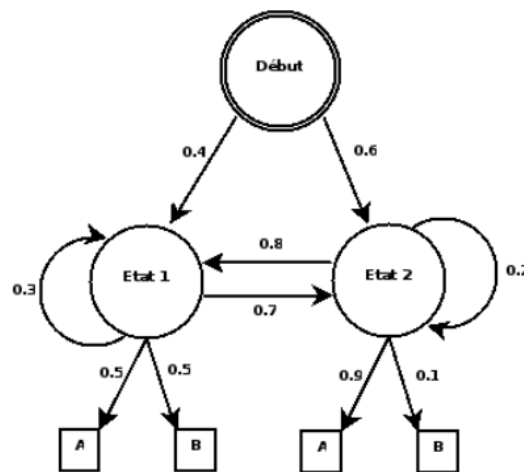


Figure 9: Représentation du modèle HMM

Au cours des quinze dernières années, les HMM sont devenus le modèle standard de reconnaissance de l'écriture manuscrite. Un HMM est construit pour chaque caractère, ce

qui permet de concaténer ces modèles pour obtenir des modèles de mots ou de phrases. La reconnaissance d'une ligne ne nécessite donc pas de segmentation préalable en caractères.

3.4.2 Les réseaux de neurones récurrents :

Les réseaux de neurones récurrents (RNN) sont des réseaux avec une notion d'état interne, évoluant dans le temps, réalisés par des connexions récurrentes. Les réseaux de Hopfield (Hopfield, 1982) constituent une forme précoce de réseau de neurones récurrents, où la récurrence est utilisée pour trouver un état stable plutôt que des séquences temporelles de processus. Saul Jordan (1990); Elman (1990) a proposé des architectures de réseau de neurones pour des séquences de vecteurs d'entrée. Ces réseaux contiennent une couche cachée, ainsi qu'une couche de contexte séparée, implémentant une mémoire dans le réseau.

Dans sa forme la plus simple, un RNN est un MLP - c'est-à-dire que les neurones sont organisés en différentes couches - avec des couches récurrentes. Une couche récurrente reçoit non seulement les entrées de la couche précédente, mais aussi d'elle-même, comme illustré ci-dessous pour l'exemple d'une seule couche.



Figure 10: Représentation du modèle RMM

Robinson (1994) a proposé des réseaux de neurones récurrents simples pour la reconnais-

sance de la parole, et Senior (1994); Lee et Kim (1995); Senior Robinson (1998) ont appliqué des réseaux de neurones récurrents simples à la reconnaissance de l'écriture manuscrite.

3.4.3 Long Short-Term Memory Units :

Dans les RNN, le problème du gradient en voie de disparition empêche le réseau d'apprendre les dépendances à long terme. Hochreiter et Schmidhuber (1997) ont proposé des neurones récurrents améliorés appelés LSTM unités de mémoire à court terme. En mode LSTM, le flux d'informations est contrôlé par un système de contrôle, mettant à l'échelle les informations d'entrée, l'activation de la sortie et la contribution de l'état interne du système à l'étape actuel, en fonction des informations d'entrée et des données récurrentes l'état interne de la cellule.

Une cellule LSTM est présentée à la figure ci-dessous et comparée à un neurone récurrent de base. L'entrée de la cellule et toutes les portes reçoivent l'activation de la couche inférieure et de la couche au pas de temps précédent.

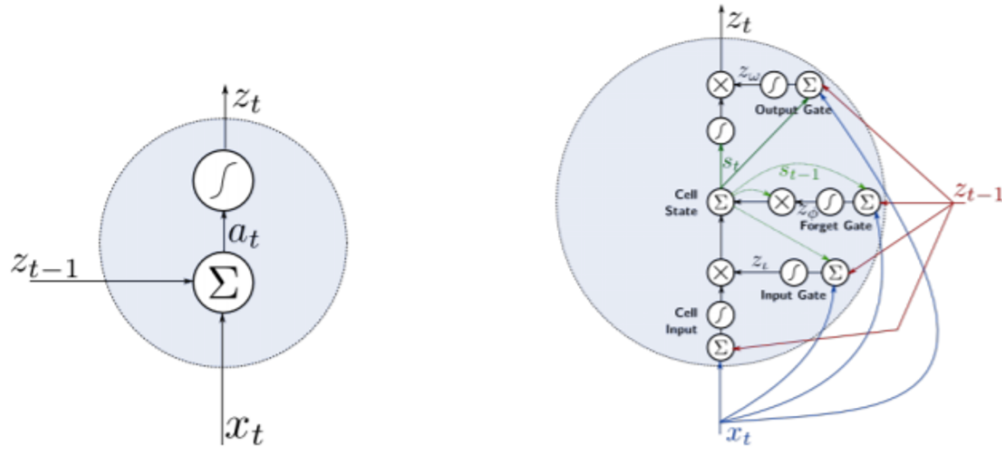


Figure 11: Comparaison entre RNN et LSTM

Les cellules LSTM offrent des capacités qui manquent aux RNN standard, telles que

l'apprentissage de grammaires simples ou la composition musicale (Eck et Schmidhuber, 2002)[12]. Les RNN LSTM ont été appliqués avec succès à la reconnaissance de phonèmes et à la parole (Graves et al., 2013a) [13], ainsi qu'à la reconnaissance de l'écriture manuscrite.

3.4.4 Réseaux de neurones convolutifs :

Les réseaux de neurones convolutifs (CNN, LeCun et al. (1989) [14] ont une structure similaire à celle des MLP [15], mais chaque couche n'est pas entièrement connectée à la précédente. Ils implémentent la notion de récepteurs locaux, via des connexions locales et un partage de poids.

L'entrée est l'image bidimensionnelle et les neurones des couches cachées sont organisés en cartes bidimensionnelles, chacune recherchant une seule caractéristique. Chaque neurone d'une carte est uniquement connecté à un petit voisinage dans les cartes précédentes (ou image d'entrée), avec les mêmes poids de connexion que les autres neurones de la carte (Figure 12). On peut l'interpréter comme des convolutions de l'image avec des filtres pouvant être entraînés.

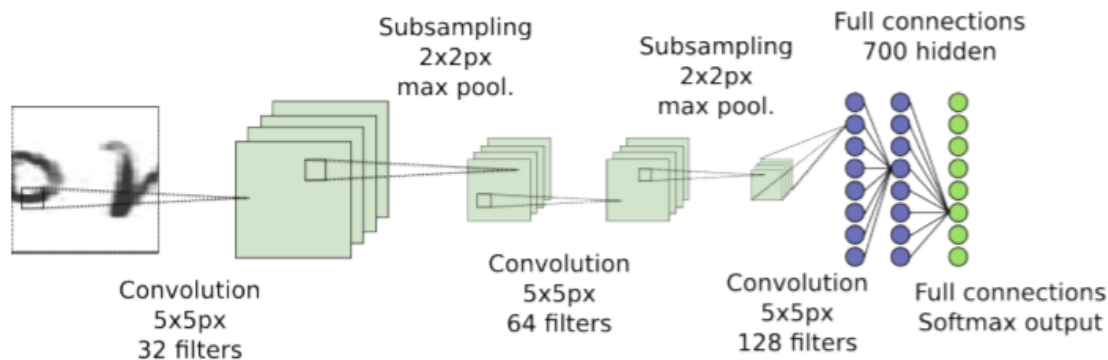


Figure 2.6: Example of Convolutional Neural Network.

Figure 12: Exemple de réseau neuronal convultif

Les couches de convolution sont souvent suivies par des couches de sous-échantillonnage,

afin de limiter la taille du réseau et de mettre en œuvre une certaine invariance aux distortions. L'opération de sous-échantillonnage peut être un max-pooling [16] ou une moyenne sur de petits quartiers. Habituellement, quelques caractéristiques (ou cartes, ou filtres pouvant être entraînés) sont extraites dans les couches inférieures, et de plus en plus, dans les couches supérieures, tandis que les dimensions des cartes diminuent (par exemple dans LeNet-5 de LeCun et al. (1998) [17]).

Les CNN conçus pour la classification peuvent facilement être étendus à des images de taille variable, produisant ainsi des séquences de prédictions, grâce aux connexions locales et aux poids partagés. Une telle structure s'appelle Réseau de neurones à déplacement d'espace. Enfin, les aspects des CNN peuvent également être mis en œuvre dans une architecture plus complexe que les MLP, comme par exemple dans les RNN multidimensionnels (MDRNN, Graves et al. (2007)).

Les CNN sont populaires et obtiennent d'excellents résultats dans diverses tâches (par exemple, la reconnaissance d'objets et la segmentation d'images. Ils ont également récemment été appliqués à la reconnaissance de la parole (AbdelHamid et al., 2012). Ils étaient initialement utilisés pour reconnaître les chiffres manuscrits (LeCun et al., 1989), puis aux textes manuscrits. Plus récemment, nous les avons combinées avec des HMM pour la reconnaissance de mots dans des bases de données publiques (Bluche et al., 2013a, b).

Le tableau ci-dessous présente un résumé et les différentes caractéristiques des modèles définit précédemment :

Modèle	Domaines d'applications	Types de données	Architectures	Caractéristiques apprises par le modèle
HMM	Séries temporelles,	Données séquentielles	Un réseau bayésien connectant chaque entité basique de la séquence (chaîne de caractères)	La relation temporelle entre l'état passé (n-1) et actuel (n). (Une mémorisation courte)
LSTM	Traitement de langage naturel, Traitement d'images, Séries temporelles	Données séquentielles, Textes et images	Un cas particulier du HMM en définissant Un réseau de neurones récurrents possédant une couche modélisant les états cachés du modèle séquentiel.	La relation temporelle entre les états passés (n-k, ..., n-1) et actuel (n). (Une mémorisation à long terme)
CNN	Traitement d'images. Traitement de langage naturel.	Données textes et images	Un réseau de convolution permettant d'effectuer des produits de convolution entre les différentes entités de base de la donnée d'entrée et ainsi cartographier les corrélations entre ces différentes entités.	La relation spatiale entre les différentes entités d'entrées (e.g. la répartition de la couleur dans les pixels d'une image, la succession de caractères dans un mot...)

Figure 13: Tableau comparatif des modèles LSTM CNN HMM

3.5 Entraînement de modèle

Après avoir défini l'algorithme d'apprentissage, on obtient une fonction hypothèse qui nous permet de classer nos images d'entrées selon le mot qu'elle contient. Cette fonction hypothèse dépend de plusieurs paramètres, ces paramètres aussi appelés poids sont susceptibles d'être variés afin d'optimiser notre fonction qu'on appelle fonction de coût. Aussi appelée fonction objective, elle définit l'erreur résultante de la comparaison entre le résultat de la fonction hypothèse et la valeur réelle on peut citer quelques exemples des algorithmes per-

mettant de minimiser la fonction objective :

3.5.1 Méthode des moindres carrés :

On considère un ensemble de test avec M comme le nombre d'occurrences et une fonction de coût J avec θ_0, θ_1 comme paramètres et une fonction hypothèse de format $h_\theta(x) = \theta_0 + \theta_1 x$. On considère aussi dans un contexte de machine learning, x^i représentant les "variables d'entrée" et y^i les "sorties" de la fonction. On peut alors considérer :

$$J(\theta_0, \theta_1) = \frac{1}{2M} \sum_{i=1}^M (h_\theta(x^i) - y^i)^2$$

3.5.2 Méthode de l'entropie croisée :

En théorie de l'information, l'entropie croisée entre deux lois de probabilité mesure le nombre de bits moyen nécessaires pour identifier un événement issu de l'« ensemble des événements » - encore appelé tribu en mathématique – sur l'univers , si la distribution des événements est basée sur une loi de probabilité q , relativement à une distribution de référence p .

L'entropie croisée pour deux distributions p et q sur le même espace probabilisé est défini de la façon suivante :

$$H(p, q) = E_p[-\log q] = H(p) + D_{KL}(p||q)$$

Où $H(p)$ est l'entropie de p , et $D_{KL}(p||q)$ est la divergence de kullback-Leibler entre q et p .

Pour p et q discrets cela signifie :

$$H(p, q) = - \sum_x p(x) \log q(x)$$

La formule est analogue pour des variables aléatoires continues :

$$- \int_x p(x) \log q(x) dx$$

3.6 Prédiction

Dans cette étape, après la définition de notre algorithme et la phase d'apprentissage il est nécessaire d'évaluer ce dernier. L'évaluation de notre modèle se fait sur les données de test à travers la performance de la classification. Dans cette partie nous présenterons différents types de métrique d'évaluation.

Afin de définir les métriques d'évaluation il est nécessaire de bien connaître les différents états des résultats de notre classification ce qui peut être résumé dans le tableau suivant :

		Classe prédite	
		Classe : oui	Classe : non
Classe réelle	Classe : oui	Vrai positive	Faux positive
	Classe : non	Faux négative	Vrai négative

Figure 14: Explication des différents types de résultat

Accuracy : est la mesure de performance la plus intuitive et il s'agit simplement d'un rapport entre les observations correctement prédites et les observations totales. On peut penser que, si nous avons une grande accuracy, notre modèle est le meilleur. Oui,

l'accuracy est une excellente mesure, mais uniquement lorsque vous disposez de jeux de données symétriques dans lesquels les valeurs de faux positif et de faux négatif sont presque identiques.

Precision : est le rapport entre les observations positives correctement prédites et le total des observations positives prédites. La haute précision concerne le faible taux de faux positifs.

Recall : est le rapport entre les observations positives correctement prédites et toutes les observations de la classe réelle « oui ».

F1 score : est la moyenne pondérée de la précision et du rappel. Par conséquent, ce score prend en compte à la fois les faux positifs et les faux négatifs. Intuitivement, il n'est pas aussi facile à comprendre que l'exactitude, mais F1 score est généralement plus utile, en particulier si la répartition des classes est inégale. La précision fonctionne mieux si les faux positifs et les faux négatifs ont un coût similaire. Si le coût des faux positifs et des faux négatifs est très différent, il vaut mieux regarder à la fois Precision et Recall.

4 Reconnaissance de l'écriture manuscrite : Cas d'étude sur la base de données IAM-DB

4.1 Introduction

Le système éducatif de nos jours compte parmi ses supports plusieurs applications informatiques qui permettent de faciliter l'apprentissage des étudiants mais aussi aider les enseignants dans leur travail quotidien. Parmi les nouvelles technologies on retrouve l'intelligence artificielle, cet outil qui ne cesse d'étendre les domaines de son application peut permettre l'archivage des documents manuscrits, automatiser la correction des examens ou encore numériser les notes des étudiants à travers la reconnaissance de l'écriture manuscrite.

Les outils de reconnaissance de l'écriture manuscrite ont connu des évolutions importantes à travers les années. Avec l'ascension de l'utilisation l'intelligence artificielle on retrouve de plus en plus d'applications qui s'inscrivent dans le domaine de la reconnaissance de l'écriture manuscrite. Ce dernier permet la segmentation des textes en lignes puis de détecter les mots et ainsi les classer ce qui nous ramène à un problème d'apprentissage supervisé et une classification multi-classes.

Nous avons présenté dans la partie précédente le processus d'apprentissage automatique qu'on peut résumer dans la figure ci-dessous, dans la suite nous allons détailler notre réalisation qui suit le même processus et qui permet d'obtenir une solution de reconnaissance de l'écriture manuscrite en se basant sur le CNN pour notre modèle d'apprentissage.

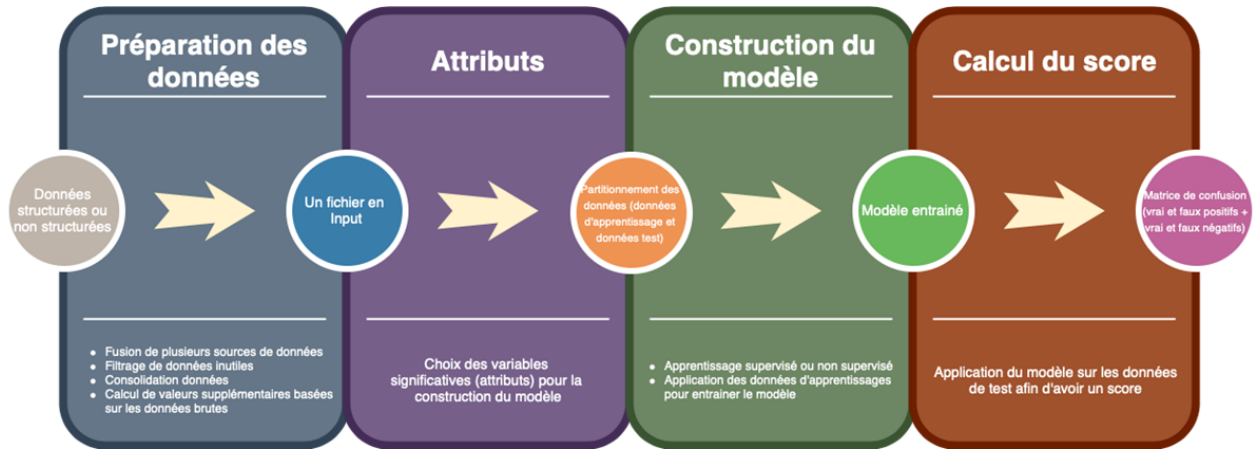


Figure 15: Représentation du processus d'apprentissage automatique

4.2 Préparation des données

4.2.1 Description de la base de données IAM-DB

La base de données sur laquelle on s'est basé pour notre étude est la base IAM Handwriting Database 3.0[18] est constitué de 1539 pages de texte numérisé, 5685 phrases isolées et étiquetées, 13353 lignes de texte isolées et étiquetées et 115320 mots isolés et étiquetés. Ces échantillons sont fournis par 657 différents écrivains qui ont permis la construction d'une base de données d'écritures manuscrites variées. Les mots constituant cette base de données ont été extraits de pages de texte numérisé en utilisant un schéma de segmentation automatique puis ils ont été vérifiés manuellement.

Toutes les lignes et mots sont fournies sous forme d'image PNG et les fichiers d'étiquette correspondants, y compris les informations de segmentation et divers paramètres estimés sont inclus dans les fichiers d'image en tant que méta-informations dans Format XML décrit dans le fichier XML et le format de fichier XML (DTD).

4.2.2 Préparation du data frame

Comme présenté ci-dessus, les données de la base IAM-DB sont formatées sous forme de fichiers (.XML ou .PNG). La suite de notre étude impose qu'ils soient adaptés et transformés en un format exploitable en entrée du modèle choisis. Les données dont nous avons besoin et sur lesquelles nous allons axé notre étude sont les images des mots et leur transcription nommée Label dans la suite du mémoire. Le fichier (.XML) relatif à l'image d'un paragraphe manuscrit, contient des références aux images des mots composant le paragraphe en question.

Depuis ces fichiers XML, pour chaque paragraphe et pour chaque mot, nous avons extrait les deux informations : référence du mot qui n'est autre que le nom du fichier image et son Label. Nous avons ensuite construit une table à 2 colonnes (Nom de l'image, Label) comme illustré ci-dessous.

	0	label
0	d06-050-00-00	Nor
1	d06-050-00-01	is
2	d06-050-00-02	she
3	d06-050-00-03	necessarily
4	d06-050-00-04	being

Figure 16: Exemple de tableau image/label

4.2.3 Définition du nombre de classes

Le tableau obtenu en résultat contient 115320 lignes, chaque ligne fait référence à une image d'un mot manuscrit et son label. Nous avons remarqué que plusieurs lignes font référence au même Label. Ceci s'explique par le fait que des mots contenus dans les documents y sont communs.

L'étape qui suit est de déterminer le nombre de Labels différents contenus dans le tableau que nous avons construit. Le résultat trouvé est de 13550 Label. Nous avons aussi remarqué qu'il existe au contraire des lignes qui ne font référence qu'à un seul Label à la fois.

Puisque notre objectif est que notre modèle apprenne à reconnaître le Label en l'entraînant sur plusieurs images ayant ce même Label, nous avons vu judicieux de retirer les images uniques des données d'étude. Le nombre de Labels a alors diminué à 6360.

4.2.4 Données d'apprentissage vs données de tests

La suite de l'étude se fait en deux étapes : l'étape d'apprentissage puis celle d'évaluation de la classification. Nous aurons besoin de scinder le lot d'images en 2 sous lots, un pour chaque étape de celles susmentionnées. En effet, nous avons choisis d'allouer 90% des images pour l'apprentissage et 10% pour l'évaluation de la performance de la classification.

Nous avons utilisé la méthode "Stratified K-fold", le k désigne le nombre de sous-groupes voulu en sortie. La particularité de cette méthode est qu'elle subdivise le lot d'images en entrée, en créant 2 sous lots en sortie de telle façon à retrouver des images de chacune des classes dans les 2 sous lots et tout en y gardant les proportions d'images dans chacune des classes.

Par exemple, si la classe de Label "That" représente 3% du nombre total d'images contenu dans le lot d'entrée, la méthode va faire en sorte de subdiviser celui-ci de telle façon que la classe "That" représente 3% dans chacun des sous lots en sortie.

4.2.5 Preprocessing sur les données

Lors de la numérisation du texte manuscrit plusieurs éléments peuvent altérer le processus de la reconnaissance de l'écriture. Afin de faciliter le traitement des données, nous avons

normalisé les données d'entrée à savoir les images de mots selon une taille standard dont on donne le résultat dans la figure ci-dessous. Une compression qui permet d'avoir des images de taille similaire 28x28 (figures 17, 18), puis nous avons converti les images, initialement numériser en RVB, en nuances de gris. La conversion en nuance de gris permet d'avoir des pixels avec des valeurs de gris comprises entre $[0,1]$, ce qui nous permet de définir l'état du pixel soit noir ou blanc selon sa proximité des extrêmes 0 ou 1.

```
train_data = x_train.reshape(-1, 28, 28, 1)
test_data = x_test.reshape(-1, 28, 28, 1)
train_data.shape, test_data.shape

((92254, 28, 28, 1), (23064, 28, 28, 1))
```

Figure 17: Exemple d'image après l'étape de pré-processing

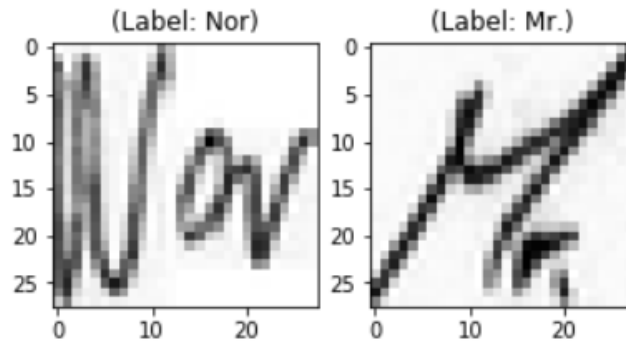


Figure 18: Distribution de notre jeux de données

4.3 Processus d'apprentissage

4.3.1 Phase de pré-entraînement : Auto-encodeur

Description de l'auto-encodeur : Notre objectif à travers le modèle qu'on a construit est d'identifier les mots contenus dans les images de notre base de données, pour cela on a prévu un modèle d'auto-encodeur. Ce dernier nous permet de d'identifier les différentes

écritures en apprenant des 657 différents styles d'écriture.

Pour la partie pré-entraînement afin d'apprendre les différentes écritures, nous avons constitué notre modèle de deux parties : un encodeur de 5 couches CNN et un décodeur de 2 couches CNN.

Notre modèle peut ainsi reconstruire les images en entrées à travers la phase d'apprentissage et par la suite définir les poids des différents neurones constituant notre modèle.

La figure suivante présente le processus d'apprentissage de notre modèle.

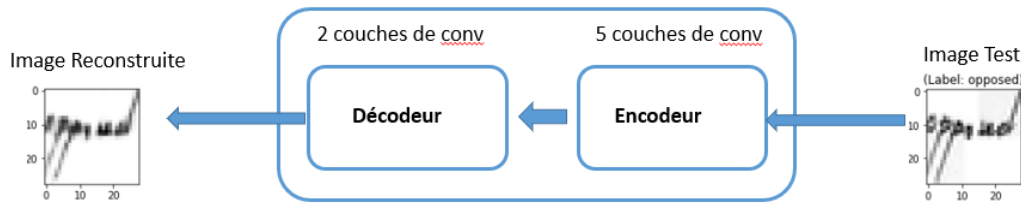


Figure 19: Descriptif de l'auto-encodeur

Définition du modèle :

Le tableau suivant présente l'architecture de notre modèle : le nombre de couches et le nombre de neurones constituant ce dernier à savoir le nombre de poids à entraîner :

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 28, 28, 1)	0
conv2d_1 (Conv2D)	(None, 28, 28, 32)	320
max_pooling2d_1 (MaxPooling2)	(None, 14, 14, 32)	0
conv2d_2 (Conv2D)	(None, 14, 14, 64)	18496
max_pooling2d_2 (MaxPooling2)	(None, 7, 7, 64)	0
conv2d_3 (Conv2D)	(None, 7, 7, 128)	73856
conv2d_4 (Conv2D)	(None, 7, 7, 128)	147584
up_sampling2d_1 (UpSampling2)	(None, 14, 14, 128)	0
conv2d_5 (Conv2D)	(None, 14, 14, 64)	73792
up_sampling2d_2 (UpSampling2)	(None, 28, 28, 64)	0
conv2d_6 (Conv2D)	(None, 28, 28, 1)	577
Total params: 314,625		
Trainable params: 314,625		
Non-trainable params: 0		

Figure 20: Architecture du modèle d'apprentissage

Définition de la fonction objectif : Après la phase d'apprentissage, on obtient une fonction hypothèse qui nous permet de reconstruire les images d'entrées. Pour optimiser notre fonction on a opté pour la méthode des moindres carrés qui nous permet de minimiser l'erreur résultante de la comparaison entre l'image d'entrée et l'image de sortie en comparant l'erreur entre les pixels des deux images, elle se base sur la comparaison euclidienne entre les deux matrices de pixels constituant les images.

```
Train on 87584 samples, validate on 9732 samples
Epoch 1/6
87584/87584 [=====] - 283s - loss: 0.0181 - val_loss: 0.0090
Epoch 2/6
87584/87584 [=====] - 279s - loss: 0.0067 - val_loss: 0.0053
Epoch 3/6
87584/87584 [=====] - 278s - loss: 0.0046 - val_loss: 0.0039
Epoch 4/6
87584/87584 [=====] - 272s - loss: 0.0038 - val_loss: 0.0044
Epoch 5/6
87584/87584 [=====] - 271s - loss: 0.0033 - val_loss: 0.0031
Epoch 6/6
87584/87584 [=====] - 274s - loss: 0.0030 - val_loss: 0.0024

l'erreur est de 0.2%
```

Figure 21: Résultat de la fonction objectif

Définition de la fonction objectif : Après avoir défini la fonction objectif, nous utiliserons la méthode de gradient descendant comme algorithme d’optimisation qui servira à trouver la valeur minimale de celle-ci. La variante choisie de ce type d’algorithme est “Adam” qui est plus appropriée à l’entraînement de modèles de réseau de neurones. C’est un algorithme itératif qui consiste à faire varier les paramètres du modèle de telle façon à faire diminuer la valeur de la fonction objectif.

Après l’application de notre algorithme d’optimisation on arrive à obtenir une erreur d’environ 0.2%, la figure suivante présente le résultat qu’on arrive à atteindre avec la fonction des moindres carrés.

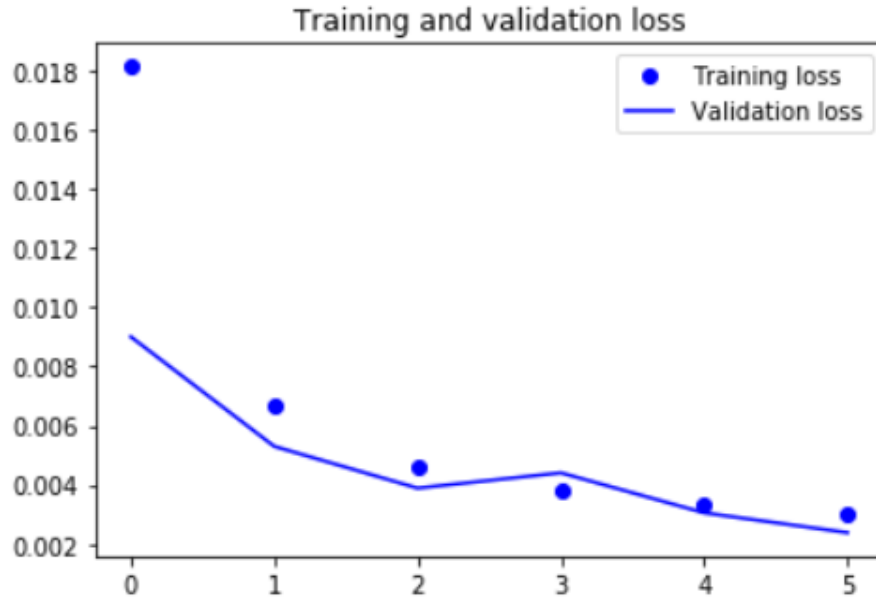


Figure 22: Training and validation loss

4.3.2 Phase d'entraînement pour la Classification

Description du modèle : Durant la première partie de pré-entraînement, l'auto-encodeur avait comme objectif d'apprendre les différents styles d'écriture manuscrite. Pour la suite de l'étude, nous allons garder sa partie encodeur à laquelle nous ajouterons une couche de classification ayant comme sortie un vecteur de probabilité de dimension le nombre de classes retenues précédemment.

Ce modèle diffère de celui d'auto-encodeur dans la mesure où le résultat du nouveau modèle nous donne une information sur la classe à laquelle appartient potentiellement le mot en entrée. En effet, cette classe est celle à laquelle correspond la probabilité la plus élevée parmi les 6360 autres valeurs du vecteur en sortie.

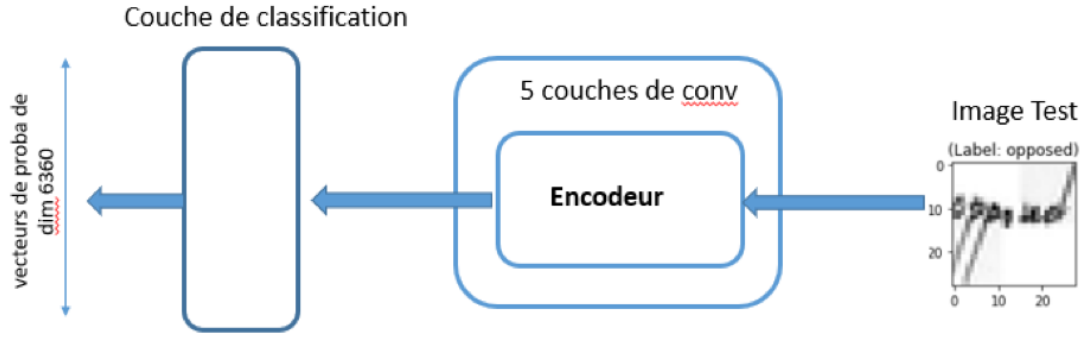


Figure 23: Architecture de notre algorithme de classification

Définition de la fonction objectif : Dans cette phase de classification, comme pour l'entraînement de notre modèle d'apprentissage, pour optimiser la fonction hypothèse on utilise la méthode de l'entropie croisée qui mesure le nombre de bits moyen nécessaires pour identifier un évènement.

Etant donné le nombre élevé de possibilité de classification des mots cette méthode est plus adéquate pour l'optimisation de notre fonction hypothèse.

```
Train on 87584 samples, validate on 9732 samples
Epoch 1/4
87584/87584 [=====] - 158s - loss: 0.4710 - acc: 0.8824 - val_loss: 3.8669 - val_acc: 0.5680
Epoch 2/4
87584/87584 [=====] - 154s - loss: 0.3563 - acc: 0.9129 - val_loss: 4.0634 - val_acc: 0.5673
Epoch 3/4
87584/87584 [=====] - 155s - loss: 0.3274 - acc: 0.9213 - val_loss: 3.9669 - val_acc: 0.5713
Epoch 4/4
87584/87584 [=====] - 154s - loss: 0.3087 - acc: 0.9283 - val_loss: 4.2416 - val_acc: 0.5706
```

Figure 24: Résultat de la fonction objectif

Définition de la métrique d'évaluation : Pour mesurer la performance on a opté pour l'accuracy qui représente le rapport entre le nombre d'images correctement classifiées sur le nombre d'images contenu dans notre base de données. Le résultat de ce rapport est égale à 57% sur nos données de test.

4.4 Conclusion

Notre architecture classe un grand nombre d'image, ce qui nous permet d'identifier les mots manuscrits (figure 25). A travers notre étude nous avons pu constater que l'accuracy de notre classification est plus élevé pour les occurrences qui ont un grand nombre. Plus la fréquence des occurrences est grande plus notre classification de notre base de données est précise (figure 26).

On peut en conclure que notre modèle favorise l'apprentissage des styles d'écriture, plus on a d'exemple d'écriture d'un mot plus notre reconnaissance de l'écriture manuscrite est plus exacte.

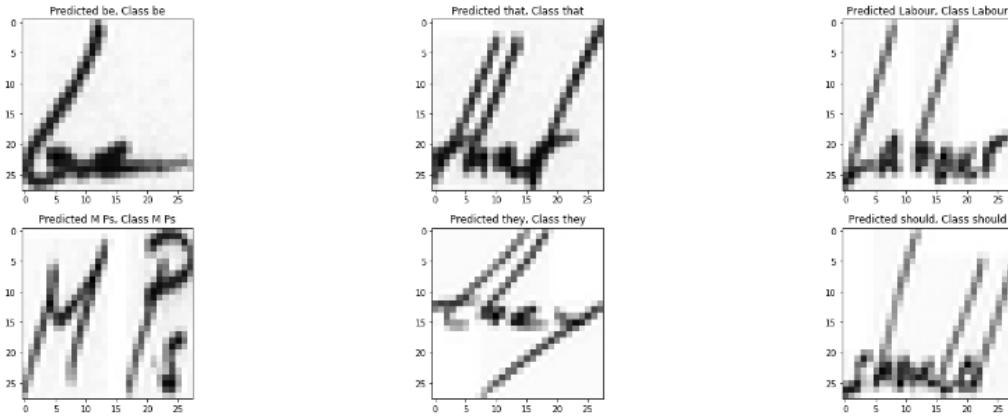
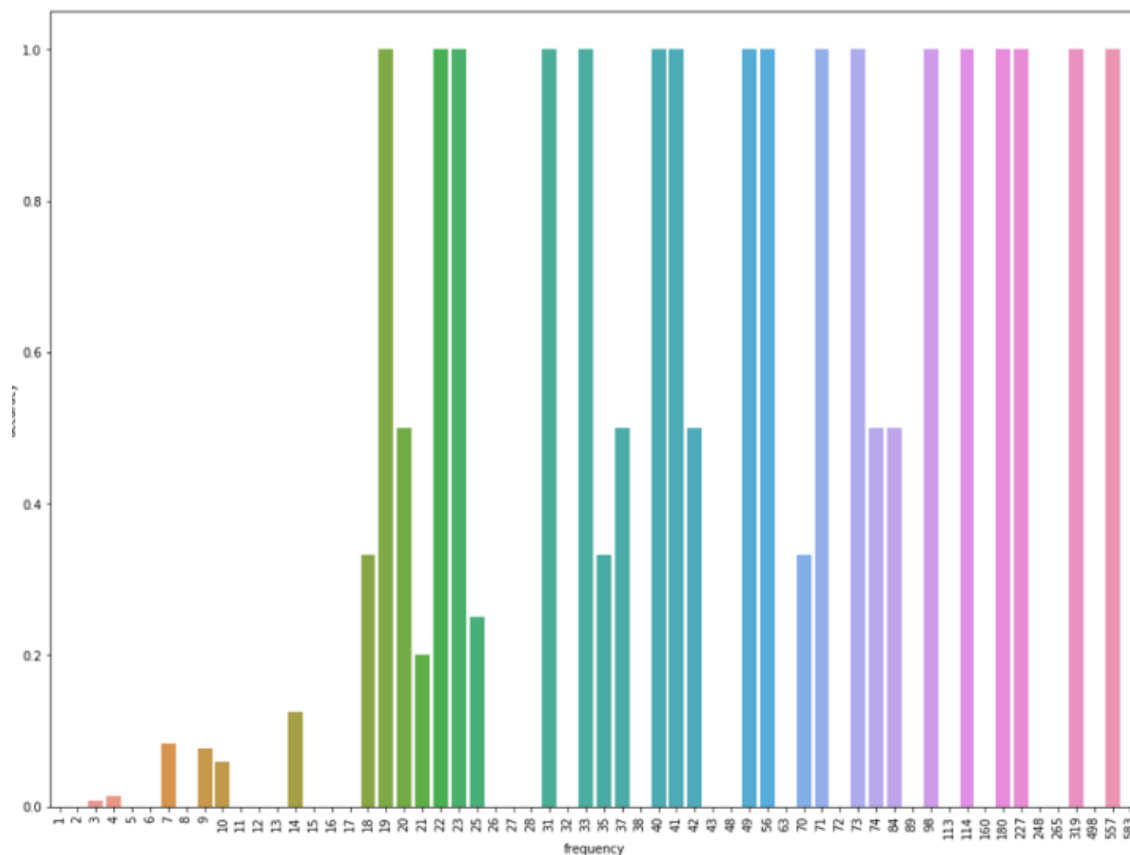


Figure 25: Exemple



5 Conclusion et Discussion

Nous avons proposé une architecture d'apprentissage en deux étapes qui permet, dans un premier temps, de modéliser l'écriture manuscrites des 657 écrivains. Ensuite un modèle de classification est entraîné pour identifier la classe du mot dessiné dans l'image. Pour répondre à ces deux objectifs, nous avons commencé par préparer nos données en les transformant en des images noir et blanc puis les normalisant en une taille unique. Batuhan et al. [19] ont proposé une méthode de normalisation de taille en ajoutant des pixel zeros en longueur et largeur de l'image jusqu'à atteindre la taille standardisée qui correspond à la longueur et largeur maximales existantes dans notre base d'images. En revanche, nous avons opté pour une option de compression d'image afin d'avoir un format unique de taille 28x28. Notre choix se justifie par le fait que l'image du mot manuscrit ne contient pas une répartition de pixel très étendue et que l'information peut facilement être comprimée sans perdre de la qualité de l'image. Une fois le pré-traitement est fini, nous avons d'abord proposé un modèle auto-encoder permettant de reconstruire les images manuscrites et ainsi d'identifier les caractéristiques de chaque écriture parmi les 657. le résultat de cette première modélisation semble très prometteur car on a atteint une erreur de reconstruction de 0.2%. Les attributs appris par ce premier modèle ont été utilisés pour assister le modèle de classification à mieux identifier la classe des mots. En effet, nous avons utilisé la technique du "Tranfert Learning" pour lier le premier modèle défini pour une tâche de reconstitution au deuxième modèle qui a pour objectif la classification des mots. Ainsi, en proposant cette architecture en deux temps novatrice, nous avons pu améliorer l'accuracy de validation de 3 points en comparaison avec les résultats de Batuhan et al.

	<i>Accuracy - Training</i>	<i>Accuracy - Validation</i>
<i>Notre modèle : E+CNN</i>	92%	57%
<i>Batuhan : RESNET-34</i>	35%	27%

Figure 27: comparaison de notre algorithme et RESNET-34

En perspective, nous aimerions augmenter la taille de notre base de données afin de fournir au modèle plus d'images pour chaque classe de mot. Les techniques comme rotation d'images, changement de contraste, zoom... ont déjà prouvé leur efficacité dans des tâches de classification d'image (S. Gunter and H. Bunke)[20]] et ainsi permettre la reconnaissance de l'écriture manuscrite.

References

- [1] Yaniv Taigman et al. DeepFace. Closing the gap to human-level performance in face verification.
- [2] Tuka Alhanai et al. Detecting depression with audio/text sequence modeling of interviews.
- [3] Zhengyao Jiang. Cryptocurrency portfolio management with deep reinforcement learning.
- [4] <https://www.gettingsmart.com/2018/08/32-ways-ai-is-improving-education/>.
- [5] Rose Luckin. Machine learning and human intelligence: The future of education for the 21st century.
- [6] G.Tauschek. Reading machine.
- [7] R.Rabiner B.H.Juang. An introduction to hidden markov models.
- [8] Tom M.Mitchell. The discipline of machine learning.
- [9] https://en.wikipedia.org/wiki/Arthur_Samuel.
- [10] N. Otsu .A. threshold selection method from gray-level histograms.
- [11] G.Chen Sargur. Removing structural noise in handwriting images using deep learning.
- [12] Eck and J. Schmidhuber. “dekf-lstm,” in proc. 10th european symposium on artificial neural networks,esann 2002.
- [13] Graves et al . Speech recognition with deep recurrent neural networks.
- [14] Lecun & convolutional network for images, speech and timeseries.
- [15] https://fr.wikipedia.org/wiki/Perceptron_multicouche.
- [16] Zhuowen Tu et al .Generalizing. Pooling functions in convolutional neural networks:mixed, gated, and tree.

- [17] LeCun al. Gradient-based learning aplied to document recognition.
- [18] <http://www.fki.inf.unibe.ch/databases/iam-handwriting-database>.
- [19] <https://pdfs.semanticscholar.org/3339/237110cd5fd3fa8206623e9b740be1d72c9e.pdf>.
- [20] Günter and H. Bunke. Off-line cursive handwriting recognition using multiple classifier systems – on the influence of vocabulary, ensemble, and training set size.