Intégration de connaissances explicites à l'apprentissage profond pour la reconnaissance et la segmentation d'écriture manuscrite d'enfants

Simon Corbillé

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Informatique*

Par

« Simon CORBILLÉ »

« Intégration de connaissances explicites à l'apprentissage profond pour la reconnaissance et la segmentation d'écriture manuscrite d'enfants »

Thèse présentée et soutenue à « Rennes », le « 28/06/2023 »
Unité de recherche : IRISA - UMR 6074

Rapporteurs avant soutenance :

Véronique EGLIN — Professeure, INSA de Lyon
Harold MOUCHERE — Professeur, Nantes Université

Composition du Jury :

Attention, en cas d'absence d'un des membres du Jury le jour de la soutenance, la composition du jury doit être revue pour s'assurer qu'elle est conforme et devra être répercutée sur la couverture de thèse

Président :	Harold MOUCHERE	Professeur, Nantes Université
Rapporteurs :	Véronique EGLIN	Professeure, INSA Lyon
	Harold MOUCHERE	Professeur, Nantes Université
Examinateurs :	Sébastien ADAM	Professeur, INSA Rouen
	Nicolas RAGOT	MdC, HDR, Université de Tours
Dir. de thèse :	Elisa FROMONT	Professeure, Université de Rennes
Co-dir. de thèse :	Eric ANQUETIL	Professeur, INSA Rennes

Remerciements

Je tiens à exprimer ma plus profonde gratitude et ma reconnaissance aux personnes suivantes, sans qui cette thèse n'aurait pas été possible.

Je tiens à remercier tout d'abord les membres du jury. Je remercie Véronique Eglin, Professeure à l'INSA de Lyon et Harold Mouchère, Professeur à Nantes Université d'avoir accepté de rapporter ce manuscrit. Je remercie de même Sébastien Adam, Professeur à l'INSA de Rouen et Nicolas Ragot, Maitre de conférence HDR à l'Université de Tours d'avoir accepté d'être les examinateurs de ma soutenance.

Je remercie ensuite mes encadrants Eric Anquetil, Professeur à l'INSA de Rennes et Elisa Fromont Professeur à l'Université de Rennes, pour leurs conseils, leurs soutiens et leurs expertises pendant ces années de thèse. Leurs commentaires perspicaces et leurs précieuses suggestions ont grandement contribué à façonner ce travail.

Je remercie chaleureusement tous les membres des équipes IntuiDoc et Lacodam et également tous les collègues de l'IRISA avec qui j'ai eu l'occasion d'échanger.

Je remercie du fond du cœur ma famille pour ses encouragements. Leur soutien constant et leur compréhension ont été déterminants pour me garder motivé pendant les phases difficiles de ce voyage. Je tiens à remercier tous mes amis qui ne liront sûrement pas ce manuscrit. Amies, amis, merci.

Enfin, je tiens à exprimer ma gratitude à toutes les personnes qui ont contribué de façon importante ou modeste, en offrant leurs conseils et leurs encouragements.

TABLE DES MATIÈRES

Introduction **9**

I Prérequis 15

1 Contexte 16

1.1 Caractéristiques de l'écriture manuscrite 17
1.1.1 La langue 17
1.1.2 Les styles script et cursif 17
1.1.3 Représentation des données 18
1.1.4 L'écriture d'enfants 19
1.2 Problématiques liées à l'écriture manuscrite 20
1.2.1 Tâche de reconnaissance d'écriture manuscrite 21
1.2.2 Tâche de segmentation d'écriture manuscrite 21
1.3 Contexte éducatif : apprentissage de l'écriture 23
1.3.1 Contexte de recopie 23
1.3.2 Contexte de dictée 24
1.4 Jeux de données utilisées 25
1.4.1 Écriture d'adultes : IAM-OnDB 25
1.4.2 Écriture d'enfants 25
1.5 Synthèse de données 27
1.5.1 Synthèse hors ligne 27
1.5.2 Synthèse en ligne 28

2 IntuiScript : système d'analyse d'écriture d'enfants 29

2.1 Historique des évolutions 30
2.1.1 Analyse de la qualité de l'écriture française 30
2.1.2 Déclinaison pour d'autres langues 30
2.1.3 Reconnaissance et analyse de l'orthographe 31
2.2 Spécifications dans un contexte de recopie 32

2.2.1 Segmentation 32
2.2.2 Analyse niveau lettre 34
2.2.3 Calcul des chemins niveau mot 36
2.2.4 Retour utilisateurs sur la qualité d'écriture 37
2.2.5 Extension pour un contexte de dictée : ajout d'un guidage phonétique 38
2.3 Bilan 41

3 Méthodes d'apprentissage profond 42
3.1 Tâche de reconnaissance de caractères 43
3.2 Tâche de reconnaissance de mots 45
3.2.1 Méthode d'entrainement : Connectionist Temporal Classification (CTC) 46
3.2.2 Réseau de neurones convolutifs récurrents 49
3.2.3 Réseau encodeur décodeur avec un modèle d'attention 50
3.2.4 Champ réceptif et convolutions 51
3.3 Tâche de détection d'objets 52
3.3.1 Réseau d'extraction de caractéristiques 53
3.3.2 Réseau de proposition de régions 54
3.3.3 Extraction des caractéristiques d'une région 55
3.3.4 Branche de prédictions 56
3.3.5 Non Maximal Suppression 57

II Contributions mineures 59

4 Reconnaissance de caractère manuscrit cursif 60
4.1 Extractions d'informations statique et dynamique dans le signal en ligne . 61
4.1.1 Information statique : la forme 62
4.1.2 Information dynamique : la direction 62
4.1.3 Information dynamique : l'orientation 63
4.2 Intégration dans un CNN 64
4.2.1 Fusion précoce 64
4.2.2 Fusion tardive 64
4.3 Expérimentations : reconnaissance de caractères 65
4.3.1 Jeu de données 65

4.3.2 Protocole . . . 67
4.3.3 Résultats . . . 67
4.4 Bilan . . . 72

5 Reconnaissance et segmentation de mots manuscrits 74
5.1 Intégration des primitives de segmentation dans un réseau de neurones . . 74
5.1.1 Représentation des entrées dans le réseau de neurones . . . 74
5.1.2 Spécification des stratégies de découpage . . . 76
5.2 Expérimentations : reconnaissance de mots d'adulte . . . 78
5.2.1 Protocole . . . 78
5.2.2 Résultats . . . 79
5.3 Segmentation en lettres d'un mot à l'aide d'un modèle de reconnaissance . 81
5.3.1 Définition de la segmentation lettre . . . 81
5.3.2 Amélioration de la qualité de segmentation . . . 83
5.4 Expérimentations : reconnaissance et segmentation de mots d'enfants . . . 86
5.4.1 Protocole . . . 86
5.4.2 Résultats . . . 86

III Contributions majeures 91

6 Intégration de la prédiction du modèle Seq2Seq dans le système de guidage du modèle IntuiScript 92
6.1 Intégration de la reconnaissance d'un modèle Seq2Seq dans le guidage du modèle IntuiScript . . . 93
6.1.1 Guidage de l'analyse par la reconnaissance du Seq2Seq . . . 93
6.1.2 Combinaison des stratégies de guidage . . . 94
6.2 Optimisation du temps de calcul : élagage du treillis de segmentation . . . 98
6.3 Retours orthographiques . . . 99
6.4 Expérimentations . . . 100
6.4.1 Protocole expérimental . . . 100
6.4.2 Évaluation de la reconnaissance et de la segmentation . . . 101
6.4.3 Évaluation de la stratégie d'élagage . . . 102
6.4.4 Évaluation de la qualité des retours . . . 103
6.5 Bilan . . . 103

7 Seq2Seg : Combinaison des modèles Seq2Seq et R-CNN **105**

7.1 Niveau A : filtrage des boîtes englobantes à l'aide d'un modèle de reconnaissance . . . 106

7.1.1 Étape 1 : calcule de toutes les séquences de prédiction du détecteur d'objets . . . 109

7.1.2 Étape 2 : filtrage des séquences en fonction de la longueur du mot prédit par le modèle de reconnaissance . . . 109

7.1.3 Étape 3 : calcule du score associé à chaque séquence . . . 110

7.2 Niveau B : amélioration de la segmentation du R-CNN à l'aide d'un treillis de segmentation . . . 111

7.3 Perspective : améliorer la robustesse . . . 112

7.4 Expérimentations . . . 114

7.4.1 Protocole . . . 114

7.4.2 Résultats quantitatifs . . . 115

7.4.3 Résultats qualitatifs . . . 118

7.4.4 Premiers résultats : intégration d'un mécanisme de rejet . . . 121

7.5 Bilan . . . 122

Conclusion et perspectives **123**

Publications **127**

Bibliography **129**

INTRODUCTION

Contexte : analyse de l'écriture des enfants

Selon le ministère de l'Éducation nationale, le niveau en orthographe des élèves en France n'a cessé de diminuer durant les dernières décennies [1]. Parallèlement, le numérique est de plus en plus utilisé en classe. De nombreux travaux s'intéressent à l'utilisation du numérique en classe où l'objectif est de concevoir des applications éducatives qui sont bénéfiques pour les élèves et les enseignants d'un point de vue pédagogique. Pour qu'une application éducative soit utilisable, il faut au préalable développer des outils informatiques fiables et robustes. Nos travaux s'intéressent au domaine de l'apprentissage de l'écriture et plus particulièrement à celui de l'orthographe. L'objectif de la thèse est de répondre à une problématique d'analyse d'écriture manuscrite d'enfants en cours de l'apprentissage de l'orthographe. La problématique d'analyse illustrée dans la figure 1, regroupe une problématique de reconnaissance et de segmentation dans le but de faire un retour visuel au niveau de la lettre à l'élève. Il est nécessaire d'avoir une analyse rapide et précise en reconnaissance et en segmentation pour donner un retour immédiat cohérent à l'élève sur la nature de ses erreurs.

Cette thèse met en relation deux équipes de recherche du laboratoire IRISA de Rennes :

— L'équipe IntuiDoc [2] s'intéresse à des problématiques d'analyse et de reconnaissance d'écriture manuscrite. Elle conçoit aussi des solutions d'aide à l'apprentissage pour l'e-éducation.

— L'équipe LACODAM [3] s'intéresse aux traitements de grands volumes de données pour en tirer de nouvelles connaissances ou pour prendre de meilleures actions à l'aide notamment de modèles d'apprentissage profond.

1. https ://www.education.gouv.fr/les-performances-en-orthographe-des-eleves-en-fin-d-ecole-primaire-1987-2007-2015-1991
2. https ://www-intuidoc.irisa.fr/
3. https ://team.inria.fr/lacodam/fr/

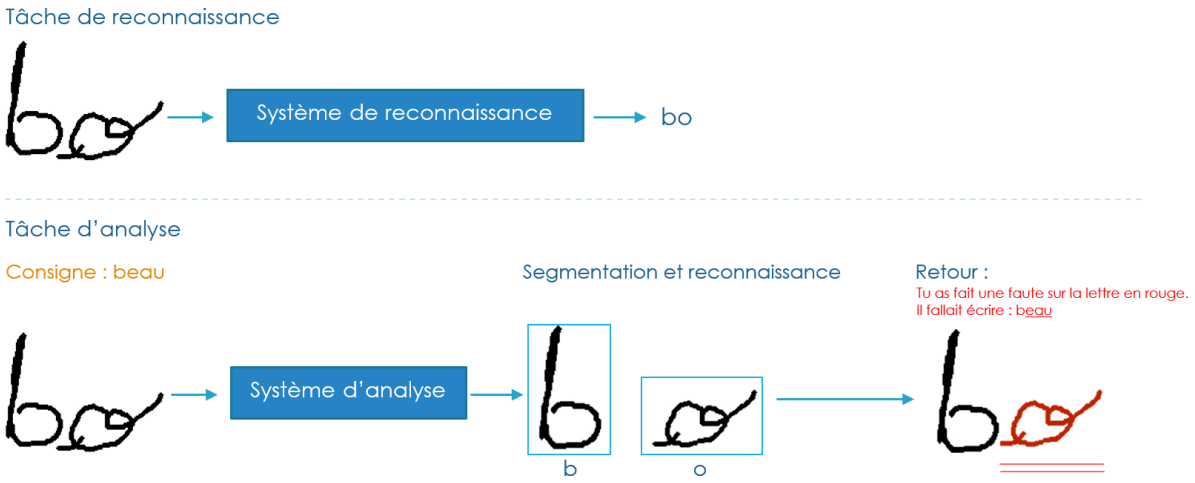

FIGURE 1 – Différence entre une tâche de reconnaissance et d'analyse d'écriture manuscrite.

Contributions

Deux axes sont abordés dans nos travaux. Le premier consiste à améliorer et à étendre le système d'analyse de l'écriture manuscrite conçu dans le contexte du projet IntuiScript [GSA17]. Pour cela, nous allons introduire des modèles d'apprentissage profond pour consolider la chaine de traitement. Le second axe vise à concevoir un nouveau système complet d'analyse de l'écriture d'enfants basé sur l'hybridation de modèles d'apprentissage profond.

Axe 1 : amélioration et extension du système d'analyse d'écriture IntuiScript

Dans ce premier axe, nous cherchons à améliorer et à étendre le système d'analyse d'écriture d'enfant IntuiScript qui sera présenté en détails dans le chapitre 2. Le système IntuiScript utilise une approche analytique se basant sur une segmentation explicite découpant le mot en hypothèses de segmentation regroupées dans un treillis. Le système analyse chaque hypothèse du treillis au niveau lettre, pour faire ensuite une analyse au niveau mot. Cette analyse est guidée par des connaissances *a priori* regroupant la consigne ainsi que des mots phonétiquement proches de la consigne. Le premier axe regroupe une contribution mineure visant à remplacer le reconnaisseur de caractères d'IntuiScript utilisé dans l'étape d'analyse lettre par un reconnaisseur plus performant et une contribution

majeure cherchant à améliorer le mécanisme de guidage d'IntuiScript.

Contribution 1 : reconnaissance de caractères isolés à l'aide d'un modèle d'apprentissage profond. Dans cette première contribution, nous développons un modèle hybride entre le système IntuiScript basé sur une segmentation explicite et les dernières évolutions en matière d'apprentissage profond pour une tâche de reconnaissance de caractères. Nous proposons de remplacer le reconnaisseur de caractères manuscrits isolés existant par un reconnaisseur plus performant dans le but de rendre l'analyse au niveau mot plus précise. Pour concevoir ce moteur de reconnaissance, notre stratégie développée dans le chapitre 4, repose sur l'intégration de l'information sur la dynamique du tracé en ligne représentant un caractère en plus de l'information statique représentant la forme dans un réseau de neurones convolutifs afin d'augmenter les performances de reconnaissance.

Contribution 2 : guidage de l'analyse par un modèle d'apprentissage profond dans un contexte de dictée. Dans un contexte de dictée, le système de guidage existant basé sur un modèle de langue n'est pas performant du fait de la nature variée des fautes d'orthographe présentes dans les mots à analyser. Dans ces travaux détaillés dans le chapitre 6, nous avons utilisé un modèle d'apprentissage profond Seq2Seq pour pouvoir étendre le fonctionnement du système IntuiScript dans un contexte de dictée. De plus, le temps d'analyse de mots longs était trop élevé pour faire un retour immédiat à l'enfant. Nous proposons d'utiliser le résultat de segmentation implicite du modèle Seq2Seq pour diminuer la complexité des calculs par élagage du treillis de segmentation afin d'accélérer le temps d'analyse.

Axe 2 : élaboration d'une approche complète d'analyse d'écriture

Dans le second axe, nous élaborons une approche complète d'analyse, *i.e.* un système de reconnaissance et de segmentation d'écriture basé sur des modèles d'apprentissage profond. Beaucoup de travaux de la littérature utilisant des modèles d'apprentissage profond s'intéressent à la partie reconnaissance, mais très peu à la partie segmentation. Le second axe regroupe une contribution mineure visant à améliorer la reconnaissance et la segmentation d'un modèle de reconnaissance de mots à l'aide de connaissances expertes liées au domaine de l'écriture et une contribution majeure proposant un système d'analyse basé sur une combinaison de deux modèles.

Contribution 3 : amélioration de la reconnaissance et de la segmentation d'un modèle de reconnaissance de mots. Les modèles de reconnaissance de mots ma-

nuscrits de l'état de l'art sont basés sur des modèles d'apprentissage profond. Ils reposent sur une approche holistique qui ne cherche pas à faire une segmentation du mot contrairement à une approche analytique. Ils dépendent de l'utilisation de filtres de convolutions pour extraire des caractéristiques spatiales dans l'image d'entrée suivie de diverses opérations selon l'architecture utilisée. Le paramétrage des filtres de convolutions induit un découpage fixe du mot à reconnaitre qui est mal adapté pour une segmentation en lettres précise. Les travaux en lien avec cette contribution sont explicités dans le chapitre 5. La première stratégie a été d'améliorer la reconnaissance de ce type de réseau en proposant un découpage du mot à l'aide d'un treillis de segmentation expert. Plusieurs stratégies ont été développées afin de chercher le meilleur moyen de découper le mot en lettres. La seconde stratégie a été d'améliorer la segmentation implicite de ce type de réseau en modifiant la fonction de coût CTC utilisée pour l'entrainement. Nos travaux ont permis de démontrer les limites de ce type de réseau pour la tâche de segmentation et la nécessité d'utiliser un réseau dédié à la tâche de segmentation qui est l'objet de notre quatrième contribution.

Contribution 4 : combinaison d'un modèle R-CNN et Seq2Seq pour l'analyse de mots manuscrits. Pour répondre au double objectif d'une bonne reconnaissance avec une segmentation précise, nous proposons dans le chapitre 7 une nouvelle méthode **Seq2Seg** qui combine un modèle de détection d'objet R-CNN dédié à la **segmentation** avec un modèle Seq2Seq dédié à la **reconnaissance** d'écriture manuscrite. Une étape importante dans le processus de détection d'objets est de définir les critères de sélection des objets détectés. Le résultat de reconnaissance du Seq2Seq permet de guider la sélection des objets détectés ainsi que de fournir un résultat de reconnaissance qui est plus précis que celui du modèle R-CNN seul. La méthode **Seq2Seg** utilise également un treillis de segmentation pour affiner la segmentation obtenue. Les expérimentations ont démontré que la méthode **Seq2Seg** surpasse le système IntuiScript de base ainsi que le système avec les améliorations proposées précédemment.

Organisation du manuscrit de thèse

Dans la **première partie**, nous présentons l'état de l'art, le contexte d'analyse d'écriture manuscrite ainsi que les architectures des modèles utilisés dans nos travaux. Le **chapitre 1** présente le contexte, les problématiques et les motivations en lien avec une tâche d'analyse de l'écriture manuscrite. Le **chapitre 2** spécifie le fonctionnement du système

existant IntuiScript qui permet de faire de l'analyse d'écriture d'enfants. Nous développons les avantages et les limites de ce système pour une application dans un contexte de dictée. Le **chapitre 3** présente les architectures de l'état de l'art des modèles d'apprentissage profond utilisées dans ces travaux pour des tâches de reconnaissance de caractères, de reconnaissance de mots et de détection d'objets.

La **deuxième partie** décrit les contributions mineures de la thèse. Le **chapitre 4** porte sur la conception d'un reconnaisseur de caractères manuscrits cursifs basé sur un réseau de neurones convolutifs intégrant l'information sur la dynamique du tracé (contribution 1). Le **chapitre 5** développe plusieurs stratégies pour améliorer les performances d'un modèle de reconnaissance d'écriture ainsi qu'une analyse de la segmentation en lettres implicite obtenue à partir d'un résultat de reconnaissance. Nous détaillons une approche modifiant la fonction de coût et une approche utilisant un treillis de segmentation pour améliorer cette segmentation (contribution 3).

La **troisième partie** décrit les contributions majeures de la thèse. Le **chapitre 6** présente une amélioration du système IntuiScript qui utilise un modèle Seq2Seq dans le mécanisme de guidage. Ce guidage permet au système d'analyser les fautes non phonétiques ainsi que d'accélérer le temps d'analyse (contribution 2). Le **chapitre 7** présente une nouvelle approche dénommée **Seq2Seg**. Elle repose sur une combinaison d'un modèle R-CNN dédié à la tâche de segmentation et d'un modèle Seq2Seq dédié à la tâche de reconnaissance dans le but de faire une analyse précise de l'écriture (contribution 4).

Pour terminer, nous ferons une conclusion des travaux de la thèse et présenterons les perspectives associées.

Première partie

Prérequis

Chapitre 1

Contexte

Au fil de son évolution, l'espèce humaine a développé une **multitude de langues** pour communiquer de manière orale et écrite. Aujourd'hui, environ 7 000 langues sont utilisées dans le monde. Les supports et les méthodes d'écriture ont évolué au cours des siècles comme l'illustre la figure 1.1 où un système d'écriture comme l'alphabet latin peut être utilisé pour écrire plusieurs langues (latin, anglais, français ...). Dans ces travaux, nous nous intéressons à la **reconnaissance et la segmentation de l'écriture manuscrite latine**. La reconnaissance d'écriture manuscrite est le fait d'associer un texte à un tracé écrit représentant une lettre, un mot ou une phrase. Dans notre contexte, la segmentation d'écriture manuscrite est le fait de localiser les caractères contenus dans un mot.

Figure 1.1 – Exemples d'écritures : 1) écriture cunéiforme, 2) hiéroglyphe, 3) alphabet grec, 4) alphabet latin, latin, 5) alphabet latin, anglais, 6) alphabet latin, français.

Dans ce chapitre, nous commençons par définir la langue, le style, le format de représentation des données d'une écriture manuscrite en général ainsi que ce qui caractérise l'écriture réalisée par des enfants en cours d'apprentissage, sujet qui est traité avec une attention particulière dans ces travaux. Puis nous spécifions les problématiques de reconnaissance et de segmentation d'écriture manuscrite. Ensuite, nous précisons notre contexte applicatif d'analyse d'écriture latine dans un but éducatif. Enfin, nous présentons les jeux de données utilisés ainsi que les techniques de synthèse de données appliquées dans nos travaux.

1.1 Caractéristiques de l'écriture manuscrite

Un **alphabet** est un système d'écriture constitué d'un ensemble de symboles, dont chacun peu représenter, un des phonèmes d'une langue [1]. Un alphabet peut servir à l'écriture de plusieurs **langues**. Nous pouvons noter que la difficulté de reconnaissance évolue selon la complexité de l'alphabet utilisé pour écrire un mot. Le nombre de caractères, la présence de caractères qui se ressemblent dans une langue définisse le niveau de complexité de l'écriture à reconnaitre. Le style d'écriture (cursif ou script) ainsi que la maitrise du scripteur (débutant ou expert) sont des paramètres à prendre en compte pour concevoir un système de reconnaissance et de segmentation performant.

1.1.1 La langue

Pour concevoir un système de reconnaissance et de segmentation efficace, il est important de connaitre les spécificités de la langue à reconnaitre comme la **taille de l'alphabet**, les **règles de ligatures inter-lettres** ainsi que le **sens d'écriture**. La figure. 1.2 représente le mot "bonjour" écrit en français, russe, mandarin et en arabe. Elle permet de mettre en lumière les spécificités des alphabets latin, cyrillique, arabe et des sinogrammes.

FIGURE 1.2 – Exemple du mot "bonjour" écrit dans quatre langues différentes. En orange l'alphabet et en bleu la langue utilisée.

1.1.2 Les styles script et cursif

Le style script consiste à écrire les lettres d'un mot sans liaison tandis que le style cursif ajoute des liaisons appelées **ligatures** au sein d'un même mot. Une personne peut utiliser un style d'écriture cursif ou script ou encore un mixte des deux. Le style d'écriture varie en fonction des personnes et de la méthode d'enseignement. La figure. 1.3 illustre la

1. https ://fr.wikipedia.org/wiki/Alphabet

différence entre le type cursif, script et le mixte des deux avec le français et l'anglais pour le mot "bonjour". Nous pouvons remarquer les différentes liaisons ou ligatures ajoutées entre les lettres pour le cursif. Certains caractères s'écrivent différemment selon le style utilisé.

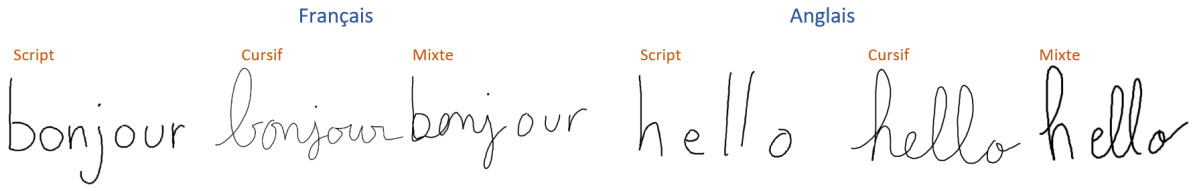

FIGURE 1.3 – Exemples de mots écrits dans le style script, cursif et le mixte des deux pour le français et l'anglais.

1.1.3 Représentation des données

L'écriture manuscrite peut être représentée au format numérique sous forme d'image appelée ici **format hors ligne** ou bien sous forme de liste de points appelée ici **format en ligne**.

Le format hors ligne

Une écriture manuscrite peut être représentée dans une **image** comme l'illustre la figure. 1.4. La difficulté de la tâche de reconnaissance et de segmentation varie selon la qualité du document. En effet, les documents peuvent être anciens et contenir une écriture dégradée (transparence, rature, effacement) ou être plus récents et contenir moins de dégradations. Nous pouvons noter que le processus d'acquisition de ces images introduit du bruit dans les données. De plus, le format hors ligne ne contient pas d'informations sur le ductus (ordre, direction, vitesse, rythme) de l'écriture.

Le format en ligne

Une écriture manuscrite peut être représentée sous forme d'une **trajectoire codée par une suite de points**. Il existe plusieurs supports pour récolter ces données de trajectoires comme une tablette tactile munie d'un stylet ou encore un tableau intelligent. La suite de points représentant le tracé manuscrit est échantillonnée selon un intervalle de temps et est définie à l'aide de quatre variables : les coordonnées (x, y) du point, la pression

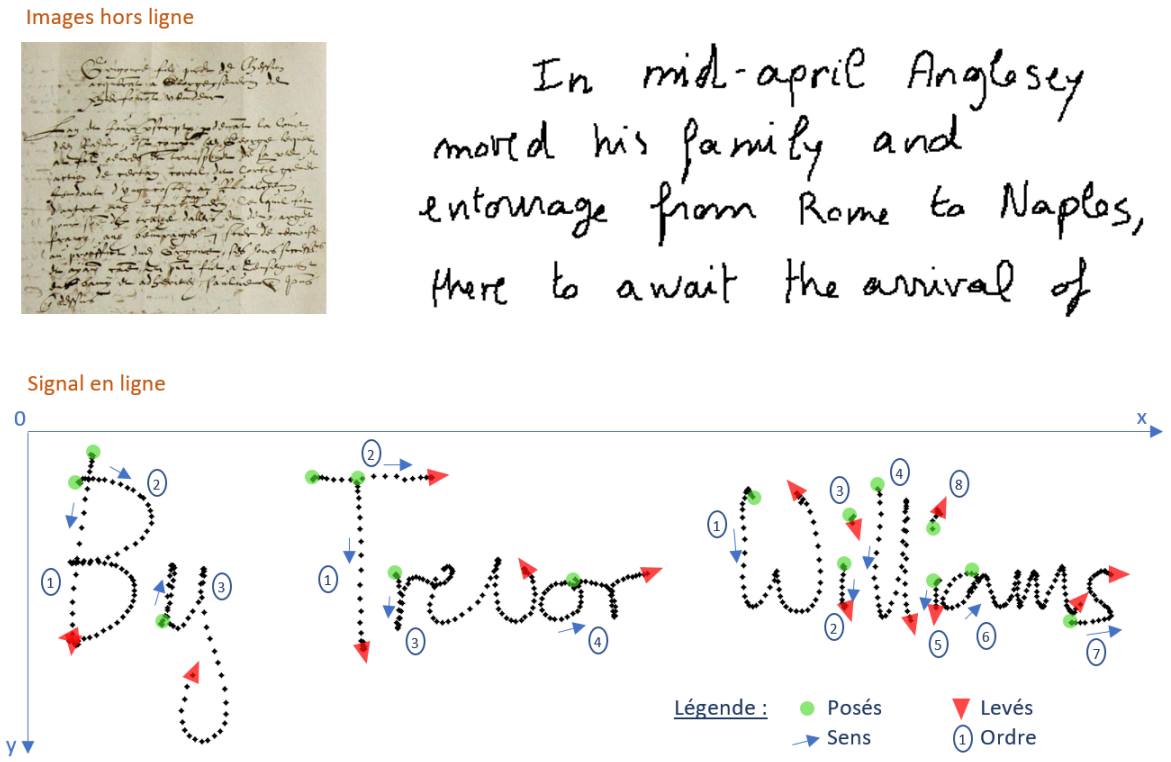

FIGURE 1.4 – Exemples d'images hors ligne et de signal en ligne représentants des écritures manuscrites.

du stylet et l'horodatage. Le tracé en ligne contient des informations supplémentaires sur le **ductus** de l'écriture comme les posés et levés de crayon, le sens, l'ordre, l'orientation et la vitesse d'écriture. La figure 1.4 montre un exemple de tracé en ligne.

1.1.4 L'écriture d'enfants

Les enfants apprenant à écrire possèdent une écriture encore mal formée et commettent des erreurs comme l'illustre la figure 1.5. La ligne *a* de la figure illustre plusieurs exemples d'écriture manuscrite dégradée. Nous pouvons voir qu'une déformation de la lettre « e » peut être interprétée comme une lettre « l » et inversement pour le mot « elle » en troisième position de cette ligne. La ligne *b* montre plusieurs exemples d'**erreurs phonétiques**. Dans le premier exemple, où l'instruction est "mes" [mɛ], l'enfant écrit "mai" [me], qui sont deux homophones en français qui se différencient généralement par le contexte d'utilisation. Ces erreurs phonétiques peuvent être anticipées par un modèle de langage qui prendrait en compte l'information contextuelle de la consigne.

Cependant, d'autres types d'erreurs illustrées à la ligne *c* telles que la **dyslexie** et les **mots hors vocabulaire** ne le peuvent pas. Le premier exemple de la ligne *c* montre une erreur courante en français où l'enfant confond les lettres "b" et "d".

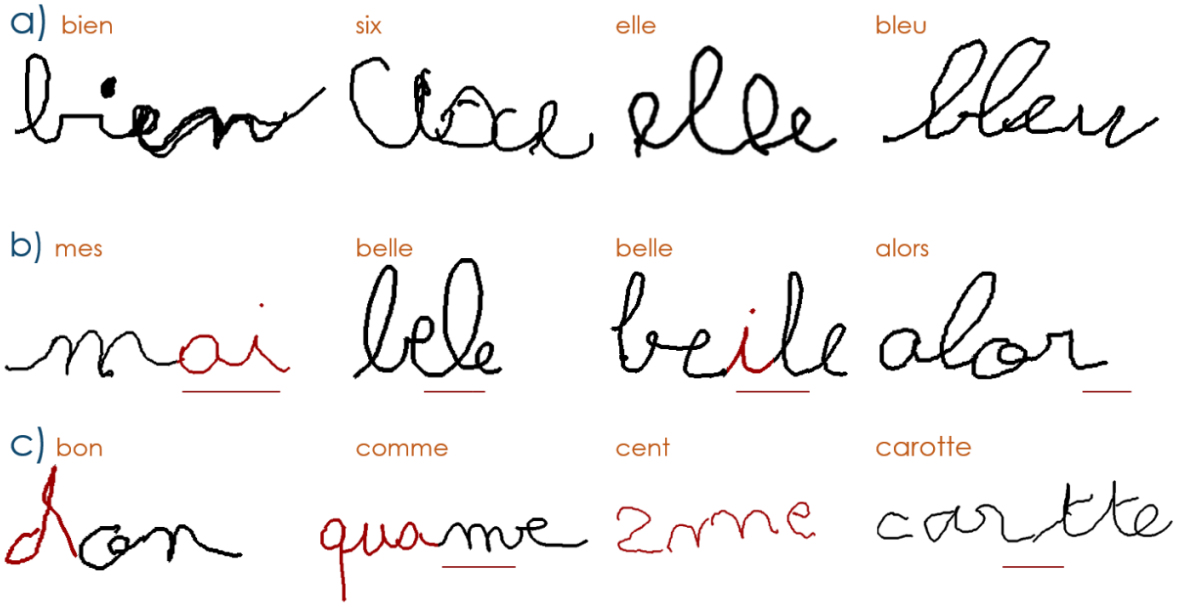

FIGURE 1.5 – Exemples d'écriture manuscrite cursive d'enfants. La consigne dictée donnée aux enfants est en orange et des exemples de corrections produites par un système d'analyse automatique sont en rouge. La ligne *a* montre une écriture dégradée, la ligne *b*, des erreurs phonétiques et la ligne *c* montre d'autres types d'erreurs.

1.2 Problématiques liées à l'écriture manuscrite

Notre objectif est de concevoir un système d'analyse d'écriture manuscrite d'enfants. L'analyse de l'écriture regroupe une tâche de **reconnaissance** qui reconnait le texte et une tâche de **segmentation** qui localise la position des caractères du texte dans le but de faire un retour à l'enfant sur la qualité de son orthographe. Cette section définie la tâche de reconnaissance et la tâche de segmentation ainsi que les métriques utilisées pour les évaluer.

1.2.1 Tâche de reconnaissance d'écriture manuscrite

Définition

La reconnaissance d'écriture manuscrite est le fait de reconnaitre le texte contenu dans une image hors ligne ou bien dans un format en ligne comme défini précédemment. La difficulté de reconnaissance varie notamment selon la taille du vocabulaire, l'alphabet à reconnaitre (latin, arabe, chinois ...), le nombre de données disponibles pour mettre au point nos modèles, la qualité des entrées (images anciennes dégradées, signal en ligne bruité ...) ainsi que la maitrise du geste graphomoteur de la personne qui écrit (débutant, expert). Dans ces travaux, nous nous intéressons à la reconnaissance d'écriture latine en ligne et hors ligne réalisée par des enfants.

Évaluation

Pour évaluer la performance d'un système de reconnaissance d'écriture, nous utilisons la distance d'édition de Levenshtein [Dam64] pour mesurer l'écart entre le texte reconnu et la vérité terrain. Cette distance est utilisée pour calculer un **taux d'erreur caractère** (en anglais Character Error Rate CER) et un **taux d'erreur mot** (en anglais Word Error Rate WER).

1.2.2 Tâche de segmentation d'écriture manuscrite

Définition

Dans notre contexte, la segmentation d'écriture manuscrite est le fait de localiser les lettres d'un mot ou d'une phrase. Dans une image, la segmentation en lettres peut être représentée à l'aide de boites englobantes ou bien de masques de segmentation sémantique, *i.e.* en classifiant tous les pixels correspondant à la lettre segmentée. Dans un signal en ligne, la segmentation en lettre est représentée par une séquence de points définissant la lettre.

Évaluation

Pour évaluer la qualité de segmentation d'un modèle, nous utilisons les métriques illustrées dans la figure 1.6. Nous distinguons l'évaluation d'une segmentation représentée par une **boite englobante** d'une segmentation représentée par **une suite de pixels** pour chaque caractère segmenté.

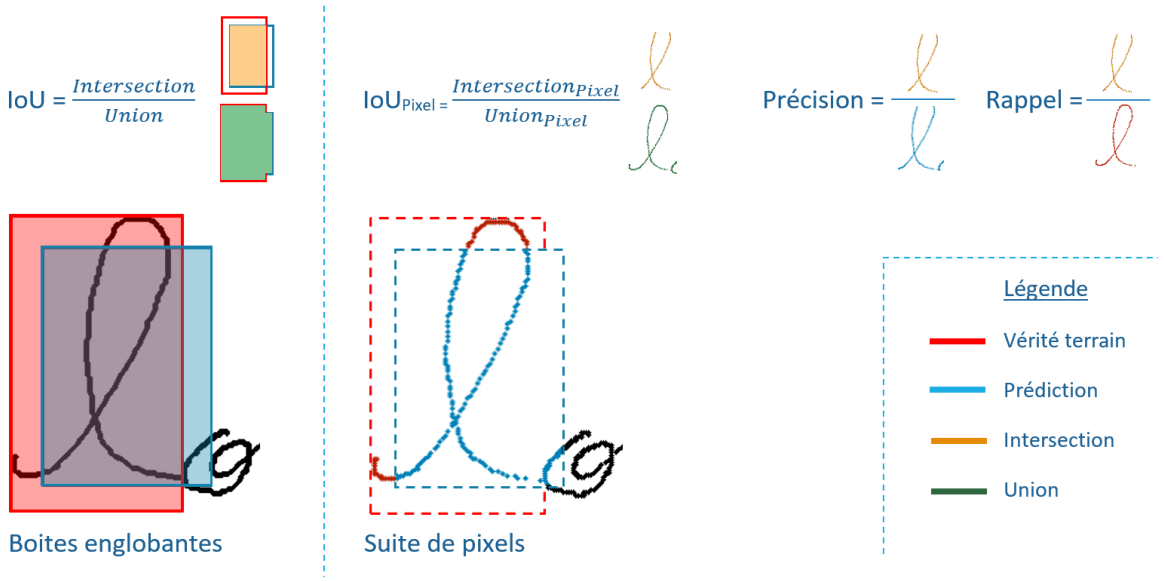

FIGURE 1.6 – Métriques considérées (IoU, IoU_{Pixel}, Précision, Rappel) pour évaluer la segmentation représentée par des boites englobantes et des suites de pixels.

La segmentation associée aux boites englobantes est évaluée à l'aide de la métrique *Intersection Over Union* (IoU) qui représente le ratio de l'air de la zone d'intersection et de la zone d'union entre la boite englobante de la vérité terrain (en rouge sur la figure) et la boite englobante de la prédiction (en bleu sur la figure). La métrique Intersection Over Union Pixel (IoU_{Pixel}) représente le ratio du nombre de **pixels d'écriture** (*i.e.* nous ne considérons que le tracé) contenus dans la zone d'intersection et du nombre de pixels d'écriture contenus dans la zone d'union. Dans le cas d'un signal en ligne, nous convertissons la séquence de points en une image. Les points sont reliés dans l'image afin d'éviter d'avoir un biais lié à la fréquence d'échantillonnage selon les tracés. La segmentation pixels d'écriture est évaluée à l'aide de métriques supplémentaires. La *précision* calcule le ratio du nombre de pixels de la zone d'intersection et du nombre de pixels de la prédiction. Le *rappel* calcule le ratio du nombre de pixels de la zone d'intersection et du nombre de pixels de la vérité terrain. Nous utilisons également le score F1 qui tient compte de la précision et du rappel de la manière suivante :

$$F1 = 2 * \frac{précision * rappel}{précision + rappel} \tag{1.1}$$

Les métriques évaluant la segmentation représentée par une suite de pixels ne tiennent

pas compte de l'arrière-plan et sont donc plus précises dans un contexte de segmentation d'écriture.

Nous pouvons noter qu'il est difficile d'évaluer précisément la qualité de segmentation d'écriture. La vérité terrain varie selon l'annotateur au niveau de la zone de ligature présente entre deux lettres d'une écriture cursive ainsi qu'au niveau de la classe associée au caractère dans le cas d'une écriture dégradée comme l'illustre la figure 1.7.

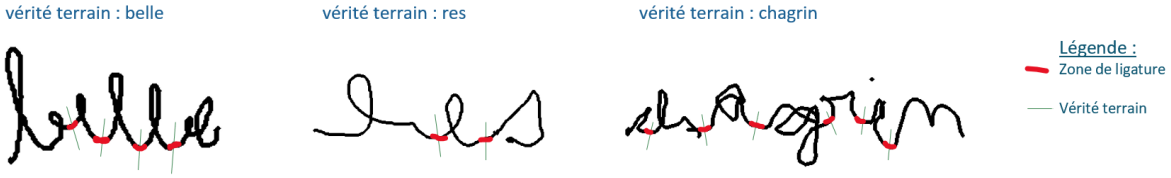

FIGURE 1.7 – Exemples d'écritures d'enfants dégradées.

1.3 Contexte éducatif : apprentissage de l'écriture

L'apprentissage de l'écriture manuscrite à l'école regroupe un ensemble de compétences à acquérir pour maitriser cette discipline essentielle. Au début, les élèves commencent par apprendre à écrire des fragments de lettres, des lettres puis des mots dans un contexte de recopie pour développer le geste graphomoteur nécessaire pour former correctement les lettres d'un mot. Ensuite, ils apprennent à travers des exercices de dictée l'orthographe. Nos travaux se situent dans l'analyse d'écriture dans un contexte de dictée dans le but de faire un retour automatique sur la qualité de l'orthographe. L'objectif ici n'est pas de remplacer le travail d'un enseignant, mais de fournir un outil d'aide à l'apprentissage pour faciliter la mise en place d'un atelier d'écriture guidé par un enseignant.

1.3.1 Contexte de recopie

Pour apprendre à écrire, il est nécessaire de maitriser le geste graphomoteur permettant de tracer des caractères. Ce geste s'apprend à travers des exercices de recopie où l'enfant voie une lettre ou un mot à recopier. L'objectif est d'évaluer la **qualité graphique** de l'écriture avec des critères de forme, de sens et d'ordre des éléments composants une lettre. La figure 1.8 montre un aperçu de Kaligo [2], une application d'aide à l'apprentissage de

2. https ://www.kaligo-apps.com/fr

l'écriture dans un contexte de recopie. Cette application a été développée dans le cadre du projet PIA Intuiscript [GSA17] et du Labcom Script & Labs [3]. L'académie de Rennes, la région Bretagne, le laboratoire Loustic, l'ESPE de Bretagne, la société Learn & Go [4] et l'équipe IntuiDoc de l'IRISA ont participé à ces projets. La figure illustre plusieurs scénarios avec des difficultés différentes. En effet, selon le niveau de l'élève, il devra écrire le mot lettre par lettre, par groupe de lettres ou le mot en entier. Le modèle d'analyse utilisé dans l'application Kaligo est détaillé dans le chapitre suivant.

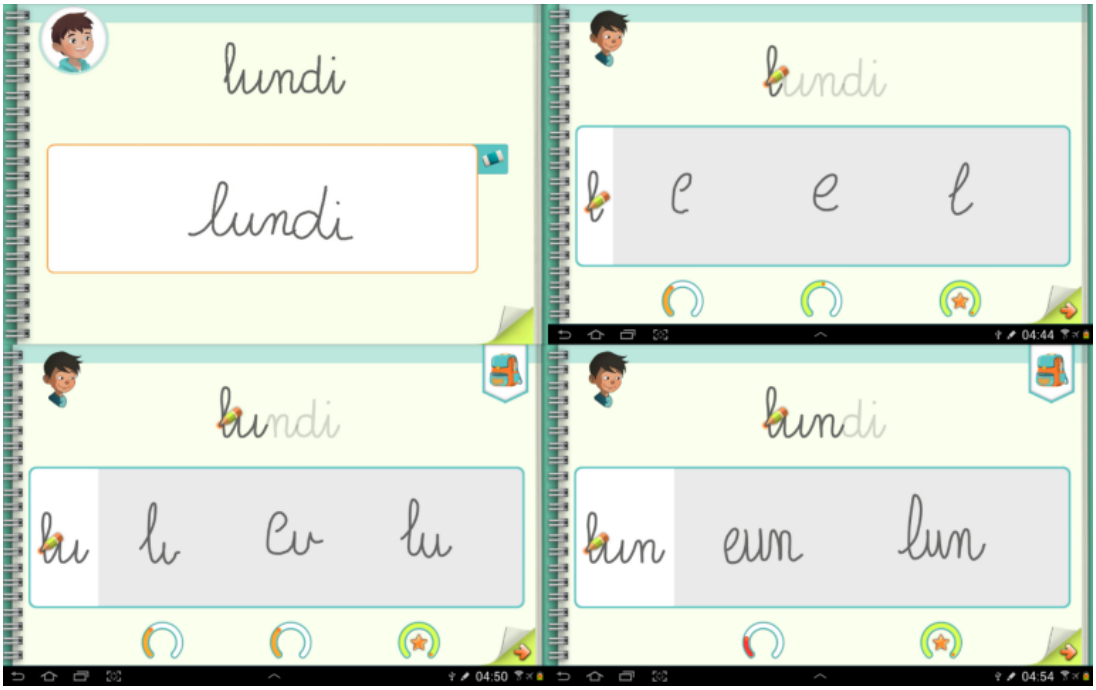

FIGURE 1.8 – Aperçu de l'application Kaligo, une application d'aide de l'apprentissage de l'écriture dans un contexte de recopie avec une granularité de difficulté différente pour le mot "lundi".

1.3.2 Contexte de dictée

Le but d'un exercice de dictée est d'évaluer la compétence **orthographique** de l'élève. Dans ce contexte, l'écriture peut encore être mal formée et contenir des erreurs orthographique (*cf.* figure 1.5) car la consigne est dictée. L'objectif est de faire un retour à l'élève sur la qualité de son orthographe, *i.e.* de mettre en évidence les erreurs.

3. https ://scriptandlabs.irisa.fr/
4. https ://learn-and-go.com/

1.4 Jeux de données utilisées

Dans cette partie, nous présentons les jeux de données d'écriture d'adultes et d'enfants utilisés dans ces travaux. Le jeu de données d'écriture d'enfants à notre disposition étant peu volumineux, nous avons utilisé également des écritures d'adultes disponibles publiquement pour développer nos systèmes.

1.4.1 Écriture d'adultes : IAM-OnDB

Le jeu de donnée IAM On-Line Handwriting (IAM-OnDB) a été publié en 2005 à la conférence ICDAR [LB05]. Il est composé d'écriture anglaise réalisée par des adultes. En plus du format hors ligne, les phrases écrites sont disponibles au format en ligne, c'est-à-dire qu'un tracé manuscrit est représenté par une suite de points. La figure 1.9 montre des exemples d'écriture du jeu de données. Le jeu de caractères contient des minuscules, des majuscules et quelques éléments de ponctuations. Le style d'écriture mélange le cursif et le script.

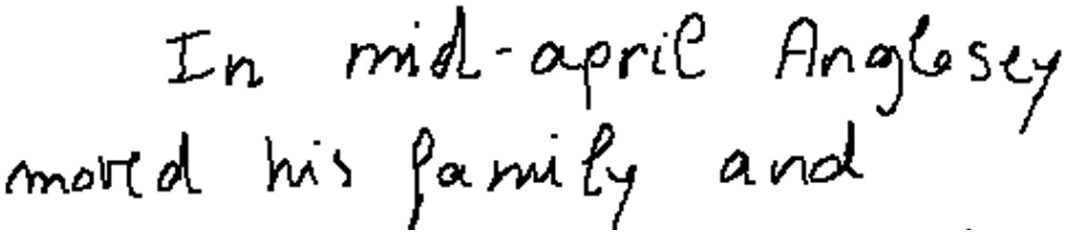

FIGURE 1.9 – Exemples d'écritures de la base IAM-OnDB.

1.4.2 Écriture d'enfants

L'équipe IntuiDoc de l'IRISA dispose de données représentant des écritures d'enfants. Ces données ont été récoltées au cours de différents projets menés en collaboration avec des écoles primaires en Bretagne, des acteurs académique, le laboratoire d'observation des usages des TIC (LOUSTIC) et la société Learn&Go. Ce jeu de données contient des lettres et des mots écrits par des enfants en cours d'apprentissage de l'écriture du français (alphabet latin). Les données ne sont pas accessibles publiquement en raison du Règlement Général sur la Protection des Données (RGPD). Les données ont été acquises dans un contexte de recopie où les enfants visualisaient le mot à écrire et un contexte de dictée où les enfants écoutent l'énoncé du mot à écrire. Les écritures ont été récoltées sur

des tablettes tactiles à l'aide d'un stylet et sont représentées par un signal en ligne. La figure 1.10 illustre différents exemples de mots contenus dans le jeu de données. Le jeu de données est composé de différents types d'écriture :

— Les mots sans fautes et bien écrit (exemple 1).
— Les mots avec une écriture bruitée (point pour l'exemple 2 et lettre "e" tracé deux fois pour l'exemple 3).
— Les mots avec des fautes phonétiques (exemple 4, 5), dyslexiques (exemple 6) ainsi que des fautes "autres" (exemple 7).
— Les écritures utilisant un style différent que celui enseigné (exemple 8).

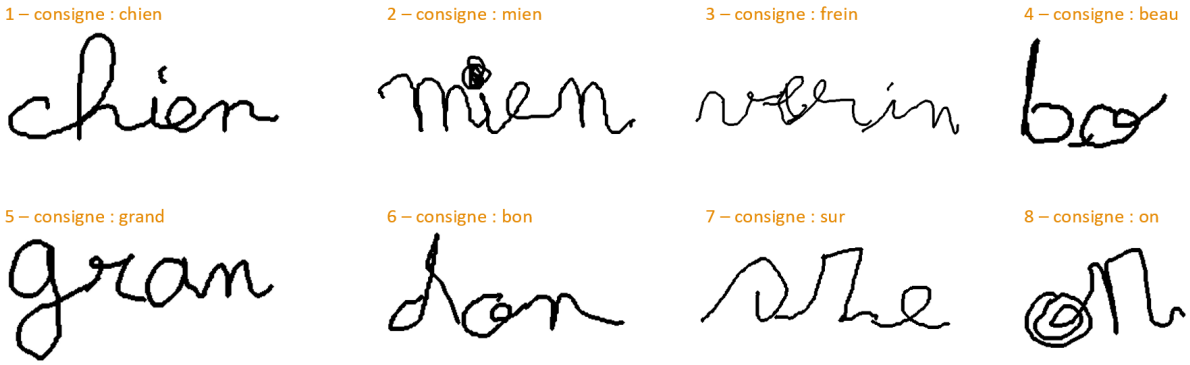

FIGURE 1.10 – Exemples d'écriture d'enfants.

Les données disponibles dans la base IntuiScript Children DataBase (ICDB) regroupent des écritures de lettres, de mots annotés au niveau mot (pas de vérité terrain de segmentation) et des mots annotés au niveau lettres. Le découpage des jeux de données utilisés dans nos expérimentations est résumé dans le tableau 1.1. Le jeu ICDB-Letters contient des caractères manuscrits écrits par 147 enfants. Il contient 27 000 caractères dans le jeu d'entrainement, 2 600 dans le jeu de validation et 7 086 dans le jeu de test. Les mots contenus dans la base ont été écrits par plus de 500 enfants pour les ensembles d'entrainement et plus de 300 enfants pour les ensembles de test. Les ensembles de données d'entrainement et de test proviennent de différentes campagnes d'acquisition de données (et de différentes salles de classe). Il n'y a pas de données d'enfants présentes à la fois dans l'ensemble d'entrainement et dans l'ensemble de test. Le jeu de données ICDB-Words-A contient 5 649 mots pour l'entrainement, 1 000 pour la validation et 913 pour le test. Le jeu de données ICDB-Words-B contient 6 812 mots pour l'entrainement et 1 242 pour le test. Nous utilisons une validation croisée avec ce jeu. Le jeu de données ICDB-Words-C contient 6 022 mots pour l'entrainement, 1 000 pour la validation et 1 032 pour le test.

Le jeu de données ICDB-Words-D est un sous-ensemble du jeu ICDB-Words-C où les ensembles d'entrainement et de validation sont annotés au niveau lettres. Pour les jeux de données mots précédents, la totalité de l'ensemble de tests, seulement est annotée au niveau lettres de manière à évaluer la reconnaissance et la segmentation. L'annotation au niveau lettres demandant beaucoup de ressources en temps, seuls les jeux de tests et une sous-partie du jeu d'entrainement ont été annotés à ce niveau de précision. Des données supplémentaires ont été récoltées et annotées au cours de ces travaux, ce qui explique la différence de découpage entre les différents jeux de données contenant des mots.

TABLE 1.1 – Type et découpages des jeux de données de la base IntuiScript Children DataBase (ICDB).

Jeu de données	Type données	Entrainement	Validation	Test
ICDB-Letters	Caractère	27 000	2 600	7 086
ICDB-Words-A	Mot	5 649	1 000	913
ICDB-Words-B	Mot	6 812	-	1 242
ICDB-Words-C	Mot	6 022	1 000	1 032
ICDB-Words-D	Mot	918	176	1 032

1.5 Synthèse de données

La synthèse de données ou augmentation de données est utilisée lorsque le nombre de données disponibles pour l'entrainement de modèle d'apprentissage automatique ou d'apprentissage profond est faible. L'objectif est de créer de nouvelles données en appliquant des déformations sur les données d'origine. Dans un contexte d'écriture manuscrite, nous distinguons deux catégories de synthèse. La synthèse hors ligne qui est appliquée sur des images et la synthèse en ligne qui est appliquée sur des suites de points. Cette partie décrit les différentes techniques de synthèse utilisées dans nos travaux.

1.5.1 Synthèse hors ligne

La première ligne de la figure 1.11 illustre les déformations d'étirement, d'inclinaison et de rotation d'une image représentant un caractère. La seconde ligne de la figure illustre un exemple de l'ajout de bruit suivant une distribution gaussienne dans une image. Cette technique permet, entre autres de réduire le "sur-apprentissage" d'un modèle d'apprentissage profond. La troisième ligne de la figure illustre un exemple de déformation de perspective. Comme nous pouvons le voir, cette technique permet de générer des images

déformées et de simuler une écriture qui n'est pas sur une ligne droite. Cette propriété est intéressante pour rendre les modèles de reconnaissance ou de segmentation plus robuste aux changements de scripteurs.

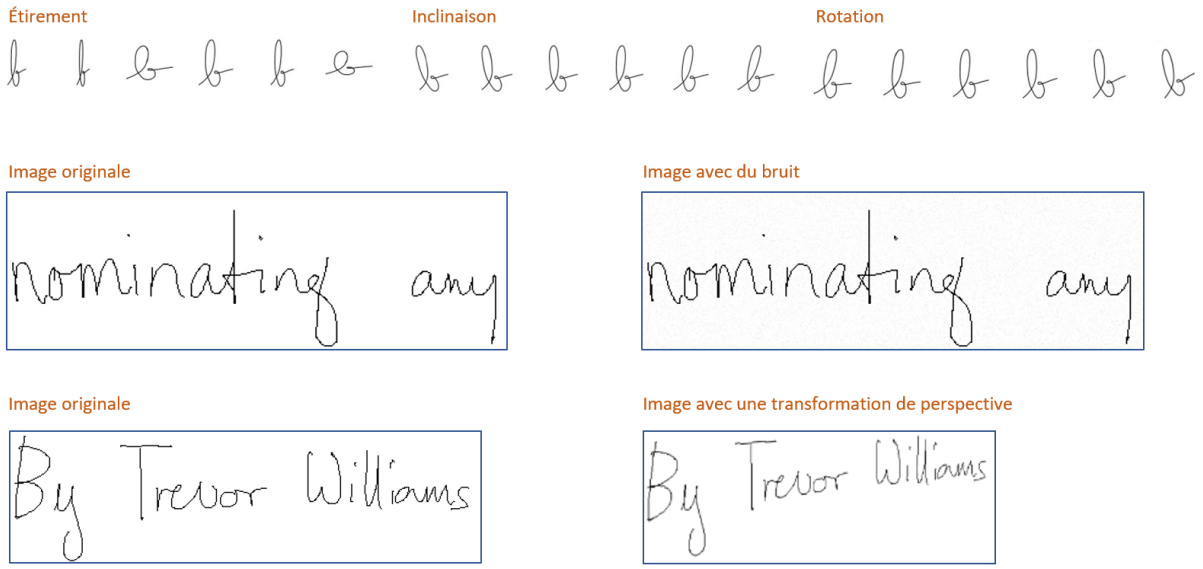

FIGURE 1.11 – Exemples de déformations hors ligne effectuées sur les mots de la base IAM-OnDB.

1.5.2 Synthèse en ligne

Nous utilisons une technique de déformation basée sur la trajectoire et la dynamique des boucles décrite dans l'article [Mou+07] et illustrée dans la figure 1.12. Une écriture au format en ligne contient les informations sur les posés et les levés de crayon ce qui permet d'appliquer des déformations indépendamment à chaque sous-partie du tracé comme la translation illustrée sur la figure dans le cas d'un caractère composé de plusieurs tracés. Nous appliquons également les déformations d'étirement, d'inclinaison et de rotation présentées précédemment.

FIGURE 1.12 – Exemples de déformations en ligne.

Chapitre 2

IntuiScript : système d'analyse d'écriture d'enfants

Le système d'**analyse d'écriture manuscrite d'enfants** IntuiScript a été développé au cours de plusieurs projets pour répondre à différents objectifs. Les utilisateurs cible sont les enfants de 4 à 7 ans (cycle 2) en cours d'apprentissage de l'écriture ainsi que les enseignants qui les accompagnent. Une conception centrée utilisateur est utilisée afin de développer un système répondant aux besoins pédagogiques des enseignants et des élèves. Un des objectifs des travaux présentés dans ce document a été d'améliorer ce système pour l'adapter à un contexte de dictée. Ce chapitre commence par faire un historique des différentes versions du système puis présente le fonctionnement général du système IntuiScript dans un contexte de recopie.

Figure 2.1 – Aperçu de l'application Kaligo qui utilise le système d'analyse d'écriture IntuiScript.

2.1 Historique des évolutions

2.1.1 Analyse de la qualité de l'écriture française

La première version du système a été développée dans le cadre du projet national d'investissements d'avenir IntuiScript. Ce projet s'est déroulé de 2014 à 2017. À travers, 40 classes pilotes, plus de 1 000 élèves de primaires de Bretagne ont participé au projet. Le projet a été dirigé conjointement par l'entreprise Learn&Go et l'équipe de recherche IntuiDoc du laboratoire IRISA. Ce projet est soutenu par la Région Bretagne, l'académie de Rennes, le laboratoire des usages LOUSTIC et l'INSPE de Bretagne. Il a pour objectif de concevoir un cahier numérique pour l'aide à l'apprentissage de l'écriture à l'école en utilisant des tablettes équipées d'un stylet. IntuiScript cible les élèves de la maternelle à l'école primaire. Ce projet a abouti au développement de l'application Kaligo qui est actuellement utilisée dans des écoles en France.

L'objectif est de délivrer des retours immédiats et précis sur la forme, le sens et l'ordre du tracé écrit par l'élève. L'application offre des scénarios pédagogiques adaptés aux difficultés de l'enfant (mot entier, groupe de lettres, lettre isolée). Le système a été développé pour des **exercices de recopies**, *i.e.* où l'enfant voit la consigne et doit la recopier. Les bénéfices en termes d'apprentissage de cette approche sont explicités dans l'article [Sim+19]. Les innovations scientifiques liées au projet IntuiScript ont donné lieu à plusieurs publications scientifiques [BA15] [SA16] [SAB17] [GSA17] [AGS17] [Bon+17].

2.1.2 Déclinaison pour d'autres langues

Le style d'écriture d'un alphabet peut varier en fonction de la langue, de la personne qui écrit comme expliqué dans la partie 1.1. Les moteurs de reconnaissance et d'analyse initialement développés pour le français ont été adaptés pour pouvoir traiter d'autres styles d'écriture pour d'autres langages (anglais et allemand).

Anglais

L'écriture de l'anglais et du français utilise l'alphabet latin. Si nous ne considérons pas les lettres accentuées et quelques caractères comme le "ç", les lettres à analyser sont les mêmes pour les deux langues. Cependant, les instructions des ministères de l'éducation anglais et français ne sont pas les mêmes en ce qui concernent les préconisations pour l'apprentissage de l'écriture. En effet, les élèves français apprennent à écrire des lettres

cursives puis des mots cursifs tandis que les élèves anglais apprennent à écrire des lettres scriptes puis des mots scripts attachés avec des ligatures. Les différences concernent la forme des lettres (script, cursif) et surtout la forme des **ligatures** pour relier les lettres entre elles. La figure 2.2 illustre des exemples de lettre "a" et "p" en anglais. La structure principale de la lettre est en vert et les ligatures en bleu. Pour pouvoir reconnaitre ce style d'écriture, les moteurs de reconnaissance et d'analyse ont été adaptés pour traiter ce nouveau style de ligature.

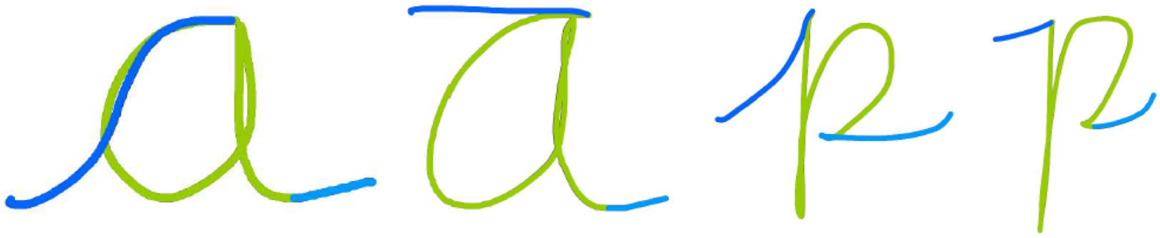

FIGURE 2.2 – Exemples de ligatures entrantes et sortantes pour les lettres "a" et "p. La structure principale de la lettre est en vert et les ligatures en bleu.

Allemand

Le système IntuiScript a été décliné pour la **langue allemande** pour un déploiement de l'application Kaligo au Luxembourg. Le système a été enrichi avec des moteurs d'analyse des chiffres, des lettres majuscules et minuscules scriptes. De **nouvelles formes de lettres** telles que la lettre Eszett (ß) ou les caractères trémas minuscules et majuscules (ä, ö, ü, Ä, Ö, Ü) ont également été ajoutés.

2.1.3 Reconnaissance et analyse de l'orthographe

Dans le cadre du projet P2IA [1] (Partenariat d'innovation et intelligence artificielle) qui s'est déroulé de nombre 2019 à août 2021, le système IntuiScript a évolué pour permettre une analyse fine de l'écriture dans un **contexte de dictée**, *i.e.* une analyse morphologique, orthographique et phonologique de l'écriture manuscrite de l'enfant. Nos travaux se situent dans ce contexte applicatif.

1. https ://eduscol.education.fr/1911/partenariat-d-innovation-et-intelligence-artificielle-p2ia

2.2 Spécifications dans un contexte de recopie

L'objectif du système est **d'analyser à partir de la consigne**, le tracé écrit par un enfant afin de lui apporter un retour immédiat sur la **qualité de son écriture**. Ce contexte est différent d'une tâche de reconnaissance d'écriture manuscrite classique où l'objectif est de **reconnaitre** le texte écrit. L'écriture de l'enfant est représentée par un signal en ligne. La figure 2.3 liste les différentes étapes du processus :

— **Segmentation** : le tracé manuscrit est segmenté à l'aide de connaissances expertes liées au domaine de l'écriture. Les hypothèses de segmentation sont organisées dans un treillis.

— **Analyse** : pour chaque hypothèse du treillis de segmentation, le système calcule les scores d'analyse lettre représentant la qualité d'écriture.

— **Calcul des chemins** : un chemin est une séquence d'hypothèses du treillis représentant un découpage du tracé d'origine. Cette étape parcourt le treillis pour définir tous les chemins et calcule un score d'analyse au niveau mot.

— **Retour utilisateur** : le chemin avec le meilleur score est choisi pour faire un retour sur la qualité de l'écriture à l'utilisateur.

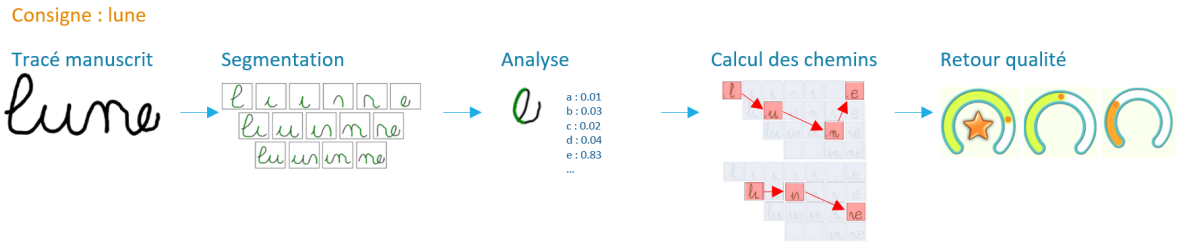

FIGURE 2.3 – Étapes du processus d'analyse d'un mot manuscrit du système IntuiScript.

Le système IntuiScript utilise une **approche analytique** pour analyser le mot, c'est-à-dire qu'il va chercher à reconnaitre les lettres pour pouvoir reconnaitre un mot. La suite de ce chapitre présente le fonctionnement général de chaque étape du processus d'analyse.

2.2.1 Segmentation

La méthode pour diviser le tracé en un treillis de segmentation est inspirée de l'article [AL97] puis consolidé dans l'article [Sim+19]. Elle s'appuie sur la notion de points d'ancrage visuels pour découper le tracé.

Méthode de segmentation du tracé

La notion de points d'ancrage visuels s'inspire du fonctionnement de l'œil chez l'Homme. Lors du processus de lecture, le cerveau va chercher à découper la séquence à lire en sous-parties. Ainsi, une phrase est découpée en mots, un mot en lettres, et une lettre en primitives. Le cerveau cherche des points particuliers du tracé pour effectuer ce découpage. Les **points d'ancrage visuels** permettent de découper un tracé manuscrit en sous-composants. Ces points permettent de délimiter deux courbes qui n'ont pas la même topologie et ainsi distinguer deux formes distinctes. Selon René Thom [Tho76], chaque changement dans une forme est une "morphogenèse", *i.e.* la génération d'une nouvelle forme qui correspond à une "catastrophe". Elle correspond à un changement brutal d'une des propriétés de la forme. Les "points catastrophes" et les points d'ancrage visuels utilisés dans ses travaux sont illustrés sur la figure 2.4 et se divisent en plusieurs catégories :

- Les points de discontinuité correspondent aux levés de crayon ;
- Les points angulaires où les dérivées à gauche et à droite sont distinctes et finies ;
- Les points de rebroussement où les dérivées à gauche et à droite sont distinctes et infinies ;
- Les extremums locaux en X et Y.

En écriture, le **ductus** est l'ordre et la direction, mais aussi la vitesse et le rythme [2], selon lesquels nous traçons les traits qui composent la lettre. Chaque type d'écriture possède un ductus propre qu'il convient de respecter pour assurer une écriture fluide et naturelle.

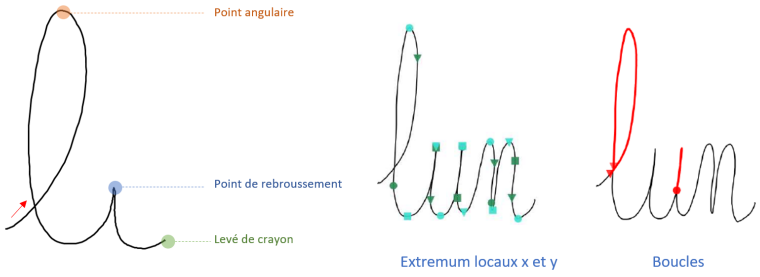

FIGURE 2.4 – Exemple de points d'ancrage visuels.

La méthode de segmentation utilise les points d'ancrage visuels pour découper le tracé en une **segmentation primaire** où chaque élément de la segmentation primaire

2. https ://fr.wikipedia.org/wiki/Ductus

peut être qualifié de tracé ascendant ou descendant. Une lettre est composée de plusieurs éléments ascendants et descendants de la segmentation primaire. Cela permet de définir des **modèles de lettres** basés sur le nombre d'éléments descendants.

Assemblage dans un treillis de segmentation

En mathématiques, un treillis est un ensemble partiellement ordonné dans lequel chaque paire d'éléments admet une borne supérieure et une borne inférieure [3]. La figure 2.5 illustre le processus de création du treillis à partir de la segmentation primaire où les tracés descendants sont entourés en vert. Le premier niveau du treillis est construit en assemblant un tracé descendant avec le tracé ascendant qui précède et le tracé ascendant qui lui succède. Les éléments des niveaux suivants résultent de la fusion de deux éléments du niveau précédent. Chaque niveau représente un niveau de granularité différent. Le premier niveau contient les segmentations des lettres contenant un tracé descendant comme le "l", le second les segmentations des lettres contenant deux tracés descendant et ainsi de suite. Ce découpage créé un treillis où le nombre d'hypothèses de segmentation est réduit. Le parcours du treillis définit les hypothèses de découpage du mot appelées **chemins de segmentation**. Un chemin de segmentation couvre la totalité de l'encre (*i.e.* le tracé) représentant le mot. Il y existe au moins un chemin dans le treillis correspondant à la segmentation lettres du mot.

Détection des diacritiques

Un signe diacritique est un signe accompagnant un symbole, une lettre pour en modifier la prononciation ou le sens. Pour le français, il s'agit des points sur les lettres "i" et "j" ainsi que les accents. Par abus de langage, le système désigne également comme diacritique la barre du "t" car elle est souvent tracée *a posteriori* de même que les accents ou les points. Avant de découper le tracé en segmentation primaire, le système détecte la présence de diacritique à l'aide de critères temporels et spatiaux. Les diacritiques sont liés aux hypothèses du treillis de segmentation les plus proches.

2.2.2 Analyse niveau lettre

L'étape d'analyse consiste à calculer les **hypothèses de lettres** pour chaque élément du treillis de segmentation. L'analyse s'effectue en deux temps. Dans un premier

3. https ://fr.wikipedia.org/wiki/Treillis_(ensemble_ordonné)

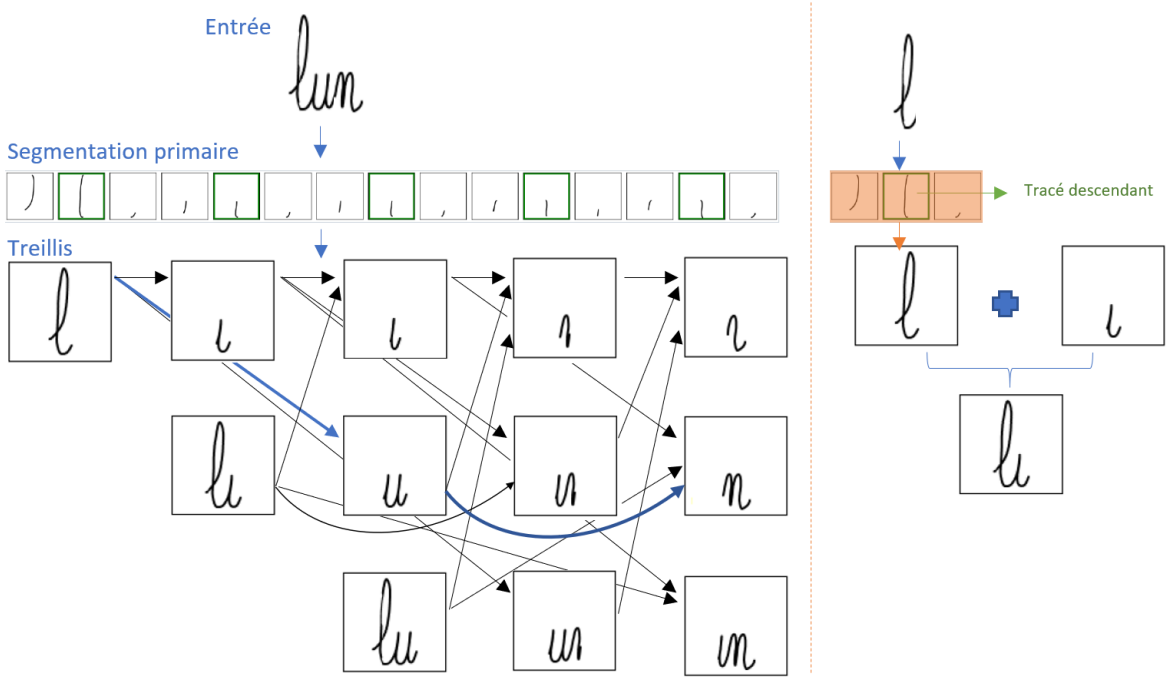

FIGURE 2.5 – Exemple de création du treillis à partir de la segmentation primaire pour la séquence de caractère "lun" à gauche et pour la lettre "l" à droite.

temps, le système extrait des caractéristiques du tracé. Ces caractéristiques sont dérivées du jeu HBF49 [DA13] qui est un ensemble de caractéristiques générique pour faire de la reconnaissance de gestes composés de caractéristiques dynamiques et visuelles [SA16]. Ces caractéristiques sont utilisées par un **classifieur de lettre** pour attribuer un **score discriminatif** pour chaque lettre de l'alphabet. Le système filtre les hypothèses de reconnaissance de manière à garder l'hypothèse de la lettre avec le meilleur score ainsi que les trois meilleures hypothèses pour les lettres présentes dans la consigne. Ce filtrage correspond à un premier **mécanisme de guidage**. Le système vérifie également que l'hypothèse de segmentation remplie les critères associés au modèle de l'hypothèse de lettre (nombre minimum et maximum de tracés descendants).

Dans un second temps, un second classifieur [SAB17] basé sur un **modèle discriminatif et génératif** est utilisé pour affiner les scores de reconnaissance des hypothèses filtrées en des **scores d'analyses**. Il utilise un score inter-classe et intra-classe comme l'illustre la figure 2.6. Le modèle discriminatif permet de déterminer la différence entre les modèles. Le modèle génératif sert à savoir si un tracé correspond bien à une lettre. Le score d'analyse représente la **qualité de l'écriture** de la lettre.

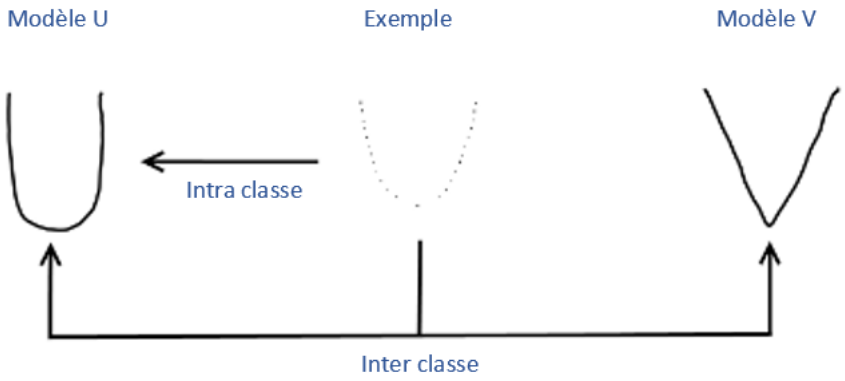

FIGURE 2.6 – Exemple de distance intra-classe entre le tracé et la classe "U". Exemple de distance inter-classe entre le tracé et les modèles "U" et "V".

2.2.3 Calcul des chemins niveau mot

L'objectif est de trouver dans le treillis le **chemin de segmentation** correspondant au découpage lettres du mot à analyser. Pour commencer, le système définit tous les chemins de segmentation possibles dans le treillis. Un chemin de segmentation est une suite d'hypothèses du treillis correspondant à un découpage possible du tracé d'origine où chaque point du tracé est utilisé une seule fois. Chaque hypothèse du treillis possède plusieurs hypothèses de lettres. Un chemin de segmentation est associé à plusieurs hypothèses de reconnaissance "mot" comme l'illustre la figure 2.7. Pour chaque hypothèse de reconnaissance "mot" le système calcule un **score d'analyse au niveau mot**, un **score n-gram** et un **score de cohérence spatiale**. Le score d'analyse au niveau mot $s_{analyse}$ est obtenu en multipliant les scores d'analyse de chaque **hypothèse de caractère**. Le score n-gram s_{ngram} prend en compte si les hypothèses de caractères font partie de la consigne. Le score de cohérence spatiale $s_{cohérencespatiale}$ utilise un modèle de logique floue pour déterminer si la correspondance de forme de deux lettres consécutives est cohérente. Le score d'une hypothèse de mot s_{mot} pour un chemin de reconnaissance est calculé en additionnant ces trois scores :

$$s_{mot} = 0.5 * s_{analyse} + 0.25 * s_{ngram} + 0.25 * s_{cohérencespatiale} \tag{2.1}$$

Le système calcule une distance d'édition entre la prédiction et la consigne pour les 100 meilleures hypothèses de mot. Le meilleur chemin est choisi de manière à maximiser le score d'analyse et de minimiser la distance d'édition. Nous pouvons noter que le temps

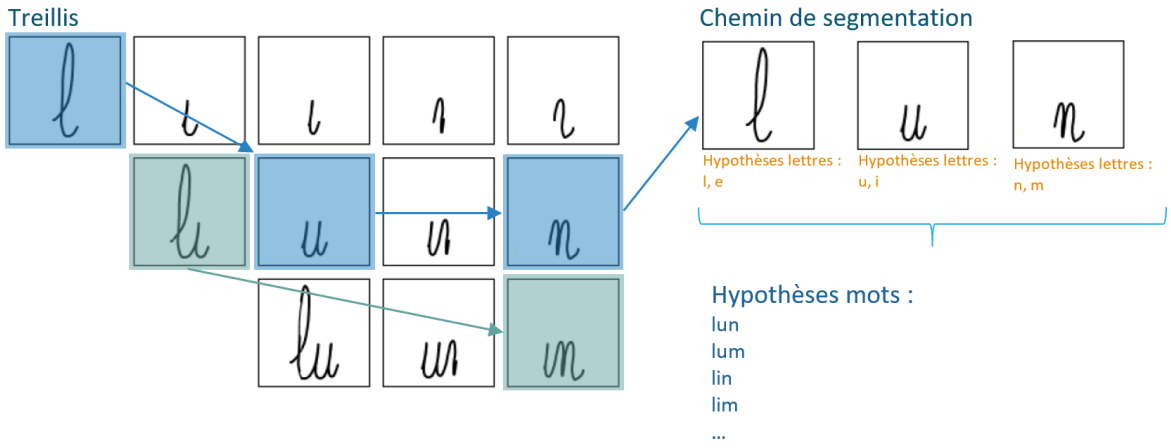

FIGURE 2.7 – Exemple de calcul de chemins de segmentation et des hypothèses de mots associées.

de calcul dépend de la longueur du mot à analyser. Plus un mot sera long, plus le treillis sera grand et plus il y aura de chemins à analyser.

2.2.4 Retour utilisateurs sur la qualité d'écriture

Le système utilise le score du meilleur chemin calculé dans l'étape précédente pour faire un retour immédiat à l'enfant sur la qualité de son écriture. La figure 2.8 illustre des exemples de retours dans l'application Kaligo pour une écriture avec une qualité haute, moyenne et basse. Le retour est représenté par une jauge utilisant un code couleur différent selon la qualité de l'écriture. Une étoile est ajoutée à la jauge pour indiquer à l'élève que son écriture remplie les attentes de l'enseignant.

FIGURE 2.8 – Exemples de retours qualités dans l'application Kaligo. La consigne est donnée sur la partie supérieure, l'écriture de l'enfant dans la partie intermédiaire et le retour dans la partie inférieure.

2.2.5 Extension pour un contexte de dictée : ajout d'un guidage phonétique

Initialement, le système IntuiScript a été développé pour analyser des mots sans fautes d'orthographe dans un contexte de recopie. Le système a besoin de connaissances *a priori* pour guider l'analyse. L'objectif de cette extension est d'adapter le système de guidage pour pouvoir analyser des mots contenant des fautes de nature phonétique. Dans cette partie, nous présentons les principales étapes du guidage phonétique décrit en détail dans l'article [Kri+21].

Intégration du guidage phonétique

Le guidage phonétique est basé sur un modèle Phonetisaurus [Nov+12] qui est un modèle Graphème vers Phonème (G2P) stochastique WFST (transducteur à état fini pondéré ou "Weighted Finite State Transducer" en anglais). Le WFST est basé sur le principe des séquences jointes [BN08] pour aligner les séquences de graphèmes avec leurs phonèmes correspondants. Un modèle N-Gram est généré à partir des séquences jointes alignées et transformé en un modèle WFST. Le modèle G2P est alors capable de prédire la prononciation d'un nouveau mot. Pour être adapté à notre problématique, le modèle G2P est combiné avec un modèle P2G (Phonème vers Graphème) afin que la sortie du modèle combiné, étant donné un nouveau mot, soit un ensemble de pseudo-mots phonétiquement similaires. Nous choisissons de générer, pour chaque mot consigne, les 50 meilleures hypothèses selon le classement du modèle Phonetisaurus. Ce modèle a été développé pour le français. L'objectif est de générer des mots contenant des fautes qu'aurait pu commettre l'enfant pour guider le système d'analyse avec une approximation de la vérité terrain plus fiable.

Les hypothèses générées par le Phonetisaurus sont utilisées pour guider l'analyse dans les étapes suivantes comme l'illustre la figure 2.9. Le calcul des hypothèses de lettres dans le treillis de segmentation est guidé par la consigne et les mots générés par le Phonetisaurus. Le score N-Gram de chaque chemin prend en compte les mots générés en plus de la consigne. Le score d'édition de chaque chemin dépendra alors des hypothèses phonétiques générées. La distance d'édition n'est pas suffisante pour choisir la bonne hypothèse puisqu'il peut y avoir deux chemins concurrents ayant le même score d'édition. Pour résoudre ce problème, le système inclut le score d'analyse dans les critères de décision, car ce score reflète la qualité de l'écriture manuscrite et permet au moteur de discriminer les chemins

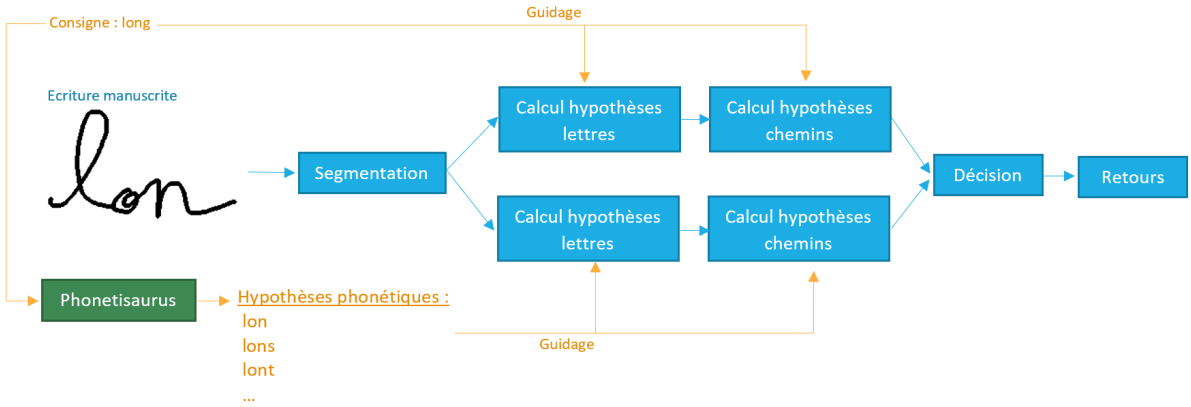

FIGURE 2.9 – Intégration du guidage phonétique dans le système IntuiScript.

concurrents. Le score d'analyse phonétique est défini comme suit :

$$Score(chemin) = \frac{1}{1 + ScoreEdition(chemin)} * 0.3 + ScoreAnalyse(chemin) * 0.7 \quad (2.2)$$

L'utilisation de toutes les lettres des hypothèses phonétiques dans le guidage dans l'étape de **calcul des hypothèses de lettres** permet d'envisager plus de possibilités dans l'étape de construction des chemins comparé à l'approche utilisant seulement les lettres de la consigne. Cependant, comme expliqué dans la partie 2.2.2, le système garde les trois meilleurs lettres contenues dans la consigne ou dans les mots phonétiquement proches. Dans certains cas, l'hypothèse correcte de la lettre n'a pas un bon score d'analyse et est ignorée à cause de ce filtrage. Pour éviter ce problème, le système intègre la notion de compétition d'analyse. Cette compétition se fait entre deux instances d'analyse afin de trouver le meilleur chemin : la première analyse (analyse de base) est guidée par la consigne et la seconde (analyse phonétique) est guidée par les hypothèses phonétiques liées au mot attendu. Si le meilleur chemin d'analyse de base est égal au mot attendu, son score d'analyse phonétique est calculé et comparé au score d'analyse phonétique du meilleur chemin d'analyse phonétique. Le chemin avec le score le plus élevé est choisi comme résultat final de l'analyse. Si dans l'analyse de base le meilleur chemin est différent du chemin attendu, le meilleur chemin de l'analyse phonétique est considéré comme le résultat final. Ce processus permet au moteur de récupérer une partie des mots correctement écrits qui ont été mal interprétés auparavant.

Retour utilisateurs sur la qualité orthographique

L'objectif d'un système d'analyse d'écriture dans un contexte de dictée est de faire des retours à l'enfant sur la qualité de l'orthographe, *i.e.* de mettre en évidence les éventuelles fautes présentes dans son écriture. Le système IntuiScript segmente en lettres un mot, ce qui permet de faire un retour précis sur les différences entre la consigne et le mot écrit comme l'illustre la figure 2.10 où le code couleur utilisé dans ces exemples est donné à titre indicatif.

FIGURE 2.10 – Exemples de retours sur la qualité de l'orthographe. Dans les exemples C à F, le système n'est pas suffisamment confiant pour produire un retour. Il s'agit de cas de rejet.

Dans l'exemple A où la consigne est "alors", le mot écrit contient des erreurs phonétiques. La lettre "l" à supprimer est en rouge et la lettre "e" à remplacer par un "s" est en orange. L'exemple B illustre des erreurs d'accentuation. Le premier "e" est en orange pour indiquer qu'il est à remplacer par un "e" avec un accent aigu, l'accent grave est en orange pour indiquer qu'il faut le remplacer par un accent aigu et l'accent circonflexe à supprimer est en rouge. Le système intègre un mécanisme de rejet. Lorsque le mot reconnu par le système est trop éloigné de la consigne ou des mots phonétiquement proche alors le mot est rejeté (code couleur bleu). L'objectif de ce rejet est d'éviter de faire des retours erronés lorsque la prédiction du système n'est pas fiable. Un exemple de retour associé au rejet est d'inviter l'enfant à réécrire le mot. L'exemple C illustre un cas où il est préférable de rejeter le mot, car l'écriture est mal formée. Les exemples D, E et F illustrent des erreurs non-phonétiques où le système ne reconnait pas les mots, car le système de guidage n'est pas adapté. Les mots reconnus étant éloignés de la consigne ou des mots phonétiquement

proches, ils sont considérés comme du rejet. Ces exemples mettent en lumière les limites du système de guidage phonétique dans un contexte de dictée.

2.3 Bilan

IntuiScript est un système d'analyse d'écriture manuscrite d'enfants basé sur une segmentation explicite ainsi qu'un mécanisme de guidage. L'objectif est d'analyser un mot à partir de connaissance *a priori* ce qui est différent d'une tâche de reconnaissance d'écriture. La version d'IntuiScript utilisant la consigne comme connaissance pour guider son analyse obtient de bons résultats dans un contexte de recopie. Cette version est actuellement utilisée dans l'application d'aide à l'apprentissage de l'écriture Kaligo. Une extension du système propose d'utiliser un guidage basé sur les mots phonétiquement proches de la consigne pour une utilisation dans un contexte de dictée. Cependant, cette extension ne permet pas d'analyser des mots avec des erreurs de nature non-phonétique. Dans nos travaux, nous avons cherché à perfectionner le système IntuiScript au niveau du mécanisme de guidage à l'aide d'un modèle d'apprentissage profond présenté dans le chapitre suivant pour pouvoir analyser des erreurs non-phonétiques. De plus, le temps d'analyse dépend de la longueur du mot à traiter. Pour des mots longs, la complexité de l'analyse augmente et le système ne peut pas fournir un retour immédiat à l'enfant. Dans nos travaux, nous avons également cherché à optimiser le temps d'analyse pour pouvoir faire un retour immédiat pour des mots longs.

Chapitre 3

Méthodes d'apprentissage profond

L'apprentissage profond est une technique d'apprentissage automatique (un sous-domaine de l'intelligence artificielle). L'apprentissage automatique vise à apprendre un modèle mathématique pour résoudre une tâche spécifique à partir d'un ensemble de données exemples dans le but de généraliser ensuite sur des données inconnues. L'apprentissage profond est grandement inspiré de la biologie [Let+59] [HW59] [HW62] [Hub95] et du fonctionnement du cerveau puisque le modèle mathématique appris est un réseau de neurones. Les modèles d'apprentissage profond sont généralement complexes et nécessitent un grand nombre de données pour fonctionner correctement. Ils font partie aujourd'hui des modèles de l'état de l'art ayant les meilleures performances dans de nombreux domaines. Ils sont utilisés notamment pour répondre à des problématiques de classification [LeC+98], de régression [Ren+15], de traduction de texte [Vas+17], de génération d'une légende pour une image [Vin+15], de résumé de texte [SLM17] ou encore de reconnaissance d'écriture [Mic+] [Bar+21]. Nous nous intéressons à des modèles d'apprentissage profond permettant de faire de la reconnaissance et de la segmentation d'écriture manuscrite. Nous considérons que le lecteur est familier avec les concepts généraux liés au domaine de l'apprentissage profond présentés en détails dans le livre [GBC16]. L'objectif de ce chapitre est de rappeler l'état de l'art et notamment le fonctionnement des différents réseaux utilisés dans nos travaux.

Dans ce chapitre, nous commençons par présenter les réseaux de neurones convolutifs répondant à une problématique de reconnaissance de caractères manuscrits. Ensuite, nous définissons des réseaux "séquence à séquence" dans le cadre de la reconnaissance de mots manuscrits. Enfin, nous détaillons le fonctionnement d'un réseau de neurones permettant de faire de la détection d'objets pour une application de détection de caractères dans un mot manuscrit.

3.1 Tâche de reconnaissance de caractères

Les réseaux de neurones convolutifs (en anglais CNN : Convolutionnal Neural Network) sont un type d'architecture d'apprentissage profond composée de couches de convolutions, de niveaux de sous-échantillonnage et de couches entièrement connectées. Nous avons choisi d'utiliser ce type de réseau, car ils obtiennent de très bonnes performances notamment pour des tâches de classification d'images qui sont proches de notre tâche cible. Nous nous intéressons à l'utilisation des CNN dans un contexte de **reconnaissance de caractères**. Dans nos travaux (*cf.* chapitre 4), nous comparons l'impact de l'utilisation d'un réseau peu profond (LeNet-5), d'un réseau profond (VGG) et d'un réseau profond avec des connexions résiduelles (ResNet). Cette partie présente ces architectures. Nous pouvons noter que plus les architectures sont complexes (*i.e.* plus elles ont de paramètres à apprendre), plus elles peuvent être utilisées pour des tâches difficiles, mais nécessitent aussi des phases d'entrainement plus longues sur des données volumineuses. Une architecture plus complexe peu également nécessiter plus de temps de calcul pour produire un résultat. Il a donc été nécessaire au début de cette thèse de calibrer la taille des réseaux offrant un bon compromis performance/temps pour notre tâche de reconnaissance de caractère.

LeNet-5. L'architecture LeNet-5 [LeC+98] est un réseau de neurones convolutifs relativement simple, mis au point, au départ, pour faire de la classification de chiffres manuscrits (jeu de données MNIST [Den12]). La figure 3.1 illustre l'architecture du réseau. LeNet-5 est un réseau peu profond composé de deux couches de convolutions, deux couches de sous-échantillonnage et de deux couches entièrement connectées. Il possède environ 60 000 paramètres à optimiser lors du processus d'apprentissage.

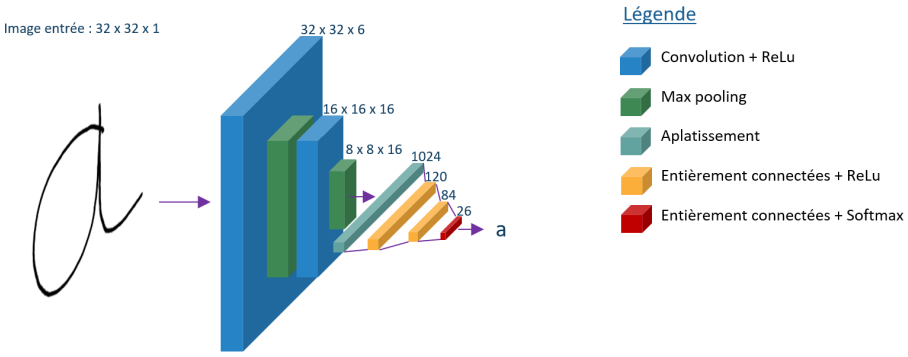

FIGURE 3.1 – Architecture du réseau LeNet-5.

VGG. L'architecture VGG [SZ15] se décline sous différentes formes plus ou moins profondes. Elle obtient de très bonnes performances sur l'épreuve de classification d'images "ImageNet" [Rus+15]. La figure 3.2 illustre l'architecture VGG avec 11 couches composée d'environ 28 millions de paramètres.

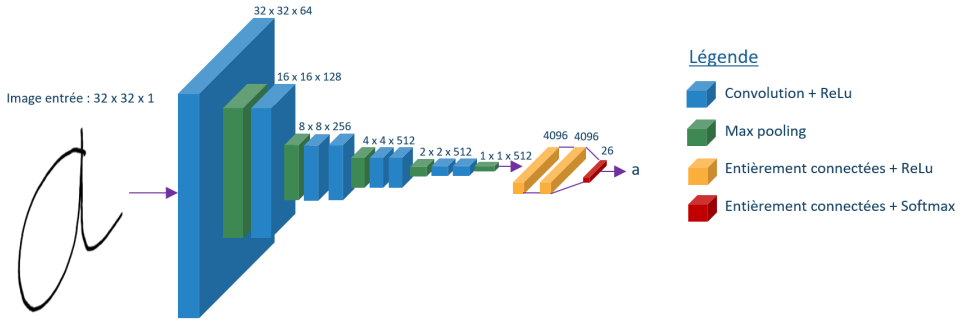

FIGURE 3.2 – Architecture du réseau VGG avec 11 couches.

ResNet. L'architecture ResNet [He+16] introduit la notion de connexion résiduelle dans le but de faciliter l'apprentissage de réseaux plus profonds. Comme VGG, cette architecture s'est montrée très performante sur l'épreuve "ImageNet". La figure 3.3 montre un exemple d'architecture ResNet avec 18 couches composé d'environ 11 millions de paramètres.

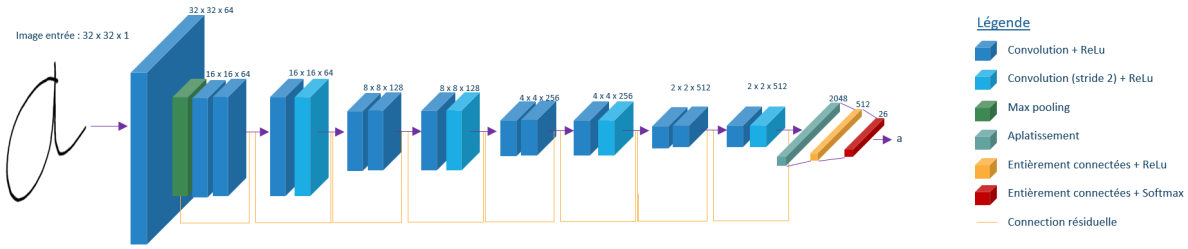

FIGURE 3.3 – Architecture du réseau ResNet avec 18 couches.

L'objectif est de concevoir un reconnaisseur de caractères manuscrits cursifs isolés basé sur un CNN pour remplacer le reconnaisseur présent dans un système d'analyse de mots existant. Notre contribution consiste à injecter la dynamique du tracé manuscrit en plus de l'information sur la forme dans un CNN afin d'améliorer la performance de reconnaissance. Nous avons choisi ces trois architectures afin de trouver un bon compromis entre performance et temps de calcul. Les travaux relatifs à la reconnaissance de caractères sont détaillés dans le chapitre 4.

3.2 Tâche de reconnaissance de mots

Les modèles d'apprentissage profond type "séquence à séquence" (Seq2Seq) permettent de résoudre notamment des tâches de traduction de texte [Vas+17], de génération d'une légende pour une image [Vin+15], de résumé de texte [SLM17] ou encore de reconnaissance d'écriture manuscrite [Mic+]. Ils font partie des approches holistiques qui considèrent la séquence d'entrée comme un tout et ne cherche pas à la segmenter pour la transformer contrairement aux approches analytiques qui segmentent la séquence d'entrée pour la transformer. Nous avons choisi ce type de modèle, car ils font partie des modèles les plus performants de l'état de l'art notamment pour faire de la reconnaissance d'écriture. Ce type de modèle permet de traiter des **séquences d'entrée et de sortie de taille variables** comme l'illustre la figure 3.4 avec un exemple d'utilisation d'un modèle Seq2Seq pour une tâche de traduction de l'allemand en français et un exemple pour une tâche de reconnaissance d'écriture manuscrite.

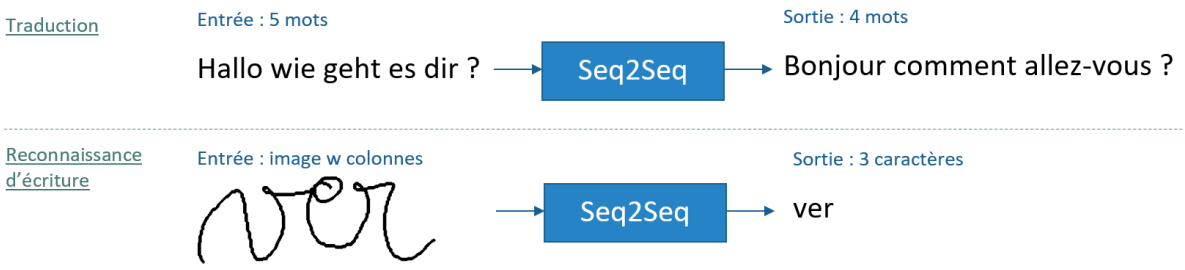

FIGURE 3.4 – Exemples d'application de modèles Seq2Seq pour une tâche de traduction et une tâche de reconnaissance d'écriture.

L'intérêt de ce type de modèle est que des données annotées au niveau séquence suffisent pour l'entrainement. Dans l'exemple de reconnaissance d'écriture, les données n'ont pas besoin d'être annotées au niveau caractère, *i.e.* qu'il n'est pas nécessaire de connaitre la position précise de chaque caractère pour apprendre le modèle. Seule l'annotation du mot contenu dans l'image est utile pour optimiser (*i.e.* apprendre) le modèle. Cela représente un gain de temps non négligeable dans le processus d'annotation des données. Dans nos travaux, nous utilisons des modèles Seq2Seq pour faire de la **reconnaissance de mots manuscrits**. Nous commençons par présenter une méthode qui permet d'entrainer un modèle pour un problème "séquence à séquence". Ensuite, nous spécifions les deux architectures utilisées dans nos travaux.

3.2.1 Méthode d'entrainement : Connectionist Temporal Classification (CTC)

Principe général

La méthode Connectionist Temporal Classification (CTC) [Gra+06] est une fonction de coût permettant d'entrainer des modèles d'apprentissage profond type Seq2Seq. Elle gère l'alignement entre une séquence de données d'entrée et une séquence de sortie de taille variable. Nous nous intéressons ici à l'utilisation de la CTC dans un contexte de reconnaissance d'écriture. La figure 3.5 illustre l'utilisation de la CTC pour entrainer un réseau Seq2Seq. Le réseau produit en sortie une séquence de longueur T où chaque élément correspond aux probabilités d'observer les caractères de l'alphabet $\mathcal{A}$ ainsi que la probabilité d'observer la classe "blank" (représenté par un "-" sur la figure) qui sert à définir un "non caractère". La sortie du réseau définit les probabilités de tous les alignements de séquences d'étiquettes de longueur T possible avec la séquence d'entrée. y_k^t représente la probabilité d'observer l'étiquette k pour l'élément t de la séquence de sortie. La probabilité d'un alignement π de longueur T définit sur l'alphabet $\mathcal{A}$' où $\mathcal{A}' = \mathcal{A} \cup \{\text{blank}\}$ en fonction des entrées x de la séquence d'entrée est défini par :

$$p(\pi|x) = \sum_{t=0}^{T} y_{\pi_t}^t, \forall \pi \in \mathcal{A}'^T \tag{3.1}$$

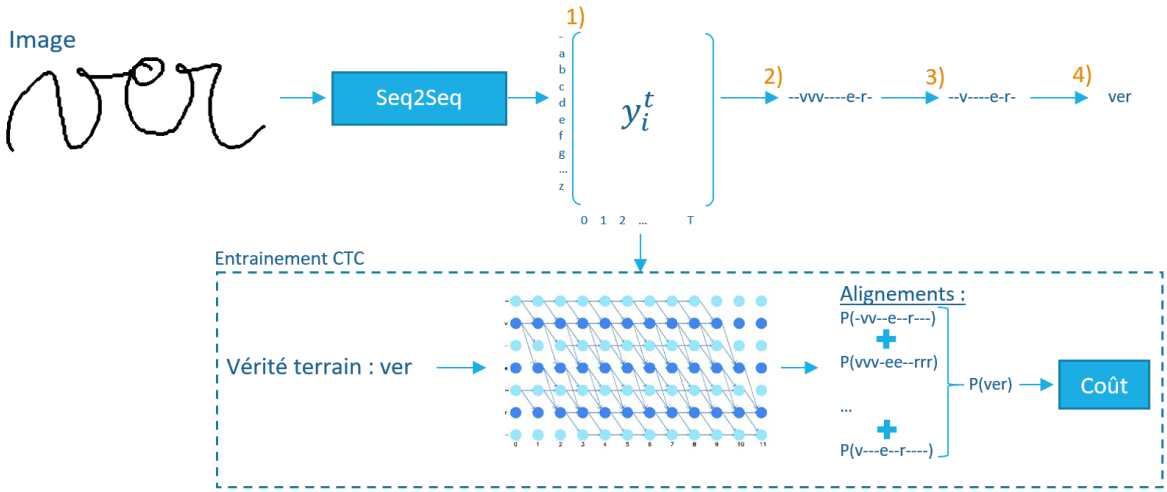

FIGURE 3.5 – Exemple de fonctionnement d'un réseau Seq2Seq utilisant la CTC.

La méthode définie une fonction $\mathcal{B}$ qui fait la correspondance entre la séquence de

caractères de sortie du réseau et la séquence de caractères finale qui sera produite. La fonction enlève simplement les répétitions de caractère et les caractères "blanks" de l'alignement comme l'illustre la figure 3.5 à travers les étapes 2, 3 et 4 ($\mathcal{B}$(- - v v v - - - - e - r -) = ver). Le modèle est conçu de manière à pouvoir prédire une séquence de caractères y de taille variable et inférieure à la longueur T de la séquence de sortie brute du modèle (étape 2). La fonction $\mathcal{B}$ permet de définir la probabilité d'une séquence de caractère $y \in \mathcal{A}^{\leq T}$ comme la somme des probabilités de tous les alignements qui lui correspondent :

$$p(y|x) = \sum_{\pi \in \mathcal{B}^{-1}(y)} p(\pi|x) \tag{3.2}$$

Lors de l'inférence, il existe différents algorithmes (best path, beam search ...) pour sélectionner les prédictions dans la matrice de probabilités de sortie du réseau Seq2Seq. Nous avons choisi d'utiliser l'algorithme "best path" parce qu'il est rapide et permet d'obtenir de bons résultats de reconnaissance. L'algorithme consiste à choisir la meilleure probabilité pour chaque élément de la séquence de sortie (étape 2 sur la figure). Ensuite, l'algorithme enlève les répétitions de caractères (étape 3) puis enlève les caractères "blank" (étape 4).

Pendant l'entrainement, la CTC génère tous les alignements possibles correspondant à séquence de caractères de la vérité terrain, *i.e.* des séquences avec des répétitions de caractères et des blanks correspond à la vérité terrain. Comme illustré sur la figure 3.5, la fonction de coût $Loss_{CTC}$ prend en compte la somme des probabilités de chaque alignement et est définie par :

$$Loss_{CTC}(x, y) = -ln\, p(y|x) \tag{3.3}$$

L'intérêt de la méthode CTC est qu'elle n'a pas besoin de connaitre le "meilleur" alignement qui nécessiterait une annotation spécifique. Elle permet d'entrainer un modèle à prédire des alignements correspondant à la vérité terrain. Nous pouvons noter que la méthode prend en compte des alignements de reconnaissance et ne tient pas compte de critère de localisation spatiale des caractères, *i.e.* qu'il n'y a pas de critères de segmentation dans la méthode CTC. Plus de détails sur le calcul de la fonction de coût sont donnés dans le papier présentant la méthode CTC [Gra+06].

Extensions : régularisation du CTC

La méthode CTC permet d'entrainer un modèle Seq2Seq à prédire les alignements correspondant aux mots à reconnaitre sans aucune notion de segmentation. Les auteurs des papiers [LJZ18] [ZSN21] proposent d'étendre la méthode CTC pour améliorer ses performances. Empiriquement, de nombreux auteurs se sont aperçus de la tendance du CTC à avoir des pics de probabilité des caractères valides, *i.e.* prédire beaucoup de blanks entre les prédictions des caractères. Ce comportement montre que le réseau surapprend un **alignement dominant**. Les auteurs de l'article [ZSN21] définissent formellement ce phénomène et pourquoi la méthode CTC agit de cette façon. Ils proposent également une méthode (CTC label prior) pour éviter ce comportement et ainsi prédire moins de blanks comme illustré dans la figure 3.6.

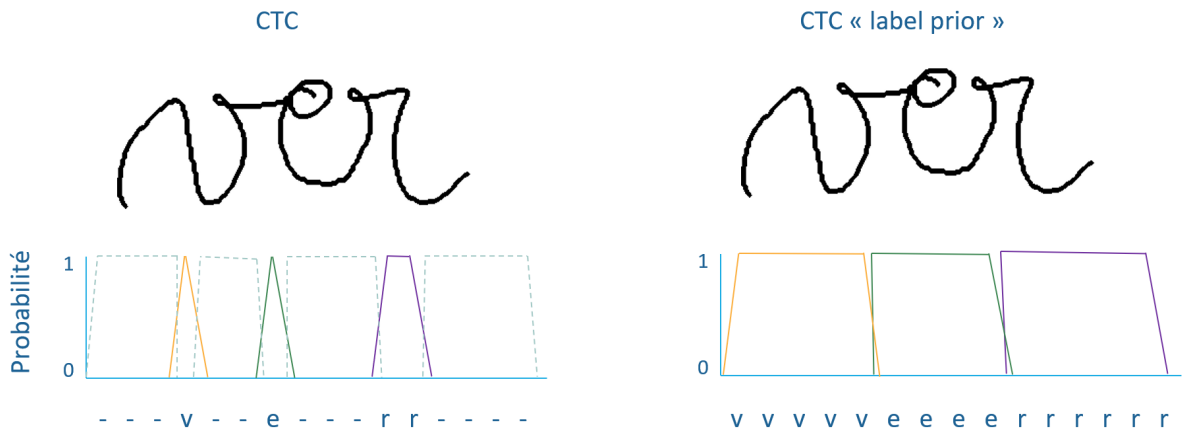

FIGURE 3.6 – Exemples d'activations pour la méthode CTC classique [Gra+06] et la méthode CTC avec un prior sur les étiquettes [ZSN21]. Les probabilités des caractères "blank" sont en pointillés.

Les auteurs de l'article [LJZ18] propose deux méthodes de régularisation du CTC pour limiter le comportement en pics (*i.e.* entropie basse). La première méthode EnCTC utilise une régularisation basée sur l'entropie conditionnelle maximale qui pénalise les distributions en pics et encourage l'exploration afin d'éviter que le modèle surapprenne sur un alignement dominant. L'entropie $H(p(\pi|y,x))$ des chemins où x est la séquence d'entrée et y le mot de la vérité terrain, est définie par :

$$H(p(\pi|y,x)) = -\sum_{\pi \in \mathcal{B}^{-1}} p(\pi|x,y)\ log\ p(\pi|x,y) \tag{3.4}$$

La loss CTC avec la régularisation est définie par :

$$L_{EnCTC} = L_{CTC} - \beta H(p(\pi|y, x) \tag{3.5}$$

ou β est un hyperparamètre qu'il faudra établir.

La seconde méthode EsCTC utilise une stratégie d'élagage basée sur l'entropie pour réduire le nombre d'alignements possibles en excluant les alignements incohérents pendant l'entrainement. Un alignement est considéré comme incohérent si au moins une prédiction caractère ne remplit pas un critère de taille (exemple pour l'alignement : vvvvvvvver où la prédiction de la lettre "v" ne remplit pas le critère, car il est peu probable que le caractère "v" représente la quasi-totalité de l'image). Le critère de taille prend en compte la longueur de la séquence de sortie T et la longueur de la vraie séquence de caractères (vérité terrain pendant l'entrainement). Nous pouvons noter que ce critère de taille est le même quel que soit le caractère et ne tient pas compte de caractères avec des largeurs variables.

À notre connaissance, beaucoup d'auteurs s'intéressent à une problématique de reconnaissances de mots, mais peu s'intéressent à une problématique de segmentation de mots en lettres. Nous nous intéresserons dans la section 5.3 à l'aspect segmentation indispensable pour notre problématique d'analyse d'écriture. Nous verrons en quoi le fait de prédire moins de blanks [ZSN21] est intéressant dans notre contexte de reconnaissance et segmentation d'écriture. De plus, nous montrerons que l'utilisation de la CTC ne permet pas directement d'obtenir une segmentation du mot en lettres précises. Nous proposerons également une extension de la méthode CTC dans le but d'améliorer les performances de segmentation associées au réseau de neurones convolutifs récurrents présenté dans la section suivante.

3.2.2 Réseau de neurones convolutifs récurrents

Les réseaux de neurones convolutifs récurrents (CRNN pour Convolutional Recurrent Neural Network) [SBY17] [BM17] [Pui17] sont des types de réseaux permettant de répondre à une problématique Seq2Seq. Nous pouvons découper leurs architectures en trois parties. La première est composée de couches de convolution servant à l'extraction de caractéristiques spatiales. La seconde est composée de couches récurrentes type BLSTM [HS97] servant à l'extraction de caractéristiques temporelles. La troisième utilise la méthode CTC pour faire la transcription (entrainement et inférence). Dans ces travaux,

nous utilisons l'architecture illustrée dans la figure 3.7 composée de moins de deux millions de paramètres pour faire de la reconnaissance et de la segmentation de mots manuscrits. Cette architecture provient de travaux préliminaires associés à l'article [Bar+21].

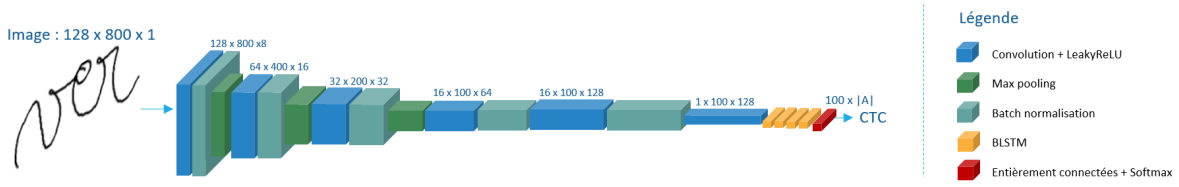

FIGURE 3.7 – Architecture CRNN [Bar+21] utilisée dans nos travaux.

3.2.3 Réseau encodeur décodeur avec un modèle d'attention

La figure 3.8 illustre l'architecture encodeur décodeur avec un modèle d'attention hybride Bahdanau [BCB15] composée de moins de trois millions de paramètres utilisée dans nos travaux pour faire de la reconnaissance et de la segmentation de mots manuscrits. Dans la suite du manuscrit, nous appelons ce réseau Seq2Seq. Sa conception s'inspire de l'article [Mic+] et est présentée dans l'article [4].

L'encodeur suit l'architecture du réseau de neurones convolutifs récurrents (CRNN) présentée précédemment. L'idée est d'étendre l'architecture CRNN en utilisant les caractéristiques extraites par un décodeur guidé par un mécanisme d'attention pour améliorer les performances de reconnaissance. Le décodeur utilise un réseau récurrent LSTM pour prédire un caractère à chaque pas t. Il utilise le caractère <sos> pour définir le début d'une séquence et le caractère <eos> pour définir la fin d'une séquence. A chaque pas t, le mécanisme d'attention focalise le décodeur sur certaines partie du vecteur de caractéristiques extrait par l'encodeur à l'aide d'un vecteur de poids d'attention. L'encodeur est entrainé avec la fonction CTC $Loss_{CTC}$ et le décodeur est entrainé à l'aide de la fonction cross entropie $Loss_{CE}$. Le coût d'entrainement du réseau Seq2Seq est défini par :

$$Loss_{Seq2Seq} = 0.5 * Loss_{CTC} + 0.5 * Loss_{CE} \tag{3.6}$$

Les poids associés aux fonctions de coût de l'encodeur et du décodeur proviennent des recommandations de l'article d'origine [Mic+].

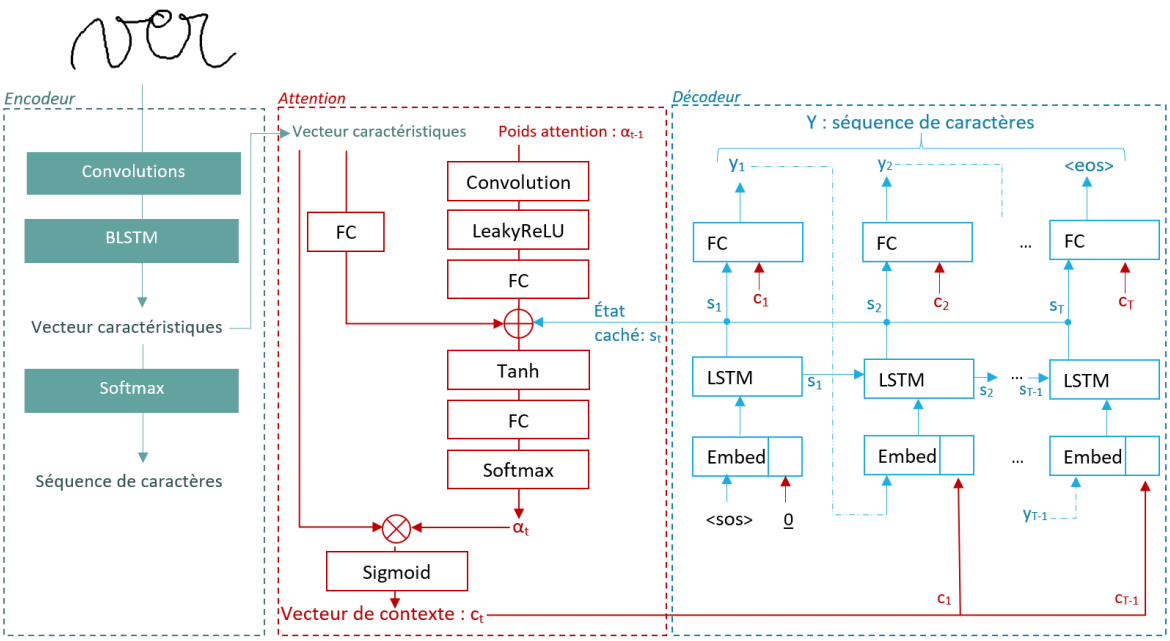

FIGURE 3.8 – Architecture encodeur décodeur avec un modèle d'attention utilisé dans nos travaux inspirée de [Mic+].

3.2.4 Champ réceptif et convolutions

Le champ réceptif associé aux convolutions d'un réseau correspond à la zone de l'image d'entrée qui a été utilisée pour extraire des caractéristiques. Sa taille r_0 est fixe et dépend du paramétrage des couches de convolution, *i.e.* du nombre de couches L, de la taille des noyaux k et des pas de déplacement s des filtres de convolutions :

$$r_0 = \sum_{l=1}^{L}\Big((k_l - 1)\prod_{i=1}^{l-1} s_i\Big)+1 \tag{3.7}$$

La figure 3.9 illustre un exemple de champs réceptifs associés au premier et second éléments de la séquence des vecteur de caractéristiques extrait par les couches de convolutions. Le paramétrage des couches de convolution induit un pas de déplacement Δx entre le champs réceptif du premier (en vert) et du second élément (en orange). Nous pouvons remarquer également qu'il y a du chevauchement entre les deux champs réceptifs.

Précédemment, nous avons défini le fonctionnement d'un modèle CRNN pour une tâche de reconnaissance de texte à l'aide de la méthode CTC. Notre objectif étant de concevoir un système de reconnaissance et de segmentation, nous spécifions dans la sec-

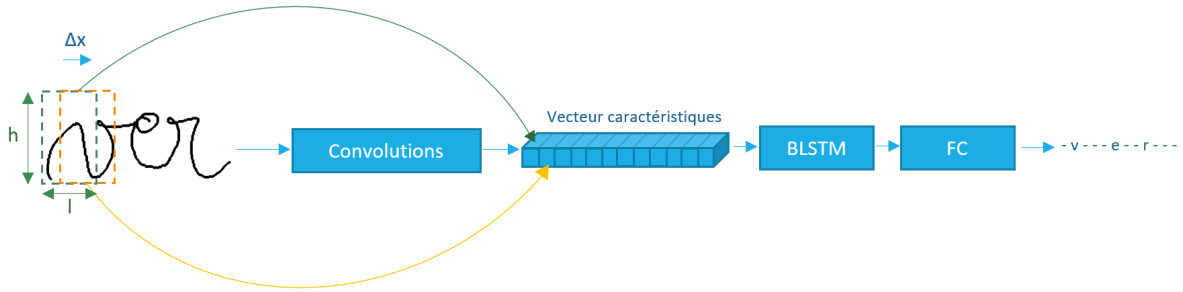

FIGURE 3.9 – Représentation des champs réceptif du premier (vert) et du second élément (orange) dans un réseau CRNN.

tion 5.3 une méthode basée sur les champs réceptifs pour obtenir une **segmentation implicite** d'un réseau de neurones dédié à la reconnaissance. Nous analysons également le rôle du CTC sur la précision de la segmentation ainsi que l'impact de différentes variantes de régularisation pour optimiser la segmentation.

3.3 Tâche de détection d'objets

La détection d'objets permet d'identifier et de localiser des objets (chats, chiens, personnes, voitures ...) dans une image ou une vidéo. Dans notre contexte applicatif, l'objectif d'un réseau de détection d'objets est de localiser et d'identifier précisément les lettres dans un mot manuscrit cursif. Une lettre est représentée par un tracé fin avec une forme assez variable pour une même classe et des formes qui se ressemblent entre plusieurs classes. De plus, certaines lettres peuvent être contenues dans des lettres plus grandes comme c'est le cas de la lettre "n" qui est contenue dans la lettre "m". Contrairement aux réseaux d'apprentissage profond présentés précédemment, ce type de réseau est entrainé pour une tâche de reconnaissance (identification) et de segmentation (localisation). Les détecteurs en deux étapes [Ren+15] [He+17] sont connus pour être un peu plus précis pour la localisation que leurs homologues fonctionnant en une étape [Red+16] [RF17] [RF18] [BWL20] [Li+22] [WBL22] même s'ils sont généralement un peu plus lents. Nous privilégierons les détecteurs en deux étapes.

La figure 3.10 illustre le fonctionnement général du modèle en deux étapes Mask R-CNN [He+17]. Le modèle utilise un réseau de neurones à convolution pour extraire des caractéristiques de l'image à analyser (étape 1). Ces caractéristiques sont utilisées dans un premier temps par un réseau de propositions de régions pour calculer les régions à analyser

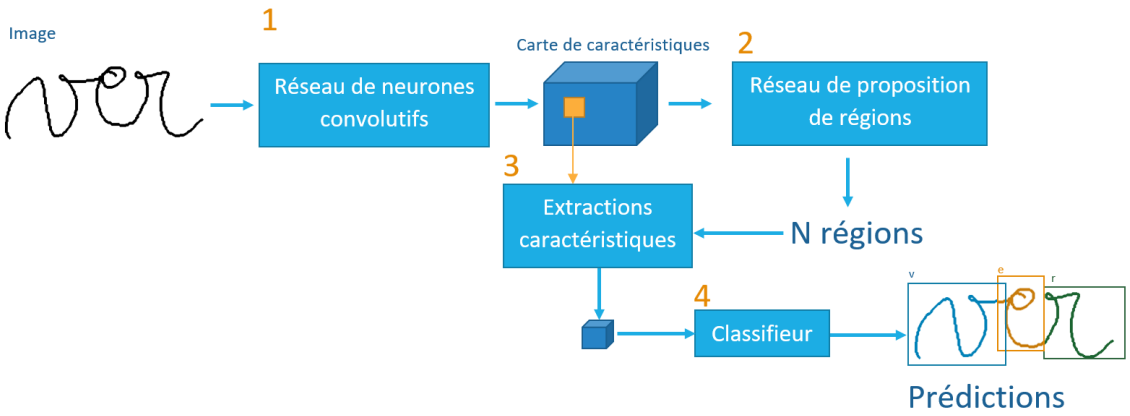

FIGURE 3.10 – Fonctionnement général du réseau Mask R-CNN appliqué à de la détection de lettres dans un mot.

(étape 2). À partir de celles-ci, le réseau extrait les caractéristiques associées à chaque région (étape 3). Enfin, pour chaque région, le réseau prédit la classe, la boite englobante ainsi que le masque de segmentation associé (étape 4). Ce modèle est de type deux étapes, car il utilise une étape de calcul des régions à analyser puis une étape d'analyse de ces régions. Nous allons utiliser le détecteur d'objets Mask R-CNN initialement conçu pour détecter des objets (chat, chien ...) dans un contexte de **détection de lettres dans un mot**. Dans cette section, nous présentons le réseau d'extraction de caractéristiques, le réseau de propositions de régions, la méthode pour extraire les caractéristiques relatives à une région, la branche qui analyse chaque région ainsi que la méthode de Non Maximal Suppression utilisé par le Mask R-CNN.

3.3.1 Réseau d'extraction de caractéristiques

Nous utilisons un réseau ResNet-FPN illustré sur la figure 3.11 comme extracteur de caractéristiques. Ce réseau est une combinaison d'un réseau ResNet [He+16] présenté dans la partie 3.1 et d'un réseau d'extraction de caractéristiques pyramidal [Lin+17] (FPN pour Feature Pyramid Network en anglais). Les flèches noires sur la figure indiquent les liens horizontaux (entre ResNet et FPN) et verticaux (entre les niveaux). L'objectif de cette combinaison est d'extraire des cartes caractéristiques avec plusieurs niveaux de granularité. Ces cartes sont utilisées par le réseau de propositions de régions pour calculer des propositions de régions à différentes échelles.

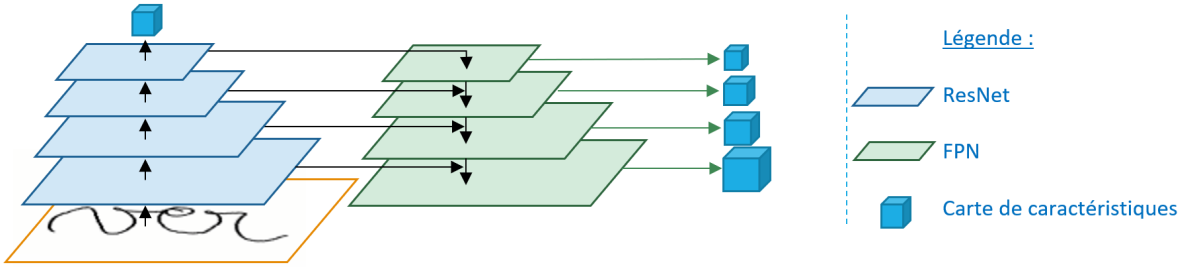

FIGURE 3.11 – Architecture du réseau ResNet-FPN.

3.3.2 Réseau de proposition de régions

À partir des cartes de caractéristiques calculées précédemment, le réseau de propositions de régions (Region Proposal Network en anglais RPN) calcule un ensemble de propositions, *i.e.* une liste de régions de l'image à analyser. Pour chaque région, le RPN prédit un score qui définit si la région est un objet à détecter ou si c'est de l'arrière-plan ainsi que les coordonnées du rectangle (*i.e.* boite englobante) correspondant à cette région dans l'image d'entrée. Le fonctionnement du RPN est illustré sur la figure 3.12. Il commence par extraire des caractéristiques à l'aide d'un filtre de convolution. Pour chaque élément (x, y) de la carte de caractéristiques (vecteur de caractéristiques en rouge sur la figure), pour chaque taille d'ancre k, le réseau produit un score qui indique si cette région est un objet ainsi que les coordonnées de la région. Une ancre est définie par sa taille (hauteur et largeur). L'intérêt d'utiliser plusieurs niveaux de cartes de caractéristiques est que nous pouvons définir des ancres de tailles plus ou moins grandes selon le niveau ce qui permet d'avoir des régions de tailles adaptées à des petits et des grands objets. Les ancres servent à mettre en correspondance les propositions de régions du modèle et la vérité terrain (boite englobante des lettres) lors de l'entrainement.

Le modèle produit H*L*k propositions de boites par niveau, ce qui représente un nombre élevé de régions. Le modèle RPN utilise la méthode Non Max Suppression (NMS) pour filtrer les régions se chevauchant. Ensuite, il calcule l'Intersection Over Union (IoU) entre les ancres et les boites englobantes de la vérité terrain. L'IoU sert à classer les ancres en trois catégories. Si l'IoU est inférieure à un seuil, l'ancre est classée comme **négative** et l'objectif est de reconnaitre la région associée comme de l'arrière-plan. Si l'IoU est supérieure à un seuil, l'ancre est classée comme **positive** et l'objectif est de détecter la région associée comme un objet. Les ancres avec un IoU entre les deux seuils ne sont pas utilisées pendant l'entrainement. Comme le nombre de propositions de régions est

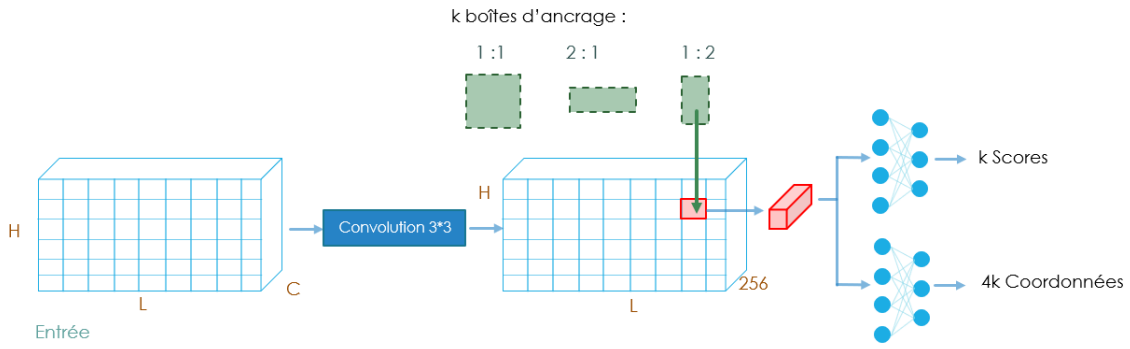

FIGURE 3.12 – Architecture du réseau de propositions de régions.

élevé, le réseau utilise un sous-ensemble choisi aléatoirement des ancres classées comme positives et négatives pour le calcul de la fonction de coût. L'objectif est d'entrainer le RPN à détecter des régions "objet" et des régions qui correspondent à l'arrière-plan. La fonction de coût regroupe une tâche de classification qui définit si la région est un objet ou de l'arrière-plan ainsi qu'une tâche de régression entre les coordonnées de la région et de la boite englobante de la vérité terrain pour les ancres positives.

En inférence, le réseau RPN utilise la méthode Non Max Suppression et sélectionne les N meilleures régions par rapport au score objet. Plus de détails sur le fonctionnement de ce composant sont donnés dans l'article d'origine [Ren+15].

3.3.3 Extraction des caractéristiques d'une région

La méthode RoIAlign illustrée dans la figure 3.13 et détaillée dans l'article [He+17] permet d'extraire une sous-partie de la carte de caractéristiques associée à toute l'image représentant les caractéristiques d'une région. La méthode redimensionne les caractéristiques extraites dans une carte de taille fixe. Ainsi, chaque région quelque soit sa taille, est représentée par le même nombre de caractéristiques. La méthode redimensionne la région (coordonnées image) en divisant chaque coordonnée par un facteur et la superpose sur la carte de caractéristiques (étape 1). Ensuite, la région est divisée en grille selon la taille de la carte de caractéristiques de sortie désirée (étape 2 : grille 3x3). Ensuite, la méthode choisit 4 points dans chaque case en utilisant une interpolation bilinéaire (étape 3). Ensuite, la valeur maximale de chaque case est gardée dans la carte de caractéristique de sortie (étape 4). Chaque canal de la carte de caractéristiques d'entrée est traité séparément.

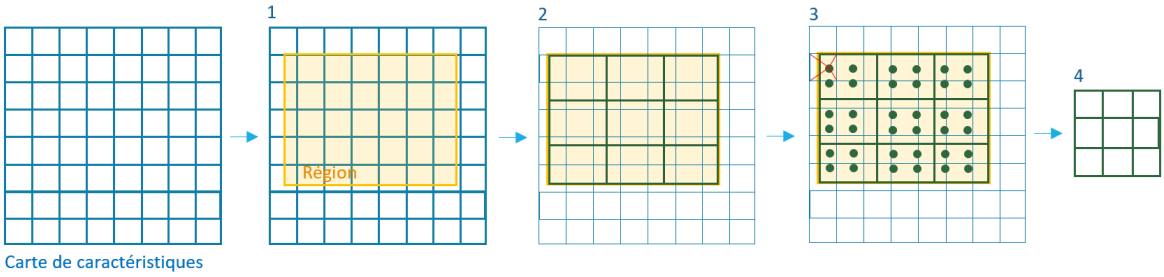

FIGURE 3.13 – Exemple d'extraction des caractéristiques associées à une région avec la méthode RoIAlign.

3.3.4 Branche de prédictions

Nous venons de voir que pour chaque région, le réseau extrait une carte de caractéristiques de taille fixe. À partir de celle-ci, le réseau va prédire la classe de l'objet, la boite englobante et le masque de segmentation sémantique associé comme l'illustre la figure 3.14. Pendant l'entrainement, de la même manière que pour le RPN, le réseau utilise un sous-ensemble des régions "objets" pour calculer la fonction de coût. La fonction de coût globale est définie par :

$$Loss = Loss_{objet} + Loss_{region_{boite}} + Loss_{classe} + Loss_{boite} + Loss_{masque} \tag{3.8}$$

Où $Loss_{objet}$ correspond à la tâche de classification "objet" du RPN, $Loss_{region_{boite}}$ correspond à la tâche de régression de la région du RPN, $Loss_{classe}$ à la tâche de classification des lettres, $Loss_{boite}$ la tâche de régression entre les coordonnées de la région et les coordonnées de la boite englobante finale et $Loss_{masque}$ à une tâche de segmentation sémantique (cross entropie pour chaque pixel où il y a deux classes : objet et fond).

En inférence, un algorithme NMS est appliqué pour réduire le nombre de prédictions puis seulement les prédictions ayant un score supérieur à un seuil sont utilisées pour obtenir la prédiction finale.

Dans nos travaux, nous n'utilisons pas la sortie de la branche de masque de segmentation. L'hypothèse est que dans un contexte de segmentation de lettres, les boites englobantes des lettres contiennent beaucoup de vide (*i.e.* de pixels blancs) comparé à des objets classiques ce qui rend la tâche de segmentation plus difficile. Nous avons préféré explorer les stratégies détaillées dans le chapitre 7. Dans la suite du manuscrit, nous appelons le réseau Mask R-CNN : "R-CNN".

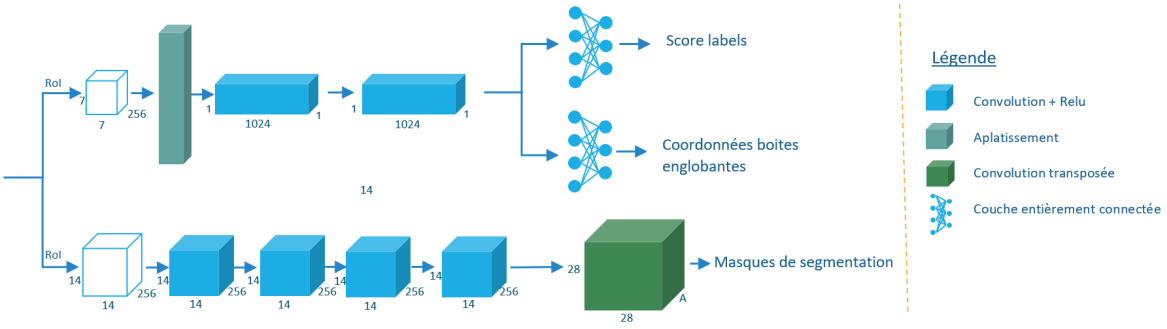

FIGURE 3.14 – Branche de prédictions de la classe, de la boite englobante et du masque de segmentation du réseau Mask R-CNN.

3.3.5 Non Maximal Suppression

Le processus de détection d'objets présenté induit un nombre important de propositions de région ainsi que de prédiction d'objets. En effet, il peut y avoir plusieurs prédictions pour un même objet comme l'illustre la figure 3.15. L'objectif de la méthode Non Maximal Suppression (NMS) est de réduire le nombre d'hypothèses pour accélérer le temps de traitement ainsi que de sélectionner la meilleure prédiction pour un même objet. À partir d'une liste de boites englobantes B, de scores associés S et d'un seuil de chevauchement N, la méthode NMS produit une liste filtrée D de la manière suivante :

1. La méthode sélectionne la boite b_{top} avec le meilleur score dans la liste B et la place dans la liste D.
2. A partir de la boite sélectionnée b_{top}, pour chaque boite restante $b_{compare}$ de la liste B, la méthode calcule l'IoU entre b_{top} et $b_{compare}$. Si l'IoU est supérieure au seuil N, la méthode enlève la boite $b_{compare}$ de la liste B.
3. La méthode réitère les deux étapes précédentes jusqu'à ce qu'il ne reste plus de boite dans la liste B.

La méthode NMS est appliquée pour filtrer les **propositions de régions** en fonction d'un score "objet" ainsi que pour filtrer les **prédictions d'objets** appartenant à la même classe. Cette dernière propriété est importante dans le cas de l'écriture. En effet, dans le cas d'une lettre déformée, le réseau peut détecter deux lettres différentes pour un même objet. Nous verrons par la suite une stratégie pour lever ces ambiguïtés.

Nous allons utiliser le détecteur d'objets présenté dans cette section dans les travaux détaillés dans le chapitre 7. L'objectif dans un premier temps va être d'évaluer ses per-

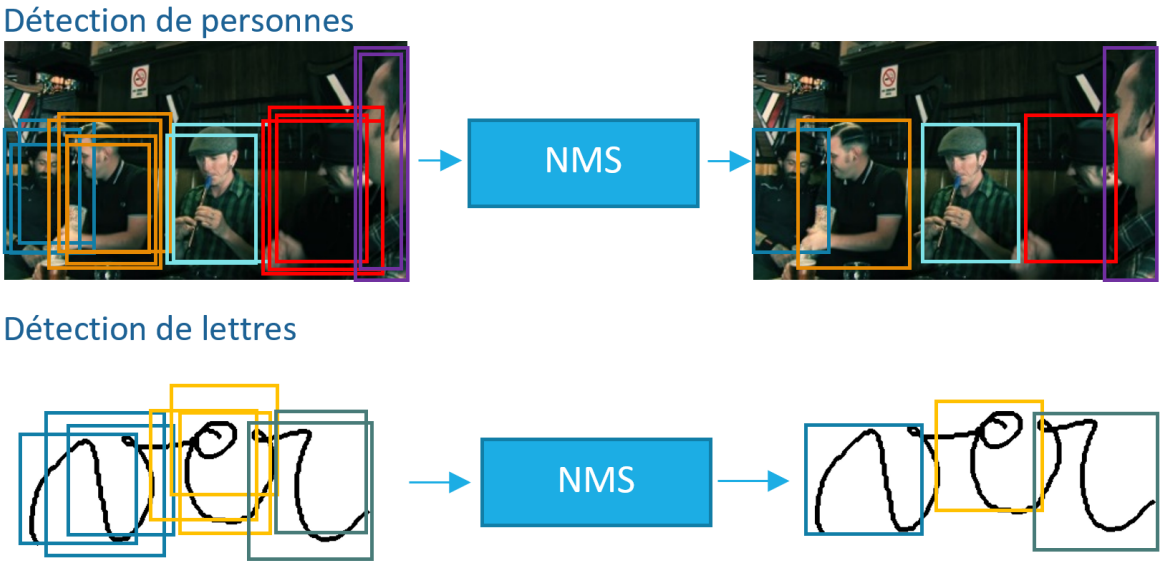

FIGURE 3.15 – Exemples d'application de l'algorithme NMS dans un contexte de détection de personnes et de détection de lettres.

formances en reconnaissance et en segmentation d'écriture. Dans un second temps, nous proposerons une combinaison du détecteur d'objets avec un modèle Seq2Seq afin d'améliorer ses performances.

Deuxième partie

Contributions mineures

Chapitre 4

Reconnaissance de caractère manuscrit cursif

Nos premiers travaux se sont intéressés à l'extension de l'approche d'analyse d'écriture manuscrite d'enfant IntuiScript présentée dans le chapitre 2. IntuiScript utilise une reconnaissance et une analyse basée sur une segmentation explicite du mot en hypothèses de caractères. La première stratégie a été d'améliorer la **reconnaissance de caractères isolés**. Nous avons utilisé un réseau de neurones convolutifs prenant en compte, en plus de l'information sur la forme du tracé, des connaissances expertes relatives à sa dynamique. Les réseaux de neurones convolutifs (CNN) peuvent être utilisés pour faire de la reconnaissance de caractère manuscrit. Même si ce type de réseaux est déjà très performant, nous avons cherché à explorer des améliorations possibles par l'**injection de connaissances expertes** liées au contexte de l'écriture manuscrite. Dans ce chapitre, nous nous cherchons à reconnaitre des caractères manuscrits cursifs français écrit par des enfants où des exemples de caractères à reconnaitre sont illustrés sur la figure 4.1.

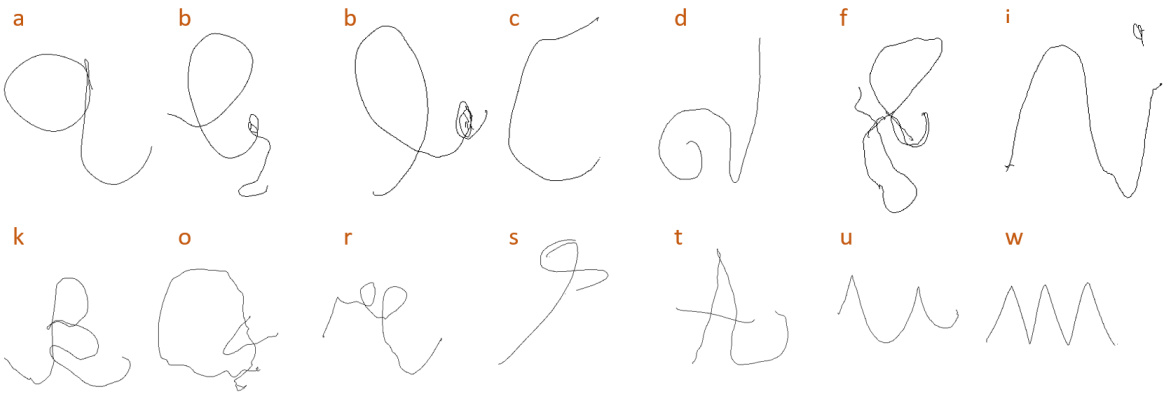

Figure 4.1 – Exemples de caractères manuscrits d'enfants déformés. En orange au-dessus du dessin, le caractère à reconnaitre.

Notre objectif est de présenter des méthodes permettant d'injecter, en plus de l'information statique sur la forme du caractère, des informations présentes dans le signal en ligne sur la dynamique du tracé représentant le ductus de l'écriture dans un réseau type CNN comme le montre la figure 4.2.

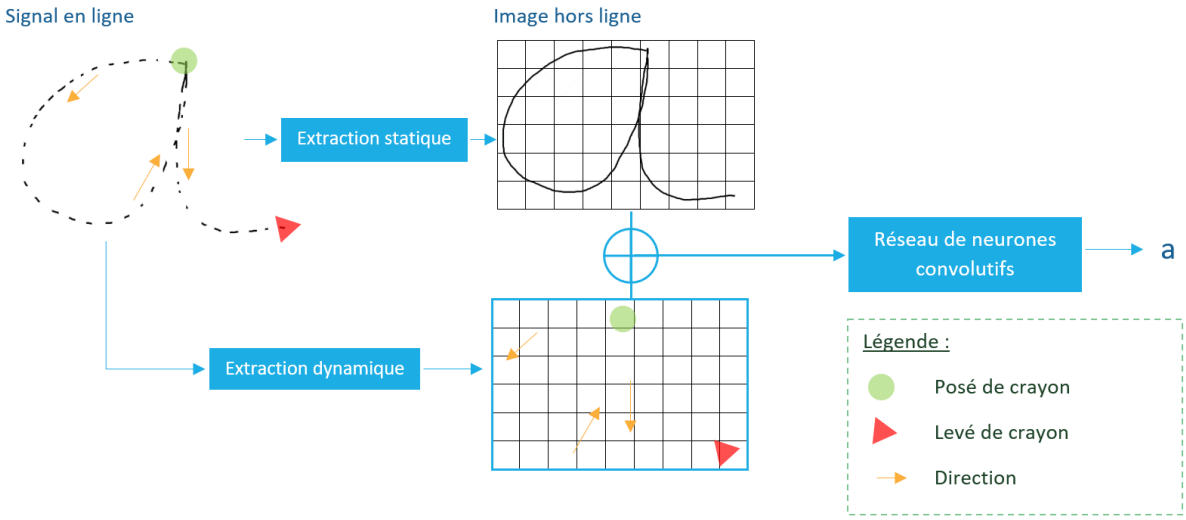

FIGURE 4.2 – Principe général de l'intégration d'informations statiques et dynamiques dans un CNN.

Ce chapitre présente les travaux en lien avec la première contribution. Il commence par spécifier la nature des informations extraites du signal en ligne et la manière de les représenter dans une image. Ensuite, il propose deux stratégies de fusion pour les intégrer dans un CNN. Enfin, une étude expérimentale présente les résultats obtenus pour ces différentes contributions.

4.1 Extractions d'informations statique et dynamique dans le signal en ligne

Un caractère manuscrit en ligne est représenté par une suite de points décrits par des coordonnées (x, y), une pression et un horodatage. Ainsi, une multitude d'informations supplémentaires peuvent être extraite à partir de ce signal en ligne comme la vitesse, la direction, l'orientation de l'écriture ou encore les posés et levés de crayon. Nos travaux s'intéressent aux informations de type forme, direction et orientation du tracé manuscrit.

4.1.1 Information statique : la forme

Le signal en ligne est acquis par un processus d'échantillonnage, *i.e.* que la tablette enregistre la position du stylet à chaque pas de temps. L'étape nécessaire pour relier les positions enregistrées et ainsi extraire l'information de base sur la forme du caractère se décompose en trois étapes illustrées dans la figure 4.3 :

1. **Normalisation :** Les points du tracé en ligne sont codés par des valeurs réelles tandis que les coordonnées des pixels d'une image sont codés par des valeurs entières. De plus, selon la personne qui écrit et la tablette utilisée pour récolter les données (résolution différente), les dimensions des caractères peuvent varier. Cette étape remet à l'échelle les coordonnées des points en les arrondissant à l'entier le plus proche puis en les stockant dans une matrice image de taille 32 x 32.
2. **Interpolation :** La suite de points récoltés par la tablette ne forme pas un tracé continu. Pour obtenir une représentation plus fidèle du caractère tracé, cette étape d'interpolation spatiale comble les vides entre deux points successifs.
3. **Centrage :** Les points sont centrés dans l'image. Nous avons fait le choix de représenter les points dans l'image par la valeur 1. Le processus d'extraction de l'information sur la forme donne en résultat une image noir et blanc où l'écriture est en blanc.

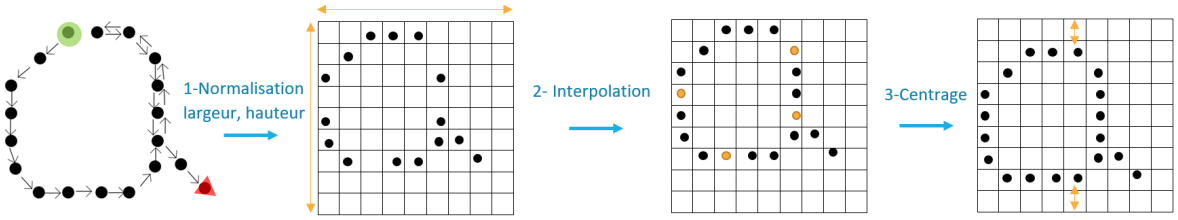

FIGURE 4.3 – Conversion du signal en ligne en une image hors ligne.

4.1.2 Information dynamique : la direction

L'horodatage du signal en ligne permet de le diviser en tracés ascendants et descendants, comme illustré dans la figure 4.4. Un tracé ascendant représente un tracé manuscrit qui a été dessiné de bas en haut et inversement pour un tracé descendant. Cette décomposition apporte de l'information sur le ductus, *i.e.* la direction du tracé manuscrit. Dans un

contexte d'écriture manuscrite cursive, les tracés descendants sont plus discriminatoires que les tracés ascendants comme expliqué dans l'article [AB02]. De la même manière que pour l'information sur la forme, l'information sur la direction (ascendant ou descendant) est représentée dans une image où les coordonnées des points (ascendant ou descendant) sont les mêmes que pour les points représentant la forme du tracé.

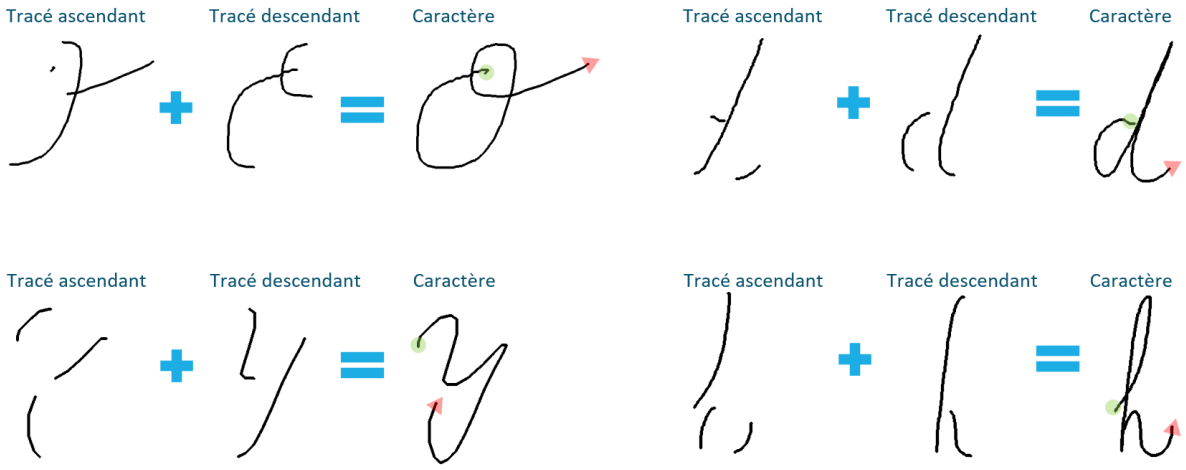

FIGURE 4.4 – Exemples de décomposition de caractères en tracés ascendants et descendants. Le point rouge représente le début (posé du stylet) du tracé et le triangle vert la fin (levé du stylet).

4.1.3 Information dynamique : l'orientation

L'orientation du tracé est représentée par l'angle α entre un point et le point suivant dans le tracé en ligne. Pour modéliser l'angle, nous utilisons le cosinus, le sinus ou l'arc tangente. L'image étant représentée par une matrice de pixels, le point suivant peut-être dans l'une des huit cases adjacentes au point courant, ce qui donne huit valeurs d'angles possibles comme illustrées sur la figure 4.5. Le cosinus et le sinus ont des valeurs comprises entre -1 et 1 et l'arc tangente entre $-\frac{\pi}{2}$ et $\frac{\pi}{2}$. Puisque nous avons déjà attribué la valeur zéro pour indiquer l'absence de tracé, nous avons normalisé le cosinus, le sinus et l'arc tangente entre 0,2 et 1 pour éviter toute confusion.

L'extraction de l'information dynamique dans des images comporte des limitations lorsque le tracé passe plusieurs fois par les mêmes coordonnées (x, y). Le fait de stocker l'information dans des images ne permet pas directement de gérer le plusieurs points pour une même position. Nous avons fait le choix de considérer uniquement le dernier point

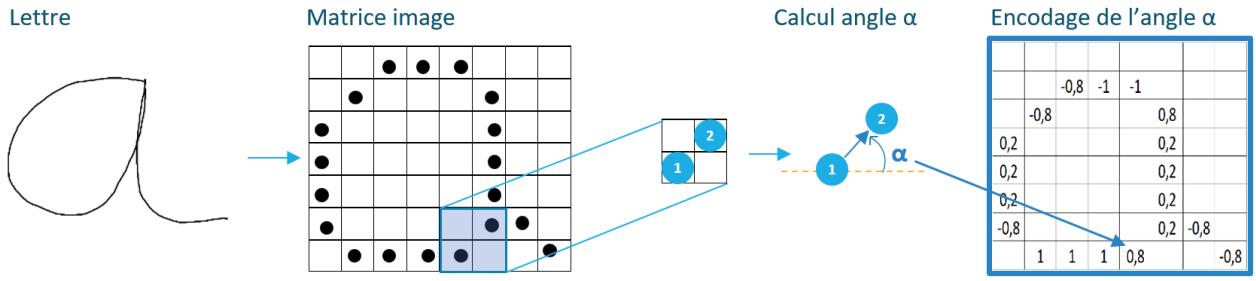

FIGURE 4.5 – Exemple de représentation de l'angle α entre les points 1 et 2 à l'aide du cosinus.

(critère horodatage).

4.2 Intégration dans un CNN

Dans ce travail, le caractère manuscrit en ligne est converti en image hors ligne représentant le caractère. L'objectif est d'enrichir l'image avec les informations dynamiques. Nous étudions deux schémas de fusion pour prendre en compte les informations extraites du tracé en ligne décrites dans la partie précédente : une fusion précoce et une fusion tardive.

4.2.1 Fusion précoce

L'approche de fusion précoce est illustrée figure 4.6. Les informations statiques et dynamiques extraites du signal en ligne sont combinées dans une image multicanal où chaque canal représente un type d'information. Les informations contenues dans les canaux sont alignées de manière à enrichir l'information sur la forme du caractère. Un réseau de neurones convolutifs (CNN) est utilisé pour reconnaitre le caractère contenu dans l'image. L'idée de cette fusion précoce est de regrouper différentes sources d'informations dès l'entrée du CNN, afin que les filtres de convolutions extraient des caractéristiques conjointes à chaque canal d'information.

4.2.2 Fusion tardive

L'approche de fusion tardive illustrée par la figure 4.7, utilise un ensemble de réseaux de neurones convolutifs où chaque réseau possède la même architecture. Pour chaque type

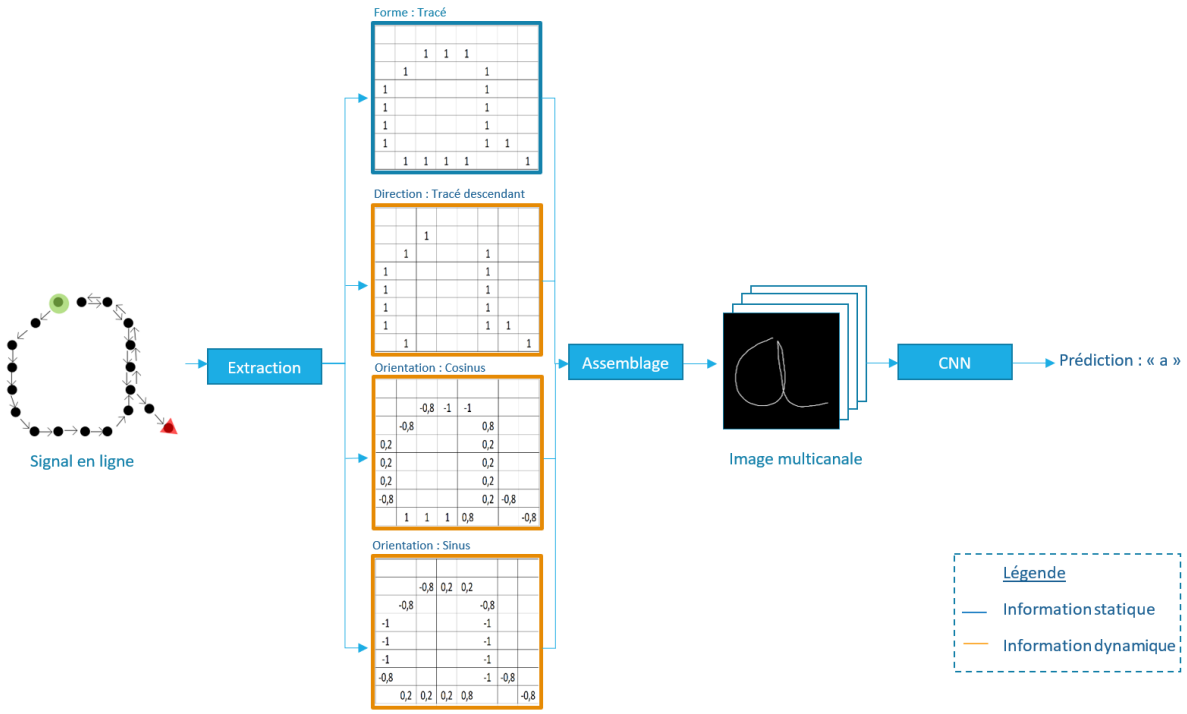

FIGURE 4.6 – Approche de fusion précoce avec les informations sur la forme, la direction (tracé descendant) et l'orientation (cosinus et sinus).

d'information extraite du tracé en ligne représenté par une image avec un seul canal, un réseau effectue une prédiction. Les prédictions sont fusionnées a posteriori pour obtenir une prédiction globale. La fusion correspond à un vote pondéré par les scores de reconnaissance de chaque réseau où chaque réseau a le même poids. L'idée de la fusion tardive est d'entrainer un **classifieur spécifique pour chaque type d'information** afin de fusionner leurs prédictions.

4.3 Expérimentations : reconnaissance de caractères

4.3.1 Jeu de données

Dans cette étude expérimentale, nous utilisons le jeu de données ICDB-Letters présenté dans la partie 1.4.2. Nous appliquons de la synthèse (étirement, inclinaison, rotation appliquées à l'image et modification de la courbure du tracé du signal en ligne) uniquement sur le jeu d'entrainement afin d'obtenir 5 000 exemples par classe (10 000 pour les lettres « e » et « x » car nous avons envisagé deux manières de les écrire). Puisque nous

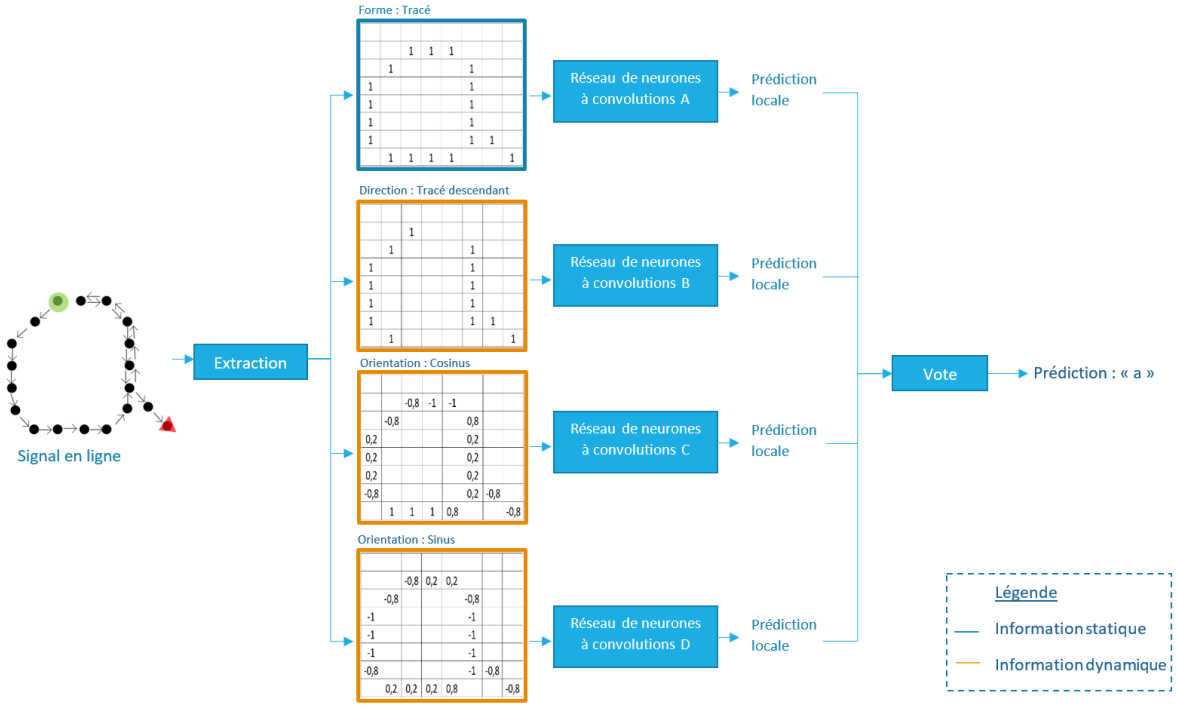

FIGURE 4.7 – Approche de fusion tardive avec les informations sur la forme, la direction (tracé descendant) et l'orientation (cosinus et sinus).

nous concentrons sur l'alphabet latin qui est composé de 26 lettres, notre jeu de données d'entraînement contient 140 000 éléments (24 x 5 000 + 2 x 10 000). Plus de détails sur les méthodes de synthèse sont donnés dans la partie 1.5. Le jeu de validation est composé de 2 600 échantillons (100 exemples par classe). Le jeu de test est composé de 7 086 échantillons et le nombre d'éléments par classe n'est pas équilibré. Aucune augmentation de donnée a été appliquée sur les jeux de validation et de tests. Le découpage du jeu de données est résumé dans le tableau 4.1. Comme détaillées précédemment, les lettres représentées par un signal en ligne sont converties en image de dimension 32 x 32.

TABLE 4.1 – Découpage du jeu de données de caractères manuscrits ICDB-Letters.

	Entrainement	Validation	Test
Nombre de caractères	27 000	2 600	7 086
Nombre de caractères (avec synthèse)	140 000	2 600	7 086

4.3.2 Protocole

Dans ces expérimentations, nous implémentons nos fusions sur différentes architectures CNN : LeNet-5, VGG-11 et ResNet-18 présentés dans la partie 3.1. Nous avons choisi une version peu profonde à la fois de VGG (11 couches pour environ 28 millions de paramètres) et de ResNet (18 couches pour environ 11 millions de paramètres) car nous pensons que les architectures plus profondes seraient plus enclines à surentrainer sur notre jeu de données.

Pour l'entrainement des réseaux de neurones, nous avons fixé de taille du batch à 128, choisi l'optimiseur ADAM et défini un taux d'apprentissage de 10^{-4} pour toutes les expérimentations. Les réseaux ont été entrainés sur des GPU et les mesures de temps d'inférence ont été effectués sur un CPU Intel Core i7-8665U. L'objectif à long terme étant l'utilisation du reconnaisseur dans une application éducative sur tablette, le temps calculé sur CPU est plus représentatif. Nous utilisons la précision (1- taux d'erreur caractère) comme mesure de performance dans cette section. Chaque expérience est exécutée deux fois pour mesurer la variance entre deux entrainements. Les tableaux de résultats contiennent la moyenne des précisions des deux modèles entrainés ainsi que la variance associée.

Le nombre d'époques d'entrainement est différent entre les expérimentations de l'étude d'ablation et les stratégies de fusion. Le nombre d'époques d'entrainement est fixé à 80 pour l'étude d'ablation, 1000 pour la stratégie de fusion précoce et 300 pour la stratégie de fusion tardive. L'objectif est de choisir le meilleur moyen de représenter l'information dynamique puis de tester la convergence des modèles avec l'approche utilisant un classifieur (fusion précoce) ou un ensemble de classifieurs (fusion tardive). Cela explique les légères différences de performances entre les résultats de l'étude d'ablation et les résultats de la fusion précoce. Le modèle ayant la meilleure précision sur le jeu de validation est évalué sur le jeu de test.

4.3.3 Résultats

Nous commençons par faire une étude d'ablation afin de choisir la manière la plus efficace de représenter l'information sur la direction (tracé ascendant, tracé descendant) et l'orientation (cosinus, sinus, arc tangente). Nous évaluons ensuite l'impact de l'ajout d'informations dynamiques dans une approche de fusion précoce puis de fusion tardive ainsi que le temps de calcul en inférence. Pour finir, nous comparons notre approche avec l'état de l'art.

Étude d'ablation

Le tableau 4.2 compare l'impact de la méthode utilisée pour encoder l'information de l'orientation. L'injection d'informations dynamiques est plus performante que l'utilisation seule de l'information sur la forme pour toutes les configurations. La combinaison du cosinus et du sinus est plus précise que l'information seule du cosinus et du sinus ou de l'arc tangente pour les trois architectures. Nous choisissons donc la **combinaison du cosinus et du sinus pour représenter l'orientation** du tracé.

TABLE 4.2 – Étude d'ablation pour l'information de l'**orientation** pour chaque architecture sur le jeu de test. La performance exprimée en % est évaluée avec la précision (une valeur plus élevée est meilleure).

Type d'information	Statique	Statique, Dynamique : Orientation			
Contenu des entrées	Forme	Forme, cosinus	Forme, sinus	Forme cosinus, sinus	Forme, arc tangente
LeNet-5					
Précision	91.37	92.48	91.54	**92.63**	92.03
Variance	0.1295	0.0784	0.5256	0.0240	0.0528
ResNet-18					
Précision	93.61	93.88	94.05	**94.25**	93.81
Variance	0.0380	0.4356	0.1680	0.0576	0.0002
VGG-11					
Précision	94.21	94.42	94.30	**94.77**	94.42
Variance	0.1024	0.1369	0.2703	0.0500	0.0168

Le tableau 4.3 compare l'impact de la méthode utilisée pour encoder l'information de la direction (tracé ascendant, tracé descendant). Nous comparons l'utilisation de l'information de tracé ascendant ou descendant ainsi que la combinaison des deux. L'injection de l'information dynamique est plus performante que l'utilisation seule de l'information sur la forme pour toutes les configurations. La partie descendante du tracé permet d'obtenir une meilleure précision que la partie ascendante ou la combinaison des deux pour les trois architectures. Cela s'explique par le fait que la partie descendante du tracé contient plus d'informations [AB02] sur le tracé que la partie ascendante. Nous choisissons donc la **partie descendante du tracé pour représenter la direction**.

Reconnaissance : fusion précoce

La stratégie de fusion précoce consiste à combiner des informations statiques et dynamiques dans une image multicanal puis d'utiliser un réseau de neurones convolutifs pour faire la classification. Nous avons testé les quatre combinaisons de canaux suivantes :

TABLE 4.3 – Étude d'ablation pour l'information de la **direction** pour chaque architecture sur le jeu de tests. La performance exprimée en % est évaluée avec la précision (une valeur plus élevée est meilleure).

Type d'information	Statique	Statique, Dynamique : direction		
Configuration des entrées	Forme	Forme, ascendant	Forme, descendant	Forme, ascendant, descendant
LeNet-5				
Précision	91.37	91.31	**92.37**	92.28
Variance	0.1295	0.0441	0.0483	0.0156
ResNet-18				
Précision	93.61	93.78	**94.34**	93.88
Variance	0.0380	0.4290	0.0961	0.0756
VGG-11				
Précision	94.21	94.44	**94.76**	94.47
Variance	0.1024	0.0840	0.0289	0.0529

— Configuration A : un seul canal qui encode le trait du stylet, *i.e.* la forme de la lettre. C'est la configuration de référence.
— Configuration B : nous ajoutons deux canaux à la configuration de référence (A) qui encodent l'information dynamique sur l'orientation du signal (cosinus et sinus).
— Configuration C : nous ajoutons un canal à la configuration de référence (A) qui encode la direction du signal (tracé descendant).
— Configuration D : nous utilisons toutes les informations disponibles : la forme, l'orientation (cosinus et sinus) et la direction (tracé descendant).

Nous notons que la dernière configuration (D) est celle qui contient le plus d'informations sur la dynamique de l'écriture manuscrite. Les configurations B et C englobent la configuration A, mais ne sont pas comparables puisque l'une (C) utilise l'information d'orientation, mais pas la direction, alors que l'autre (B) utilise la direction sans information d'orientation.

Les résultats obtenus en utilisant les différentes configurations d'entrée (A, B, C, D) pour les trois architectures de réseau sont donnés dans le tableau 4.4. Pour les trois architectures de réseau, l'injection d'informations dynamique sur l'orientation (B) ou la direction (C) améliore la performance comparée à l'approche de référence (A). La combinaison de l'information sur l'orientation et de la direction avec l'information sur la forme (D) permet d'obtenir les meilleurs résultats. Le réseau VGG-11 avec le plus grand nombre de paramètres obtient la meilleure précision de 95 %.

TABLE 4.4 – Résultats de la classification pour chaque architecture et configuration d'entrée avec une **fusion précoce** sur le jeu de test. La performance exprimée en % est évaluée avec la précision (une valeur plus élevée est meilleure).

Type d'information	Statique	Statique + Dynamique		
Configuration des entrées	A (forme)	B (forme, cosinus, sinus)	C (forme, descendant)	D (forme, cosinus, sinus, descendant)
LeNet-5				
Précision	90.62	91.35	91.93	**92.66**
Variance	0.0240	0.1599	0.0121	0.0049
ResNet-18				
Précision	94.05	94.06	94.33	**94.64**
Variance	0.0100	0.1764	0.0030	0.0030
VGG-11				
Précision	94.54	94.64	94.84	**95.00**
Variance	0.0066	0.0580	0.0323	0.042

Reconnaissance : fusion tardive

Pour chaque type d'architecture, nous avons entrainé un réseau de neurones spécifique pour chacun type d'information. Le résultat global de l'ensemble des réseaux est obtenu en faisant un vote sur l'ensemble des prédictions où chaque réseau a un poids égal. Les résultats sont donnés dans le tableau 4.5. Comme pour la fusion précoce, l'utilisation d'une fusion tardive de plusieurs canaux dans un ensemble de classifieur combinant des informations de forme, direction et orientation donne de meilleurs résultats que la prédiction du classifieur utilisant l'information de la forme et les prédictions des classifieurs pris séparément. Nous pouvons voir également que l'utilisation unique de l'information de direction (tracé descendant) dans un réseau donne de très mauvaises performances pour toutes les architectures. Cette information n'est pas suffisante en elle-même pour la classification. Les résultats de précision d'ensemble sont similaires à ceux obtenus avec une approche de fusion précoce. Cela montre que la stratégie de fusion est moins importante que le type d'information injecté dans le réseau. Nous pouvons noter que les ensembles de classifieurs LeNet et ResNet obtiennent des résultats légèrement meilleurs avec la fusion précoce, ce qui n'est pas le cas pour l'ensemble VGG. Pour les deux stratégies de fusions, les réseaux les plus profonds (ResNet et VGG) sont plus performants que le réseau le plus léger (LeNet). Nous avons également essayé d'apprendre un poids associé à chaque classifieur dans l'ensemble, mais les résultats n'étaient pas meilleurs que ceux présentés dans le tableau 4.5 et ne sont donc pas présentés ici.

TABLE 4.5 – Résultats de la classification pour chaque **architecture et configuration d'entrée** avec une fusion tardive sur le jeu de test. La performance exprimée en % est évaluée avec la précision (une valeur plus élevée est meilleure).

Type d'information	Statique	Dynamique			Statique + Dynamique
Configuration des entrées	Forme	Descendant	Cosinus	Sinus	Ensemble
LeNet-5					
Précision	91.10	77.49	90.78	88.15	**93.35**
ResNet-18					
Précision	93.35	74.44	93.60	93.19	**94.87**
VGG-11					
Précision	94.24	75.30	94.24	93.77	**94.76**

Temps d'inférence

Nous avons calculé le temps d'inférence moyen sur le jeu de données de test (7 086 échantillons) pour les configurations définies précédemment. Les mesures ont été effectuées avec un processeur Intel i7-7600U, 2,80 GHz. Les résultats sont présentés dans le tableau 4.6. Ce tableau montre que le temps d'inférence est faible pour toutes les configurations de fusion et répond à l'exigence de temps réel pour utiliser notre reconnaisseur dans un système d'analyse de mots. Sans surprise, le temps d'inférence pour LeNet est beaucoup plus faible que pour les deux autres architectures. L'approche de fusion tardive prend beaucoup plus de temps que celle de fusion précoce et peut ne pas répondre aux exigences en temps réel dans un pipeline plus complexe. L'ajout de canaux qui encodent la dynamique dans le schéma de fusion précoce n'augmente pas significativement le temps d'inférence. Il existe un compromis clair entre la complexité du modèle d'apprentissage en profondeur et le temps d'inférence. Nous pensons que l'architecture ResNet, qui est un peu moins précise que celle de VGG, pourrait encore être préférée pour son meilleur temps d'inférence.

TABLE 4.6 – Temps d'inférence moyen (en ms) pour chaque architecture, chaque configuration d'entrée de fusion précoce (A, B, C, D) et pour la fusion tardive.

Entrée / Architecture	Fusion précoce A	Fusion précoce B	Fusion précoce C	Fusion précoce D	Fusion tardive
LeNet-5	0.68	0.71	0.77	0.76	2.80
ResNet-18	13.33	13.96	14.35	15.33	59.39
VGG-11	18.89	18.51	18.94	19.46	76.43

Comparaison avec l'état de l'art

Cette partie compare la précision de notre reconnaisseur de caractères manuscrits avec l'état de l'art (présenté dans la partie 2.2.2). Nous utilisons l'approche utilisant une fusion précoce avec une architecture VGG ainsi que l'utilisation de l'information dynamique en entrées du réseau. Le reconnaisseur de l'état de l'art correspond au classifieur de lettre utilisé dans l'analyseur de mot IntuiScript [Sim+19] et présenté dans la partie 2.2.2. Le jeu de test est le même que pour les expérimentations précédentes. Les résultats présentés dans le tableau 4.7 montrent que notre classifieur est plus performant de 1.25 % en précision.

TABLE 4.7 – Comparaison avec l'état de l'art. La performance exprimée en % est évaluée avec la précision (une valeur plus élevée est meilleure).

Méthode	Entrée	Précision
État de l'art [Sim+19]	Signal en ligne	93.75
VGG-11 avec fusion précoce	Configuration D	**95.00**

Des expérimentations ont été faites pour mesurer l'impact de l'utilisation de notre reconnaisseur de caractères dans le modèle d'analyse de mots IntuiScript. Malgré une meilleure précision au niveau caractère, son utilisation a un impact mineur sur les performances de l'analyse de mot (détaillée dans la partie 2). Cela s'explique par le fait que la reconnaissance de lettre intervient au début de la chaine de traitement d'IntuiScript et est utilisée pour **sélectionner les hypothèses** de lettres à considérer par la suite pour chaque case du treillis de segmentation. De plus, plusieurs étapes de guidage interviennent par la suite, ce qui permet de garder les hypothèses avec des scores de reconnaissance plus faibles.

4.4 Bilan

Les travaux détaillés dans ce chapitre présentent le fonctionnement d'**un reconnaisseur de caractères manuscrits cursifs** écrit par des enfants dans l'objectif de remplacer le reconnaisseur de caractères du système IntuiScript et ainsi améliorer la précision de l'analyse de mot. Nous proposons une approche originale d'**injection d'informations dynamiques** dans un réseau de neurones convolutifs. Deux méthodes de fusion sont présentées pour injecter ces informations : au début ou à la fin du processus de classification. Les expérimentations montrent une amélioration des performances en termes de précision en ajoutant des informations dynamiques pour les deux méthodes de fusions. Les

résultats sont similaires pour les deux méthodes. Cela démontre que le choix et le type des informations à injecter dans le réseau est plus important que la méthode de fusion. Avec un schéma de fusion précoce et une architecture VGG, nous avons atteint 95.00 % de précision avec un temps d'inférence moyen de 20 ms. Ces résultats **surpassent le reconnaisseur de l'état de l'art** de 1.25 % de précision dans un contexte de reconnaissance de caractères isolés et ont donné lieu à une publication lors de la conférence ICFHR [2]. Dans un contexte d'analyse de mots, notre reconnaisseur n'a pas d'impact significatif sur la précision finale du système IntuiScript. Le prochain chapitre traite de la reconnaissance et de la segmentation d'un mot en caractères à l'aide d'**un modèle d'apprentissage profond**. Notre objectif est de concevoir un modèle qui surpasse le système IntuiScript en précision de reconnaissance et de segmentation.

Chapitre 5

Reconnaissance et segmentation de mots manuscrits

Dans le chapitre précédent, nous avons présenté un système de reconnaissance de caractères utilisant les informations dynamiques présentes dans le tracé en ligne. Ici, nous étendons cette approche avec un modèle d'apprentissage profond dans un contexte de **reconnaissance de mots** à travers deux stratégies. La première stratégie d'amélioration regroupe différentes méthodes d'injection de connaissances dynamiques présentes dans le tracé en ligne dans un modèle d'apprentissage profond dans le but d'améliorer ses performances en reconnaissance. Dans la seconde stratégie, nous expliquons comment obtenir une **segmentation lettres** à partir du résultat de reconnaissance mots. Ensuite, nous proposons plusieurs méthodes hybrides d'amélioration de la segmentation pour l'analyse d'écriture d'enfants. Ce chapitre présente les travaux en lien avec la troisième contribution.

5.1 Intégration des primitives de segmentation dans un réseau de neurones

Nous nous basons cette fois sur un réseau CRNN présenté dans la partie 3.2.2. Le paramétrage des filtres de convolutions du CRNN induit un **découpage fixe du mot** à reconnaitre. Un mot manuscrit en ligne peut être découpé en primitives de segmentation à l'aide de connaissances expertes comme expliqué dans la partie 2.2.1. Nous décrivons plusieurs stratégies d'injection des primitives de segmentation dans un reconnaisseur de mots de type CRNN (présenté dans la partie 3.2.2).

5.1.1 Représentation des entrées dans le réseau de neurones

Le processus de reconnaissance d'un CRNN utilise des couches de convolutions pour extraire des caractéristiques spatiales d'une image d'entrée. Nous considérons donc des

images de mots. Les images sont représentées sous forme d'une séquence de vecteurs de caractéristiques où chaque vecteur correspond aux caractéristiques d'une **sous-partie de l'image** de taille fixe. La taille de cette sous-partie dépend du paramétrage des couches de convolutions. Nous pouvons découper l'image d'entrée en une séquence d'imagettes et traiter indépendamment chaque imagette avec les mêmes couches de convolutions du réseau comme illustré dans la figure 5.1.

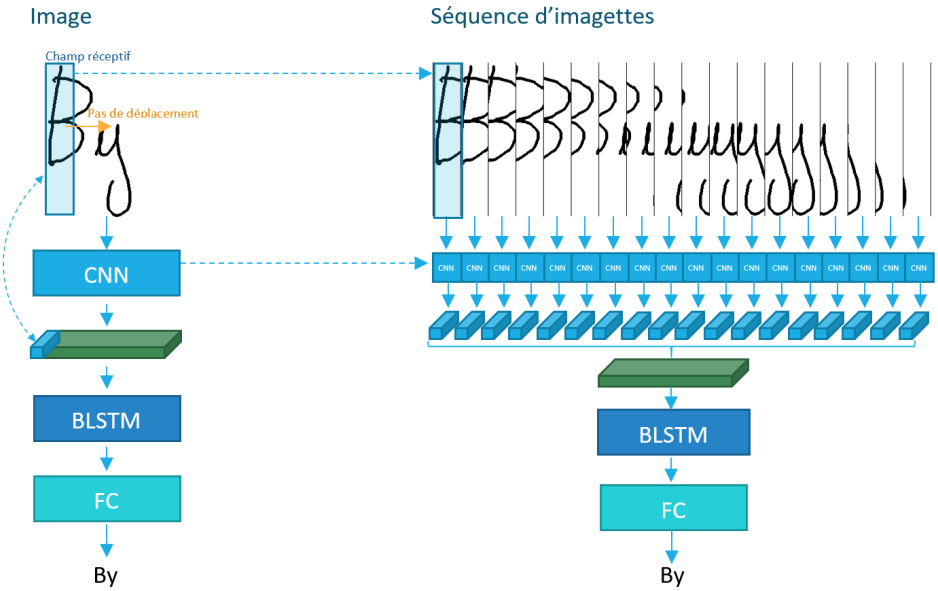

FIGURE 5.1 – Exemple de représentation d'une image et d'une séquence d'imagettes dans un CRNN composé de couches de convolutions (CNN), de couches récurrentes bidirectionnelles (BLSTM) et de couches totalement connectées (FC). Dans la séquence d'imagettes, les traits verticaux représentent la séparation entre deux imagettes. Ils sont utilisés uniquement pour la visualisation et ne sont pas présents dans les entrées du réseau.

L'extraction des caractéristiques pour une séquence d'imagettes ou une image est la même. Cependant, le découpage en séquence augmente la taille des entrées (duplication de certaines zones de l'image) et donc augmente le nombre de calculs à effectuer. L'objectif de ce découpage en séquence est de pouvoir injecter par la suite des connaissances expertes ou de modifier ce découpage pour focaliser le réseau sur des zones spatiales spécifiques. Ensuite, le réseau fonctionne de la même manière pour l'approche image et séquence d'imagettes. Les caractéristiques spatiales extraites par les couches de convolutions (représentées en vert sur la figure) sont utilisées par les couches récurrentes pour extraire des caractéristiques temporelles puis par les couches totalement connectées pour faire la prédiction de mots.

5.1.2 Spécification des stratégies de découpage

À partir de connaissances expertes liées au domaine de l'écriture décrites dans la partie 2.2.1, le signal en ligne représentant un mot ou un groupe de mots peut être segmenté en primitives regroupées dans un **treillis de segmentation**, comme l'illustre la figure 5.2. Ces primitives matérialisent des fragments de lettres, des lettres entières ou bien des fusions de fragments de deux lettres distinctes. Dans cette partie, nous avons exploré trois stratégies d'hybridation d'un treillis appliqué au **signal en ligne** et d'un réseau CRNN utilisant des images **hors ligne** dans le but d'améliorer la reconnaissance. La première stratégie intègre les primitives du premier niveau dans une image, la seconde focalise le réseau par rapport aux positions de début de fin et de primitives et la troisième intègre les primitives dans une séquence d'imagettes multicanales.

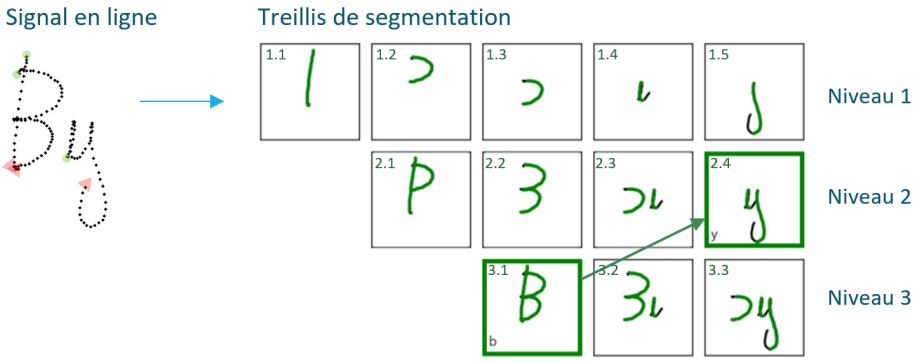

FIGURE 5.2 – Exemple d'un treillis de segmentation basé sur des connaissances expertes. La segmentation lettres parfaite est toujours présente dans les hypothèses du treillis.

Stratégie 1 : Intégration des primitives du premier niveau dans une image

La première stratégie illustrée dans la figure 5.3 utilise les primitives du premier niveau du treillis de segmentation. Les primitives sont mises bout à bout dans une image sur l'axe x avec un espacement d'un pixel en préservant également les espaces inter-mots. Cette nouvelle image est utilisée par le CRNN pour faire la prédiction de mots. Plutôt que d'utiliser un découpage fixe illustré dans la figure 5.1, nous proposons d'utiliser un découpage dynamique basé sur le treillis de segmentation dans le but de guider le CRNN.

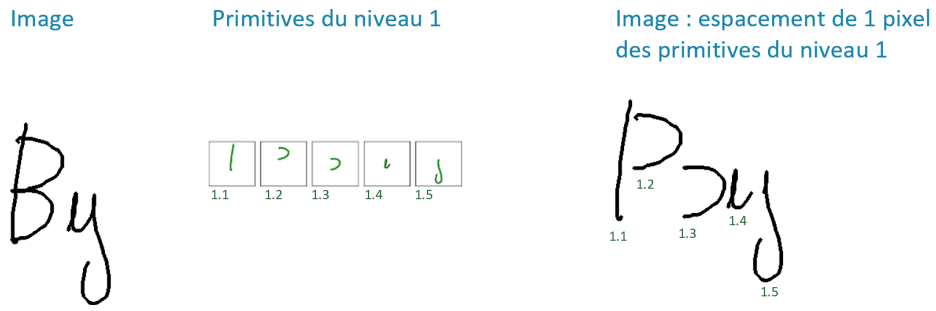

FIGURE 5.3 – Exemple de représentation des primitives du premier niveau du treillis de segmentation avec un espacement d'un pixel dans une image.

Stratégie 2 : Extraction de sous-parties de l'image à partir des positions de début et fin des primitives

La seconde stratégie illustrée dans la figure 5.4 utilise l'abscisse des positions de début et de fin de chaque primitive présente dans le treillis de segmentation pour extraire une zone de l'image originale de largeur w fixe. L'abscisse délimite le début de la zone à extraire. La stratégie ordonne les positions par leur abscisse et enlève les doublons (la fin d'une primitive peut être le début de la suivante). Les zones extraites forment une séquence d'imagettes où chaque image est traitée indépendamment par les couches de convolutions du réseau. L'idée est de focaliser le réseau sur les zones du mot contenant le plus d'informations.

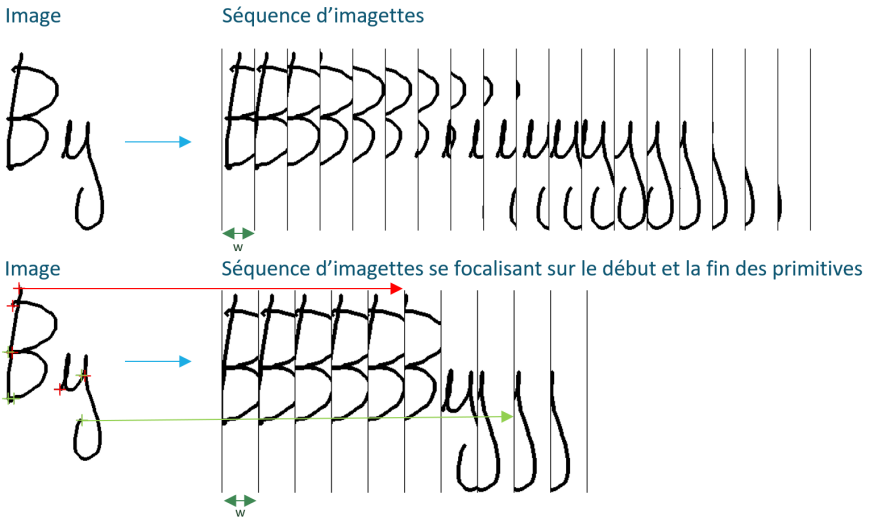

FIGURE 5.4 – Exemple d'extraction de sous-parties de l'image à partir des positions de début (rouge) et fin (vert) des primitives.

Stratégie 3 : Intégration des primitives dans une séquence d'imagettes multicanales

La troisième stratégie illustrée dans la figure 5.5 intègre les primitives des différents niveaux du treillis de segmentation dans une séquence d'imagettes multicanales. Le premier canal contient une séquence d'imagettes représentant le tracé d'origine. Les autres canaux servent à intégrer les primitives de segmentation des différents niveaux. Chaque primitive est intégrée une seule fois dans une imagette de la séquence dans laquelle elle est la mieux incluse (distance en x entre le centre de l'imagette et le centre de la primitive). L'objectif est de traiter le mot comme une séquence d'imagettes et d'enrichir certaines imagettes dans lesquelles il y a le plus d'information, *i.e.* celles qui contiennent le mieux les primitives et donc potentiellement une lettre.

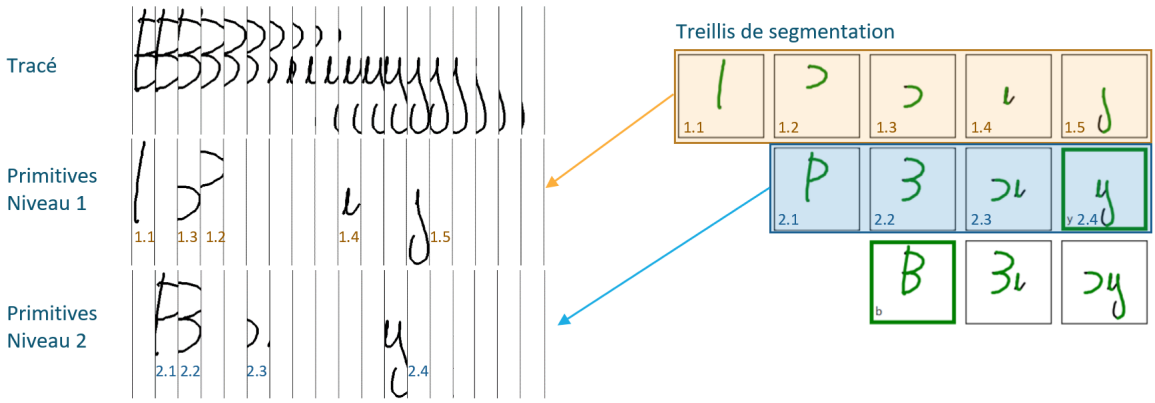

FIGURE 5.5 – Exemple d'une séquence d'imagettes multicanales avec le tracé et les primitives des niveaux 1 et 2.

5.2 Expérimentations : reconnaissance de mots d'adulte

5.2.1 Protocole

Dans cette étude expérimentale, les réseaux sont entrainés et évalués sur des **écritures anglaises d'adulte** (jeu de données IAM-OnDB présenté dans la partie 1.4.1). Nous avons choisi le jeu de données IAM-OnDB, car il est composé de plus de mots et l'écriture est moins déformée que pour le jeu de données d'écriture d'enfant à notre disposition. L'objectif, dans un premier temps, est de vérifier que les stratégies énoncées précédemment apportent une amélioration de la reconnaissance. Les images d'entrée sont

redimensionnées à une hauteur de 128 pixels et nous ajoutons du blanc pour obtenir une largeur fixe de 1280 pixels. Nous appliquons de l'augmentation de données (bruit gaussien et inclinaison) aléatoirement à chaque époque de l'entrainement d'un réseau CRNN (cf. partie 3.2.2). Suite à la recherche des meilleurs hyperparamètres sur l'ensemble de validation, pour l'entrainement, nous avons fixé la taille du batch à 4, choisi l'optimiseur ADAM et défini le taux d'apprentissage sur 10^{-3} pour toutes les expérimentations. Pour évaluer la reconnaissance, nous utilisons le taux d'erreur caractère (CER) et le taux d'erreur mot (WER) détaillé dans la partie 1.2.1. Les réseaux sont entrainés pendant 400 époques et sont évalués sur le jeu de validation.

5.2.2 Résultats

Le tableau 5.1 compare les résultats de reconnaissance obtenus pour chaque stratégie présentée précédemment. Les deux premières lignes du tableau correspondent à l'approche de référence, *i.e.* en utilisant uniquement le tracé de l'écriture sous la forme d'une image ou d'une séquence d'images. Nous pouvons observer que l'injection d'informations dynamiques utilisée dans le chapitre 4 pour la reconnaissance de caractères dégrade la reconnaissance et n'est donc pas adaptée dans un contexte de reconnaissance de mots (lignes 3 à 5). L'utilisation unique des primitives du niveau 1 dans une image à la place du tracé (stratégie 1) dégrade la reconnaissance (ligne 4). Le fait de déconstruire le tracé enlève une cohérence spatiale dans l'écriture et rend l'extraction de caractéristiques par les couches de convolutions moins performante. L'utilisation des positions de début et fin des primitives pour extraire les sous-parties de l'image (stratégie 2) dégrade également la reconnaissance (ligne 5). En extrayant seulement des sous parties de l'image, le réseau dispose sans doute de moins d'information et cela ajoute probablement des déplacements artificiels pour l'extraction de caractéristiques temporelles par les couches récurrentes du réseau. La suite des résultats (lignes 8 à 11) compare les stratégies d'injection de primitives dans une séquence d'imagettes multicanales (stratégie 3). Nous remarquons que l'utilisation de l'information sur le tracé (lignes 8 et 9) est indispensable pour obtenir une bonne reconnaissance. L'injection des primitives du niveau 1 obtient le meilleur résultat et surpasse légèrement l'approche de référence.

Les stratégies d'injection de l'information dynamique développées pour faire de la reconnaissance de caractères ne sont donc pas avérées intéressantes dans un contexte de reconnaissance de mots. La plupart des méthodes développées dans ce chapitre déconstruisent le tracé d'entrée et enlèvent une cohérence spatiale dans l'écriture. Cela a pour

TABLE 5.1 – Résultat de reconnaissance du modèle CRNN avec les différentes configurations d'entrée. La reconnaissance est évaluée avec le taux d'erreur caractère (CER) sur le jeu de validation en % (une valeur plus faible est meilleure).

Stratégie	Contenu des entrées	Représentation des entrées	CER (%) val
Référence	Tracé	Image	**6.5**
Référence	Tracé	Séquence d'images	**6.5**
Chapitre 4	Tracé, Tracé descendant	Image	7.2
Chapitre 4	Tracé, cosinus et sinus	Image	7.4
Chapitre 4	Tracé, Tracé descendant, cosinus et sinus	Image	7.6
Stratégie 1	Primitive niveau 1, espacement 1 px	Image	7.3
Stratégie 2	Pas adaptatif	Séquence d'images	9.0
Stratégie 3	Tracé, primitives niveau 1	Séquence d'images	**6.4**
Stratégie 3	Tracé, primitives niveau 1 et 2	Séquence d'images	6.7
Stratégie 3	Primitives niveau 1	Séquence d'images	9.2
Stratégie 3	Primitives niveau 1 et 2	Séquence d'images	8.4

effet de dégrader les performances de reconnaissance. Seule l'injection des primitives du premier niveau du treillis de segmentation avec le tracé d'origine permet d'améliorer légèrement la reconnaissance du réseau.

Dans cette partie, nous venons de voir que les modalités d'intégration des connaissances expertes dans un CRNN ne sont pas simples et seulement une des stratégies proposées permet un léger gain de reconnaissance sur des écritures d'adultes. Notre objectif à long terme est de concevoir un système d'analyse d'écriture d'enfants. Cette analyse est possible en combinant une tâche de reconnaissance (identification des caractères) avec une tâche de segmentation (localisation des caractères dans le mot). Dans la suite de ce chapitre, nous nous intéresserons à la seconde tâche liée à une tâche d'analyse d'écriture : la segmentation. Nous commençons par définir la segmentation lettres implicite obtenue à partir du modèle de reconnaissance mots. Nous nous focaliserons sur l'amélioration de cette segmentation en modifiant la fonction de coût d'entrainement du modèle ainsi que l'utilisation des primitives en post traitement pour améliorer cette segmentation.

5.3 Segmentation en lettres d'un mot à l'aide d'un modèle de reconnaissance

Dans la partie précédente, nous nous étions intéressés à l'aspect de reconnaissance. Dans cette partie, nous intéressons à l'aspect segmentation. Nous commençons par décrire comment obtenir une segmentation lettres à partir des prédictions d'un modèle de reconnaissance mots. Nous présentons ensuite deux stratégies pour améliorer la qualité de la segmentation. La première consiste à modifier la méthode d'entrainement CTC décrite dans la partie 3.2.1 tandis que la seconde propose d'utiliser un treillis de segmentation après l'entrainement du modèle de reconnaissance pour augmenter la précision de la segmentation. Ces travaux utilisent les architectures CRNN et Seq2Seq présentées dans la partie 3.2.

5.3.1 Définition de la segmentation lettre

Dans cette partie, nous expliquons comment obtenir une segmentation lettres à partir du résultat d'un modèle de reconnaissance d'un réseau CRNN ou Seq2Seq.

Le processus de reconnaissance du modèle CRNN peut être résumé de la manière suivante. Des couches de convolutions extraient des caractéristiques de l'image d'entrée sous forme d'une **séquence de vecteurs de caractéristiques** où chaque élément correspond aux caractéristiques extraites pour une fenêtre de l'image. Cette fenêtre équivaut au champ réceptif des convolutions du modèle. Sa taille est fixe et est définie par le nombre et la configuration des couches de convolutions. Les caractéristiques sont ensuite utilisées par les couches récurrentes (BLSTM) et entièrement connectées (FC) pour calculer la **séquence de caractères** (avec la classe blank et la répétition possible d'un même caractère) contenue dans l'image. Nous mettons en correspondance la séquence de vecteurs de caractéristiques de sortie des couches de convolutions et la séquence de caractères comme illustrées dans la figure 5.6. Une prédiction caractère est associée à un vecteur de caractéristiques ayant le même indice dans la séquence et ainsi à une zone spécifique de l'image. Cette zone ou fenêtre représente la segmentation du caractère. Cette séquence de caractères n'est pas le résultat de reconnaissance définitif, car elle contient les prédictions de classe blank et la succession de prédictions pour un même caractère (*cf.* lettre "y" dans la figure 5.6). Dans le cas où il y a une répétition de prédiction de caractères, les fenêtres associées sont fusionnées. Cela a pour effet d'augmenter la largeur de la zone

de segmentation du caractère. L'utilisation directe de cette zone pour la segmentation finale est limitée, car elle ne tient pas compte de l'influence des couches récurrentes. En effet, elles peuvent induire un décalage spatial entre la séquence de caractéristiques issue des convolutions et la séquence de caractères prédite. Pour essayer de limiter ce décalage, nous explorons également la segmentation d'un modèle Seq2Seq utilisant un mécanisme d'attention.

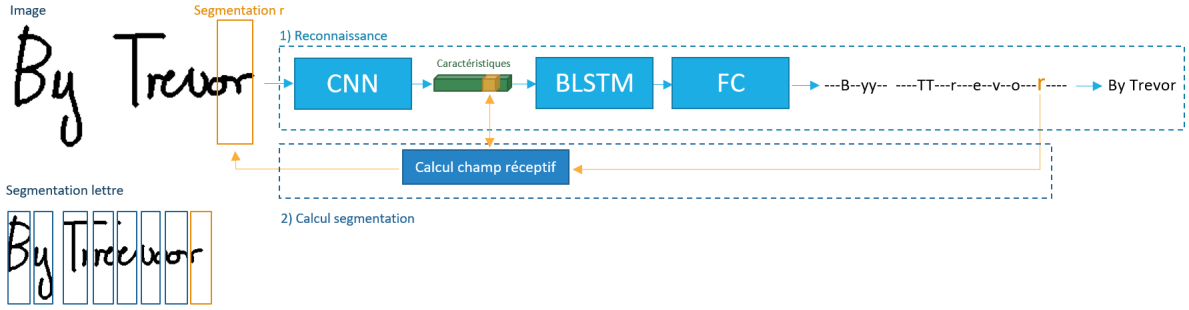

FIGURE 5.6 – Exemple de segmentation en lettres avec le modèle CRNN.

Le modèle Seq2Seq utilisé dans ces travaux utilise un encodeur CRNN et un décodeur utilisant un système d'attention. La figure 5.7 illustre le processus pour définir la segmentation en lettres du décodeur. Pour chaque lettre prédite par le décodeur, nous utilisons le vecteur de poids d'attention qui a servi à prédire ce caractère. Le vecteur de poids d'attention sert à focaliser le décodeur sur une sous-partie du **vecteur de caractéristiques de sortie de l'encodeur**. De la même manière que pour le CRNN, pour chaque prédiction caractère du décodeur, nous mettons en correspondance le vecteur de poids associé à ce caractère et la séquence de vecteur de caractéristiques de sortie avec celle extraite par les couches de convolutions dans l'image entière. Ainsi, si **un poids** du vecteur d'attention est supérieur à un seuil, la fenêtre de l'image associée au vecteur de caractéristiques lui-même associé à ce poids fait partie de la segmentation lettre. Une limite de cette approche, c'est qu'elle ne vérifie pas la cohérence spatiale de la segmentation d'une lettre. En effet, pour un caractère, les fenêtres de l'image correspondant à la segmentation peuvent être discontinues, ce qui n'est pas le cas pour l'encodeur CRNN.

Les modèles CRNN et Seq2Seq sont entrainés de manière supervisée uniquement pour une tâche de reconnaissance et donc l'aspect segmentation n'est pas pris en compte lors de l'entrainement. La précision de la segmentation lettres obtenue dépend de la dimension du champ réceptif des convolutions associée à une prédiction lettre. Dans un contexte

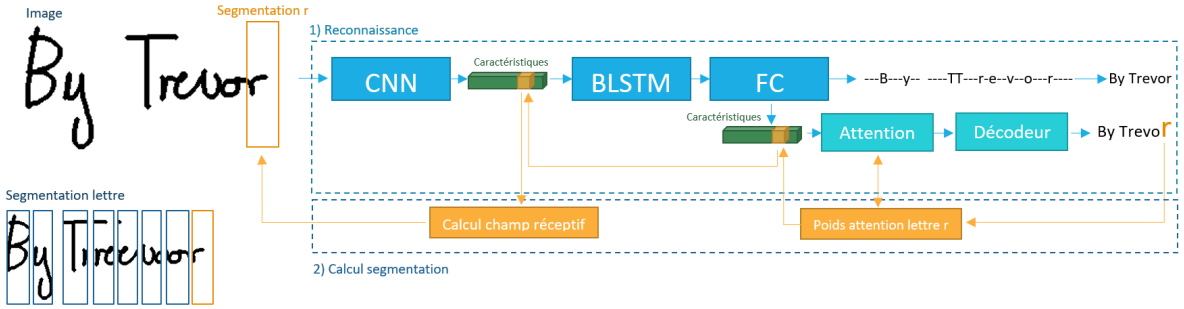

FIGURE 5.7 – Segmentation lettre avec le décodeur du Seq2Seq.

d'écriture manuscrite cursive, les caractères ont des largeurs variables. La dimension du champ réceptif étant fixe, la segmentation implicite des modèles CRNN et Seq2Seq ne permettent pas d'obtenir de segmentation précise notamment pour des caractères de largeur supérieure ou inférieure à celle du champ réceptif. Pour une largeur supérieure, la segmentation du caractère est coupée et pour une largeur inférieure la segmentation contient des morceaux des caractères adjacents. De plus, comme expliqué dans la partie 3.2.1 la méthode d'entrainement CTC utilisé par le CRNN à tendance à prédire une ou deux fenêtres par caractère ce qui limite les dimensions de chaque caractère segmenté. Par ailleurs, un champ réceptif étant délimité par un rectangle, cette approche ne permet pas de segmenter précisément les lettres inclinées. Pour toutes ces raisons, les segmentations lettres des modèles CRNN et Seq2Seq ne sont pas assez précise pour une utilisation dans un contexte d'analyse de l'écriture.

La suite de ce chapitre présente deux stratégies pour améliorer la qualité de la segmentation lettres. La première s'intéresse à la modification de la méthode d'entrainement CTC utilisée pour optimiser une tâche de reconnaissance d'une séquence de caractères. La seconde stratégie utilise le treillis de segmentation introduit précédemment en post traitement pour augmenter la précision de segmentation.

5.3.2 Amélioration de la qualité de segmentation

Modification de la méthode CTC

Nous avons vu dans la partie 3.2.1, que la méthode d'entrainement CTC a tendance à avoir des pics d'activation, *i.e.* prédire beaucoup de "blank" entre les prédictions des caractères. Ce comportement montre que le réseau sur apprend un alignement de recon-

naissance dominant. Pour l'approche CTC classique, le réseau prédit une séquence de caractères (avec la classe "blank") qui permet d'obtenir le bon résultat de reconnaissance. Cependant, la segmentation implicite liée à cette séquence ne couvre pas toute l'écriture du mot ce qui a pour effet d'obtenir une segmentation imprécise comme illustrée dans la figure 5.8 pour l'approche "CTC classique".

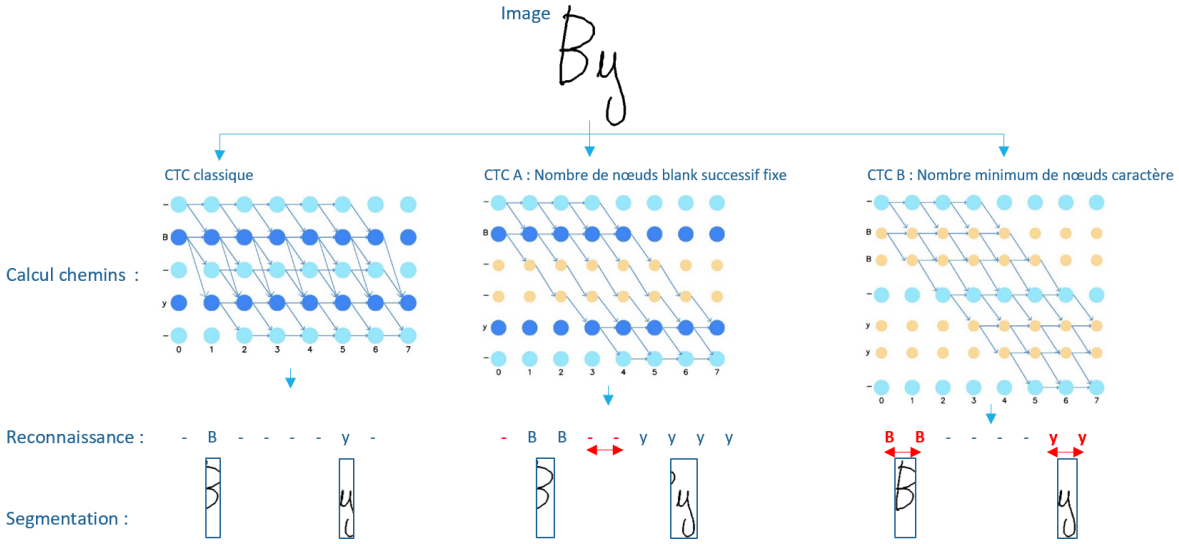

FIGURE 5.8 – Exemple de calcul des chemins avec le CTC classique, la stratégie "CTC A" avec maximum deux nœuds blank, la stratégie "CTC B" avec minimum deux nœuds caractère et les résultats de reconnaissance et segmentation associés.

Notre approche, illustrée dans la figure 5.8, propose de modifier le calcul de coût de la méthode CTC afin d'augmenter le taux de couverture de l'écriture de la segmentation lettres associée au résultat de reconnaissance. Nous distinguons trois stratégies :

— CTC A : Fixer le nombre de nœuds prédit "blank" consécutivement. L'objectif est de définir un nombre maximal de fenêtres "blank" entre chaque prédiction lettre. Le fait de limiter le nombre de prédictions de "blank" total dans la séquence augmente le nombre de prédictions de caractères. Le but est d'augmenter le nombre de fenêtres par caractère pour étendre le taux de couverture et ainsi améliorer la segmentation.

— CTC B : Forcer la prédiction de plusieurs nœuds lettres. L'objectif est d'augmenter le taux de couverture en prédisant un nombre minimal de fenêtres par caractère.

— CTC C : Cette stratégie est une spécialisation de l'approche CTC B dans le cas où nous définissons le nombre minimum de caractères à deux. Pour chaque lettre,

nous définissons une classe "début" et une classe "fin". Le nombre de classes est multiplié par deux. L'objectif est de spécialiser le CTC afin que le réseau prédise le début et la fin d'un caractère dans le but d'augmenter le taux de recouvrement et donc d'améliorer la segmentation.

Post traitement : utilisation d'un treillis de segmentation

Cette méthode illustrée dans la figure 5.9 propose d'utiliser la segmentation lettre obtenue d'un modèle de reconnaissance (CRNN ou Seq2Seq) puis de chercher dans un treillis de segmentation (détaillée dans la partie 2.2.1) l'hypothèse de segmentation la plus proche pour améliorer la segmentation finale. Nous utilisons le critère d'IoU pixel pour définir l'hypothèse du treillis la plus proche. Le segmenteur utilise un signal en ligne en entrée et les hypothèses de segmentation sont définis par une liste de points. En plus d'améliorer la précision de segmentation, cette méthode fournit une segmentation sémantique du mot (calculée à l'aide des points des hypothèses du treillis sélectionnées).

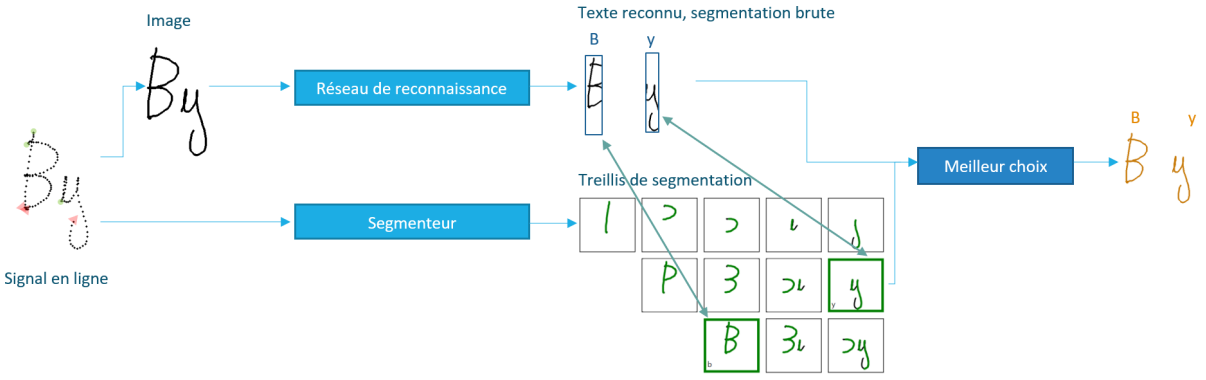

FIGURE 5.9 – Exemple d'utilisation d'un treillis de segmentation en post traitement pour améliorer la qualité de segmentation d'un modèle reconnaissance.

5.4 Expérimentations : reconnaissance et segmentation de mots d'enfants

5.4.1 Protocole

Pour pouvoir évaluer la qualité de la segmentation en lettres d'une méthode, il est nécessaire d'avoir des données annotées avec une segmentation niveau lettres ce qui n'est pas le cas des jeux de données publics comme IAM-OnDB [LB05] utilisé au chapitre précédent. Cette étude expérimentale utilise un jeu de données privé ICDB-Words-A présenté dans la partie 1.4.2 contenant des **mots écrit pas des enfants** en cours d'apprentissage de l'écriture. Le jeu de données contient 5 649 mots pour l'entrainement, 1 000 pour la validation et 913 pour le test. L'annotation niveau "lettres" étant beaucoup plus coûteuse (en temps) à obtenir, seul le jeu de test est annoté au niveau lettres pour les expérimentations de ce chapitre. Nous verrons au chapitre suivant que l'annotation du jeu d'entrainement s'est avéré nécessaire. Le nombre de données d'entrainement étant limité, nous utilisons un jeu de données d'écriture d'adulte IAM-OnDB pour pré-entrainé les modèles d'apprentissage profond. Suite à la recherche des meilleurs hyperparamètres sur l'ensemble de validation, pour l'entrainement des réseaux de neurones, nous avons fixé de taille du batch à 16, choisi l'optimiseur ADAM et défini le taux d'apprentissage sur 10^{-3} pour toutes les expérimentations. Dans cette étude, nous évaluons l'aspect de reconnaissance avec le taux d'erreur "caractère" (CER) et le taux d'erreur "mot" (WER), ainsi que l'aspect segmentation au niveau lettre avec les métriques en ligne de précision, rappel, f1 et la métrique hors ligne d'intersection over union (IoU). Les métriques sont présentées en détail dans la partie 1.2.2. Les modèles sont entrainés pendant 400 époques. Il n'y a pas de supervision de la segmentation pendant l'entrainement. Nous utilisons le modèle ayant le meilleur CER sur le jeu de validation pour l'évaluation sur le jeu de tests. Le réseau Seq2Seq fait une prédiction pour l'encodeur et une pour le décodeur. Dans cette étude, nous séparons les deux pour évaluer laquelle est la plus performante.

5.4.2 Résultats

Cette étude expérimentale commence par évaluer la performance de reconnaissance des modèles CRNN et Seq2Seq par rapport à l'état de l'art sur des écritures d'adultes ainsi que l'impact du pré-entrainement pour la reconnaissance d'écriture d'enfants. Elle évalue ensuite la performance en reconnaissance et segmentation sur des écritures d'enfants

des différentes contributions détaillées précédemment. Enfin, elle compare la meilleure approche avec l'état de l'art sur des écritures d'enfants.

Le tableau 5.2 présente les résultats de reconnaissance d'écriture d'adulte des modèles CRNN et Seq2Seq utilisés dans ces travaux comparés à l'état de l'art. Ces résultats montrent que l'architecture Seq2Seq est plus performante que l'architecture CRNN et surpasse l'état de l'art (BLSTM) de +0.8 % CER et obtient un WER équivalent pour l'approche n'utilisant pas de modèle de langue.

TABLE 5.2 – Comparaison avec l'état sur des écritures adultes (jeu de test de IAM-OnDB) pour les modèles CRNN et Seq2Seq. La reconnaissance est évaluée avec le taux d'erreur caractère (CER) et le taux d'erreur mot (WER). Une valeur plus faible est meilleure.

Architecture	Modèle de langue	CER (%)	WER (%)
BLSTM [Car+20]	Non	5.9	**18.6**
BLSTM [Car+20]	Oui	4.0	10.6
CRNN 3.2.2	Non	8.1	29.6
Seq2Seq Encodeur 3.2	Non	**5.1**	**18.7**
Seq2Seq Décodeur 3.2	Non	5.5	20.1

Le tableau 5.3 montre l'intérêt d'utiliser des données d'écriture d'adultes pour pré-entrainer les modèles CRNN et Seq2Seq. Sans le pré-entrainement, la performance de reconnaissance des modèles est faible, car le nombre de données d'entrainement est probablement insuffisant.

TABLE 5.3 – Résultat de reconnaissance sur le jeu de test d'écriture d'enfants pour les modèles CRNN et Seq2Seq. Certains modèles sont préentrainés sur le jeu de données IAM-OnDB contenant des écritures d'adulte anglaise.

Architecture	Préentrainement	CER (%)	WER (%)
CRNN	Non	43.4	77.3
CRNN	Oui	5.5	18.7
Seq2Seq Encodeur	Non	24.7	53.4
Seq2Seq Décodeur	Non	24.9	53.4
Seq2Seq Encodeur	Oui	**4.7**	**16.4**
Seq2Seq Décodeur	Oui	6.7	22.1

Le tableau 5.4 détaille les résultats de reconnaissance et de segmentation obtenus par les stratégies spécifiées précédemment. Les lignes 1 à 3 correspondent aux modèles sans modification. Ce sont les approches de référence. Ensuite, Les lignes 4 à 6 présentent les résultats avec les modifications apportées sur la fonction CTC (CTC A : fixe le nombre

de nœuds blank à 5, CTC B : minimum de 2 nœuds caractère et CTC C : dédoublement lettres début et fin). Les résultats présentés correspondent aux valeurs d'hyper paramètres ayant les meilleures performances. Les lignes 7 à 12 présentent les résultats avec des méthodes de régularisation du CTC de l'état de l'art. Avec les approches de références, la prédiction de l'encodeur du Seq2Seq est la plus performante. Au niveau de la segmentation, le décodeur à un score F1 plus important (+9.9 %) que l'encodeur, mais un score d'IoU plus faible (-2.2 %). L'approche CTC A dégrade la reconnaissance et la segmentation. La valeur optimale du nombre de "blank" entre chaque caractère est spécifique à chaque image, pour certaind mots la méthode ajoute des chemins incohérents ce qui explique la dégradation des résultats. Un chemin est incohérent lorsque l'alignement entre les caractères à reconnaitre et les caractères contenus dans l'image est erroné. L'approche CTC B et C dégradent la reconnaissance et améliorent légèrement la segmentation (+1.1 % F1, +2.3 % IoU pour l'approche B et +2.8 % F1). Les modifications apportées par ces approches ajoutent également des chemins incohérents dans le calcul du CTC pour des mots où la largeur des lettres est inférieur à deux fenêtres. Les légers gains en segmentation s'expliquent, car le réseau prédit deux fenêtres au lieu d'une pour chaque caractère et la segmentation est donc mécaniquement plus précise. La régularisation proposée par [ZSN21] et détaillée dans la partie 3.2.1 permet de prédire plus de fenêtres par caractères. Cependant, dans l'architecture Seq2Seq utilisée, il y a du chevauchement entre les fenêtres. Cela a pour effet d'utiliser des mêmes parties de l'image dans la segmentation de plusieurs caractères, ce qui dégrade la qualité de segmentation. De plus, cette approche force à reconnaitre des caractères dans des fenêtres contenant une sous-partie du caractère, ce qui dégrade les performances de reconnaissance. La régularisation EnCTC proposée par [LJZ18] ($\beta = 0.125$) et détaillée dans la partie 3.2.1 améliore légèrement la performance de segmentation et ne dégrade pas la performance de reconnaissance. La régularisation EsCTC proposée par [LJZ18] ($\tau = 1.5$) dégrade la performance de reconnaissance et de segmentation. Cette dernière se base sur un espacement de taille fixe entre les caractères et n'est pas adaptée aux caractères manuscrits. Contrairement aux caractères imprimés utilisés dans l'article, les caractères manuscrits ont des largeurs variables. Enfin, la méthode de post-traitement avec le treillis de segmentation permet d'obtenir un gain important sur la qualité de segmentation.

Le tableau 5.5 montre l'intérêt d'utiliser un treillis de segmentation en post-traitement. Le tableau présente les résultats pour la stratégie ayant les meilleures performances en reconnaissance et segmentation (Seq2Seq encodeur régularisation EnCTC). Le post-

TABLE 5.4 – Résultat de reconnaissance et de segmentation sur le jeu de test d'écriture d'enfant. La reconnaissance est évaluée avec le taux d'erreur caractère (CER) et le taux d'erreur mot (WER). Une valeur plus faible est meilleure. La segmentation est évaluée avec la précision, le rappel, le score F1 et l'IoU. Une valeur plus haute est meilleure.

Approche	Architecture	CER (%)	WER (%)	Précision (%)	Rappel (%)	F1 (%)	IoU (%)
Référence	CRNN	5.5	18.7	49.1	77.7	56.8	51.4
Référence	Seq2Seq Encodeur	**4.7**	**16.4**	49.2	78.2	56.9	53.2
Référence	Seq2Seq Décodeur	6.7	22.1	62.2	78.3	66.0	51.0
CTC A	Seq2Seq Encodeur	11.2	31.9	50.9	50.6	48.2	35.9
CTC B	Seq2Seq Encodeur	5.9	20.7	53.2	73.3	**58.0**	**55.5**
CTC C	Seq2Seq Encodeur	6.0	19.8	51.9	83.8	**59.7**	53.2
CTC [ZSN21]	Seq2Seq Encodeur	7.0	23.4	59.5	57.9	55.4	40.3
CTC [ZSN21]	Seq2Seq Décodeur	8.7	27.3	61.8	66.0	59.5	48.1
EnCTC $\beta = 0.125$ [LJZ18]	Seq2Seq Encodeur	**4.5**	**16.8**	51.1	76.8	**58.0**	**55.1**
EnCTC $\beta = 0.125$ [LJZ18]	Seq2Seq Décodeur	7.6	22.6	59.2	62.0	57.2	50.5
EsCTC $\tau = 1.5$ [LJZ18]	Seq2Seq Encodeur	6.9	19.4	46.0	73.4	53.0	49.3
EsCTC $\tau = 1.5$ [LJZ18]	Seq2Seq Décodeur	8.8	29.1	56.4	57.3	52.9	45.9

traitement permet d'améliorer significativement la segmentation (+ 10.6% F1 et + 24.9% IoU). Nous pouvons noter que l'utilisation du treillis en post-traitement améliore les performances de segmentation pour toutes les stratégies, mais les résultats associés ne sont pas présentés dans le tableau.

TABLE 5.5 – Résultats de reconnaissance et de segmentation sur le jeu de test d'écriture d'enfants avec le post-traitement utilisant le treillis de segmentation.

Approche	Architecture	CER (%)	WER (%)	Précision (%)	Rappel (%)	F1 (%)	IoU (%)
EnCTC $\beta = 0.125$ [LJZ18]	Seq2Seq Encodeur	**4.5**	**16.8**	51.1	76.8	58.0	55.1
EnCTC $\beta = 0.125$ + Treillis [LJZ18]	Seq2Seq Encodeur	**4.5**	**16.8**	64.6	81.4	**68.6**	**80.0**

Le tableau 5.6 compare la meilleure stratégie (Seq2Seq encodeur régularisation EnCTC et post-traitement avec le treillis de segmentation) avec l'état de l'art présenté dans la partie 2.2.2 avec deux approches de guidage. Notre approche surpasse l'état de l'art en termes de reconnaissance (+ 7.1% CER pour le guidage phonétique), mais propose une segmentation lettre moins précise (- 23.1% F1 et -8.4% IoU).

TABLE 5.6 – Résultats de reconnaissance et de segmentation sur le jeu de test d'écriture d'enfants avec l'état de l'art.

Approches	Architecture	CER (%)	WER (%)	Précision (%)	Rappel (%)	F1 (%)	IoU (%)
Guidage consigne [Kri+21]	IntuiScript	13.2	34.7	91.4	91.1	91.1	87.8
Guidage module phonétique [Kri+21]	IntuiScript	11.6	29.35	**92.4**	**91.2**	**91.7**	**88.4**
EnCTC $\beta = 0.125$ + Treillis [LJZ18]	Seq2Seq Encodeur	**4.5**	**16.8**	64.6	81.4	68.6	80.0

Les résultats des expérimentations ont montré que les modèles (CRNN et Seq2Seq) dédiés à la reconnaissance de mots manuscrits sont performants pour faire de la recon-

naissance et surpassent l'état de l'art dans le cadre d'analyse d'écriture d'enfants, mais ils sont imprécis pour segmenter en lettres. Les stratégies développées pour modifier le fonctionnement de la fonction d'entrainement CTC de ces modèles se sont révélées inefficaces, car en cherchant à améliorer la qualité de segmentation, elles dégradent la qualité de reconnaissance. L'utilisation d'un treillis de segmentation en post-traitement est plus efficace que l'injection de connaissances dynamiques dans les entrées et permet d'améliorer significativement la qualité de segmentation, mais en restant en dessous des performances de l'état de l'art. Elles ont montré qu'il est indispensable d'avoir un **modèle dédié à la segmentation** pour obtenir un système performant en segmentation, que ce soit pour le modèle de l'état de l'art IntuiScript basé sur une segmentation explicite ou bien sur l'utilisation d'un treillis de segmentation en post-traitement avec un modèle de reconnaissance. Dans un contexte de dictée d'analyse de mots manuscrits d'enfants, nous avons donc d'un côté le système IntuiScript qui n'est pas assez performant en reconnaissance et d'un autre côté un système Seq2Seq couplé à un treillis qui n'est pas assez précis en segmentation pour fournir un retour de qualité à l'enfant.

La suite des travaux explore de nouvelles stratégies pour obtenir un système à la fois performant en reconnaissance et en segmentation. La première stratégie consiste à améliorer la reconnaissance du modèle IntuiScript en renforçant le système de guidage par un modèle de reconnaissance Seq2Seq. La seconde stratégie propose d'utiliser un modèle R-CNN (Region Proposal CNN) dédié à la tâche de segmentation et de le combiner avec un modèle Seq2Seq dédié pour la tâche de reconnaissance.

Troisième partie

Contributions majeures

Chapitre 6

Intégration de la prédiction du modèle Seq2Seq dans le système de guidage du modèle IntuiScript

Dans cette approche, nous proposons d'utiliser comme point de départ le modèle existant IntuiScript présenté dans le chapitre 2. Le système a été conçu au départ pour une application dans un **contexte de recopie** dans le but de faire une analyse de la qualité de l'écriture. Le système a été étendu pour un **contexte de dictée** dans le but de faire une analyse de la qualité de l'orthographe en utilisant comme connaissances a priori la consigne et les mots phonétiquement proches de la consigne pour guider son analyse. Le système est performant en segmentation, mais il a besoin de guidage pour faire une analyse précise du mot. Notre premier objectif est d'enrichir ce guidage avec la prédiction d'un modèle Seq2Seq performant en reconnaissance présenté dans la partie 3.2.3. Par ailleurs, le temps de calcul du modèle IntuiScript augmente en fonction de la taille du mot à analyser. Pour de longs mots, le système n'est pas toujours capable de faire un retour immédiat à l'enfant. Le second objectif consiste à utiliser la segmentation approximative du réseau Seq2Seq comme connaissance pour optimiser et accélérer le temps l'analyse. Ces approches forment un système hybride entre le modèle IntuiScript et un modèle Seq2Seq appris.

Ce chapitre présente les travaux en lien avec la seconde contribution [4]. Il commence par spécifier les deux nouvelles **stratégies de guidage** du modèle IntuiScript en intégrant le résultat de reconnaissance du modèle Seq2Seq. Ensuite, il propose une méthode d'optimisation du temps de calcul en utilisant la segmentation approximative du Seq2Seq pour **élaguer le treillis de segmentation** utilisé dans le modèle IntuiScript.

6.1 Intégration de la reconnaissance d'un modèle Seq2Seq dans le guidage du modèle IntuiScript

Le modèle IntuiScript utilise des connaissances a priori pour guider son analyse. Ces connaissances regroupent la consigne du mot dicté à l'enfant ainsi que les mots, pseudo phonétiquement proches de la consigne pour anticiper les potentielles fautes d'orthographe. Mais ce guidage devient obsolète et inefficace lorsque l'enfant commet des erreurs non-phonétique. Le résultat de reconnaissance du modèle Seq2Seq (sans modèle de langue) est une approximation fiable de ce que l'enfant a écrit. Nous pouvons donc la prendre en compte pour guider le processus d'analyse d'IntuiScript. Cette partie présente une stratégie dans laquelle le processus d'analyse est guidé par la reconnaissance du modèle Seq2Seq ainsi qu'une stratégie qui combine le guidage de la consigne, des mots phonétiquement proches de la consigne et le résultat de reconnaissance du Seq2Seq.

6.1.1 Guidage de l'analyse par la reconnaissance du Seq2Seq

La figure 6.1 illustre la stratégie de guidage par la **reconnaissance** du modèle Seq2Seq. Le guidage par la reconnaissance du Seq2Seq intervient dans l'étape de **l'analyse lettre** où la séquence de lettres reconnue par le Seq2Seq guide l'analyse pour chaque hypothèse du treillis de segmentation (guidage 1). Il intervient également dans l'étape de **l'analyse mot** où le mot reconnu par le Seq2Seq guide l'analyse lors du calcul des chemins dans le treillis de segmentation (guidage 2). Le modèle Seq2Seq est très performant en reconnaissance et n'utilise pas de modèle de langue. C'est donc une bonne approximation du mot écrit par l'enfant, ce qui permet un guidage efficace du système d'analyse.

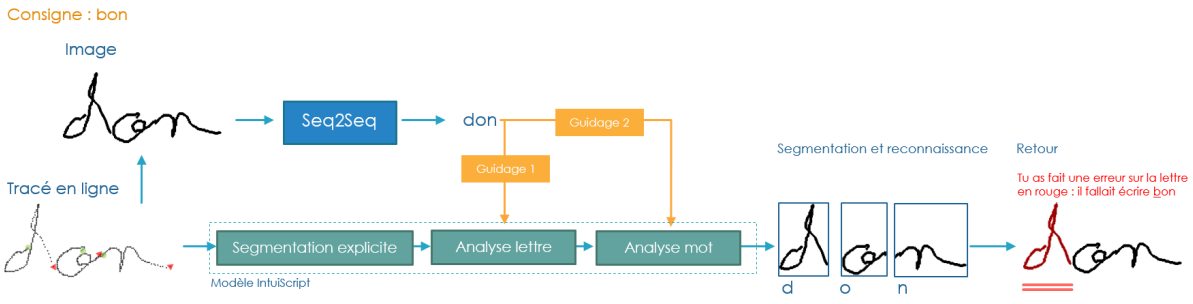

FIGURE 6.1 – Exemple d'analyse d'un mot écrit par un enfant dans un contexte de dicté avec un guidage par la reconnaissance du Seq2Seq.

La figure 6.2 met en lumière l'intérêt du guidage par le Seq2Seq. Dans cet exemple où

la consigne dictée est "cent", l'enfant a écrit le mot éloigné de la consigne "zme". La figure regroupe les résultats de trois stratégies de guidage :

— Stratégie A : le processus d'analyse est guidé par la consigne "cent". Cette stratégie est adaptée lorsqu'il n'y a pas d'erreurs, mais devient obsolète dans le cas de mots avec des fautes.

— Stratégie B : le processus d'analyse est guidé par les mots proches phonétiquement de la consigne "cent", "san", "sand" ... Cette stratégie est adaptée pour des erreurs phonétiques. Dans cet exemple ou l'erreur n'est pas phonétique, la stratégie de guidage ne fonctionne pas.

— Stratégie C : le processus d'analyse est guidé par la reconnaissance du Seq2Seq "zme". Le guidage permet au processus d'analyse de reconnaître et segmenter correctement le mot.

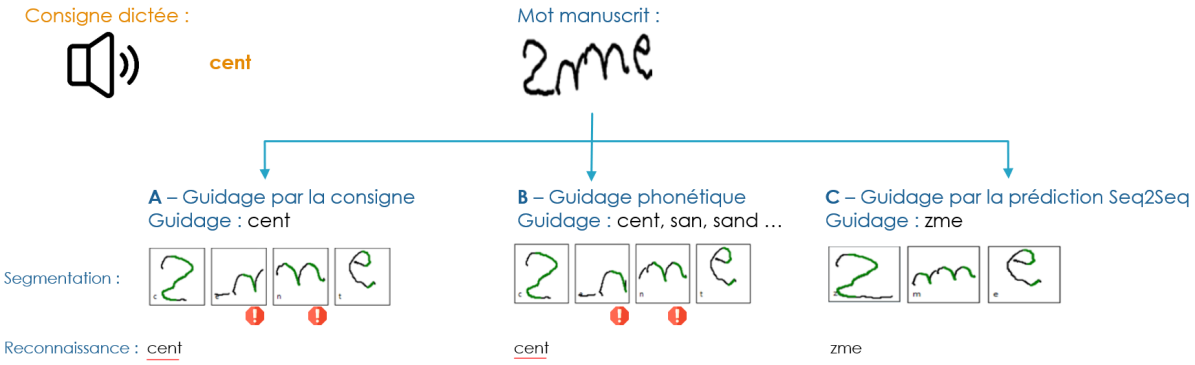

FIGURE 6.2 – Exemple d'analyse d'un mot différent de la consigne dictée avec trois stratégies de guidage : A) guidage par la consigne, B) guidage phonétique, C) guidage par la prédiction Seq2Seq.

6.1.2 Combinaison des stratégies de guidage

Le bon fonctionnement de la stratégie précédente repose sur le fait que le mot reconnu par le Seq2Seq ne contienne pas d'erreur. La figure 6.3 illustre un exemple dans lequel le Seq2Seq commet une erreur de reconnaissance. Dans cet exemple où la consigne dictée est "bien", l'enfant a écrit "biin". Nous pouvons voir que les deux premières stratégies (A et B) ont prédit correctement la première lettre "b" alors que la stratégie C a fait une erreur pour la première lettre, mais a bien reconnu le reste du mot. Dans cet exemple, aucune des stratégies de guidage ne permet de segmenter et de reconnaitre correctement le mot.

Intuitivement, nous pouvons voir que chaque stratégie est adaptée à un scénario spécifique. Nous supposons qu'elles peuvent être complémentaires. Nous proposons de combiner ces trois stratégies dans une nouvelle stratégie de guidage, appelée **fusion et compétition**. Cette dernière représente deux façons de combiner les stratégies, d'abord une conjonction en fusionnant ces connaissances préalables, puis une disjonction en introduisant une notion de concurrence entre les stratégies de prédiction. La suite de cette partie spécifie les deux étapes de cette nouvelle stratégie.

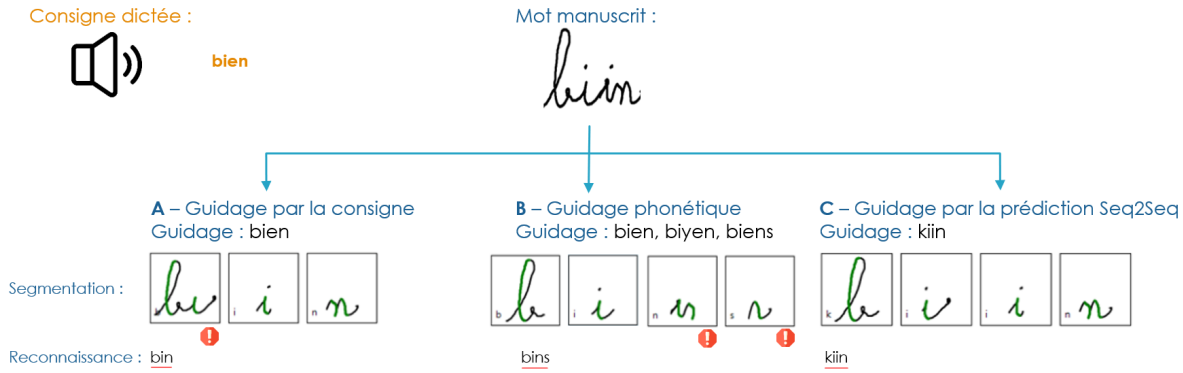

FIGURE 6.3 – Exemple d'analyse d'un mot où aucune des trois stratégies de guidage (A, B, C) ne fonctionne.

Fusion

Les résultats de **reconnaissance** des stratégies de guidage par la consigne (A), du guidage phonétique (B) et par le Seq2Seq (C) sont fusionnées pour générer une **approximation alternative** de la vérité terrain qui servira de nouvelle source de connaissance préalable pour le guidage. Ce nouveau guidage permet de traiter les cas où aucune des stratégies de guidage à elle seule ne permet de reconnaitre et de segmenter correctement le mot. Cette fusion se fait en deux étapes :

— Alignement des séquences de caractères résultantes des trois stratégies (A, B, C), de la consigne et de la prédiction brute du Seq2Seq à l'aide de techniques de programmation dynamique. Le résultat de l'analyse guidée par le Seq2Seq (C) peut être différent de la prédiction brute Seq2Seq, c'est pourquoi nous intégrons ici en plus la prédiction Seq2Seq.

— Application d'un algorithme de vote Rover [SG00] qui choisit le caractère le plus fréquent dans l'alignement.

La figure 6.4 illustre l'alignement et la fusion des stratégies de guidage présentées précédemment, avec l'ajout de la consigne et du résultat brut du Seq2Seq. Dans cet exemple, le résultat de la fusion correspond à la vérité de terrain. Ce qui permet d'obtenir un guidage efficace du système.

Mot manuscrit

Prédiction Seq2Seq	k	i	i	n	
Consigne dictée	b	i	e	n	
A - Analyse guidage consigne	b		i	n	
B - Analyse guidage phonétique	b		i	n	s
C – Analyse guidage Seq2Seq	k	i	i	n	
Fusion : alignement et Rover	b	i	i	n	

FIGURE 6.4 – Exemple de fusion pour la consigne "bien" et le mot écrit "biin".

Compétition

La fusion nous donne une alternative intéressante aux propositions des trois stratégies, néanmoins, selon les cas ce n'est pas toujours la meilleure proposition. Nous allons donc mettre en place une ultime étape de compétition pour optimiser le choix entre les stratégies. L'étape de compétition permet au système de choisir la meilleure stratégie de guidage parmi le guidage par la consigne (A), le guidage phonétique (B), la reconnaissance du Seq2Seq (C) et la fusion (D). La figure 6.5 illustre l'architecture générale du système avec une approche fusion et compétition. Le système commence par analyser le mot avec les stratégies de guidage par la consigne (A), le guidage phonétique (B) et la reconnaissance du Seq2Seq (C). Ensuite, le système analyse le mot avec le guidage provenant de la fusion (D) des résultats des stratégies de guidages précédent.

Pour évaluer chaque stratégie, nous allons calculer un **score d'édition** et un **score d'analyse**. Le score d'édition basé sur une distance d'édition Damerau-Leveinshtein (détaillé dans la partie 1.2.1), est calculé entre le mot de guidage et le mot représenté par le chemin optimal. Son rôle est de déterminer si le résultat d'une stratégie est proche de la connaissance utilisée pour le guidage. Le score d'analyse S_a d'un chemin P_n de longueur n évalue la confiance de l'analyse d'un mot par le système. Il est défini par :

$$S_a(P_n) = \sqrt{\Pi_{i=0}^{n} S_a(i)} \tag{6.1}$$

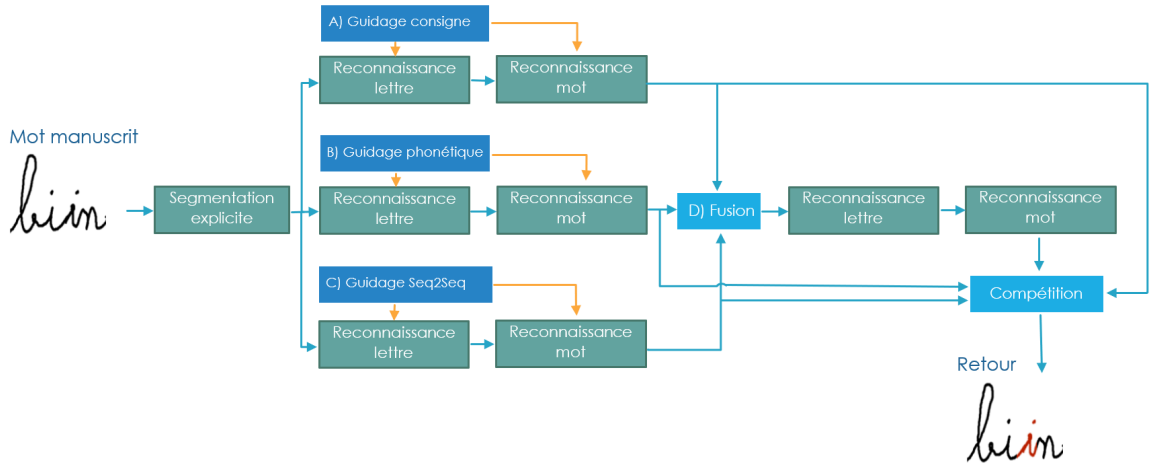

FIGURE 6.5 – Architecture du système avec une approche fusion et compétition.

où $S_a(i)$ est le score d'analyse de l'élément i du chemin.

Nous définissons un **score phonétique** qui combine le score d'édition et le score d'analyse S_a de la manière suivante :

$$ScorePhonétique(P_n) = S_a(P_n) * 0.7 + \frac{1}{1 + ScoreEdition(P_n)} * 0.3 \tag{6.2}$$

L'étape de compétition sélectionne le résultat de la stratégie de guidage ayant le meilleur score phonétique. Les coefficients 0.7 et 0.3 ont été choisis de manière empirique afin de donner plus de poids au score d'analyse pour chaque stratégie de guidage. Nous pouvons noter que cette approche effectue le calcul des chemins de segmentation pour chaque stratégie de guidage, ce qui augmente le temps d'analyse.

Dans cette partie, nous avons présenté l'intégration du résultat de reconnaissance du Seq2Seq dans le processus d'analyse. Le temps de calcul du système dépend de la longueur du mot à analyser. Plus un mot sera long, plus le treillis sera grand et plus il y aura de chemins à analyser. La partie suivante détaille une méthode d'optimisation qui utilise le résultat de segmentation du Seq2Seq pour élaguer le treillis de segmentation. L'objectif est de diminuer la complexité du processus et d'accélérer le temps de calcul.

6.2 Optimisation du temps de calcul : élagage du treillis de segmentation

La complexité du processus d'analyse dépend de la longueur du mot. Pour des mots longs, le processus d'analyse ne permet pas toujours de faire un retour immédiat à l'utilisateur. L'idée de cette approche est d'optimiser l'efficacité de l'analyse en utilisant la segmentation approximative du Seq2Seq pour élaguer le treillis de segmentation dans le but de diminuer le nombre d'hypothèses de lettre à considérer et de réduire ainsi le nombre de chemins à explorer. L'objectif est d'obtenir un bon compromis entre performance et la rapidité du temps de calcul. La figure 6.6 illustre la méthode d'élagage sur le mot "alors" qui se déroule en deux temps.

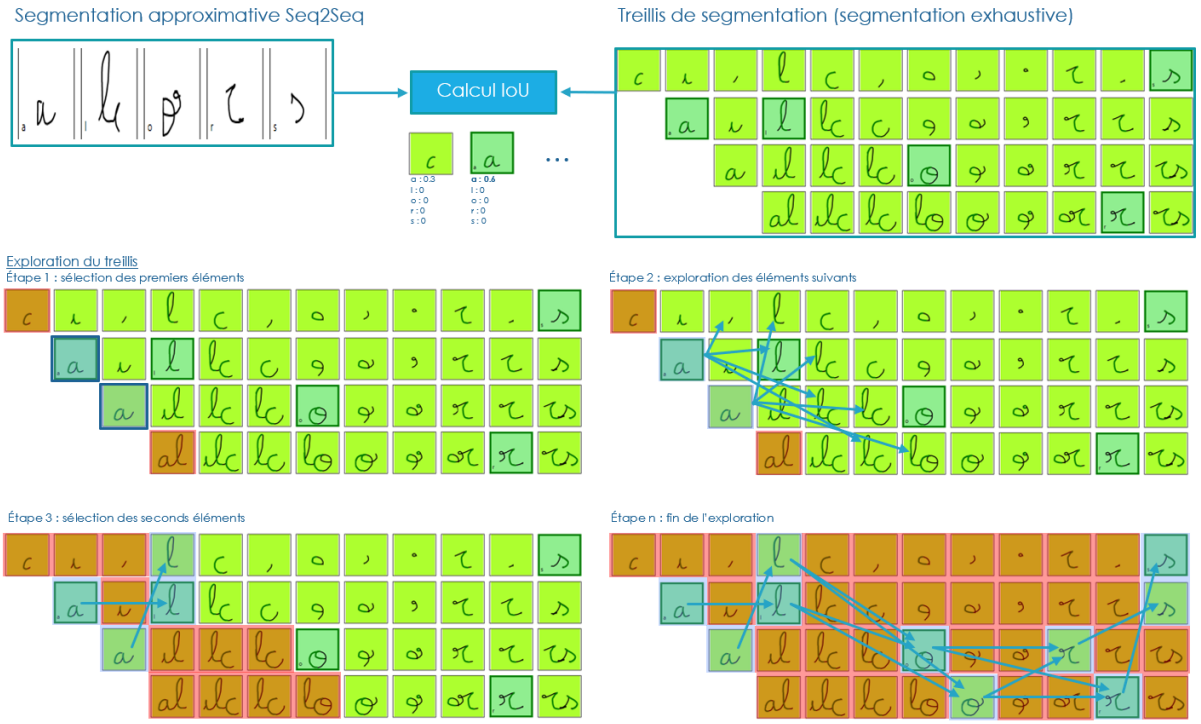

FIGURE 6.6 – Élagage du treillis de segmentation du mot "alors". En rouge les hypothèses élaguées et en bleu les chemins à explorer.

Dans un premier temps, nous calculons pour chaque hypothèse du treillis un score d'IoU (*cf.* partie 1.2.2) avec toutes les segmentations approximatives caractères du Seq2Seq. Pour chaque hypothèse de segmentation du treillis, nous gardons uniquement le caractère du SeqSeq le plus proche (score IoU le plus haut). Dans un second temps, l'élagage inter-

vient lors de l'exploration du treillis pour calculer les chemins. Lors de la première étape, la méthode d'exploration sélectionne le premier élément de chaque niveau du treillis. L'élagage consiste à garder uniquement les éléments du treillis pour lesquels la lettre du caractère du SeqSeq le plus proche fait partie des hypothèses d'analyse. La seconde étape consiste à explorer les prochains éléments des chemins et l'étape trois sélectionne les éléments à garder (élagage). L'exploration se poursuit jusqu'à ce que la méthode ait parcouru tout le treillis.

6.3 Retours orthographiques

Le degré de précision et de détails du retour visuel affiché à l'enfant dépend de la confiance du résultat l'analyse. Lorsque la confiance de l'analyse est faible, nous générons des retours plus génériques, *i.e.* un avertissement sur une zone d'incertitude, ou même aucun retour. La typologie du retour est illustrée dans la figure 6.7 et décomposée en trois niveaux différents :

- (1) Confiance élevée : lorsque le chemin du mot prédit correspond à la stratégie de guidage, le système donne un retour précis.
- (2) Confiance modérée : lorsqu'une lettre diffère entre le mot prédit et la stratégie de guidage, le système génère un avertissement sur une zone d'incertitude.
- (3) Rejet : lorsqu'aucune des conditions ci-dessus n'est remplie, aucun retour n'est généré dans le but d'éviter de faire des retours erronés à l'enfant.

L'alignement entre le mot prédit par le système et la consigne permet de mettre en évidence les lettres en erreurs.

Consigne	Confiance	Type d'erreur	Aperçu retour	Alignement consigne / prédiction
alors	Haute	Ajout, substitution		a | | | o | r | s a | l | | o | r | e
alors	Modérée	Substitution, doute reconnaissance		a | l | o | r | s a | l | o | r | d
bonjour	Rejet			b | o | n | j | o | u | r l | e | z | | o | u | r

FIGURE 6.7 – Exemple de retours fait à l'utilisateur avec un niveau de confiance du système haut, modéré et rejet

6.4 Expérimentations

6.4.1 Protocole expérimental

Jeu de données

Ce travail nécessite des données **mots annotés au niveau des caractères** pour évaluer le système sur l'aspect de segmentation. Pour nos expériences, nous utilisons le jeu de données privé ICDB-Words-B, composé de mots en cursif français écrits par des enfants présenté dans la partie 1.4.2. Chaque mot est une séquence de points représentés par leurs coordonnées (x et y), leur pression et leur temps. Le signal en ligne est converti en images qui sont redimensionnées à une hauteur de 128 pixels. Le modèle IntuiScript utilise le signal en ligne tandis que le modèle Seq2Seq utilise directement les images. Notre ensemble de données est divisé en 6 812 mots écrits par plus de 500 enfants pour l'ensemble d'entrainement et 1 242 mots écrits par plus de 300 enfants pour l'ensemble de test. Les ensembles de données d'entrainement et de test proviennent de différentes campagnes d'acquisition de données (et de différentes salles de classe). Il n'y a pas de données d'enfants présentes à la fois dans l'ensemble d'entrainement et dans l'ensemble de test, ce qui nous permet de vérifier la capacité du système à généraliser sur des styles d'écriture non présents dans le jeu d'entrainement.

Entrainement et évaluation

Nous utilisons une **validation croisée** avec k=10 blocs sur le jeu d'entrainement pour évaluer la robustesse du système. Lors de la validation croisée, le jeu d'entrainement est composé de 8 blocs, le jeu de validation de 1 bloc et le jeu de test de 1 bloc. Nous entrainons et évaluons le système avec chaque permutation pour le jeu de test. La reconnaissance est évaluée avec le taux de reconnaissance mots et la segmentation avec l'IoU. Nous utilisons le réseau Seq2Seq présenté dans la partie 3.2.3. Le nombre de mots écrit par des enfants étant limité, nous pré-entrainons le réseau avec des écritures adultes de la base IAM-OnDB [LB05]. Ensuite, le réseau est affiné pendant 200 époques avec un taux d'apprentissage de 0,001 avec l'optimiseur Adam.

Cette étude expérimentale commence par évaluer les performances de reconnaissance et de segmentation du modèle Seq2Seq. Elle présente ensuite les résultats des stratégies d'hybridation entre le modèle Seq2Seq et IntuiScript. Après, elle mesure l'impact de l'utilisation de l'élagage du treillis de segmentation. Pour finir, elle présente une analyse des

retours faits par le système.

6.4.2 Évaluation de la reconnaissance et de la segmentation

Le tableau 6.1 regroupe les résultats de reconnaissance et de segmentation pour le modèle Seq2Seq. Le taux de reconnaissance est meilleur dans le jeu de test de la validation croisée parce que les données représentent des mots écrit par des enfants ont servi à l'entrainement (même scripteurs). Les résultats montrent que le réseau Seq2Seq est performant en reconnaissance, mais ne permet pas d'obtenir une segmentation précise pour faire un retour à l'utilisateur. Nous utilisons la prédiction de l'encodeur (texte et segmentation) pour les prochaines expériences, car son taux de reconnaissance sur le jeu de test de la validation croisée est meilleur.

TABLE 6.1 – Moyenne et écart type pour l'évaluation de la reconnaissance et de la segmentation sur des mots des enfants. Le taux de reconnaissance est évalué sur l'ensemble de test de la validation croisée et sur l'ensemble de test complet. Le taux de segmentation est évalué sur l'ensemble de test complet. L'encodeur et le décodeur de Seq2Seq sont évalués en %. Une valeur plus haute est meilleure.

Modèle	Taux de reconnaissance mots Validation croisée (%)	Taux reconnaissance mots Test (%)	Taux de segmentation lettres IoU (%)
Seq2Seq : Encodeur	**86.65** ± 1.17	**75.08** ± 1.16	**51.14** ± 7.06
Seq2Seq : Décodeur	86.32 ± 1.25	69.20 ± 2.14	45.91 ± 3.19

Le tableau 6.2 détaille les résultats de reconnaissance et de segmentation pour les différentes stratégies développées dans ce chapitre. Le modèle Seq2Seq et les stratégies d'hybridation sont testés sur les dix modèles générés de la validation croisée. Les résultats des moyennes et des écart-types sont présentés pour ces approches. Les trois premières lignes du tableau correspondent à des approches utilisant un seul modèle tandis que les deux dernières correspondent aux stratégies d'hybridation. La performance en reconnaissance du Seq2Seq surpasse l'approche de guidage par la consigne ou par le guidage phonétique, mais avec une segmentation très approximative (IoU de 51,14%). Lorsque nous intégrons les résultats de la reconnaissance du Seq2Seq dans le moteur IntuiScript, le taux de reconnaissance est moins haut que le Seq2Seq et la segmentation s'améliore avec un IoU de 90,4%. Cette première stratégie d'hybridation surpasse la stratégie de guidage par la consigne et la stratégie de guidage phonétique. La différence du score de reconnaissance s'explique par le fait que le résultat du Seq2Seq est utilisé comme guidage et non comme résultat de reconnaissance définitif. La stratégie de fusion et de combinaison des guidages

est encore meilleure en reconnaissance (+11.10 % taux de reconnaissance) et en segmentation (+2.42 % d'IoU) que l'approche de guidage par le Seq2Seq. Cela démontre l'intérêt d'utiliser les différents type de guidage (consigne, phonétique, Seq2Seq et fusion) pour guider l'analyse du système.

TABLE 6.2 – Évaluation de la reconnaissance et de la segmentation de différentes stratégies de guidage. Une valeur plus haute est meilleure.

Stratégie	Taux de reconnaissance mots (%)	Taux de segmentation lettres (IoU) (%)
Encodeur Seq2Seq	75.08 ± 1.16	51.14 ± 7.06
IntuiScript : guidage par la consigne	64.09	88.66
IntuiScript : guidage phonétique	66.42	88.72
IntuiScript : guidage Seq2Seq	72.18 ± 0.73	90.40 ± 0.54
IntuiScript : fusion et compétition	**83.28** ± 0.51	**92.82** ± 0.28

6.4.3 Évaluation de la stratégie d'élagage

Cette partie évalue le temps d'analyse des stratégies d'hybridation avec et sans élagage sur un ordinateur avec un CPU Intel Core i7-8665U. Le modèle Seq2Seq prédit le mot contenu dans une image en 73 millisecondes en moyenne. Ce calcul étant très rapide, il n'est pas inclus dans les calculs de temps suivant. Le tableau 6.3 présente les performances de reconnaissance et de segmentation des stratégies d'hybridation proposées, ainsi que leur temps moyen d'analyse par mot. Comme nous l'avons vu, la stratégie de fusion et de compétition fournit les meilleurs résultats de reconnaissance et de segmentation, mais le temps d'analyse (4,74 s par mot) est 3 fois plus important que la stratégie de guidage par prédiction du Seq2Seq. L'utilisation de l'élagage permet de réduire le temps d'analyse de la stratégie de fusion à un niveau acceptable de 0,67 s en moyenne, tout en perdant environ 2 % des performances de reconnaissance (79,87 %) et 1 % de la précision de la segmentation (91.44 %). Cette baisse s'explique par la nature approximative de la segmentation du Seq2Seq. Il en va de même pour l'élagage avec la stratégie de guidage par le Seq2Seq. Nous pouvons donc conclure que l'élagage constitue un bon compromis entre temps d'analyse et performance.

TABLE 6.3 – Évaluation de la reconnaissance, de la segmentation et du temps de calcul des stratégies d'hybridation avec et sans élagage. Un taux de reconnaissance et de segmentation plus haut est meilleur. Un temps de calcul plus faible est meilleur.

Stratégie	Élagage	Taux de reconnaissance (%)	Taux de segmentation (IoU) (%)	Temps de calcul moyen (s)
Guidage Seq2Seq	Non	72.18 ± 0.73	90.40 ± 0.54	1.34
Fusion et compétition	Non	83.28 ± 0.51	92.82 ± 0.28	4.74
Guidage Seq2Seq	Oui	70.87 ± 2.30	89.30 ± 1.12	**0.37**
Fusion et compétition	Oui	79.87 ± 0.94	91.44 ± 0.98	**0.67**

6.4.4 Évaluation de la qualité des retours

Cette partie présente les résultats de la méthode d'hybridation de fusion et compétition avec élagage dans un contexte applicatif. Le tableau 6.4 présente les résultats de l'analyse en fonction du niveau de confiance du système . D'une part, le système a un degré de confiance élevé pour 88,88 % des mots sur l'ensemble de test ce qui permet de faire un retour précis. D'autre part, le système a un faible degré de retour de confiance modérée et de rejet (4,5 et 6,7 % respectivement). L'utilisation du rejet permet d'améliorer le taux de reconnaissance mot du système de 5.5% et ainsi de délivrer un retour plus précis à l'enfant.

TABLE 6.4 – Évaluation de la pertinence des retours générés sur le jeu de tests pour la méthode d'hybridation de fusion et compétition avec élagage.

Confiance	Nb de mots	% de mots	Taux de reconnaissance mot (%)
Haute	1 103.9 ± 11	88.8	-
Modérée	55.4 ± 4.4	4.5	-
Rejet	82.7 ± 9.48	6.7	-
Haute, Modérée, Rejet	1 242	100	79.8
Haute, Modérée	1 146	93.3	**85.3**

6.5 Bilan

Dans ce chapitre, nous avons présenté une méthode de reconnaissance et segmentation en lettres, de mots écrits par des enfants, dans un contexte de dictée. Elle permet de faire une analyse fine de l'écriture pour faire des retours orthographiques immédiats sur les différentes erreurs qu'a pu commettre l'enfant. Cette méthode s'appuie sur une hybridation

d'un modèle Seq2Seq qui est performant en reconnaissance, mais approximatif en segmentation, et d'un modèle IntuiScript qui est performant en segmentation, mais qui a besoin de guidage pour faire une analyse précise du mot. L'utilisation du modèle Seq2Seq permet d'injecter une nouvelle source de guidage dans le modèle IntuiScript ainsi que d'optimiser le temps de calcul à l'aide d'une stratégie d'élagage. Les expérimentations ont démontré l'intérêt de la méthode dans un contexte éducatif d'apprentissage de l'écriture dans un exercice de dictée. La méthode intègre un système de rejet basé sur la comparaison du texte prédit et les différents mots de guidages utilisés par le système permettant ainsi de détecter les mots où le système a un degré de confiance faible dans le but d'éviter de faire des retours erronés à l'enfant. Ces expérimentations ont donné lieu à une publication dans le journal IJDAR [4].

Le bon fonctionnement de cette approche repose sur la qualité des connaissances injectées dans le guidage, ce qui peut être une limite pour l'adaptation de la méthode pour une autre langue où le guidage phonétique doit être mis à jour. De plus, l'utilisation de guidages augmente le temps d'analyse, ce qui ne permet pas toujours de faire un retour immédiat à l'enfant pour des mots longs. La stratégie d'élagage permet de résoudre cette limitation tout en dégradant légèrement les performances du système. Le chapitre suivant présente une nouvelle approche de reconnaissance et segmentation d'écriture manuscrite qui n'utilise pas de modèle de langue. Elle repose sur la combinaison d'un modèle Seq2Seq dédié à la reconnaissance et d'un modèle R-CNN dédié à la segmentation. Le premier objectif est de proposer un système sans guidage, rapide et au moins aussi performant en reconnaissance et segmentation. Le second objectif est d'étendre la notion de rejet afin d'augmenter la précision des retours fait à l'enfant.

Chapitre 7

Seq2Seg : Combinaison des modèles Seq2Seq et R-CNN

Ce chapitre présente une nouvelle approche que nous appelons **Seq2Seg**. C'est une combinaison d'un modèle Seq2Seq et R-CNN pour segmenter et reconnaître avec précision des mots manuscrits écrits par des enfants. Dans notre travail, nous utilisons l'architecture Seq2Seq définie dans la partie 3.2.3 comme **modèle de reconnaissance de texte** et l'architecture R-CNN définie dans [He+17] et présentée dans la partie 3.3 comme **détecteur d'objets**. Comme nous l'avons vu précédemment, le Seq2Seq est performant pour une tâche de reconnaissance, mais ne permet pas d'obtenir une segmentation lettre précise. Au contraire, le modèle R-CNN est performant pour une tâche de segmentation, mais est moins précis en reconnaissance que le Seq2Seq (*cf.* partie 7.4.2 pour les résultats détaillés). L'idée est d'exploiter les atouts de chaque modèle pour fournir une segmentation précise des mots dans le but de faire un retour immédiat à l'enfant sur la qualité orthographique. L'architecture générale de la méthode est illustrée dans la figure 7.1. La première contribution (niveau A) utilise un modèle dédié à la tâche de reconnaissance (Seq2Seq) en tant qu'oracle pour guider le filtrage des boîtes englobantes prédites par le détecteur d'objets (R-CNN). La seconde contribution (niveau B) utilise un treillis de segmentation expert [AL97] [Sim+19] pour affiner la segmentation des lettres associées aux boîtes englobantes prédites par le détecteur d'objets et permet d'obtenir une segmentation sémantique du mot.

Ce chapitre présente les travaux en lien avec la quatrième contribution publiés à la conférence ICDAR [3]. Il commence par spécifier le fonctionnement du filtrage guidé par le Seq2Seq (niveau A) puis détaille l'utilisation du treillis de segmentation dans le but d'améliorer la précision de segmentation (niveau B). Ensuite, nous proposons une méthode de rejet pour améliorer la qualité du retour fait à l'élève. Enfin, une étude expérimentale présente les résultats quantitatifs, qualitatifs ainsi que l'impact du rejet sur l'approche Seq2Seg.

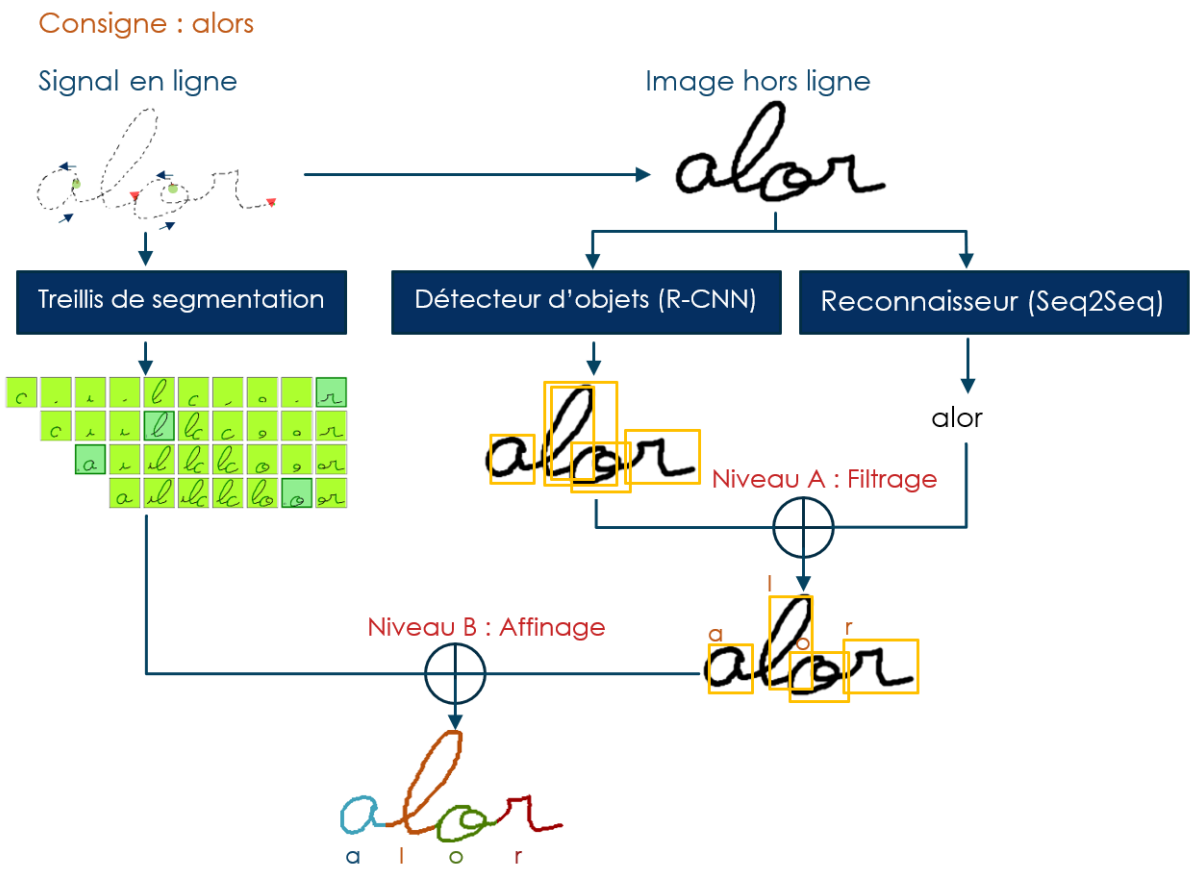

FIGURE 7.1 – Architecture générale de la méthode Seq2Seg. Niveau A : filtrage des boites englobantes guidé par le reconnaisseur. Niveau B : affinage de la segmentation à l'aide d'un treillis de segmentation.

7.1 Niveau A : filtrage des boîtes englobantes à l'aide d'un modèle de reconnaissance

Le modèle de reconnaissance Seq2Seq est entrainé uniquement pour une tâche de reconnaissance et prédit le mot contenu dans l'image. De ce mot, nous pouvons déduire notamment le nombre potentiel de lettres à segmenter. Ces informations permettent de sélectionner un nombre déterminé de prédictions de segmentations du détecteur d'objets **pendant l'inférence** et d'utiliser le résultat de reconnaissance du modèle de reconnaissance **plus précis** que celui du détecteur d'objets. Le processus de sélection des prédictions du détecteur d'objets peut être ambigu dans certains cas, comme illustré sur la figure 7.2. Dans l'exemple 1, une lettre « m » et deux lettres « n » peuvent être reconnues, ce qui

peut être considéré comme vrai dans un contexte local. Ici, une vue plus globale est nécessaire pour choisir la bonne segmentation. L'utilisation du modèle de reconnaissance peut permettre de lever ces ambiguïtés.

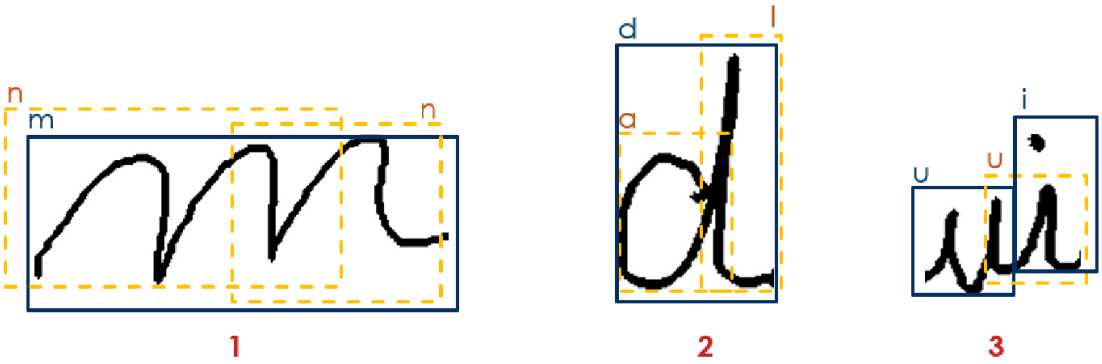

FIGURE 7.2 – Exemple d'ambiguïtés dans les prédictions du détecteur d'objets (R-CNN) : la prédiction correcte est la boite englobante en ligne continue, la prédiction incorrecte en pointillée et l'étiquette de la classe associée à chaque prédiction au-dessus.

Le détecteur d'objets fait plusieurs prédictions de boites englobantes qui peuvent être identifiées individuellement par la coordonnées en x du coin inférieur gauche de chaque boite. Le but est de sélectionner la prédiction du détecteur d'objets correspondant au mieux à la "vrai" segmentation de la lettre. Cette méthode illustrée dans la figure 7.3 se décompose en trois étapes :

— Étape 1 : **calcule de toutes les séquences (*i.e.* les listes de boites englobantes ordonnées sur l'axe des x) de prédiction du détecteur d'objets** ;
— Étape 2 : **filtrage des séquences en fonction de la longueur du mot prédit par le modèle de reconnaissance** ;
— Étape 3 : **calcule du score associé à chaque séquence**. La séquence finale sélectionnée est celle avec le score le plus élevé.

La détection de lettres dans un mot cursif est différente de la détection d'objets classiques comme présents dans le jeu de données COCO [Lin+14]. Les lettres ne partagent jamais de partie commune sauf au niveau du tracé reliant deux lettres *i.e.* sur la ligature. Il y a donc peu de chevauchement entre les objets quand ceux-ci sont des lettres.

Le modèle R-CNN inclut nativement deux phases de suppression non maximale (NMS) pour filtrer ses prédictions détaillées en section 3.3.5. La première est appliquée aux **propositions de régions** (coordonnées de la région et score "est un objet"), tandis que la

seconde est appliquée aux **prédictions** (boîtes englobantes et étiquettes) ayant la **même étiquette**. Afin de gérer les cas où plusieurs prédictions de lettres sont imbriquées comme illustré sur la figure 7.2, nous avons ajouté un NMS sur les prédictions du modèle qui est **indépendant de l'étiquette** afin de sélectionner une prédiction par lettre. Notre méthode utilise les prédictions **avant** ce dernier NMS pour avoir une plus grande variété de prédictions à filtrer avec des ambiguïtés de segmentation. Le but de la méthode est de supprimer ces ambiguïtés et ainsi de choisir la bonne segmentation lettre.

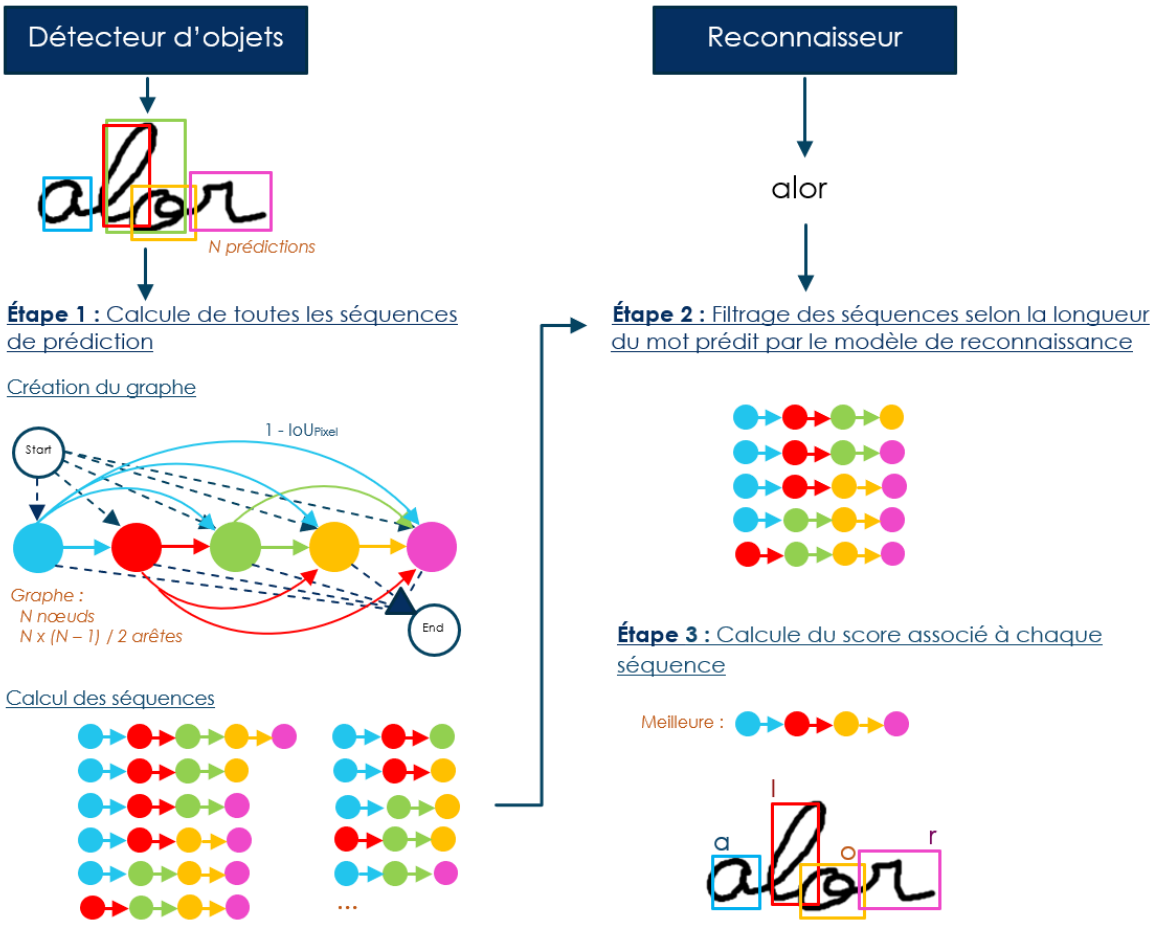

FIGURE 7.3 – Niveau A : exemple des trois étapes du processus de filtrage des prédictions du détecteur d'objets guidé par le résultat d'un modèle de reconnaissance.

7.1.1 Étape 1 : calcule de toutes les séquences de prédiction du détecteur d'objets

Nous considérons un graphe orienté $G(V, E)$, où V et E correspondent aux ensembles de sommets et d'arêtes. Pour chaque prédiction du détecteur d'objets ordonnée par x_{min} à partir des coordonnées de la boîte englobante, un sommet est ajouté dans G comme illustré à la figure 7.3. Le poids d'une arête $e_{ij} = (v_i, v_j) \in E$ est calculé comme $e_{ij} = 1 - \text{IoU}_{Pixel}$ entre les prédictions ordonnées par x_{min} associées aux sommets. IoU_{Pixel} représente l'intersection sur l'union des pixels d'écriture manuscrite contenus dans les deux boîtes englobantes correspondant aux deux sommets. Les prédictions avec un chevauchement représentées par l'IoU_{Pixel} plus élevé ont un lien plus faible. Une séquence de prédictions ordonnées par x_{min} correspond à un chemin du graphe, *i.e.* une liste de sommets connectés dans le graphe.

7.1.2 Étape 2 : filtrage des séquences en fonction de la longueur du mot prédit par le modèle de reconnaissance

Il existe trois scénarios de correspondance entre les prédictions du détecteur d'objets et du texte prédit par le reconnaisseur (le tableau 7.3 détaille le résultat de chaque type de scénario) :

- **Correspondance parfaite** : le nombre de lettres prédites par le détecteur d'objets et par le modèle de reconnaissance est toujours égal. Dans ce cas, nous nous attendons à ce que notre filtrage n'améliore que la partie reconnaissance du détecteur d'objets (que nous n'utilisons pas explicitement).
- **Correspondance** : le nombre de lettres prédites par le détecteur d'objets et par le modèle de reconnaissance peut-être différent, mais il existe au moins une séquence de prédiction du détecteur d'objets dans le graphe $G(V, E)$ de longueur égale au nombre de lettres prédites par le modèle de reconnaissance. Dans ce cas, nous nous attendons à ce que l'utilisation du mot prédit par le reconnaisseur comme oracle aide à lever certaines ambiguïtés pour le détecteur d'objets. Cela peut améliorer à la fois la reconnaissance et la segmentation.
- **Pas de correspondance** : le nombre de lettres prédites par le détecteur d'objets et par le modèle de reconnaissance est différent et il n'y a pas de correspondance possible. Dans ce cas, à la fois la reconnaissance et la segmentation du détecteur d'objets sont utilisées (le reconnaisseur est ignoré). En pratique, dans ce cas, nous

avons remarqué que le reconnaisseur prédisait une lettre en plus ou en moins. Il est donc important que le détecteur d'objets puisse ignorer la prédiction de l'oracle lorsqu'il y a un fort conflit entre les deux modèles. Ce filtrage pourrait ainsi améliorer les résultats globaux de reconnaissance puisque le détecteur d'objets prendra le relais du reconnaisseur, mais uniquement pour les prédictions les plus difficiles.

7.1.3 Étape 3 : calcule du score associé à chaque séquence

Le score d'une séquence de taille N_a prend en compte un critère de **chevauchement** inter-lettres et un critère de **couverture** de l'écriture. Le score de chevauchement, $s_{overlap}$, est le produit de tous les poids des arêtes *weight* v dans le chemin du graphe $G(V, E)$ correspondant à une séquence :

$$s_{overlap} = \Pi_{i=1}^{N_a} weight\ v_i \tag{7.1}$$

Plus le chevauchement est important, plus le score est bas. Pour calculer le score de couverture s_{cover}, nous additionnons le nombre de pixels N_p contenus dans chaque prédiction *pred* et nous soustrayions le nombre de pixels contenus à l'intersection *inter* des deux prédictions du nombre de pixels contenus dans chaque prédiction afin de compter chaque pixel une seule fois. Ensuite, le nombre de pixels prédit est divisé par le nombre total de pixels comme suit :

$$s_{cover} = (\Sigma_{i=1}^{N_a} N_p\ pred_i - \Sigma_{i=2}^{N_a} N_p\ inter(pred_{i-1}, pred_i))/N_p\ total \tag{7.2}$$

Le score d'alignement final $s_{alignment}$ est défini comme suit :

$$s_{alignement} = s_{overlap} + s_{cover} \tag{7.3}$$

La séquence avec le score $s_{alignment}$ le plus élevé sert à calculer la sortie de la méthode Seq2Seg.

Le résultat de **segmentation sémantique** correspond aux pixels contenus dans chaque boite englobante de la séquence (calculé à l'aide du **signal en ligne**). Le résultat de **reconnaissance** correspond à la prédiction du reconnaisseur s'il y a correspondance, sinon aux prédictions lettres associées à chaque boite englobante de la séquence.

Nous pouvons noter que l'efficacité de cette méthode en termes de temps de calcul

dépend de la taille des graphes générés. Dans notre contexte d'écriture manuscrite d'enfants, les graphes associés aux mots sont petits, car la longueur des mots est inférieure à 10 lettres. Le temps de calcul de cette méthode est donc suffisamment faible pour fournir un retour immédiat à l'enfant.

7.2 Niveau B : amélioration de la segmentation du R-CNN à l'aide d'un treillis de segmentation

L'écriture manuscrite **en ligne** peut être découpée précisément en différentes hypothèses de segmentation regroupées dans un treillis de segmentation [AL97] [Sim+19]. Cette méthode, présentée dans la partie 2.2.1, n'a pas besoin de reconnaissance des lettres pour produire un ensemble organisé d'hypothèses plausibles de segmentation. De plus, le signal en ligne qui modélise la trajectoire permet d'extraire de l'image une segmentation sémantique automatique pour chaque hypothèse en séparant les pixels de l'arrière-plan et de l'écriture manuscrite. Notre objectif est d'utiliser ce treillis pour trouver les hypothèses du tracé en ligne "les plus proches" du treillis de segmentation associées aux boîtes englobantes (image) prédites par le détecteur d'objets comme illustré sur la figure 7.4.

Pour chaque boite englobante du R-CNN, la méthode sélectionne l'hypothèse du treillis la plus proche (étape 1). L'IoU$_{Pixel}$, *i.e.* une intersection sur l'union entre les pixels d'écriture manuscrite contenus dans une boîte englobante (facilement accessible comme expliqué précédemment) et ceux d'une reconstruction d'une image à partir du tracé en ligne, est utilisé comme critère pour sélectionner l'hypothèse la plus proche. Ensuite, les coordonnées de la boite englobante prédites par le R-CNN sont mises à jour avec celles de la boite englobante de l'hypothèse du treillis sélectionné ce qui permet d'ajuster les coordonnées (étape 2). Le treillis étant construit à partir du signal en ligne, la segmentation sémantique associée aux hypothèses est précise et permet de segmenter correctement les lettres obliques (*cf.* "l" et "o" de la figure 7.4) ce qui n'est pas le cas pour la segmentation sémantique associée aux boites englobantes prédit par le R-CNN.

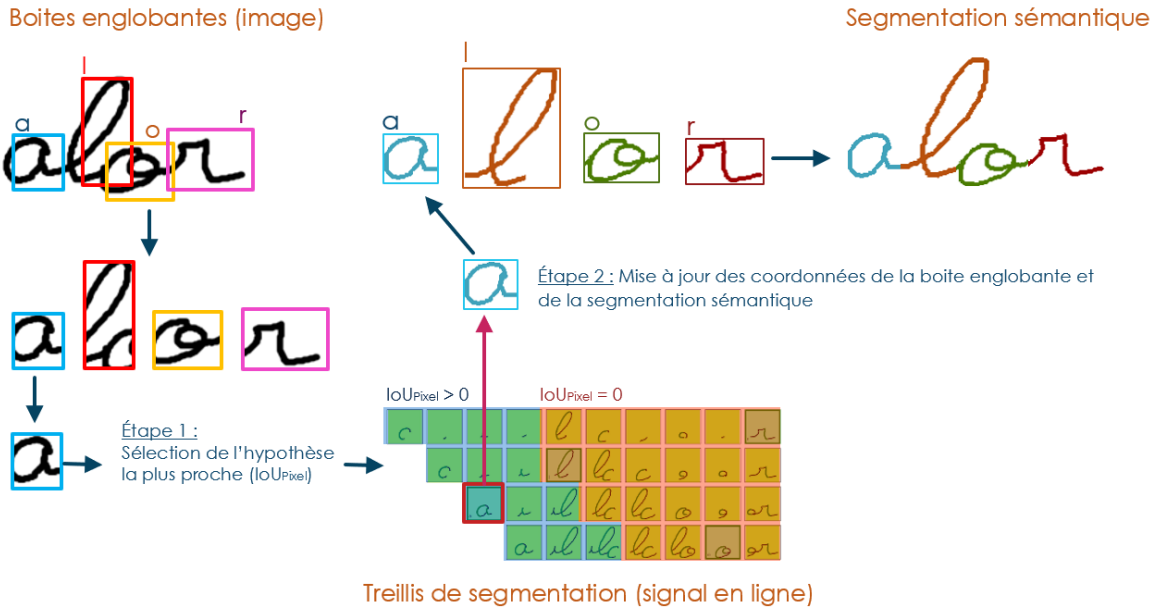

FIGURE 7.4 – Exemple d'amélioration des boîtes englobantes avec le treillis de segmentation en ligne. IoU_{Pixel} est utilisé comme critère pour sélectionner la meilleure hypothèse de réseau.

7.3 Perspective : améliorer la robustesse

L'objectif de la méthode Seq2Seg est de faire des retours de nature orthographique. Cependant, la méthode commet des erreurs de reconnaissance et de segmentation. Dans un contexte d'apprentissage, il est préférable de demander à l'élève de réécrire le mot quand la confiance de la méthode est faible plutôt que de l'induire en erreur. Cette section présente des travaux préliminaires cherchant à détecter si la prédiction de la méthode Seq2Seg contient des erreurs. Nous différencions deux types d'erreurs :

— Une **erreur de segmentation** où la segmentation lettres d'un mot par la méthode Seq2Seg n'est pas assez précise. Cette erreur induit un mauvais positionnement des caractères reconnus et donc un retour de mauvaise qualité.

— Une **erreur de reconnaissance** où le mot reconnu contient des erreurs de reconnaissance. Cette erreur est très grave, car le système peut indiquer des corrections fausses à l'élève ou même indiquer en erreur un mot correctement écrit.

Dans cette section, nous proposons une stratégie de rejets basée sur des critères de segmentation représentés par un score de couverture, un score de chevauchement et un critère de reconnaissance se basant sur une distance d'édition entre le mot reconnu par le

R-CNN et le Seq2Seq. Ces scores sont exprimés de la manière suivante :

— Le **score de couverture** représente le ratio entre le nombre de pixels prédits et le nombre total de pixels de l'écriture. L'objectif est de faire un retour visuel sur tout le tracé de l'enfant.

— Le **score de chevauchement** calcul le nombre de pixels commun à plusieurs prédictions. L'objectif est de ne pas attribuer le tracé à plusieurs lettres, sauf pour le cas des ligatures inter-lettres.

— La **distance d'édition** calcul le nombre de caractères différents entre le mot prédit par le Seq2Seq et par le R-CNN. Si la prédiction du caractère est différente entre les deux modèle et que les caractères prédit sont de même nature le coût et de un, sinon il est de deux. Nous entendons par nature de lettres, les lettres qui ont des dimensions similaires, *i.e.* avec ou sans hampe, jambe comme illustrées sur la figure 7.5. L'objectif de la distance d'édition et de vérifier qu'il y a une certaine cohérence entre les deux résultats de reconnaissance.

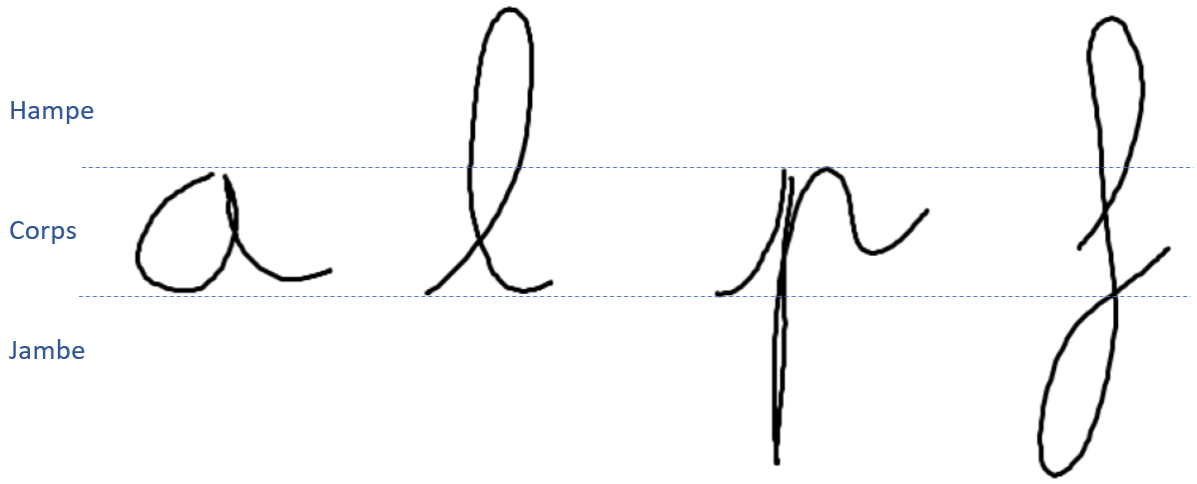

FIGURE 7.5 – Exemple de corps de lettres avec/sans hampe et jambe.

Notre stratégie utilise les scores énoncés présentés précédemment de façon indépendante. La prédiction du Seq2Seg est rejetée si au moins un des scores est inférieur ou supérieur à un des **seuils fixes**.

7.4 Expérimentations

7.4.1 Protocole

Jeu de données

Les mots manuscrits sont représentés sous la forme d'un **signal en ligne** encodé par une séquence de points représentés par deux coordonnées (x, y), une pression et un horodatage. Le signal en ligne est utilisé à la fois pour calculer le treillis de segmentation et est converti en une image avec une épaisseur d'écriture de 2 pixels. Le tableau 7.1 détaille les **jeux de données d'écriture d'enfants** utilisés pour entrainer/tester les modèles d'apprentissage profond (qui sont tous des variantes des mots enfants acquis). Le jeu de données original ICDB-Words-C (*cf.* section 1.4.2) est composé de 8 054 mots cursifs manuscrits français annotés au **niveau du mot** qui sont utiles pour entraîner le réseau Seq2Seq. Dans le jeu ICDB-Words-D, une sous-partie du jeu ICDB-Words-C, 2 126 mots sont annotés (*i.e.* segmentés) au **niveau des lettres** pour entrainer le détecteur d'objets R-CNN. Le nombre de données annotées lettres étant limité pour l'apprentissage du détecteur d'objets, nous avons déplacé 210 mots du jeu de test dans le jeu d'entrainement par rapport au découpage utilisé dans le chapitre précédent. Les enfants sont différents pour le jeu d'entrainement et le jeu de tests et le jeu de test est le même pour tous les modèles.

Table 7.1 – Détails des données d'écriture d'enfants utilisées pour l'entrainement et le test des modèles d'apprentissage profond.

Modèles	Type d'annotation	Entrainement	Validation	Test	Total
Seq2Seq	Mots	6 022	1 000	1 032	8 054
R-CNN	Lettres	918	176	1 032	2 126
R-CNN avec synthèse	Lettres	27 540	176	1 032	28 748

Pour mieux entrainer le modèle R-CNN, nous effectuons une synthèse de données uniquement sur l'ensemble d'apprentissage (appelé "avec synthèse" dans le tableau). Parmi une liste de déformations hors ligne usuelles (étirement, inclinaison) et plus récentes (étirement du trait, courbure) détaillées dans la partie 1.5, chaque mot est augmenté 30 fois avec des paramètres aléatoires.

Entrainement

Le modèle Seq2Seq suit la même architecture et le même protocole d'apprentissage que pour le chapitre précédent. Le modèle est pré-entraîné sur un ensemble de données d'écriture manuscrite d'adultes en anglais IAM-OnDB [LB05], puis entrainé sur l'ensemble de données d'écriture manuscrite d'enfants ICDB-Words-C. Le modèle est entrainé pendant 200 époques avec un batch de 16. L'optimiseur Adam est utilisé avec un taux d'apprentissage de 0,001. Comme le jeu de test est différent, nous réévaluons la méthode décrite dans le chapitre précédent sur ce jeu de données. Le R-CNN avec un réseau d'extraction de caractéristiques ResNet-FPN est entrainé pendant 60 époques avec une taille de lot de 4. L'optimiseur AdamW [LH19] est utilisé avec un taux d'apprentissage de 0,0001. Les paramètres R-CNN sont indiqués dans l'article original [He+17] sauf que nous ignorons la branche Mask (cf. 3.3.4). Nous utilisons le critère Complete Intersection Over Union (CIoU) [WS21] pour faire correspondre les vérités terrain et les prédictions pendant la phase d'entraînement. Nous ajoutons également un filtrage NMS indépendant de la classe sur les sorties pour gérer les prédictions imbriquées comme expliqué en section 7.1. Nous avons défini tous les paramètres du modèle R-CNN sur l'ensemble de validation à l'aide du score de performance Mean Average Precision (MAP) avant d'évaluer le meilleur modèle sur l'ensemble de tests.

Evaluation

Pour évaluer les performances de notre approche Seq2Seg, nous utilisons le taux d'erreur de caractères (CER) et le taux d'erreur de mots (WER) pour les performances de reconnaissance. Nous utilisons l'Intersection Over Union (IoU) et l'IoU au niveau du pixel pour évaluer les performances de segmentation. L'IoU_{Pixel} se concentre sur les lignes d'écriture manuscrite et ignore l'arrière-plan (principalement blanc).

7.4.2 Résultats quantitatifs

Nous réalisons d'abord une étude d'ablation pour mesurer l'impact des différentes contributions ainsi que le choix d'architecture pour l'extracteur de caractéristiques dans le détecteur d'objets. Ensuite, nous comparons l'approche Seq2Seg avec l'approche présentée dans le chapitre précédent. Toutes les expériences sont évaluées en termes de reconnaissance, de segmentation et de vitesse de calcul sur l'ensemble de test. Alors que les réseaux sont entrainés sur GPU, le temps de traitement est calculé sur un ordinateur

portable avec un processeur Intel Core i7-8665U. En effet, les applications éducatives sont exécutées sur une tablette avec stylet, où une connexion Internet n'est pas toujours disponible. Par conséquent, l'analyse temporelle est plus pertinente sur un ordinateur portable équipé d'un processeur. Nous considérons comme acceptable un temps d'analyse inférieur à 2 secondes pour fournir un retour immédiat aux enfants.

Le tableau 7.2 montre les résultats de l'étude d'ablation, où le niveau A correspondant à la méthode de filtrage des prédictions du détecteur d'objets par le résultat du modèle de reconnaissance présenté dans la section 7.1 et le niveau B correspondant au raffinement des coordonnées des boîtes englobantes du détecteur d'objets à l'aide d'un treillis de segmentation présenté dans la section 7.2. Nous notons Seq2Seq comme le résultat de l'encodeur et n'utilisons que le résultat de l'encodeur dans ce travail comme pour le chapitre précédent. Nous pouvons voir dans le tableau que le choix d'un extracteur plus profond dans le détecteur d'objets (R-CNN) améliore les performances en reconnaissance et en segmentation (-1,3% de CER et +1,9% d'IoU$_{Pixel}$ de ResNet 18 à ResNet 101). Par contre, le temps de calcul augmente de plus de 2 secondes. Le modèle Seq2Seq reste beaucoup plus précis en reconnaissance (CER/WER) que toutes les versions du R-CNN. Le choix de l'extracteur n'a pas eu d'impact significatif sur nos contributions (voir partie inférieure du tableau). Nous avons choisi l'extracteur ResNet-34 FPN pour les prochaines expériences en raison de sa vitesse et de ses performances légèrement meilleures en reconnaissance.

Niveau A : le filtrage des prédictions du détecteur d'objets avec le texte reconnu par le Seq2Seq permet d'obtenir des résultats légèrement meilleurs en reconnaissance (CER de 5,0%) que le Seq2Seq seul (CER de 5,3%). Les raisons de cette amélioration sont données dans le cas « pas de correspondance » de la deuxième étape de la première contribution présentée à la section 7.1. De plus, cette méthode sélectionne les boîtes englobantes pour maximiser la couverture et minimiser le chevauchement de l'écriture manuscrite et améliore ainsi la segmentation du détecteur d'objets. Le tableau 7.3 détaille les différents scénarios de filtrage et leurs contributions à la performance par rapport au détecteur d'objets et à la performance Seq2Seq seule :

— Dans le scénario où le nombre de prédictions des deux modèles est égal (**correspondance parfaite**), le niveau A améliore uniquement les performances de reconnaissance comme prévu. Ce scénario concerne la plupart des mots.
— Dans le scénario où le nombre de prédictions est différent et qu'il existe une **correspondance**, le gain est le plus élevé. En effet, la stratégie de niveau A permet

TABLE 7.2 – Étude d'ablation de l'extracteur de caractéristiques du détecteur d'objets et de l'impact des contributions A et B. La reconnaissance est évaluée avec le taux d'erreur de caractères (CER) et le taux d'erreur de mots (WER) (les valeurs inférieures sont meilleures). La segmentation est évaluée avec Intersection Over Union (IoU) et IoU_{Pixel} (les valeurs élevées sont meilleures). Le temps moyen est le nombre moyen de secondes pour qu'une méthode analyse un mot.

		Reconnaissance		Segmentation		Temps
Méthode	Extracteur de caractéristiques	CER (%)	WER (%)	IoU (%)	IoU_{Pixel} (%)	Temps Moyen / mot (s)
Seq2Seq		**5.3**	**19.4**	48.4	59.4	0.12
R-CNN	ResNet-18 FPN	12.0	36.4	78.6	80.3	1.25
	ResNet-34 FPN	12.2	37.7	79.6	81.3	1.29
	ResNet-50 FPN	11.4	34.7	80.5	81.5	1.61
	ResNet-101 FPN	10.7	34.2	81.0	82.2	2.17
Niveau A - section 7.1	ResNet-18 FPN	5.2	19.0	81.7	83.6	1.43
	ResNet-34 FPN	**5.0**	**18.6**	82.3	84.0	1.47
	ResNet-50 FPN	5.2	18.9	82.8	84.0	1.79
	ResNet-101 FPN	5.1	19.0	82.0	83.5	2.35
Niveau B - section 7.2	ResNet-18 FPN	12.0	36.4	82.6	85.0	1.40
	ResNet-34 FPN	12.2	37.7	83.3	85.9	1.44
	ResNet-50 FPN	11.4	34.7	83.8	86.3	1.77
	ResNet-101 FPN	10.7	34.2	84.4	87.1	2.32
Seq2Seg : Niveau A + B	ResNet-18 FPN	**5.2**	**19.0**	**85.9**	**88.3**	**1.58**
	ResNet-34 FPN	**5.0**	**18.6**	**86.1**	**88.9**	**1.62**
	ResNet-50 FPN	**5.2**	**18.9**	**86.3**	**89.0**	1.95
	ResNet-101 FPN	**5.1**	**19.0**	**85.6**	**88.4**	2.50

de filtrer les mauvaises prédictions du détecteur d'objet.

— Pour quelques mots où il n'y a **pas de correspondance**, rien n'est filtré et donc cette contribution n'améliore pas les performances du détecteur d'objet. En pratique, cela correspond à des mots pour lesquels le reconnaisseur fait plus d'erreurs que le détecteur d'objets.

TABLE 7.3 – Nombre de mots par scénario de filtrage entre le R-CNN et le Seq2Seq. Comparaison des performances des modèles seuls et contribution de niveau A. Le R-CNN utilise l'extracteur de caractéristiques ResNet34-FPN.

		R-CNN		Seq2Seq		Level A	
Type de filtrage	#Mots	CER (%)	IoU (%)	CER (%)	IoU (%)	CER (%)	IoU (%)
Correspondance parfaite	857	8.0	**85.6**	**3.8**	50.2	**3.8**	**85.6**
Correspondance	164	34.7	47.9	**11.5**	40.0	**11.5**	**64.6**
Pas de correspondance	11	**1.8**	**87.2**	36.6	31.3	**1.8**	**87.2**

Niveau B : l'affinage des coordonnées des boîtes englobantes par l'utilisation du treillis de segmentation améliore les performances de segmentation R-CNN pour un faible coût de temps de calcul.

Les résultats de comparaison avec les méthodes du chapitre précédent sont présentés dans le tableau 7.4. Les meilleures performances de reconnaissance et de segmentation sur

ce jeu de données sont données par la combinaison d'un modèle Seq2Seq et IntuiScript avec une petite marge par rapport à Seq2Seg (+0,1% CER, +1,6% IoU_{Pixel}) et un temps de calcul élevé (+3,45s par rapport à Seq2Seg). Pour pallier ce coût de calcul, une stratégie d'élagage (montrée en deuxième ligne) est utilisée. Cette stratégie dégrade la performance de reconnaissance ainsi que celle de segmentation qui la rend significativement inférieure à Seq2Seg pour la reconnaissance et la segmentation (-2,6% CER, -4,3% WER, +1,3% IoU, +2,6% IoU_{Pixel}). La section suivante présente une analyse qualitative des résultats obtenus par la méthode Seq2Seg.

TABLE 7.4 – Comparaison avec les approches du chapitre 6. La reconnaissance est évaluée avec le taux d'erreur de caractère (CER) et le taux d'erreur de mot (WER) (les valeurs inférieures sont meilleures). La segmentation est évaluée avec Intersection Over Union (IoU) et IoU_{Pixel} (les valeurs élevées sont meilleures). Le temps moyen est le nombre moyen de secondes pour qu'une méthode analyse un mot. ML signifie "Modèle de langage".

		Reconnaissance		Segmentation		Temps
Méthode	ML	CER (%)	WER (%)	IoU (%)	IoU_{Pixel} (%)	Temps moyen (s)
IntuiScript et Seq2Seq : Fusion compétition	Oui	**4.9**	**16.1**	**89.2**	**90.5**	5.07
IntuiScript et Seq2Seq : Fusion compétition (élagage)	Oui	7.6	22.9	84.8	86.3	**0.72**
Seq2Seg	Non	**5.0**	18.6	86.1	**88.9**	**1.62**

7.4.3 Résultats qualitatifs

Cette section présente une analyse qualitative des résultats obtenus par la méthode Seq2Seg. L'objectif est de visualiser l'impact des contributions de niveau A et de niveau B. Dans ces exemples de visualisation, la sortie du détecteur d'objets correspond aux prédictions avant le dernier NMS.

La figure 7.6 met l'accent sur la pertinence de la contribution niveau A. Le processus de filtrage par le modèle de reconnaissance sélectionne le nombre correct de lettres en minimisant le chevauchement et en maximisant le taux de couverture. De plus, l'utilisation du treillis de segmentation dans la contribution niveau B produit une segmentation précise des mots manuscrits, notamment dans l'exemple 1 pour les lettres « i » et « t » et dans l'exemple 2 pour les lettre « i » et « e » où les boîtes englobantes s'entremêlent. Dans ces deux exemples, la méthode Seq2Seg fait une analyse précise du mot manuscrit.

La figure 7.7 illustre des exemples où le modèle de reconnaissance commet des erreurs. Dans l'exemple 1, il n'y a pas de correspondance entre la prédiction du modèle de reconnaissance et le détecteur d'objets. Nous pouvons voir que le modèle de reconnaissance fait des erreurs de reconnaissance et donc le filtrage associé serait erroné. Dans ce cas (*cf.* cas

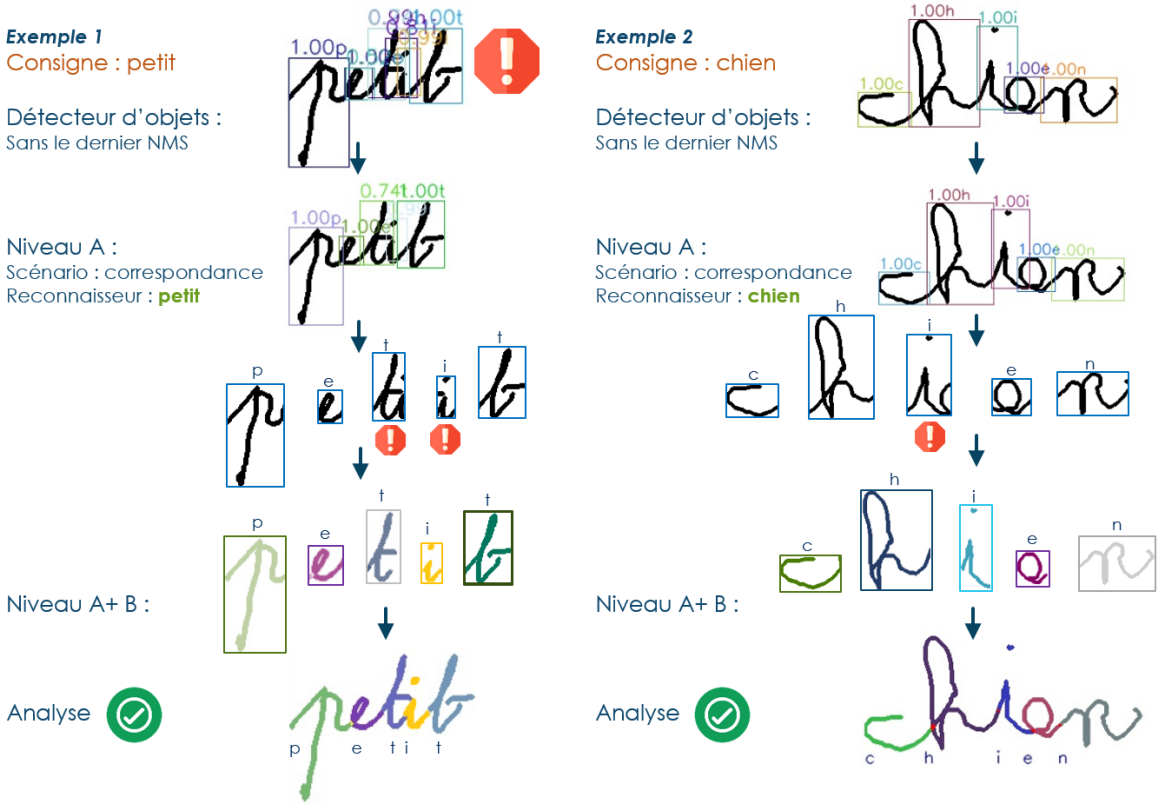

FIGURE 7.6 – Exemples qualitatifs de la méthode Seq2Seg avec une reconnaissance et une segmentation précise.

pas de correspondance section 7.1.2), les boîtes englobantes et les étiquettes prédites par le détecteur d'objets sont utilisées et fournissent un résultat précis en reconnaissance et en segmentation. La méthode Seq2Seg produit une analyse précise du mot. Dans un contexte applicatif, le système pourrait faire un retour à l'élève sur la faute d'orthographe commise. L'exemple 2 montre un cas où le filtrage par le modèle de reconnaissance conduit à une erreur de segmentation. Dans ce cas, la méthode Seq2Seg produit une analyse erronée du mot. De plus, on peut noter l'omission du tracé du point du "i" dans l'exemple 2 ce qui est assez fréquent dans un contexte d'apprentissage de l'écriture.

Cette analyse qualitative illustre l'intérêt de la méthode Seq2Seg. Les erreurs persistantes sont dues à des erreurs de reconnaissance du Seq2Seq. Nous pouvons noter qu'il est difficile d'évaluer la qualité de la segmentation de l'écriture manuscrite avec les métriques actuellement utilisées. En effet, il n'est pas facile de définir une vérité terrain de segmentation absolue pour certaines lettres en raison de la zone de ligature entre les lettres. Ainsi, une prédiction peut avoir un score IoU inférieur à 100% avec la vérité terrain alors

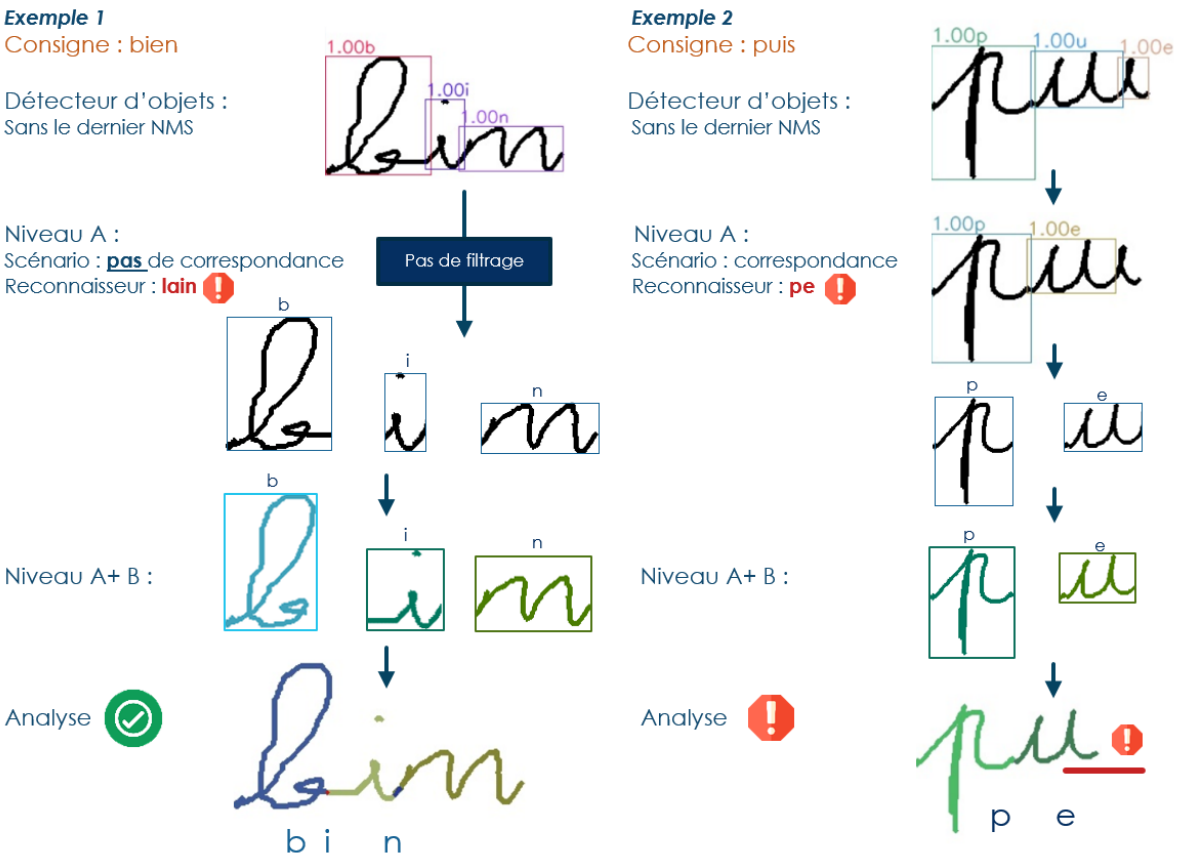

FIGURE 7.7 – Exemples où le modèle de reconnaissance fait des erreurs : dans l'exemple 1, il n'y a pas de correspondance entre le modèle de reconnaissance et le détecteur d'objets ce qui permet de faire une analyse correcte. Dans l'exemple 2 le filtrage conduit à une analyse erronée contenant une erreur de sous-segmentation.

que la segmentation associée est correcte visuellement. De plus, la classe de vérité terrain associée à une lettre dégradée peut varier selon l'annotateur (confusion entre la lettre « e » et la lettre « l », « a » et « o »...). Tenir compte de l'incertitude dans les prédictions pourrait être utile pour savoir quand un enseignant (humain) devrait prendre le relais sur la méthode Seq2Seg pour fournir un retour plus utile à l'enfant. La section suivante présente les résultats associés à une méthode de rejet tenant compte de l'incertitude dans les prédictions.

7.4.4 Premiers résultats : intégration d'un mécanisme de rejet

Dans cette partie, nous évaluons l'efficacité de la stratégie automatique de rejet présentée précédemment. Nous considérons une prédiction de la méthode Seq2Seg comme pertinente si elle ne contient pas d'erreur de reconnaissance (CER = 0), son score de couverture est supérieur à 90%, son score de chevauchement est inférieure à 30% et si l'IoU est supérieur à 80%. Comme expliqué précédemment, il est difficile d'évaluer précisément la qualité de la segmentation de l'écriture manuscrite. Nous avons fait le choix d'utiliser des valeurs de seuils tolérantes pour les métriques évaluant la segmentation. Il y a 364 prédictions "mots" de la méthode Seq2Seg à rejeter sur l'ensemble de tests de 1 032 mots avec ces critères. Nous déterminons les seuils sur le jeu de validation. Le seuil de couverture est fixé à 0.95 et le seuil de chevauchement à 0.3. Nous utilisons deux valeurs pour le seuil de la distance d'édition. Nous comparons une **approche de rejet souple** (seuil de couverture = 0.95, seuil de chevauchement = 0.3, seuil distance d'édition =2) avec une **approche de rejet strict** (seuil de couverture = 0.95, seuil de chevauchement = 0.3, seuil distance d'édition =1). L'objectif est de trouver un compromis afin de faire le plus de retours possibles avec le moins d'erreurs.

Le tableau 7.5 présente les résultats de classification de rejets des deux approches où un exemple classé comme vrai positif est un mot correctement analysé (reconnaissance et segmentation), un vrai négatif est un mot correctement rejeté, un faux positif est un mot mal analysé non rejeté et un faux négatif est un mot correctement analysé et rejeté. L'approche de rejet souple fait moins d'erreurs de rejet, i.e. rejeter un mot correctement analysé (faux négatif), que l'approche de rejet strict, néanmoins elle détecte moins d'erreurs de prédiction (faux positif).

TABLE 7.5 – Comparaison des performances de classification de rejet entre l'approche de rejet souple et l'approche de rejet strict.

Stratégie	Vrai positif	Vrai négatif	Faux positif	Faux négatif	Précision (%)	Rappel (%)	F1 (%)
Approche de rejet souple	**565**	202	162	**103**	77.7	**84.5**	**81.0**
Approche de rejet strict	447	**258**	**106**	221	**80.8**	66.9	73.2

Le tableau 7.6 illustre l'impact de l'utilisation du rejet sur les performances de reconnaissance et de segmentation de la méthode Seq2Seg. Les deux approches de rejet améliore les performances de reconnaissance et de segmentation. Le gain de reconnaissance est plus faible avec l'approche de rejet souple, cependant elle permet de faire un retour sur plus de mots. Ces résultats préliminaires montrent l'intérêt d'intégrer un mécanisme de rejet dans la méthode Seq2Seg.

TABLE 7.6 – Comparaison des performances de la méthode Seq2Seg avec et sans stratégie de rejet sur le jeu de test.

Stratégie	Nb mots	CER (%)	WER (%)	IoU (%)	IoU_{Pixel} (%)
Sans rejet	1032	5.2	18.9	86.5	89.1
Approche de rejet souple	727	3.9	14.3	**88.5**	**91.5**
Approche de rejet strict	553	**3.1**	**11.5**	**88.8**	**91.7**

7.5 Bilan

Dans ce chapitre, nous avons présenté la méthode Seq2Seg, une stratégie de combinaison qui utilise un modèle dédié à la reconnaissance en tant qu'oracle pour filtrer les prédictions de segmentation d'un détecteur d'objets, puis affine la segmentation à l'aide d'un treillis de segmentation. Seq2Seg produit le meilleur des deux mondes : la reconnaissance précise d'un Seq2Seq et la segmentation précise fournie par un détecteur d'objets R-CNN. Seq2Seg est suffisamment efficace pour fournir un retour immédiat aux enfants qui apprennent à écrire et il surpasse les résultats obtenus avec la méthode présentée dans le chapitre précédent sur cette tâche sans l'utilisation d'un modèle de langage. Ce dernier point rend Seq2Seg beaucoup plus flexible à d'autres contextes d'apprentissage. Ces travaux ont donné lieu à une publication lors de la conférence SIFED [6] (présentation de résultats préliminaires) et une publication à la conférence ICDAR [3] (présentation de la méthode Seq2Seg). Des résultats préliminaires ont montré l'intérêt d'intégrer un mécanisme de rejet dans la méthode Seq2Seg afin d'améliorer la qualité du retour fait à l'élève. Cette piste d'amélioration est développée dans les perspectives associées à ces travaux.

Conclusion et perspectives

Conclusion

Dans la littérature, de nombreux auteurs s'intéressent à une problématique de reconnaissance d'écriture manuscrite, mais très peu s'intéressent à une problématique d'analyse d'écriture regroupant une tâche de reconnaissance et une tâche de segmentation d'écriture. Dans cette thèse, nous nous sommes intéressés à l'analyse d'écriture manuscrite d'enfants en cours d'apprentissage de l'orthographe. L'objectif est de concevoir un système à la fois précis en reconnaissance et en segmentation pour faire un retour visuel immédiat à l'élève sur la qualité de son orthographe. La segmentation du mot en lettres permet de localiser précisément les fautes dans le tracé réalisé par l'élève. Les stratégies développées dans nos travaux se regroupent à travers deux axes.

Le premier axe présente plusieurs stratégies pour consolider le système d'analyse d'écriture manuscrite existant IntuiScript à l'aide de modèles d'apprentissage profond. La première contribution consiste à remplacer le modèle de reconnaissance de caractères isolés présent dans IntuiScript par un modèle plus performant. Ce nouveau modèle est basé sur une architecture de réseau de neurones convolutifs. L'originalité de notre travail a été d'injecter des informations décrivant la dynamique du tracé représentant un caractère en plus de l'information statique représentant la forme dans le but d'améliorer la performance de reconnaissance. Bien que le nouveau reconnaisseur de caractère soit plus performant, son intégration dans le système d'analyse de mots IntuiScript n'apporte pas d'amélioration significative. En effet, le système de guidage présent dans le système arrive à compenser ses erreurs de reconnaissances. Le système de guidage a été enrichi par le modèle de langue Phonetisaurus afin de pouvoir analyser des mots contenant des fautes d'orthographe. Le Phonetisaurus génère des mots phonétiquement proche de la consigne dans le but de guider l'analyse de mots contenant des fautes de nature phonétique. Dans la seconde contribution, nous proposons d'améliorer le système de guidage afin de pouvoir analyser des mots contenant des erreurs non-phonétique. Notre stratégie repose sur l'utilisation du résultat de reconnaissance d'un modèle de reconnaissance de mots Seq2Seq n'utilisant pas de modèle de langue. Cette stratégie améliore significativement les perfor-

mances de reconnaissance et de segmentation, néanmoins en contrepartie, elle augmente également le temps d'analyse. Nous proposons également une méthode pour optimiser le temps de calcul afin de pouvoir faire un retour immédiat sur des mots longs. Cette méthode permet d'avoir un bon compromis entre performance et temps d'analyse. Le système avec l'optimisation du temps de calcul obtient un CER de 7.6 % et un IoU de 84.8 %.

Le second axe regroupe des nouveaux systèmes d'analyse d'écriture basés sur des modèles d'apprentissage profond. Dans la troisième contribution, nous cherchons à améliorer les performances d'un modèle de reconnaissance de mots type Seq2Seq. La première stratégie a été d'utiliser en entrée le tracé segmenté selon des connaissances expertes liées au domaine de l'écriture. Dans la seconde stratégie, nous proposons d'améliorer la qualité de la segmentation explicite obtenue par le modèle à l'aide de deux méthodes. La première se base sur la modification de la fonction d'entrainement CTC du modèle tandis que le seconde utilise un treillis de segmentation en post traitement pour améliorer la segmentation. Les expérimentations associées ont permis de démontrer que la segmentation de l'écriture en entrée du modèle n'améliore pas les performances de reconnaissance et de segmentation ainsi que la modification de la CTC améliore légèrement la précision de segmentation, mais diminue la précision de reconnaissance. Seule l'utilisation d'un treillis de segmentation augmente significativement la précision de segmentation ce qui démontre la nécessité d'utiliser un modèle dédié à la segmentation pour obtenir à la fois une reconnaissance et une segmentation précise. La quatrième contribution propose un système basé sur la collaboration de deux modèles d'apprentissage profond afin de répondre au double problème de segmentation et reconnaissance d'écriture. Les travaux associés représentent la contribution majeure de la thèse. Dans ce nouveau système appelé Seq2Seg, un modèle R-CNN est dédié à la tâche de segmentation et un modèle Seq2Seq est dédié à la tâche de reconnaissance. La combinaison des deux modèles permet de faire une analyse précise de l'écriture dans un contexte orthographique. Le système est le plus performant en termes de reconnaissance et segmentation de toutes les stratégies développées dans nos travaux. Le système obtient un CER de 5.0 % (-2.6 % comparé à la seconde contribution) et un IoU de 86.1 % (+1.3% comparé à la seconde contribution). Même si la méthode Seq2Seg est performante en reconnaissance et en segmentation, il reste des leviers d'amélioration à explorer afin de rendre la méthode plus robuste.

Perspectives

Amélioration de la robustesse de la méthode Seq2Seg. La première perspective associée à nos travaux regroupe des stratégies visant à améliorer et à tester la robustesse de la méthode Seq2Seg. L'objectif est de faire un retour à l'élève sur la qualité de son orthographe. Il est préférable de demander à l'élève de réécrire le mot quand la confiance de la méthode est faible, plutôt que de lui faire un retour erroné. La première stratégie vise à intégrer la notion de rejet dans la méthode Seq2Seg. Les travaux préliminaires sur le rejet de la section 7.3 ont démontré l'intérêt d'ajouter la notion de rejet. L'objectif est de consolider la stratégie de rejet soit à l'aide d'un classifieur utilisant les scores décrits dans la section 7.3 ou bien d'explorer des méthodes de rejet plus évoluées [GE19]. La seconde stratégie est de vérifier la robustesse de la méthode avec le rejet à l'aide d'expérimentation en classes. Les expérimentations présentées dans ce manuscrit ont démontré l'efficacité de la méthode Seq2Seg pour l'analyse de l'écriture manuscrite française d'enfant. La troisième stratégie est de tester la robustesse de la méthode sur des écritures utilisant un autre alphabet comme l'arabe [Mah+18] ou le vietnamien [NNN18].

Amélioration de la coopération des modèles de la méthode Seq2Seg. La seconde perspective est de développer un cercle vertueux entre les modèles de la méthode Seq2Seg. La stratégie repose sur l'intégration du résultat de segmentation de la méthode Seq2Seg dans un nouveau mécanisme d'attention dans le modèle Seq2Seq. L'objectif est de focaliser le Seq2Seq sur les caractères segmentés afin d'améliorer la performance de reconnaissance. Le mécanisme d'attention peut intervenir au niveau des couches de convolutions [Woo+18] de l'encodeur, à la sortie des couches récurrentes de l'encodeur ou bien dans le module d'attention déjà présent dans le modèle. Ainsi, l'entrainement des modèles en plusieurs cycles améliorait les performances globales de la méthode Seq2Seg.

Amélioration des retours. Nos travaux ont montré l'intérêt de la méthode Seq2Seg pour faire des retours sur la qualité de l'orthographe de l'écriture. Dans la version actuelle, le système permet d'identifier et de localiser les fautes d'orthographe à partir d'une consigne. Le but de cette troisième perspective est d'enrichir ce retour avec la notion de type de fautes (confusion de phonème, dyslexie, ...) afin de cibler les notions à renforcer pour chaque élève. Ce diagnostique servirait également à l'enseignant *a posteriori* afin de détecter des éventuelles troubles spécifiques du langage écrit.

PUBLICATIONS

Workshop

[1] Omar Krichen, Simon Corbillé, Eric Anquetil, Nathalie Girard, Pauline Nerdeux. Online analysis of children handwritten words in dictation context. 14th International Workshop on Graphics Recognition, Sep 2021, Lausanne, Switzerland. https ://hal.science/hal-03448357v1

Conférences internationales

[2] Simon Corbillé, Elisa Fromont, Eric Anquetil, Pauline Nerdeux. Integrating Writing Dynamics in CNN for Online Children Handwriting Recognition. 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), Sep 2020, Dortmund, Germany. https ://hal.science/hal-02940236v1

[3] Simon Corbillé, Elisa Fromont, Eric Anquetil. Precise Segmentation for Children Handwriting Analysis by Combining Multiple Deep Models with Online Knowledge. The 17th International Conference on Document Analysis and Recognition (ICDAR). August 2023, San José, California, USA

Revues internationales

[4] Omar Krichen, Simon Corbillé, Eric Anquetil, Nathalie Girard, Elisa Fromont, Pauline Nerdeux. Combination of explicit segmentation with Seq2Seq recognition for fine analysis of children handwriting. International Journal on Document Analysis and Recognition, 2022 https ://hal.science/hal-03845144v1

Symposiums et colloques nationaux

[5] Simon Corbillé, Eric Anquetil, Elisa Fromont. Hybridation d'approches «transparentes» et basées «Deep Learning» pour l'analyse automatisée de productions graphiques

d'élèves dans le contexte de l'éducation. SIFED 20220 - Symposium International Francophone sur l'Ecrit et le Document, Jul 2020, En ligne. https ://hal.science/hal-03992931

[6] Simon Corbillé, Elisa Fromont, Eric Anquetil. Recognition and segmentation of children handwriting. SIFED 2022 - Symposium International Francophone sur l'Ecrit et le Document, Oct 2022, Rennes, France. https ://hal.science/hal-03879307v1

BIBLIOGRAPHIE

[AB02] Éric ANQUETIL et H. BOUCHEREAU, « Integration of an On-Line Handwriting Recognition System in a Smart Phone Device », in : *16th International Conference on Pattern Recognition, ICPR 2002, Quebec, Canada, August 11-15, 2002*, IEEE Computer Society, 2002, p. 192-194, DOI : 10.1109/ICPR.2002.1047827, URL : https://doi.org/10.1109/ICPR.2002.1047827.

[AGS17] Eric ANQUETIL, Nathalie N. GIRARD et Damien SIMONNET, « IntuiScript : suivi et aide à l'apprentissage de l'écriture », in : *Symposium International sur la Littératie à l'Ecole / International Symposium for Educational Literacy (SILE/ISEL)*, Ajaccio, Corse, France, juin 2017, URL : https://hal.inria.fr/hal-01551684.

[AL97] Éric ANQUETIL et Guy LORETTE, « Perceptual Model of Handwriting Drawing Application to the Handwriting Segmentation Problem », in : *4th International Conference Document Analysis and Recognition (ICDAR '97), 2-Volume Set, August 18-20, 1997, Ulm, Germany, Proceedings*, IEEE Computer Society, 1997, p. 112, DOI : 10.1109/ICDAR.1997.619824, URL : https://doi.org/10.1109/ICDAR.1997.619824.

[BA15] Manuel BOUILLON et Eric ANQUETIL, « Handwriting Analysis with Online Fuzzy Models », in : *17th Biennal Conference of the International Graphonomics Society (IGS)*, Drawing, Handwriting Processing Analysis : New Advances and Challenges, Pointe-à-Pitre, Guadeloupe, France, juin 2015, URL : https://hal.archives-ouvertes.fr/hal-01238091.

[Bar+21] Killian BARRERE et al., « Transformers for Historical Handwritten Text Recognition », in : *Doctoral Consortium - ICDAR 2021*, Nibal Nayef and Jean-Christophe Burie, Lausanne, Switzerland, sept. 2021, URL : https://hal.science/hal-03485262.

[BCB15] Dzmitry BAHDANAU, Kyunghyun CHO et Yoshua BENGIO, « Neural Machine Translation by Jointly Learning to Align and Translate », in : *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA,*

USA, May 7-9, 2015, Conference Track Proceedings, sous la dir. d'Yoshua Bengio et Yann LeCun, 2015, URL : http://arxiv.org/abs/1409.0473.

[BM17] Théodore Bluche et Ronaldo O. Messina, « Gated Convolutional Recurrent Neural Networks for Multilingual Handwriting Recognition », in : *14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15, 2017*, IEEE, 2017, p. 646-651, DOI : 10.1109/ICDAR.2017.111, URL : https://doi.org/10.1109/ICDAR.2017.111.

[BN08] Maximilian Bisani et Hermann Ney, « Joint-sequence models for grapheme-to-phoneme conversion », in : *Speech Commun.* 50.*5* (2008), p. 434-451, DOI : 10.1016/j.specom.2008.01.002, URL : https://doi.org/10.1016/j.specom.2008.01.002.

[Bon+17] Nathalie Bonneton-Botté et al., « Apprendre à écrire des lettres cursives sur tablette numérique : rôle du feedback. », in : *Symposium International sur la Littéracie à l'Ecole*, Universite Paris-Est Creteil, Ajjacio, France, juin 2017, URL : https://hal.univ-rennes2.fr/hal-01763977.

[BWL20] Alexey Bochkovskiy, Chien-Yao Wang et Hong-Yuan Mark Liao, « YOLOv4 : Optimal Speed and Accuracy of Object Detection », in : *CoRR* abs/2004.10934 (2020), arXiv : 2004.10934, URL : https://arxiv.org/abs/2004.10934.

[Car+20] Victor Carbune et al., « Fast multi-language LSTM-based online handwriting recognition », in : *Int. J. Document Anal. Recognit.* 23.*2* (2020), p. 89-102, DOI : 10.1007/s10032-020-00350-4, URL : https://doi.org/10.1007/s10032-020-00350-4.

[DA13] Adrien Delaye et Éric Anquetil, « HBF49 feature set : A first unified baseline for online symbol recognition », in : *Pattern Recognit.* 46.*1* (2013), p. 117-130, DOI : 10.1016/j.patcog.2012.07.015, URL : https://doi.org/10.1016/j.patcog.2012.07.015.

[Dam64] F. Damerau, « A technique for computer detection and correction of spelling errors », in : *Commun. ACM* 7.*3* (1964), p. 171-176, DOI : 10.1145/363958.363994, URL : https://doi.org/10.1145/363958.363994.

[Den12] Li Deng, « The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web] », in : *IEEE Signal Process. Mag.* 29.*6* (2012), p. 141-142, doi : 10.1109/MSP.2012.2211477, url : https://doi.org/10.1109/MSP.2012.2211477.

[GBC16] Ian Goodfellow, Yoshua Bengio et Aaron Courville, *Deep Learning*, http://www.deeplearningbook.org, MIT Press, 2016.

[GE19] Yonatan Geifman et Ran El-Yaniv, « SelectiveNet : A Deep Neural Network with an Integrated Reject Option », in : *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, sous la dir. de Kamalika Chaudhuri et Ruslan Salakhutdinov, t. 97, Proceedings of Machine Learning Research, PMLR, 2019, p. 2151-2159, url : http://proceedings.mlr.press/v97/geifman19a.html.

[Gra+06] Alex Graves et al., « Connectionist temporal classification : labelling unsegmented sequence data with recurrent neural networks », in : *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, sous la dir. de William W. Cohen et Andrew W. Moore, t. 148, ACM International Conference Proceeding Series, ACM, 2006, p. 369-376, doi : 10.1145/1143844.1143891, url : https://doi.org/10.1145/1143844.1143891.

[GSA17] Nathalie Girard, Damien Simonnet et Eric Anquetil, « IntuiScript a new digital notebook for learning writing in elementary schools : 1st observations », in : *18th International Graphonomics Society Conference (IGS2017)*, Proceedings of IGS 2017, Gaeta, Italy, juin 2017, p. 201-204, url : https://hal.inria.fr/hal-01548200.

[He+16] Kaiming He et al., « Deep Residual Learning for Image Recognition », in : *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, IEEE Computer Society, 2016, p. 770-778, doi : 10.1109/CVPR.2016.90, url : https://doi.org/10.1109/CVPR.2016.90.

[He+17] Kaiming He et al., « Mask R-CNN », in : *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, IEEE

Computer Society, 2017, p. 2980-2988, DOI : 10.1109/ICCV.2017.322, URL : https://doi.org/10.1109/ICCV.2017.322.

[HS97] Sepp HOCHREITER et Jürgen SCHMIDHUBER, « Long Short-Term Memory », in : *Neural Comput.* 9.*8* (1997), p. 1735-1780, DOI : 10.1162/neco.1997.9.8.1735, URL : https://doi.org/10.1162/neco.1997.9.8.1735.

[Hub95] David H. HUBEL, *Eye, Brain, and Vision*, Scientific American Library/Scientific American Books, 1995.

[HW59] D. H. HUBEL et T. N. WIESEL, « Receptive fields of single neurones in the cat's striate cortex », in : *The Journal of Physiology* 148.*3* (1959), p. 574-591, DOI : https://doi.org/10.1113/jphysiol.1959.sp006308, eprint : https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/jphysiol.1959.sp006308, URL : https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1959.sp006308.

[HW62] D. H. HUBEL et T. N. WIESEL, « Receptive fields, binocular interaction and functional architecture in the cat's visual cortex », in : *The Journal of Physiology* 160.*1* (1962), p. 106-154, DOI : https://doi.org/10.1113/jphysiol.1962.sp006837, eprint : https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/jphysiol.1962.sp006837, URL : https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1962.sp006837.

[Kri+21] Omar KRICHEN et al., « Online Analysis of Children Handwritten Words in Dictation Context », in : *Document Analysis and Recognition, ICDAR 2021 Workshops, Lausanne, Switzerland, September 5-10, 2021, Proceedings, Part I*, sous la dir. d'Elisa H. Barney SMITH et Umapada PAL, t. 12916, Lecture Notes in Computer Science, Springer, 2021, p. 125-140, DOI : 10.1007/978-3-030-86198-8_10, URL : https://doi.org/10.1007/978-3-030-86198-8%5C_10.

[LB05] Marcus LIWICKI et Horst BUNKE, « IAM-OnDB - an On-Line English Sentence Database Acquired from Handwritten Text on a Whiteboard », in : *Eighth International Conference on Document Analysis and Recognition (ICDAR 2005), 29 August - 1 September 2005, Seoul, Korea*, IEEE Computer Society, 2005, p. 956-961, DOI : 10.1109/ICDAR.2005.132, URL : https://doi.org/10.1109/ICDAR.2005.132.

[LeC+98] Yann LeCun et al., « Gradient-based learning applied to document recognition », in : *Proceedings of the IEEE* 86.*11* (1998), p. 2278-2324, DOI : 10.1109/5.726791.

[Let+59] J. Y. Lettvin et al., « What the Frog's Eye Tells the Frog's Brain », in : *Proceedings of the IRE* 47.*11* (1959), p. 1940-1951, DOI : 10.1109/JRPROC.1959.287207.

[LH19] Ilya Loshchilov et Frank Hutter, « Decoupled Weight Decay Regularization », in : *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, 2019, URL : https://openreview.net/forum?id=Bkg6RiCqY7.

[Li+22] Chuyi Li et al., « YOLOv6 : A Single-Stage Object Detection Framework for Industrial Applications », in : *CoRR* abs/2209.02976 (2022), DOI : 10.48550/arXiv.2209.02976, arXiv : 2209.02976, URL : https://doi.org/10.48550/arXiv.2209.02976.

[Lin+14] Tsung-Yi Lin et al., « Microsoft COCO : Common Objects in Context », in : *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, sous la dir. de David J. Fleet et al., t. 8693, Lecture Notes in Computer Science, Springer, 2014, p. 740-755, DOI : 10.1007/978-3-319-10602-1_48, URL : https://doi.org/10.1007/978-3-319-10602-1%5C_48.

[Lin+17] Tsung-Yi Lin et al., « Feature Pyramid Networks for Object Detection », in : *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, IEEE Computer Society, 2017, p. 936-944, DOI : 10.1109/CVPR.2017.106, URL : https://doi.org/10.1109/CVPR.2017.106.

[LJZ18] Hu Liu, Sheng Jin et Changshui Zhang, « Connectionist Temporal Classification with Maximum Entropy Regularization », in : *Advances in Neural Information Processing Systems 31 : Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, sous la dir. de Samy Bengio et al., 2018, p. 839-849.

[Mah+18] Sabri A. Mahmoud et al., « Online-KHATT : An Open-Vocabulary Database for Arabic Online-Text Processing », in : *The Open Cybernetics & Systemics Journal* 12 (2018), p. 42-59.

[Mic+] Johannes MICHAEL et al., « Evaluating Sequence-to-Sequence Models for Handwritten Text Recognition », in : *2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20-25, 2019*, IEEE, p. 1286-1293, DOI : 10.1109/ICDAR.2019.00208, URL : https://doi.org/10.1109/ICDAR.2019.00208.

[Mou+07] Harold MOUCHÈRE et al., « Synthetic On-line Handwriting Generation by Distortions and Analogy », in : *in 13th Conference of the International Graphonomics Society (IGS2007)*, Melbourne, Australia, nov. 2007, p. 10-13, URL : https://hal.science/inria-00300700.

[NNN18] Hung Tuan NGUYEN, Cuong Tuan NGUYEN et Masaki NAKAGAWA, « ICFHR 2018 - Competition on Vietnamese Online Handwritten Text Recognition using HANDS-VNOnDB (VOHTR2018) », in : *16th International Conference on Frontiers in Handwriting Recognition, ICFHR 2018, Niagara Falls, NY, USA, August 5-8, 2018*, IEEE Computer Society, 2018, p. 494-499, DOI : 10.1109/ICFHR-2018.2018.00092, URL : https://doi.org/10.1109/ICFHR-2018.2018.00092.

[Nov+12] Josef R. NOVAK et al., « Improving WFST-based G2P Conversion with Alignment Constraints and RNNLM N-best Rescoring », in : *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, ISCA, 2012, p. 2526-2529, URL : http://www.isca-speech.org/archive/interspeech%5C_2012/i12%5C_2526.html.

[Pui17] Joan PUIGCERVER, « Are Multidimensional Recurrent Layers Really Necessary for Handwritten Text Recognition ? », in : *14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15, 2017*, IEEE, 2017, p. 67-72, DOI : 10.1109/ICDAR.2017.20, URL : https://doi.org/10.1109/ICDAR.2017.20.

[Red+16] Joseph REDMON et al., « You Only Look Once : Unified, Real-Time Object Detection », in : *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, IEEE Computer Society, 2016, p. 779-788, DOI : 10.1109/CVPR.2016.91, URL : https://doi.org/10.1109/CVPR.2016.91.

[Ren+15] Shaoqing Ren et al., « Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks », in : *Advances in Neural Information Processing Systems 28 : Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, sous la dir. de Corinna Cortes et al., 2015, p. 91-99, URL : https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html.

[RF17] Joseph Redmon et Ali Farhadi, « YOLO9000 : Better, Faster, Stronger », in : *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, IEEE Computer Society, 2017, p. 6517-6525, DOI : 10.1109/CVPR.2017.690, URL : https://doi.org/10.1109/CVPR.2017.690.

[RF18] Joseph Redmon et Ali Farhadi, « YOLOv3 : An Incremental Improvement », in : *CoRR* abs/1804.02767 (2018), arXiv : 1804.02767, URL : http://arxiv.org/abs/1804.02767.

[Rus+15] Olga Russakovsky et al., « ImageNet Large Scale Visual Recognition Challenge », in : *Int. J. Comput. Vis.* 115.*3* (2015), p. 211-252, DOI : 10.1007/s11263-015-0816-y, URL : https://doi.org/10.1007/s11263-015-0816-y.

[SA16] Damien Simonnet et Eric Anquetil, « Handwriting Quality Analysis of Block Letters and Cursive Words », in : *Handwriting Today, Journal of the National Handwriting Association 15* (déc. 2016), p. 15-21, URL : https://hal.archives-ouvertes.fr/hal-01484924.

[SAB17] Damien Simonnet, Éric Anquetil et Manuel Bouillon, « Multi-criteria handwriting quality analysis with online fuzzy models », in : *Pattern Recognit.* 69 (2017), p. 310-324, DOI : 10.1016/j.patcog.2017.04.003, URL : https://doi.org/10.1016/j.patcog.2017.04.003.

[SBY17] Baoguang Shi, Xiang Bai et Cong Yao, « An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition », in : *IEEE Trans. Pattern Anal. Mach. Intell.* 39.*11* (2017), p. 2298-2304, DOI : 10.1109/TPAMI.2016.2646371, URL : https://doi.org/10.1109/TPAMI.2016.2646371.

[SG00] H. Schwenk et J.-L. Gauvain, « Improved ROVER using Language Model Information », in : (nov. 2000).

[Sim+19] Damien Simonnet et al., « Evaluation of children cursive handwritten words for e-education », in : *Pattern Recognit. Lett.* 121 (2019), p. 133-139, DOI : 10.1016/j.patrec.2018.07.021, URL : https://doi.org/10.1016/j.patrec.2018.07.021.

[SLM17] Abigail See, Peter J. Liu et Christopher D. Manning, « Get To The Point : Summarization with Pointer-Generator Networks », in : *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1 : Long Papers*, sous la dir. de Regina Barzilay et Min-Yen Kan, Association for Computational Linguistics, 2017, p. 1073-1083, DOI : 10.18653/v1/P17-1099, URL : https://doi.org/10.18653/v1/P17-1099.

[SZ15] Karen Simonyan et Andrew Zisserman, « Very Deep Convolutional Networks for Large-Scale Image Recognition », in : *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, sous la dir. d'Yoshua Bengio et Yann LeCun, 2015, URL : http://arxiv.org/abs/1409.1556.

[Tho76] René Thom, « Structural stability and morphogenesis », in : *Pattern Recognit.* 8.*1* (1976), p. 61, DOI : 10.1016/0031-3203(76)90030-3, URL : https://doi.org/10.1016/0031-3203(76)90030-3.

[Vas+17] Ashish Vaswani et al., « Attention is All you Need », in : *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, sous la dir. d'Isabelle Guyon et al., 2017, p. 5998-6008.

[Vin+15] Oriol Vinyals et al., « Show and tell : A neural image caption generator », in : *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, IEEE Computer Society, 2015, p. 3156-3164, DOI : 10.1109/CVPR.2015.7298935, URL : https://doi.org/10.1109/CVPR.2015.7298935.

[WBL22] Chien-Yao Wang, Alexey Bochkovskiy et Hong-Yuan Mark Liao, « YOLOv7 : Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors », in : *CoRR* abs/2207.02696 (2022), DOI : 10.48550/arXiv.2207.02696, arXiv : 2207.02696, URL : https://doi.org/10.48550/arXiv.2207.02696.

[Woo+18] Sanghyun Woo et al., « CBAM : Convolutional Block Attention Module », in : *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, sous la dir. de Vittorio Ferrari et al., t. 11211, Lecture Notes in Computer Science, Springer, 2018, p. 3-19, DOI : 10.1007/978-3-030-01234-2_1, URL : https://doi.org/10.1007/978-3-030-01234-2%5C_1.

[WS21] Xufei Wang et Jeong-Young Song, « ICIoU : Improved Loss Based on Complete Intersection Over Union for Bounding Box Regression », in : *IEEE Access* 9 (2021), p. 105686-105695, DOI : 10.1109/ACCESS.2021.3100414, URL : https://doi.org/10.1109/ACCESS.2021.3100414.

[ZSN21] Albert Zeyer, Ralf Schlüter et Hermann Ney, « Why does CTC result in peaky behavior ? », in : *CoRR* abs/2105.14849 (2021), arXiv : 2105.14849, URL : https://arxiv.org/abs/2105.14849.

UNIVERSITE DE RENNES

ATTESTATION DE REUSSITE AU DIPLOME

Le Responsable de la Scolarité atteste que

le doctorat en informatique
a été décerné à

Monsieur SIMON CORBILLÉ
né le 11 octobre 1993 à LE MANS (072)

au titre de l'année universitaire 2022/2023

Titre des travaux	:	Intégration de connaissances explicites à l' apprentissage profond pour la reconnaissance et la segmentation d' écriture manuscrite d' enfants
Date de soutenance	:	28 juin 2023
Etablissement soutenance	:	UNIVERSITE DE RENNES 1
Jury	:	M. HAROLD MOUCHERE, Président du jury, PROFESSEUR DES UNIVERSITES Université de Nantes M. SÉBASTIEN ADAM, Membre du jury, PROFESSEUR DES UNIVERSITES INST NAT SC APPLIQ ROUEN UNIVERSITE ROUE M. NICOLAS RAGOT, Membre du jury, PROFESSEUR DES UNIVERSITES UNIVERSITE TOURS FRANCOIS RABELAIS M. ERIC ANQUETIL, Co directeur, PROFESSEUR DES UNIVERSITES INSA DE RENNES Mme VERONIQUE EGLIN, Rapporteur avant soutenance, PROFESSEUR DES UNIVERSITES INST NAT SC APPLIQ LYON Mme ELISA FROMONT, Directeur de travaux, PROFESSEUR DES UNIVERSITES UNIVERSITE DE RENNES 1
Ecole doctorale	:	Mathématiques, informatique, signal et électronique et télécommunications.
Formation doctorale	:	INFORMATIQUE
Section CNU	:	27 - Informatique

Fait à Rennes, le 30 juin 2023

Fabrice LE GOUGUEC

UNIVERSITE DE RENNES

N° étudiant : 20116899

Avis important: Il ne peut être délivré qu'un seul exemplaire de cette attestation. Aucun duplicata ne sera fourni.

UNIVERSITE DE RENNES

PROCES VERBAL DE SOUTENANCE DU 28/06/2023 A 09h15

ANNEE UNIVERSITAIRE 2022/2023

Etudiant	:	M. SIMON CORBILLÉ né le : 11/10/1993
Diplôme	:	DT UNIVERSITE DE RENNES 1 spécialité INFORMATIQUE
Titre des travaux	:	Intégration de connaissances explicites à l'apprentissage profond pour la reconnaissance et la segmentation d'écriture manuscrite d'enfants
Secteur disciplinaire	:	Département Sciences et technologies de l'information et de la communication
Ecole doctorale	:	Mathématiques, informatique, signal et électronique et télécommunications.
Formation doctorale	:	INFORMATIQUE
Section CNU	:	27 - Informatique
Unité de recherche	:	INSTITUT DE RECHERCHES EN INFORMATIQUE ET SYSTEMES ALEATOIRES
Directeur	:	Mme ELISA FROMONT
Codirecteur	:	M. ERIC ANQUETIL
Lieu de soutenance	:	IRISA, Campus de Beaulieu, Rennes

La soutenance est publique.

Résultat : Admis

Avis de reproduction : en l'état

Membres du Jury

Nom	Qualité	Etablissement	Rôle	Signature
M. SÉBASTIEN ADAM	PROFESSEUR DES UNIVERSITES	INST NAT SC APPLIQ ROUEN UNIVERSITE ROUE	Membre	
M. NICOLAS RAGOT	PROFESSEUR DES UNIVERSITES	UNIVERSITE TOURS FRANCOIS RABELAIS	Membre	
M. ERIC ANQUETIL	PROFESSEUR DES UNIVERSITES	INSA DE RENNES	Co.Dir.	
Mme VERONIQUE EGLIN	PROFESSEUR DES UNIVERSITES	INST NAT SC APPLIQ LYON	Rap.av.sou	
Mme ELISA FROMONT	PROFESSEUR DES UNIVERSITES	UNIVERSITE DE RENNES 1	Dir.Thèse	
M. HAROLD MOUCHERE	PROFESSEUR DES UNIVERSITES	Université de Nantes	Rap.av.sou	

Le (la) Président(e) du jury M./~~Mme~~ H. Bouchéne atteste que :

~~M./Mme~~ Mme Elisa Fromont, directeur de thèse
~~M./Mme~~ Eric Anquetil, co-directeur de thèse
M./Mme ....., co-encadrant de thèse
M./Mme ....., co-encadrant de thèse
Était (étaient) présent(s) à la soutenance et n'a (n'ont) pas pris part à la décision.

Après y avoir été invité, le candidat :
☒ a prêté serment
☐ n'a pas prêté serment

Signature du président de jury

UNIVERSITE DE RENNES

Université de Rennes
DRI/Pôle Espace Doctoral – Scolarité des Doctorants
Campus de Beaulieu CS 74205 35042 RENNES cedex

Doctorat de l'Université de Rennes – mention **Informatique**

Soutenu le **28 juin 2023** par **CORBILLE Simon**

-=-=-=-=-=-=-=-=-=-=-

RAPPORT DE SOUTENANCE (Signature obligatoire de tous les membres du jury)

Nom du Président de jury: Harold Mouchère

Simon Corbillé a exposé son travail au travers d'une présentation très pédagogique, très claire et parfaitement illustrée, à l'image de son manuscrit.

Il a su faire des choix pertinents pour mettre en valeur et expliquer ses contributions principales au sein d'un travail de thèse très riche.

Lors des nombreux échanges avec le jury, Simon Corbillé a montré son recul et sa maîtrise du sujet sur les différentes thématiques travaillées pendant sa thèse: l'analyse de l'écriture manuscrite et l'apprentissage profond.

Cette soutenance a mis en lumière des qualités scientifiques et pédagogiques qui correspondent à celles d'un enseignant-chercheur.

Pour toutes ces raisons, le jury lui décerne le titre de docteur en informatique de l'Université de Rennes.

UNIVERSITÉ DE RENNES

Signatures des membres du jury :

ADAM Sébastien	RAGOT Nicolas	ANQUETIL Eric	EGLIN Véronique	FROMONT Elisa	MOUCHERE Harold
Po H. Mouchère			Po Harold Mouchère		

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Intégration de connaissances explicites à l'apprentissage profond pour la reconnaissance et la segmentation d'écriture manuscrite d'enfants

Mot clés : Reconnaissance, Segmentation, Apprentissage Profond, Ecriture manuscrite d'enfant, Education

Résumé : Notre objectif est de concevoir un système de reconnaissance et de segmentation d'écriture manuscrite d'enfants dans le but d'analyser précisément l'écriture afin de faire des retours orthographiques immédiats à l'enfant. Les contributions de cette thèse reposent sur l'hybridation de modèles d'apprentissage profond avec des modèles utilisant des connaissances expertes explicites. La première contribution consiste à intégrer la dynamique de l'écriture contenue dans le signal en ligne dans un réseau de neurones convolutifs pour faire de la reconnaissance de caractères. La seconde contribution porte sur l'amélioration d'un système existant d'analyse de mots. Ce système utilise un mécanisme de guidage par la consigne ainsi que les mots phonétiquement proches de la consigne pour aiguiller son analyse. Le principe est d'intégrer la prédiction d'un modèle de reconnaissance Seq2Seq dans le système de guidage. L'objectif est de palier les manques du mécanisme de guidage quand les mots d'entrée contiennent des erreurs non-phonétiques. La troisième contribution propose un nouveau système de reconnaissance et de segmentation. Il repose sur la combinaison d'un modèle dédié à la reconnaissance et d'un modèle dédié à la segmentation. Le système intègre également les connaissances contenues dans le signal en ligne afin d'améliorer la précision de la segmentation. Enfin, nous avons développé un mécanisme de rejet dans le but d'améliorer la qualité du retour effectué à l'enfant. Les résultats des expérimentations démontrent l'intérêt et l'efficacité de ces contributions.

Title: Integration of explicit knowledge with deep learning for the recognition and segmentation of children's handwriting

Keywords: Recognition, Segmentation, Deep learning, Children handwriting, Education

Abstract: Our goal is to design a tool for children's handwriting recognition and segmentation in order to accurately analyze the handwriting and provide immediate orthographic feedback to the child. The contributions of this thesis are based on the hybridization of deep learning models with models using explicit expert knowledge. The first contribution consists in integrating the writing dynamics of the online signal in a convolutional neural network for character recognition. The second contribution concerns the improvement of an existing word analysis system. This system uses a guidance mechanism based on the instruction and the words phonetically close to the instruction. It integrates the prediction of a Seq2Seq recognition model into the guidance system. The objective is to overcome the shortcomings of the guidance mechanism when the input words contain non-phonetic errors. The third contribution proposes a new recognition and segmentation framework. It is based on the combination of a model dedicated to recognition and a model dedicated to segmentation. The system also integrates the knowledge contained in the online signal in order to improve the accuracy of the segmentation. Finally, we have developed a rejection mechanism to improve the quality of the feedback given to the child. The results of the experiments demonstrate the interest and efficiency of these contributions.