Conversion d'écriture hors-ligne en écriture en-ligne et réseaux de neurones profonds

Elmokhtar Mohamed Moussa

▶ **To cite this version:**

Elmokhtar Mohamed Moussa. Conversion d'écriture hors-ligne en écriture en-ligne et réseaux de neurones profonds. Autre [cs.OH]. Nantes Université, 2024. Français. NNT : 2024NANU4001 . tel-04583082

HAL Id: tel-04583082

https://theses.hal.science/tel-04583082v1

Submitted on 22 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

COLLEGE ED MATHEMATIQUES
DOCTORAL STIC
PAYS DE LA LOIRE

Nantes Université

THÈSE DE DOCTORAT DE

NANTES UNIVERSITÉ

ÉCOLE DOCTORALE N° 641
Mathématiques et Sciences et Technologies du numérique de l'Information et de la Communication
Spécialité : *Informatique*

Par
Elmokhtar MOHAMED MOUSSA

Conversion d'écriture hors-ligne en écriture en-ligne et réseaux de neurones profonds

Thèse présentée et soutenue à Nantes, le 16 Janvier 2024
Unité de recherche : Laboratoire des Sciences du Numérique de Nantes

Rapporteurs avant soutenance :

Dr. Laurence LIKFORMAN	Maître de conférences HDR à Telecom Paris
Pr. Eric ANQUETIL	Professeur des universités à INSA Rennes

Composition du Jury :

Président :	Pr. Andreas FISCHER	Professeur des universités à la HEIA de Fribourg, Suisse
Examinateurs :	Dr. Clément CHATELAIN	Maître de conférences HDR à INSA Rouen Normandie
Dir. de thèse :	Pr. Harold MOUCHERE	Professeur des universités à Nantes Université
Encadrant :	Dr. Thibault LELORE	Ingénieur R&D à MyScript SAS

Remerciement

Je souhaiterais tout d'abord remercier mon directeur et mon co-encadrant de thèse Harold Mouchère et Thibault Lelore. Le doctorat n'était pas un long fleuve tranquille, mais c'est grâce à vous que j'en ressors une expérience extrêmement enrichissante et positive. J'ai énormément appris à vos côtés. Je tiens aussi à remercier Robin Mélinand pour avoir suivi de prêt mes travaux aussi. Je remercie également Pierre-Michel Lallican pour m'avoir fait confiance et de m'avoir donné cette opportunité à MyScript. Je remercie également beaucoup Nibal pour son soutien et ses encouragements tout au long de ma thèse.

Je remercie également tous les collègues de MyScript aux côtés desquels j'ai vécu cette aventure, tout d'abord mon équipe interactivité : Romain, Nicolas, Gregory, Jean-Christophe et l'ensemble de l'équipe de recherche : Zsolt, Loïs, Freddy, Guillermo, Prajol, Udit, Gaël, David, Nicolas, Lei, Mathieu, Fred, Bastien. Et aussi Marie de l'équipe UX pour son aide indispensable et son soutien. Je remercie également Emilie et Alexandros, j'ai eu la chance d'apprendre beaucoup en travaillant avec vous. Et bien sur, je remercie l'ensemble de l'équipe IT : Pierre-Yves, Stéphane et Martial.

Je remercie également tous les doctorants de l'équipe IPI et DUKE : Yejing, Tristan, Mathieu, Yassin, Alexandre, Ali, Noémie, Gaëlle, Mona, Jingwen, Kevin, Andreas, Sarah et bien sûr Mathilde, merci de m'avoir soutenu au quotidien à Polytech et chez nous. Merci pour tout ton aide et ta positivité. Je souhaite également remercié les autres collègues permanents de l'équipe IPI : Jeanpierre, Patrick, Nicolas, Vincent, Toinon, Matthieu, Sébastien, Alexandre et Pierre.

Par-dessus tout, je tiens à remercier mes parents d'avoir toujours été des modèles de persévérance. Je suis reconnaissant à ma mère d'avoir toujours été présente à me motiver et me pousser à aborder les défis du doctorat. C'est ta force qui m'a permis d'arriver là où j'en suis. Je remercie mon père qui est un puits de connaissance et un exemple d'éthique pour moi. Je remercie ma petite sœur Soumeya, tes grandes réussites m'ont toujours beaucoup inspiré pour me dépasser moi-même. Je remercie également mon petit frère Mohamed malgré ton jeune âge, tu nous rattrapes déjà à de très grands pas avec ton enthousiasme et énergie débordante. Je serai très heureux d'être à vos soutenances

respectives de doctorats et je suis très fière de vous deux et de ma plus petite sœur Mroum et mon plus petit frère Ibrahim. Je dédie ce manuscrit à la mémoire de ma cousine Nezha Abdeljelil. Tu étais une personne extrêmement généreuse, précieuse et exemplaire pour ma génération.

En hommage à Refaat Alareer, universitaire Palestinien de Gaza et poète décédé avec sa famille le 6 décembre 2023 à Gaza. L'université de Gaza a été entièrement détruite quelques semaines plus tard par les bombardements. Il était co-fondateur du workshop We Are Not Numbers (WANN) pour mettre en avant les jeunes écrivains palestiniens. Il écrit le poème suivant qui a fait le tour du monde :

"si je dois mourir
vous devez vivre
pour raconter mon histoire
pour vendre mes choses
pour acheter un morceau de tissu
et quelques fils pour faire un cerf-volant
faites qu'il soit blanc avec une longue traine
pour qu'un enfant, quelque part à gaza, alors qu'il regarde le paradis dans les yeux, en attendant son père qui est parti dans la gloire (dans le feu/dans une explosion)—
et qui n'a pas pu dire adieu
pas même à sa chaire (sa famille/son corps)
pas même à lui-même—
voie le cerf-volant, le cerf-volant que tu as fait en mon nom, voler dans les cieux
et (qu'il) pense, pour un instant, qu'un ange est là
ramenant l'amour
si je dois mourir
que ma mort amène l'espoir
que ma mort devienne un conte"

Table of Contents

1 Introduction **7**

1.1 Histoire du ductus d'écriture 7

1.2 Prise de notes et Nebo® 10

1.3 Conversion d'écriture hors-ligne en en-ligne et réseaux de neurones profonds 13

2 État de l'art **17**

2.1 Reconstruction de signal en-ligne et protocole d'évaluation 17

2.1.1 Définition 18

2.1.2 Métriques d'évaluation hors-ligne et en-ligne 19

2.1.3 Génération écriture hors-ligne synthétique 25

2.2 Généralités sur les réseaux de neurones profonds appliqués à l'écriture. . . 26

2.2.1 CNN pour l'écriture hors-ligne 26

2.2.2 RNN pour modélisation d'écriture en-ligne 28

2.2.3 Transformers 31

2.3 Travaux existants 38

2.3.1 Squelettisation et approche globale d'optimisation de parcours graphe 38

2.3.2 Résolutions locales des ambiguïtés 42

2.3.3 Approche à réseaux de neurones 43

2.4 Conclusion 48

3 Contributions **51**

3.1 Données et Métriques 51

3.1.1 Données et pré-traitements 51

3.1.2 Métrique invariante à la fréquence d'échantillonnage avec DTW-seg 54

3.1.3 Conclusion 57

3.2 FCNN et généralisation aux images de taille variable 58

3.2.1 Modèle multitâche entièrement convolutif 58

3.2.2 Évaluation sur les symboles isolés et expressions hors-ligne 62

3.2.3 Discussion 64

3.3 Transformer au niveau sous-trait 66
3.3.1 Vue d'ensemble 67
3.3.2 Extraction des sous-traits 67
3.3.3 Vecteurs descripteurs de sous-traits avec SET 71
3.3.4 Ordonnancement de sous-traits avec SORT 75
3.3.5 Apprentissage des modèles et inférence 77
3.3.6 Évaluations sur les expressions hors-ligne et ligne de texte 78
3.3.7 Conclusion 89

4 Conclusion **93**

Liste des figures **95**

Liste des tableaux **97**

Bibliographie **99**
Publications de l'auteur 99
À paraître 99
Bibliographie 110

Chapitre 1

Introduction

1.1 Histoire du ductus d'écriture

En écriture, le ductus désigne l'ensemble des caractéristiques importantes pour tracer les différents traits composant une lettre. Le ductus englobe plusieurs informations dont l'ordre, la direction, la vitesse et l'orientation, selon lesquels on trace les traits qui composent la lettre avec un outil d'inscription donné (calame, stylo, etc.). Chaque type d'écriture possède un ductus propre qu'il convient de respecter pour assurer une écriture fluide et naturelle. La figure 1.1 illustre le ductus moderne de l'alphabet latin [1].

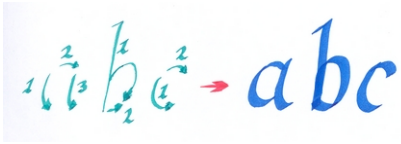


Figure 1.1 – Exemple du ductus d'écriture de lettres de l'alphabet latin avec un feutre.

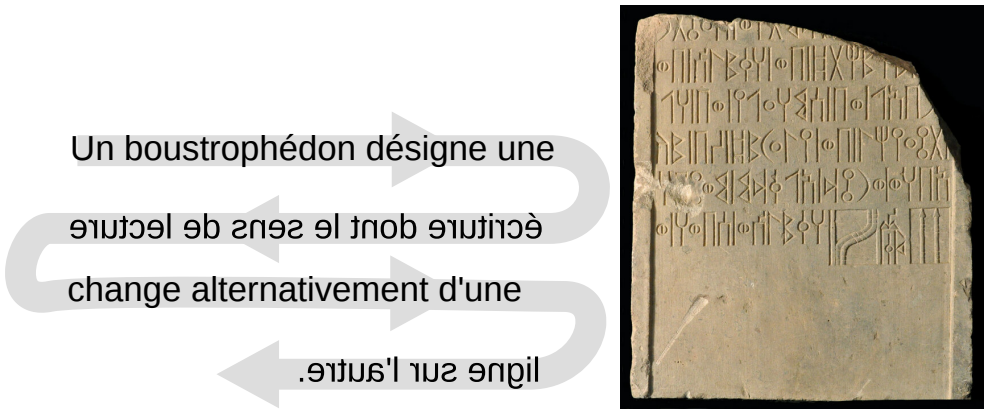
De nos jours, la majorité des systèmes d'écriture, tels que le latin, suivent un sens d'écriture de gauche à droite, avec les traits souvent tracés dans le sens des aiguilles d'une montre. Ces conventions d'écriture varient selon les cultures et les époques. Par exemple, l'arabe et l'hébreu utilisent un sens d'écriture opposé, allant de la droite vers la gauche. Dans le passé, il existait davantage de systèmes d'écriture avec des orientations différentes. Parmi les premiers systèmes d'écriture développés indépendamment à travers l'histoire, on trouve :

— Cunéiforme de la Mésopotamie (bassins versants du Tigre et de l'Euphrate) vers 3400 av. J.-C.
— Idéophonographique Chinois apparu entre 1200 et 1050 av. J.-C.
— Hiéroglyphique Olmèque de la Mésoamérique (Amérique centrale précolombienne) remontant à 650 av. J.-C.

1. Source de l'image `https://fr.wikipedia.org/wiki/Ductus`

Les systèmes d'écriture Cunéiforme et Idéophonographique étaient basés sur une forme de gravure sur un support rocheux (pierre, argile), ces systèmes emploient un ductus d'écriture de la droite vers la gauche. Le script chinois antique est gravé sur la pierre à l'aide d'un ciseau et un marteau. Le scribe saisit le marteau avec la main droite et le ciseau avec la main gauche et creuse les lettres en commençant depuis la droite pour aller vers sa gauche. L'écriture cunéiforme emploie les tablettes d'argile humide comme support sur lequel le scribe tamponne l'extrémité d'un roseau taillé en biais (appelé calame aujourd'hui) pour dessiner les lettres en forme de clous. Au départ ce système cunéiforme s'écrivait de haut en bas et de droite à gauches. Cette écriture verticale évolue plus tard en écriture horizontale en ligne écrite de gauche à droite. Le script Olmèque est quant à lui constitué de glyphes généralement disposés en colonnes et gravés sur la pierre. Il n'existe pas de consensus clair sur le ductus de ce script, que ce soit de droite à gauche ou gauche droite ou même une combinaison des directions.

Ces premiers systèmes d'écritures anciens ont ensuite suivi une évolution progressive et complexe en passant par plusieurs stades intermédiaires pour donner l'écriture moderne d'aujourd'hui. Le Boustrophédon, terme d'origine grecque signifiant "les trajets des bœufs lors du labour dans un champ", est un système d'écriture utilisé par plusieurs écritures à leurs stades anciens intermédiaires. Le sens d'écriture est alterné d'une ligne à l'autre (entre droite-gauche et gauche-droite) en inversant le ductus des lettres (voir Fig.1.2) [2].



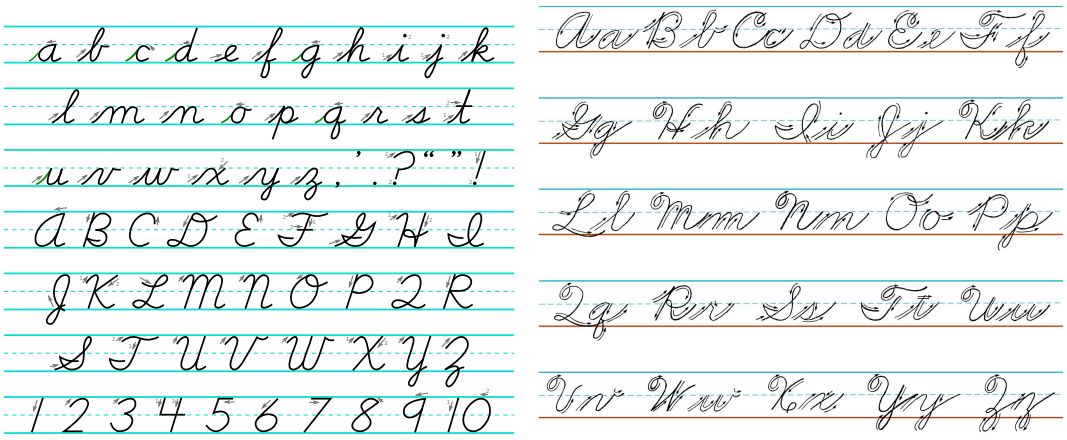
(a) (b)

FIGURE 1.2 – (a) Ductus de l'écriture boustrophédon. Ce ductus est aussi comparable au trajet de la tête d'impression d'une imprimante. (b) exemple d'inscription d'ancien Sabéen (actuel Yémen) du VIIIe siècle av. J.-C, avec un ductus boustrophédon.

2. Source des images `https://fr.wikipedia.org/wiki/BoustrophÃľdon`

Les langues Gréco-latines modernes ainsi que la majorité des systèmes d'écriture suivent aujourd'hui un ductus de gauche à droite (sens dextroverse) de haut en bas, alors que les scripts sémitiques Arabe et Hébreu sont restés sur un ductus de gauche à droite (sens sinistroverse). Les idéogrammes chinois historiquement écris verticalement de haut en bas de la droite vers la gauche sont aujourd'hui écrit plus fréquemment écrit aussi de la gauche vers la droite pour s'accommoder à l'écriture latine et aux limitations techniques des formats de documents informatiques.

Bien que le système d'écriture latin soit aujourd'hui omniprésent (Amérique, Europe et Afrique), la forme des lettres manuscrites ainsi que leur ductus varient d'une région à l'autre et parfois présentent également des variations au sein des individus d'une même région. Par exemple, aux États-Unis deux variantes d'écritures manuscrites de l'anglais sont apprises à l'école primaire : la méthode de Zaner-Bloser et la méthode de D'Nealian [3] [4]. Ces deux méthodes d'enseignement d'écriture cursive présentent beaucoup de similarité, la forme des lettres est presque identique, mais ils divergent sur le ductus de quelques lettres comme illustré par la figure 1.3.



(a) Alphabet D'Nealian. (b) Alphabet Zaner-bloser.

FIGURE 1.3 – Les deux alphabets cursifs de l'anglais américain. L'écriture de D'Nealian (a) trace lettres minuscules en commençant par un trait partant du bas (la ligne de base). En écriture cursive Zaner-Bloser (b), les lettres minuscules c, d, g, o et q commencent par un trait partant du haut. Les deux méthodes divergent aussi sur le ductus de quelques lettres majuscules comme B, P and R.

3. D'Nealian vs. Zaner-Bloser: how do their cursive fonts differ?
4. `https://en.wikipedia.org/wiki/D'Nealian`

1.2 Prise de notes et Nebo®

La prise de notes est le processus consistant à écrire des informations importantes, des idées, des points clés ou des détails lors d'un échange oral tel qu'une réunion de travail, un cours magistral ou une conversation. Les notes servent à capturer et à organiser des informations et pensées volatiles pour une utilisation ultérieure. Les notes qui peuvent être des documents manuscrits qui sont écrits à la main à l'aide d'un outil d'inscription (stylo, feutre, etc.) et d'un support d'écriture physique (papier, tableau, etc.), ou des documents tapuscrits (ou dactylographiés) qui sont saisie sur un ordinateur à l'aide d'un clavier [5].

Depuis plusieurs décennies, l'ordinateur a pris une place importante dans les activités d'écritures humaines à travers des outils de traitements de textes puissants et rapides. Néanmoins, lorsqu'il s'agit de prendre des notes, l'utilisation exclusive du tapuscrit n'est pas toujours unanime : aujourd'hui le clavier est souvent utilisé en combinaison avec le stylo dans nos habitudes de prise de notes. L'étude [MO14] conclut que le tapuscrit rend la fonction d'encodage superficielle et donc par conséquent peut nuire à l'apprentissage. Il montre que les notes saisies au clavier résulteraient souvent en une transcription longue et en verbatim contrairement la prise de notes manuscrite qui favorise la reformulation, le traitement et la fonction de stockage de l'information. Cependant, les travaux de [MDR19 ; Urr+21 ; Leb22] ont montré des résultats mitigés sur les performances de mémorisation ou compréhension en fonction de l'outil de prise de note. De plus, certains travaux [Luo+18] ont même indiqué une supériorité du tapuscrit dans certains contextes. En effet, la vitesse de saisie au clavier est très rapide et permet d'exprimer plus d'idées. Dans des conditions d'exploitation des notes sans révisions ou lorsque la mémoire de travail du sujet est moins performante, le manuscrit devient moins optimal que le tapuscrit.

Les avancées technologiques telles que les stylos intelligents Anoto [Fer+11], les surfaces tactiles et les stylets numériques ont ouvert de nouvelles perspectives pour améliorer la prise de notes manuscrites. Ces technologies représentent l'écriture comme un signal indiquant les mouvements du stylo dans un espace en fonction du temps. Ce signal caractérise le tracé en enregistrant les propriétés suivantes :

— Positions du stylo : les capteurs de position, tels que des capteurs électromagnétiques, capacitifs ou optiques enregistrent les coordonnées X et Y du stylet à intervalles réguliers, créant ainsi une séquence de points correspondant à la trajectoire du stylo sur la tablette.

5. https ://fr.wikipedia.org/wiki/Tapuscrit

— La pression appliquée par le stylo sur la tablette est aussi capturée, cette information permet de varier l'épaisseur des lignes ou des traits en fonction de la pression.
— Certains appareils peuvent aussi détecter l'inclinaison du stylet, permettant des effets de dessin plus naturels.

On parle ainsi d'écriture en-ligne par opposition à l'écriture hors-ligne obtenue en capturant des images de notes écrites avec des supports physiques (par exemple avec un stylo sur un papier). Bien que le stylo et le papier offrent toujours un confort inégalé comme outil d'inscription pour écrire des notes manuscrites hors-ligne, l'exploitation automatisée des documents hors-ligne est aujourd'hui limitée. Les documents hors-ligne sont généralement numérisés en capturant des images avec un scanner ou un appareil photo. C'est une étape peu pratique pour la plupart des utilisateurs et qui ajoute également du bruit en fonction des conditions d'acquisitions. Les documents en-ligne constituent quant à eux une alternative nativement numérique. Ce sont des signaux acquis sur une surface tactile (tablette) à l'aide d'un stylet électronique. En plus de l'information spatiale portée par les deux modalités, les documents en-lignes disposent d'une information supplémentaire qui est la temporalité des mouvements du stylo(pression, inclinaison et vitesse). Plusieurs logiciels dédiés à la prise de notes en-lignes sur tablette avec un stylet ont été proposés. Ces logiciels de prise de notes en-ligne sont en constante amélioration, notamment avec l'émergence des modèles de langues larges et des modèles génératifs, et surpassent aujourd'hui leurs analogues hors-lignes. Ils comprennent plusieurs avantages, parmi eux :

— Une interactivité fluide et plus intuitives des utilisateurs avec leurs notes.
— Une organisation efficace des notes grâce à leur indexation automatisée.
— Un moteur de reconnaissance d'écriture plus performant.

Nebo, proposée par MyScript, est une application de prise de notes disponible sur les tablettes iOS™, Android™et Windows©. Nebo offre la possibilité de créer et rédiger des notes manuellement. Nebo permet d'allier également le manuscrit avec la saisie au clavier ainsi que les traitements de textes tapuscrits. Nebo permet de créer trois types de notes :

— Les notes régulières peuvent s'apparenter à une page web de largeur fixe et une longueur infinie. En effet, de nouvelles lignes vierges sont ajoutées à la fin si les utilisateurs arrivent en bas de l'écran de la tablette.
— Les notes libres offrent un espace de rédaction infini dans les deux dimensions, sans contraintes de structuration (absence de ligne droite à suivre).
— Les imports PDF, quant à eux, sont des fichiers sur lesquels il est possible de rédiger. Pour annoter un document existant par exemple.

Lors de la rédaction de contenu dans Nebo, le contenu manuscrit est stocké par une encre numérique interactive appelée MyScript *interactive ink* (MyScript *iink*®). Cette encre offre les deux intérêts suivants :

— Encre convertissable de manuscrit en tapuscrit grâce au moteur de reconnaissance multilingue. Par conséquent, les utilisateurs peuvent éditer leur écriture tapuscrite avec un clavier dans Nebo.

— Encre éditable grâce à des gestes réalisables avec le stylet. Par exemple : en raturant un mot, il est supprimé. Un geste de stylo vertical entre deux mots permet d'ajouter un saut à la ligne.

Nebo permet de combiner l'écriture manuscrite et tapuscrite (converties ou tapé au clavier) dans le même support pour rédiger des notes. Il fonctionne sur différents types de contenus comme le texte, les mathématiques, les diagrammes. Les utilisateurs ont la possibilité d'inclure des médias dans leurs notes et de les partager en les exportant vers différents formats (tels que Word ou PDF).

Récemment, des appareils hybrides ont été conçus avec des caractéristiques matérielles pour reproduire la même expérience d'écriture et de prise de notes que sur du papier. Ils disposent d'écran *E-Ink* offrant une lisibilité plus élevée en plein soleil et une faible fatigue oculaire par comparaison aux écrans LCD ou OLED des tablettes. Ils sont fournis avec un stylet actif qui procure une légère résistance lors de son utilisation sur l'écran, créant ainsi une expérience proche de celle de l'écriture sur du papier avec un stylo. Ils permettent de conserver la simplicité (et réduire les distractions numériques) et le confort de l'écriture manuscrite traditionnelle tout en profitant des avantages de la technologie numérique.

Cependant, ces appareils restent aujourd'hui limités en termes de ressources de calcul par rapport à d'autres appareils tactiles. Une alternative à cette solution hardware qui simule la prise de notes sur papier a été étudiée par plusieurs chercheurs dans l'effort de proposer un algorithme capable de reconstruire automatiquement la dynamique des mouvements de stylos à partir d'images de notes papier dans l'objectif de permettre la conversion de document hors-ligne en document en-ligne. Ces travaux s'inscrivent dans la même optique de réflexion que les écrans *E-Ink* : combiner l'ergonomie des outils d'inscription physiques (stylo et papier) avec les avantages des technologies numériques de traitements de notes.

Cette thèse CIFRE se focalise sur cette problématique dans le contexte de Nebo : comment concevoir un tel système de conversion d'image de notes manuscrites en signal afin de permettre aux utilisateurs d'inclure automatiquement leurs notes papier dans

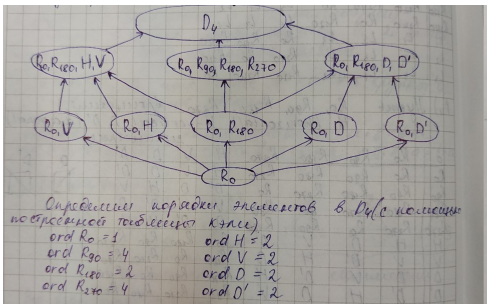
Nebo ?

1.3 Conversion d'écriture hors-ligne en en-ligne et réseaux de neurones profonds

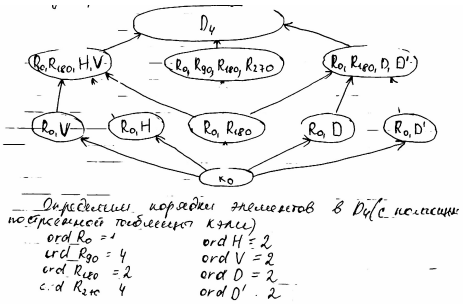
La conversion automatique d'images de scans de notes manuscrites en signal temporel de l'écriture en-ligne revient à reconstruire la trajectoire spatio-temporelle du stylo à partir uniquement des informations visuelles. Plusieurs étapes de traitement des images sont nécessaires afin de pouvoir effectuer la reconstruction de la trajectoire du stylo. Dans l'ordre, ces étapes incluent généralement la correction de perspective et des distorsions dues à la numérisation, la binarisation et le nettoyage d'artefacts (par exemple du quadrillage). La figure 1.4 illustre toutes ces étapes de traitements depuis la capture jusqu'à la conversion en signal d'écriture en-ligne. Ces étapes sont loin d'être triviales, c'est pour cela que plusieurs travaux portés sur la conversion d'écriture hors-ligne en écriture en-ligne se positionnent sur des cas d'études en avals de ces traitements où l'image en entrée est supposée dans un état propre afin de se focaliser sur le problème de fond. La complexité du problème réside dans l'ambiguïté qui se produit lorsqu'il y a des zones de l'écriture où la direction ou l'intention du mouvement du stylo n'est pas claire. Cela peut survenir pour plusieurs raisons :

— Zone de jonction ou croisements : Lorsque plusieurs traits se croisent ou se chevauchent, il peut être difficile de déterminer l'embranchement à continuer avec le stylo.
— Zone de rebroussement : désigne les endroits où la direction du trait de l'écriture change brusquement, créant des points d'inflexion ou des traits tracés en doubles.
— Variabilité individuelle dans l'ordre et dans l'orientation des traits. Les préférences et habitudes individuelles influencent la façon dont chacun écrit.

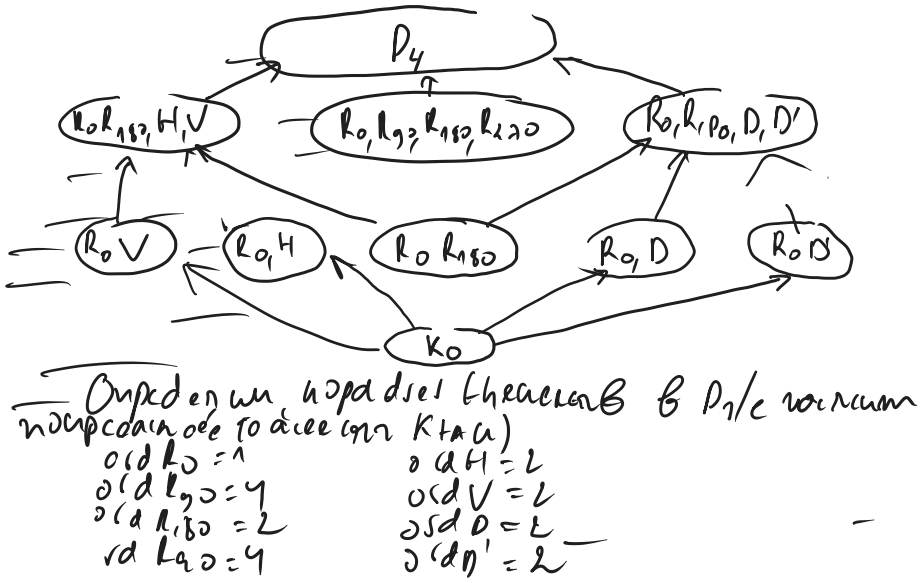
Pour surmonter ces difficultés, des systèmes à règles basées sur des heuristiques ont été proposés en premier dans l'état de l'art. Néanmoins, la conception manuelle d'algorithme universel avec un ensemble fini de règles strictes capables de reconstituer la trajectoire du stylo pour toutes les écritures s'est avérée très difficiles. Ces systèmes sont très difficiles à maintenir et ne se généralisent pas à suffisamment de différentes langues ou contenus. Récemment, de nombreuses approches basées sur des apprentissages automatiques de réseaux de neurones à partir de couples de données hors-ligne et en-ligne ont été proposées dans la littérature afin de résoudre ce problème. Ces efforts ont démontré des résultats



(a) Photo capturée d'une note manuscrite.



(b) Binarisation.



(c) Conversion en note manuscrite en-ligne.

FIGURE 1.4 – Chaîne de traitements depuis la capture d'un document hors-ligne jusqu'à sa conversion en en-ligne pour permettre potentiellement son importation dans un logiciel de prise de note en-ligne comme Nebo.

prometteurs surpassant les approches heuristiques sur les images de lettres isolées. Cette thèse s'appuie sur ce constat pour approfondir les approches neuronales et répondre aux problématiques suivantes :

- Comment généraliser l'approche neuronale pour la reconstruction de trajectoire de stylo sur des contenus plus large tel que des mots, des lignes de texte et plus diversifiés et complexes comme des expressions mathématiques ?
- Étant donné le manque de jeu de données avec une correspondance image-tracés, principalement en raison du coût élevé de la collecte de ces données, quelles stratégies d'apprentissage devraient être mises en œuvre ?
- Comment évaluer quantitativement la qualité de l'écriture en-ligne reconstruite de manière objective ?

Ce mémoire de thèse est découpé de la façon suivante : nous débutons en passant en revue l'état actuel des méthodes utilisées pour cette conversion. La section 2.1 énonce clairement le problème et présente le protocole d'évaluation nécessaire. Ensuite, dans la section 2.2, nous examinons les réseaux de neurones, mettant en lumière leurs applications réussies dans la modélisation de l'écriture. Nous explorons également les approches de l'état de l'art pour la reconstruction du signal temporel des mouvements de stylo dans la section 2.3, analysant les avantages et les limites de ces méthodes.

Le chapitre 3 se concentre sur les contributions de cette thèse. Dans la section 3.1, nous définissons notre contexte d'application, mettant l'accent sur la généralisation des approches neuronales aux échelles plus larges de phrases et d'expressions mathématiques. Nous présentons les bases de données en-ligne considérées et le protocole d'évaluation avec une métrique d'alignement adaptée aux variations de fréquences d'échantillonnage ? Ensuite, la section 3.2 introduit une approche novatrice utilisant un réseau de neurones entièrement convolutif multitâches, démontrant son avantage pour la squelettisation et discutant des limites du paradigme basé sur la séquence de caractéristiques visuelles. Dans la section 3.3, un paradigme alternatif basé sur l'information positionnelle des différents segments de l'image est proposé et implémenté avec un réseau Transformer. Nous montrons comment cette méthode surpasse plusieurs limitations de l'état de l'art, notamment dans le contexte des phrases cursives et des expressions mathématiques.

Chapitre 2

État de l'art

Dans ce chapitre, nous passerons en revue l'état de l'art des méthodes utilisées pour la conversion d'écriture hors-ligne en écriture en-ligne. Dans la section 2.1 nous énoncerons le problème à résoudre et présenterons le protocole d'évaluation nécessaire pour comparer les différentes méthodes proposées dans la littérature.

Par la suite, dans la section consacrée aux réseaux de neurones (section 2.2), nous offrirons un aperçu global des réseaux de neurones et illustrerons leurs applications réussies dans des contextes liés à la modélisation de l'écriture.

Nous explorerons également, dans la section 2.3, les différentes approches de l'état de l'art pour la reconstruction du signal temporel des mouvements de stylo à partir d'image. Nous parcourrons d'abord les méthodes classiques basées sur des heuristiques, comme l'optimisation globale de graphe et la résolution locale des ambiguïtés, pour finalement aborder les approches plus récentes exploitant l'apprentissage automatique, en particulier les réseaux de neurones. Nous analyserons les avantages et les limites de ces approches afin de mieux comprendre pourquoi les méthodes neuronales semblent mieux adaptées à notre contexte d'application.

2.1 Reconstruction de signal en-ligne et protocole d'évaluation

Dans cette section, nous allons d'abord définir le problème de reconstruction du signal d'écriture en-ligne à partir d'images d'écriture hors-ligne. Nous montrerons que cette tâche est complexe en raison des ambiguïtés liées aux intentions de mouvement du stylo. Ensuite, nous présenterons les images d'écriture hors-ligne synthétiques utilisées dans cette étude, ainsi que la nature simplifiée du signal en-ligne à reconstruire. Enfin, nous examinerons les métriques d'évaluation couramment employées dans l'état de l'art pour quantifier la qualité des reconstructions.

2.1.1 Définition

Dans cette, nous étudions la tâche de conversion d'écriture hors-ligne en écriture en-ligne. Cette tâche consiste en la reconstruction du signal temporel des mouvements du stylo à partir d'une image statique d'écriture. Cette tâche est couramment employée pour extraire des caractéristiques temporelles pertinentes pour l'ensemble des systèmes hors-ligne comme la reconnaissance, la vérification de signature [PP99], l'identification du scripteur. La synthèse d'écriture hors-ligne quant à elle est parfois effectuée en convertissant d'abord le hors-ligne vers l'en-ligne pour prendre avantage des systèmes de synthèses en-lignes plus performants [May+20]. La figure 2.1 illustre un exemple d'utilisation d'une étape de conversion d'écriture hors-ligne vers en-ligne dans le système de transfert de style d'écriture hors-ligne.

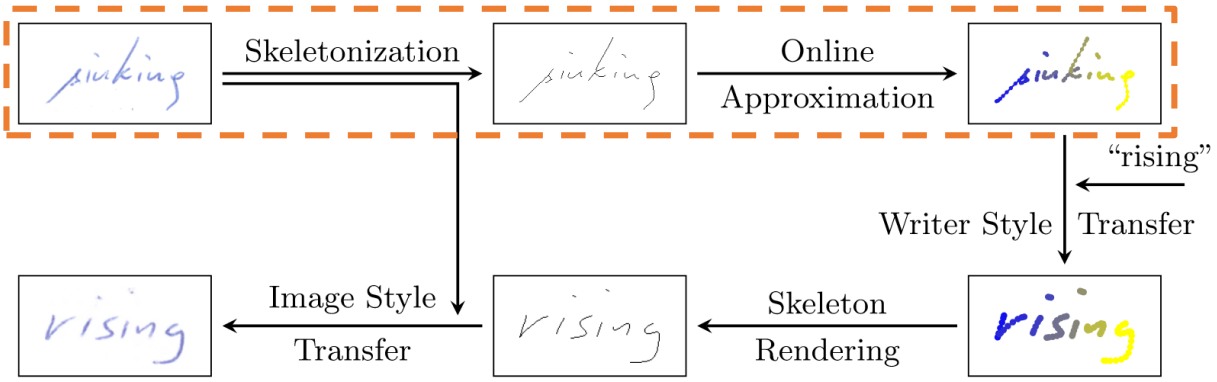


FIGURE 2.1 – Pipeline de Transfert de style d'écriture hors-ligne à hors-ligne (illustration de [May+20]). L'image est convertie d'abord en signal en-ligne (encadré en pointillé orange), puis le transfert de style est effectué dans le domaine en-ligne avant d'être rastériser en une image d'écriture hors-ligne.

Notre objectif étant de reconstruire la trajectoire du stylo à partir d'image d'écriture manuscrite hors-ligne, idéalement, nous disposerions d'un ensemble d'images hors-ligne avec les données en-ligne correspondantes (comme dans les bases IRONOFF[VG+99] et CROHME2023[Xie+23]). Cependant, la collecte de cette donnée est relativement plus difficile et coûteuse. De plus, un tel alignement est très difficile à garantir, par exemple il n'est pas disponible dans les bases citées précédemment. Par conséquent, dans la littérature, une approche couramment adoptée consiste à générer des données hors-lignes simplifiées en rastérisant les données en-ligne en images et en les dégradant (voir section 2.1.3).

Un signal en-ligne est un enregistrement des mouvements d'un stylet sur une surface

tactile, contenant diverses informations telles que les coordonnées 2D (x, y), la vitesse, l'accélération et la pression appliquée. Dans les travaux existants (voir section 2.3), les systèmes de conversion se concentrent, généralement, uniquement sur la reconstruction de la forme et de la séquence temporelle des symboles, en utilisant une fréquence d'échantillonnage constante, sans tenir compte de la vitesse ou de la pression en raison de la complexité supplémentaire que cela entraînerait. Il convient de noter que bien que l'on parle ici de la reconstruction du "signal en-ligne", il s'agit en réalité de la reconstruction d'une forme simplifiée de ce signal.

Bien que des simplifications soient opérées sur les entrées et sorties de cette tâche de conversion hors-ligne vers en-ligne, la difficulté reste dans l'expression des mouvements des stylos dans les zones ambiguës suivantes :

— Zone de jonction : zones où plusieurs traits s'entrecroisent ou un trait s'entrecroise avec lui-même pour créer une intersection. Une décision est à prendre pour relier les entrées et sorties correctes à cette intersection.
— Zone de rebroussement : désigne les endroits où la direction du trait de l'écriture change brusquement, créant des points d'inflexion ou des traits tracés en doubles.
— Détection de lever de stylo : les extrémités des traits peuvent être cachées ou occultées par d'autres traits, complexifiant considérablement la détection de début et fin de trait.
— Variabilité individuelle dans l'ordre des traits : les préférences individuelles pour l'ordonnancement des traits est caractérisée par une grande diversité. Cette diversité peut ajouter de la complexité à la tâche.

La figure 2.2 illustre les différentes difficultés rencontrées par les systèmes de reconstruction de signaux en-ligne. Trois scripteurs écrivent la même lettre de différentes manières, mais le résultat visuel est identique en image. Trois systèmes génèrent trois reconstructions différentes, les deux premières sont plausibles, mais la dernière est incohérente (ressemblant davantage au ductus d'un "a").

2.1.2 Métriques d'évaluation hors-ligne et en-ligne

Pour évaluer la performance d'un système de récupération de trajectoires, il est nécessaire de comparer les trajectoires reconstruites avec la vérité terrain en-ligne. Cela peut être réalisé soit en utilisant le signal en-ligne capturer simultanément lors du tracé hors-ligne [VG+99 ; NDPH05], soit en générant des images hors-ligne à partir de données en-ligne [QNY06] (cf. section 2.1.3). Dans la littérature, la performance des techniques

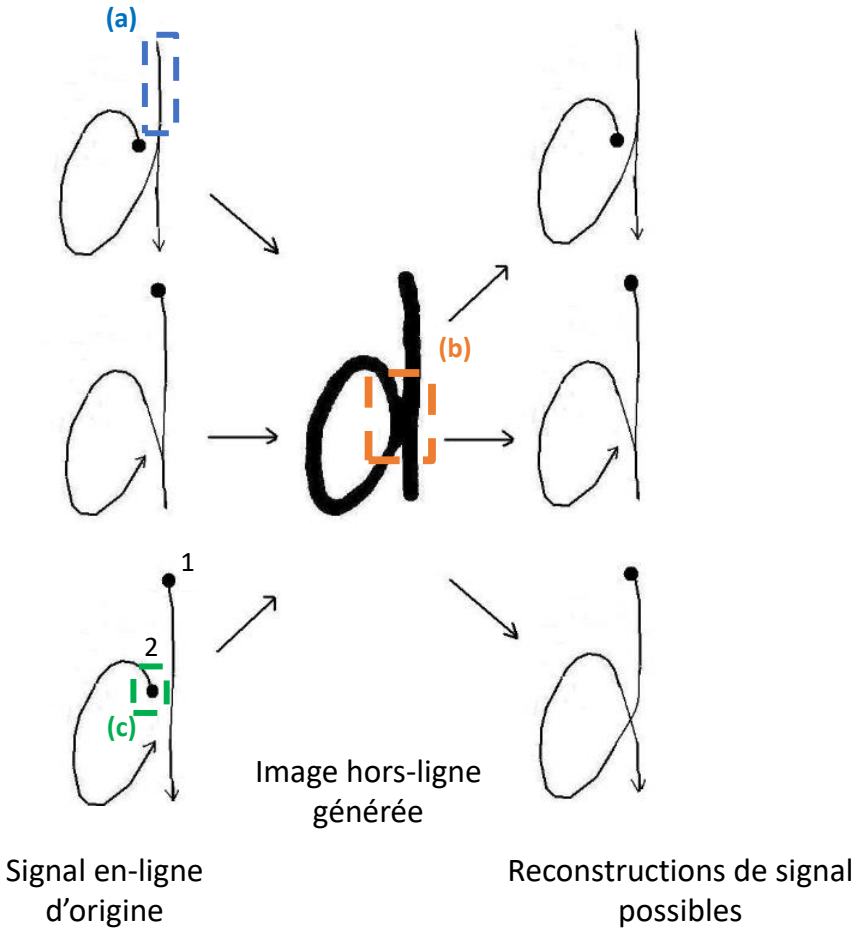


FIGURE 2.2 – Différentes variantes de ductus de la lettre "d" générant la même image hors-ligne et quelques reconstructions possibles proposé par un système de conversion hors-ligne vers en-ligne. Les points noirs indiquent le début du trait et la flèche sa direction. (a) une zone de rebroussement, (b) une zone de jonction et (c) un lever et poser de stylo "caché". Illustration par [Rou07]

de reconstruction de trajectoires est parfois évaluées par une étude qualitative par des utilisateurs humains [PP99 ; Boc+93 ; DR95 ; KY00]. Cette évaluation présente l'avantage de mieux juger la variabilité individuelle de l'écriture. En effet, la mise au point de cette perception subjective en mesures numériques précises est complexe et nécessite le développement de métriques appropriées. Ce type de métrique présente des difficultés importantes et a été peu explorée dans la littérature. Cependant, l'évaluation visuelle nécessite des moyens importants pour sa mise en œuvre. Dans la littérature [DR95 ; Jäg98], elle est réalisable uniquement avec un nombre d'échantillons de test relativement faible. De plus, l'évaluation visuelle est subjective, non quantitative et sujette aux erreurs. Le

manque de standardisation rend difficile la comparaison des résultats et conclusions des différentes approches. Quelques protocoles d'évaluation quantitative de performance ont été proposés pour surmonter ces limitations.

L'erreur quadratique moyenne *RMSE* entre la trajectoire prédite et la trajectoire vérités est proposée comme métriques d'évaluation dans [HAMB13 ; Tri ; Dia+22]. La *RMSE* est une métrique simple peu coûteuse en temps et permet de vérifier la précision des coordonnées de la trajectoire prédites. Cependant, comme il s'agit d'une comparaison stricte point à point, les deux séquences doivent être de mêmes longueurs. Les trajectoires prédites par les systèmes proposés dans la littérature ne vérifient pas toujours cette condition, rendant ainsi cette métrique peu pratique. Un post-traitement est nécessaire pour ré-échantillonner les deux signaux à comparer afin d'avoir la même longueur pour les deux. Cette étape peut être complexe, car un échantillonnage naïf peut introduire des artefacts ou des distorsions dans la trajectoire entraînant une perte d'informations importantes des détails subtils de l'écriture. De plus, si les deux trajectoires ne sont pas parfaitement synchronisées, par exemple en raison de retards ou de décalages temporels, cela peut entraîner une augmentation significative de la *MSE*, et fausser l'évaluation de la qualité globale de la trajectoire.

La *DTW* (*Dynamic Time Warping* en anglais) [SC78] est une technique d'alignement élastique de deux séquences temporelles. DTW est couramment utilisée dans le domaine de la reconnaissance de la parole et de l'analyse de séries temporelles. Bien que plus complexe à calculer par rapport à la *RMSE*, elle est particulièrement utile lorsque les séquences à comparer ont des longueurs différentes ou présentent des variations temporelles importantes dû à la vitesse ou un déphasage. [NV06] propose d'utiliser la DTW comme métrique d'évaluation des trajectoires proposées par leur système. La figure 2.3 illustre un exemple d'évaluation de reconstruction de trajectoire avec la *RMSE* et la DTW.

L'algorithme DTW calcule l'alignement optimal entre deux signaux de longueurs différentes x et $\hat{x}$ grâce à un appariement élastique de type un-vers-plusieurs. L'algorithme détermine le chemin de recalage $w_p = \{(i,j) \in \mathbb{N}^{N\times M}\}_{p=1}^{P}$ établissant la correspondance optimale des deux séquences. Ce chemin est le résultat de la minimisation de l'équation 2.1 avec des contraintes de continuité temporelles afin de respecter l'ordre temporel. Ces contraintes sont les suivantes :

— Frontières : $w_1 = (1,1) \wedge w_P = (N,M)$. Les paires de départ et de début doivent apparier.
— Monotonie : soit $w_p = (i,j)$ et $w_{p+1} = (i',j')$ alors $i \leq i' \wedge j \leq j'$. Les paires

apparaissent dans l'ordre croissant.

— Continuité : $i' \leq i+1 \wedge j' \leq j+1$. Le décalage entre les paires consécutives dans w_p est au maximum de 1. Cela assure qu'il n'y a aucune discontinuée en utilisant tous les éléments de chaque séquence au moins une fois.

La distance euclidienne est généralement utilisée comme fonction de coût f dans l'équation 2.1 et permet d'initialiser la matrice $C_{N\times M}$ des coûts cumulés. Un algorithme dynamique effectue l'optimisation en mettant à jour itérativement de la matrice C avec l'équation 2.2. Le parcours de C s'arrête à sa dernière case C_{NM}. Le coût total du meilleur alignement entre x et $\hat{x}$ est exprimé par C_{NM}. Ce coût est souvent normalisé par la longueur $|w_P|$ du chemin de recalage.

$$\begin{aligned} DTW(x,\hat{x}) &= \min_{w\in W}\left\{\sum_{p=1}^{P} f(w_p)\right\} \\ f(w_p) &= f(x_i,\hat{x}_j) \end{aligned} \tag{2.1}$$

$$\begin{aligned} C_{ij} &= f(x_i,\hat{x}_j) + \min\begin{cases} C_{i,j-1} \\ C_{i-1,j-1} \\ C_{i-1,j} \end{cases}, \\ C_{0,0} &= f(x_0,\hat{x}_0) \end{aligned} \tag{2.2}$$

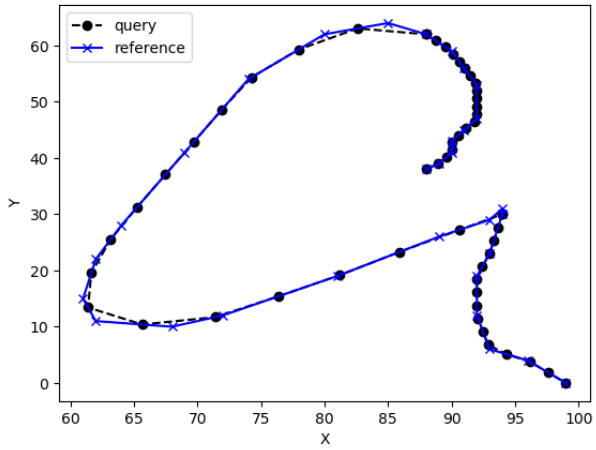
Les variations de fréquence d'échantillonnage observées dans les systèmes de reconstruction de trajectoire de stylo présenter un défi pour l'emploi DTW comme métrique d'évaluation. Les variations de fréquence d'échantillonnage peuvent introduire des coûts supplémentaires qui ne sont pas nécessairement liées aux caractéristiques d'écriture réelle. Cela peut affecter la capacité de la DTW à évaluer correctement la similarité entre les trajectoires. De plus, la DTW se concentre principalement sur la similarité temporelle et ne capture pas nécessairement la similarité structurelle ou la fidélité des glyphes.

L'utilisation d'un moteur de reconnaissance en-ligne permet de vérifier la cohérence linguistique des glyphes reconstruit. En effet, les systèmes de reconstruction de trajectoire s'avèrent bénéfique pour la reconnaissance de l'écriture hors-ligne. Cette étape intermédiaire permet d'employer les moteurs de reconnaissances en-ligne plus performants. Le taux de reconnaissance en-ligne peut être comparé ensuite avec celui obtenu à partir du signal en-ligne vérité. Il offre un aperçu de la cohérence sémantique du signal reconstruit. Même si le taux de reconnaissance reflète indirectement la qualité du signal, cette métrique

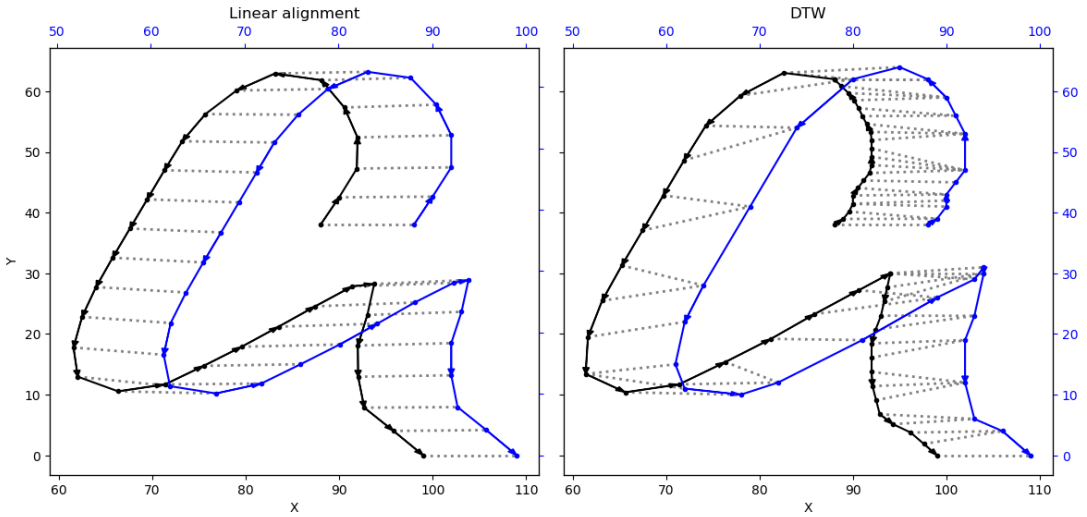
présente un inconvénient majeur : en fonction de la complexité des systèmes de reconnaissance employée, il est possible que le mot correct soit reconnu même si la reconstruction en-ligne est infidèle, par exemple les moteurs de reconnaissances sont souvent invariants à certaines transformations géométriques (inclinaisons, rotations, etc.). De plus, ces moteurs encapsulent un modèle de langue capable de compenser des erreurs présentes dans la trajectoire estimée. D'autres métriques focalisées sur la précision de la reconstruction du signal d'écriture en-ligne sont très présent dans la littérature. Parmi ces métriques on peut citer :

— Précision de la résolution des intersections [Bhu+18].
— Le taux de restauration correcte de la trajectoire du stylo [QNY06].
— Distance d'édition (par exemple la distance de Levenshtein) [Jäg98] sur les parcours de graphe pour les approches basé sur l'optimisation du chemin de coût minimale (voir section 2.3.1).

Pour évaluer des systèmes de reconstruction de trajectoire du stylo avec les métriques mentionnées précédemment, il est important de disposer de jeux de données d'images hors-ligne et le signal en-ligne vérité associé. Cela implique donc la nécessité de capturer simultanément le signal en-ligne et le tracé hors-ligne d'un contenu manuscrit lors de la phase d'acquisition des données. Pour ce faire, [VG+99] utilise une feuille de papier fixée sur une tablette de numérisation, permettant ainsi l'écriture en temps réel tout en enregistrant les mouvements du stylo. Cependant, il est important de noter que lors du processus d'enregistrement, du bruit peut s'introduire à tout moment (absence d'encre, erreur d'acquisition). De plus la numérisation de la feuille et la binarisation ajoutent du bruit supplémentaire. Ces étapes d'acquisition et de traitement altèrent ainsi les deux modalités indépendamment et conduisent à un mauvais alignement entre le script statique et son équivalent dynamique (comme le montre la figure 2.4). Des techniques spécifiques de réalignement doivent être appliquées afin de compenser ces différences et garantir une correspondance adéquate entre les représentations en-ligne et hors-ligne. [VG+99] emploie une méthode itérative de réalignement à l'aide des gradients pour ajuster la position et l'orientation du script statique afin de le rapprocher autant que possible de son équivalent dynamique. Cependant, cette méthode permet de diminuer les écarts, mais des décalages peuvent persister.



(a) Signal référence et signal reconstruit. On note la différence de la fréquence d'échantillonnage entre les deux.



(b) À gauche, l'alignement linéaire RMSE qui nécessite un rééchantillonnage des deux signaux, et à droite, alignement par DTW.

FIGURE 2.3 – Deux méthodes classiques d'évaluation de l'écriture manuscrite en-ligne par alignement : $RMSE$ et DTW.

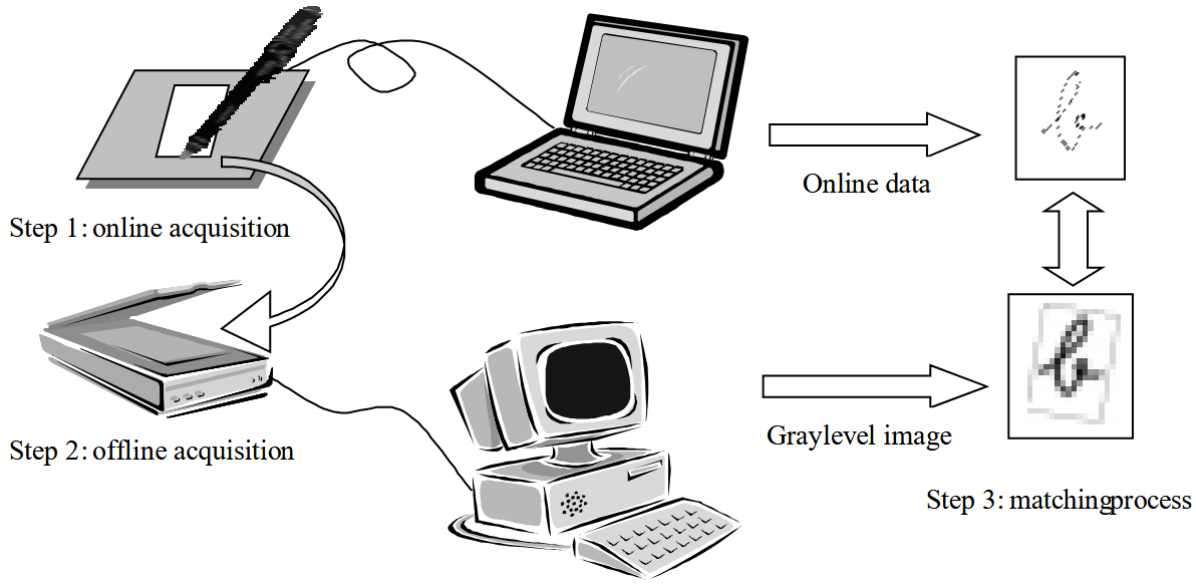


FIGURE 2.4 – Exemple de désalignement entre une image et le signal brut associé. En vert les projections des points du signal sur l'image. Source [VG+99]

2.1.3 Génération écriture hors-ligne synthétique

En pratique, le signal en-ligne est souvent utilisé pour générer des images hors-lignes. Cela est motivé par les difficultés associées à l'acquisition simultanée du signal en-ligne et de l'image hors-ligne énoncées précédemment dans la section 2.1.2. Ce processus appelé la rastérisation, est le problème inverse à notre tâche de conversion d'image statique en écriture dynamique. Il s'agit d'un processus beaucoup plus simple garantissant un alignement entre un signal et son rendu hors-ligne. La rastérisation est omniprésente dans la littérature pour obtenir des images hors-ligne synthétique à partir d'un jeu de données en-ligne [Bhu+18 ; Arc+21 ; ZYT18]. Avant la rastérisation, les coordonnées continues du signal en-ligne doivent être normalisées et discrétisées à une échelle d'écriture fixe. Le signal est en général rastérisé avec une épaisseur de trait constante de 1 à 3 pixels [Dia+22 ; Cha20 ; HE17]. Une stratégie de rastérisation avec des épaisseurs de traits variables pour des rendus synthétique plus proche du réaliste est proposée par [Guo+19].

L'anti-aliasing et autres opérateurs morphologiques de lissage peuvent être combinés avec la rastérisation afin d'obtenir une écriture plus fluide [Kov07]. L'utilisation des GANs [Gra14] (réseaux antagonistes génératifs) pour la génération réaliste de document hors-ligne présente un moyen plus puissant pour une rastérisation plus réalise, imitant de près le changement d'épaisseur et d'intensité d'encre attribué aux caractéristiques individuelles

FIGURE 2.5 – Quatre styles de dessin. De gauche à droite : épaisseur croissante, épaisseur croissante puis décroissante, épaisseur aléatoire, épaisseur constante. Illustration par [Guo+19]

de l'écriture manuscrite. [Mad+22] propose par exemple un modèle génératif capable de transférer le style d'une base de données hors-ligne réel à une autre. Cette technique demeure inexplorée dans la littérature sur les systèmes de conversion hors-ligne vers en-ligne.

2.2 Généralités sur les réseaux de neurones profonds appliqués à l'écriture.

Dans cette section, nous présentons les différentes architectures de réseaux de neurones en prenant en exemples les travaux sur les modélisations de l'écriture. Le lecteur intéressé par plus de détails peut se référer aux livres sur apprentissage profonds [GBC16 ; Zha+23].

2.2.1 CNN pour l'écriture hors-ligne

La vectorisation de dessin est une étape cruciale de la création d'animations et de croquis en 2D. Elle consiste à convertir les sketchs dessinés en format vectoriels. Les artistes commencent souvent par dessiner un brouillon de leurs œuvres sur papier, puis les vectorisent manuellement pour finaliser leurs conceptions sur des logiciels adaptés. Cependant, la vectorisation de croquis bruts et complexes peut s'avérer difficile, car il faut combiner plusieurs lignes qui se chevauchent en une seule et supprimer les lignes superflues et le bruit de fond. [SS+16] propose un réseau de simplification entièrement convolutif (FCNN) bout-en-bout pour simplifier les croquis en haute définition. En contraste avec les modèles CNN classiques qui utilisent des couches entièrement connectées et ne permettent pas de traiter des images de résolution arbitraire, ici seules les couches de convolutions

sont exploitées permettant ainsi le traitement d'images de résolution variable. Les couches entièrement connectées dans les CNN sont utilisées pour intégrer l'information globale de l'image et apprendre des relations complexes entre les caractéristiques extraites par le CNN. Le réseau FCNN proposé ici est composé d'un encodeur et d'un décodeur. L'encodeur emploie une stratégie alternative pour intégrer des informations contextuelles plus globales dans sa carte de caractéristiques. La figure 2.6 montre l'architecture du réseau qui se décline comme suit :

— L'encodeur constitué de plusieurs blocs de convolution avec un pas de 2 et de *maxpooling* réduisant par un facteur de $\frac{1}{8}$ la résolution spatiale de la carte de caractéristique extraite. Cela permet d'augmenter exponentiellement la taille des champs réceptive d'une couche à l'autre, ainsi intégrant des contextes spatiaux plus larges.

— Un décodeur constitué de couches d'*upsampling* utilisées pour augmenter la résolution spatiale des caractéristiques. Elles permettent de restaurer les détails et de reconstruire une représentation à la résolution originale de l'image à partir des caractéristiques à basse résolution.

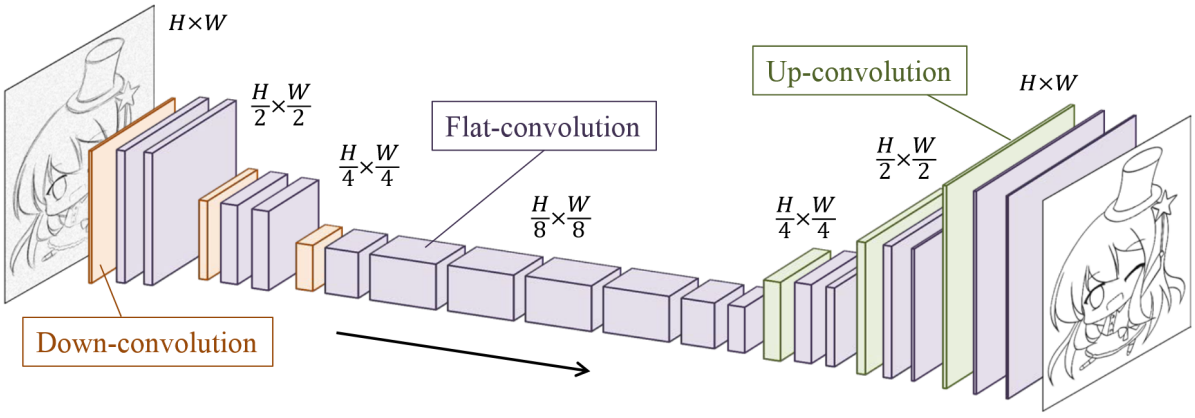


FIGURE 2.6 – Architecture FCNN encodeur-décodeur de [SS+16].

Leur méthode se généralise bien sur une variété de croquis approximatifs tels que des dessins numérisés au crayon et papier, ainsi que des croquis structurés avec détaillés complexes. Le succès observé des FCNNs pour la vectorisation a motivé d'autres approches telles que celle de [Guo+19]. Ils ont développé un système en deux étapes employant deux FCNNs pour vectoriser des dessins plus propres. Le premier CNN multitâches prédit les images de squelette et de jonction et le second CNN résout la connectivité des segments

autour des jonctions (cf. figure 2.7). Ils ont démontré des résultats supérieurs à l'état de l'art sur le jeu de donnée *Quick, Draw !* [HE17]. Cependant, leur méthode est limitée à des jonctions de degrés relativement faibles 3 à 6 qui tiennent dans une fenêtre de 32×32, limitant ainsi leur application au contenu comme l'écriture manuscrit.

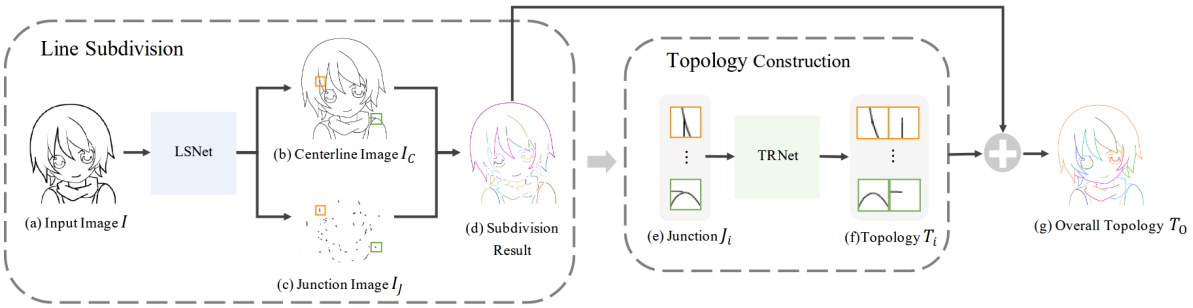


FIGURE 2.7 – Architecture FCNN encodeur-décodeur de [SS+16].

Par la suite, dans la section 3.2 nous proposons une première approche FCNNs inspirée des travaux précédents pour généraliser le modèle CNN de [ZYT18] à des images de tailles arbitraires.

2.2.2 RNN pour modélisation d'écriture en-ligne

Les RNNs (réseaux de neurones récurrents) sont un type de réseaux de neurones utilisés pour traiter des données séquentielles, telles que des séquences de texte, de temps ou de données audio ou encore l'écriture en-ligne. Contrairement aux réseaux de neurones traditionnels qui traitent chaque entrée de manière indépendante, les RNN ont une mémoire interne qui leur permet de maintenir des informations sur les états précédents et de les utiliser pour inférer le traitement des entrées actuelles. L'un des types les plus couramment utilisés de RNN est le LSTM (Long Short-Term Memory), qui a été conçu pour surmonter les limitations des RNN traditionnels, notamment les problèmes d'instabilité du gradient qui se produit lors de l'entraînement de RNNs profonds. Les LSTM introduisent des mécanismes de portes dans les cellules récurrentes pour contrôler le flux d'informations. Chaque cellule LSTM dispose de trois portes principales : la porte d'oubli, la porte d'entrée et la porte de sortie. L'équation 2.3 défini la propagation avant d'une

cellule LSTM. La figure 2.3 illustre ces mécanismes de portes.

$$
\begin{aligned}
f_t &= \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right) \\
i_t &= \sigma_g\left(W_i x_t + U_i h_{t-1} + b_i\right) \\
o_t &= \sigma_g\left(W_o x_t + U_o h_{t-1} + b_o\right) \\
\tilde{c}_t &= \sigma_c\left(W_c x_t + U_c h_{t-1} + b_c\right) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
h_t &= o_t \odot \sigma_h\left(c_t\right)
\end{aligned}
\tag{2.3}
$$

Avec les variables suivantes :

- $x_t \in \mathbb{R}^d$: vecteur en entrée de la cellule LSTM.
- $f_t \in (0,1)^h$: porte d'oubli.
- $i_t \in (0,1)^h$: porte d'entrée.
- $o_t \in (0,1)^h$: porte de sortie.
- $h_t \in (-1,1)^h$: état caché et sortie de la cellule LSTM.
- $\tilde{c}_t \in (-1,1)^h$: vecteur d'information à sauvegarder.
- $c_t \in \mathbb{R}^h$: état caché.
- $W \in \mathbb{R}^{h \times d}, U \in \mathbb{R}^{h \times h}$ and $b \in \mathbb{R}^h$: matrices de poids et vecteur de biais (paramètres à apprendre) d et h sont la dimension du vecteur en entrée et le nombre d'unités cachés respectivement.

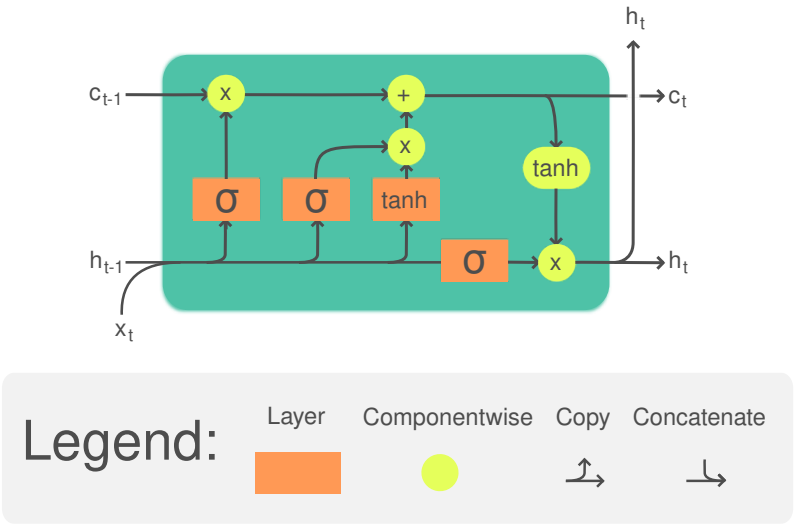
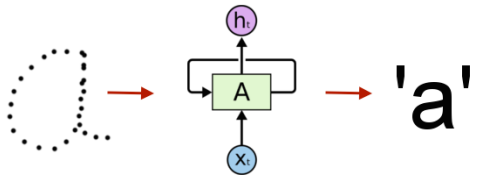
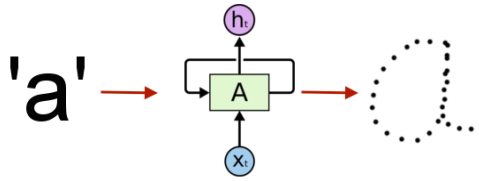


FIGURE 2.8 – Illustration des portes du LSTM. Source de l'image : Wikipédia.

Les LSTM sont capables de capturer les dépendances à long terme dans les séquences ce qui en fait un choix populaire pour des tâches de modélisation séquentielle telles que la reconnaissance d'écriture en-ligne [Gra+09] et la synthèse d'écriture en-ligne [Gra14] (fig.2.9).



(a) Reconnaissance d'écriture en-ligne.



(b) Synthèse d'écriture en-ligne.

FIGURE 2.9 – Modèle LSTM pour la reconnaissance en-ligne et le problème inverse de synthèse d'écriture en-ligne. Illustration de Greydanus.

Le modèle d'écriture de Graves (figure 2.10) se constitue de trois modèles différents, empilé l'un après l'autre et appris de simultanément avec la rétro propagation du gradient :

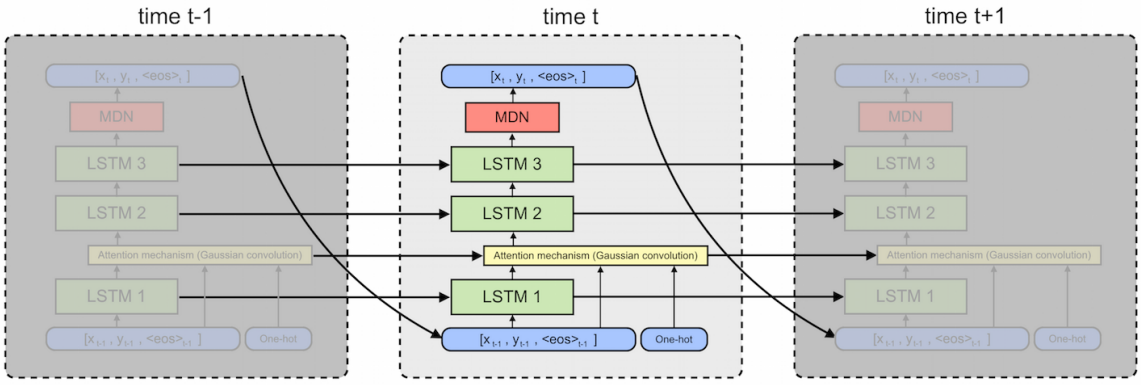


FIGURE 2.10 – Modèle LSTM de synthèses d'écriture en-ligne de [Gra14]. Illustration de Greydanus.

1. Au cœur du modèle de Graves, on retrouve un LSTM avec trois couches cachées, qui permet de modéliser la temporalité du signal à générer.

2. Un mécanisme d'attention est utilisé pour obtenir des informations sur les caractères constituant la phrase à générer. Ce modèle utilise un mécanisme d'attention différentiable, une convolution gaussienne sur un encodage 1 parmi n (*one-hot* en anglais) des caractères de la phrase. Le modèle apprend à déplacer la fenêtre de caractère en caractère pour bien guider la génération.
3. Un réseau de mélange de distributions (*Mixture Density Network*) [Bis94] est un réseau neuronal capable de mesurer sa propre incertitude. Un MDN est un réseau de neurone qui paramétrise des distributions de probabilité de gaussiennes. Ses sorties sont μ, σ, ρ, π pour plusieurs composantes gaussiennes multivariées. Le MDN apprend à capturer la variabilité importante de l'écriture.

Dans la section 2.3.3, nous étudierons des approches basées sur des modèles combinant un CNN et un LSTM pour la reconstruction de la trajectoire du stylo à partir d'image d'écriture hors-ligne.

2.2.3 Transformers

Alors que les LSTM ont été largement utilisés pour la synthèse d'écriture en raison de leur capacité à modéliser des séquences longues et complexes, les Transformers [Vas+17] ont émergé entre temps comme une alternative prometteuse. Les Transformers héritent de l'architecture encodeur-décodeur des modèles *Seq2seq*, leur architecture (figure 2.11) est composée quasi-exclusivement de modules à mécanisme d'attention multitête pour mieux capturer les relations à longue distance entre les éléments d'une séquence. Sans aucune récurrence ou mécanisme de mémoire séquentielle, il s'agit d'une architecture parallélisable contrairement au RNNs. Chaque encodeur se compose de deux parties :

— une couche dite d'auto-attention multitête : elle porte son attention uniquement sur l'ensemble de la séquence source pour encoder un élément précis.
— Une couche de neurones à propagation avant (*feedforward* en anglais) responsable de la transformation non linéaire nécessaire.

Un décodeur possède une couche d'attention supplémentaire par rapport à l'encodeur. C'est la couche d'attention croisée multitête Encoder-Décoder. Elle permet au décodeur d'effectuer une attention entre la séquence d'entrée encodée (mémoire) et la séquence de sortie (à décoder).

Contrairement aux modèles de séquence récurrents, les Transformers n'ont pas de mécanisme de récurrence qui capture naturellement l'ordre des éléments dans une séquence.

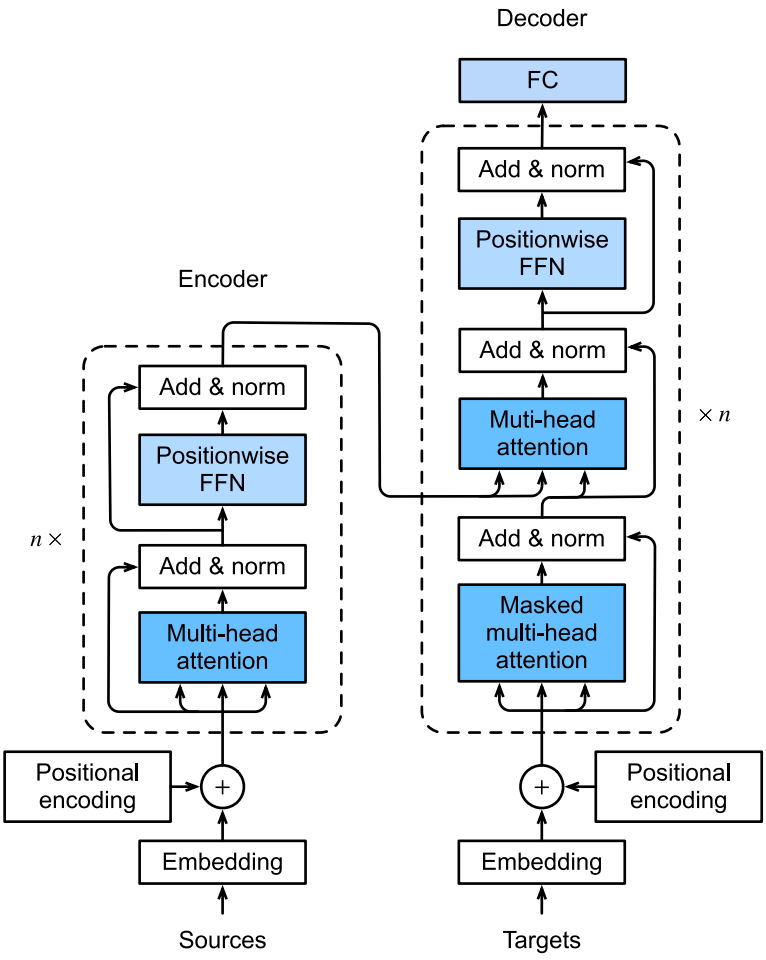


FIGURE 2.11 – Architecture Transformer encodeur-décodeur [Dev+19].

Les encodages de position (*positional encodings* en anglais) sont introduits pour remédier à cette limitation. Dans [Vas+17], un encodage sinusoïdal à différentes fréquences est proposé :

$$\begin{aligned} \mathrm{PE}_{(\mathrm{pos},2i)} &= \sin\left(\mathrm{pos}/10000^{2i/d_{\mathrm{model}}}\right) \\ \mathrm{PE}_{(\mathrm{pos},2i+1)} &= \cos\left(\mathrm{pos}/10000^{2i/d_{\mathrm{model}}}\right) \end{aligned} \tag{2.4}$$

Le composant clé du Transformer, le module d'attention multitête, est défini par l'équation 2.5. Il prend en entrée une matrice de requêtes (**Q**uery) et une paire de matrices clés (**K**ey) et valeurs (**V**alue).

$$\begin{aligned} \mathrm{MultiTête}(Q,K,V) &= \mathrm{Concat}(\mathrm{tête}_1,\ldots,\mathrm{tête}_h)W^O \\ \mathrm{tête}_i &= \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V) \\ \mathrm{Attention}(Q',K',V') &= \mathrm{softmax}\left(\frac{Q'K'^T}{\sqrt{d_k}}\right)V' \end{aligned} \tag{2.5}$$

Avec les projections linaires qui sont les matrices apprenables $W_i^Q \in \mathbb{R}^{d_{\mathrm{model}} \times d_k}, W_i^K \in \mathbb{R}^{d_{\mathrm{model}} \times d_k}, W_i^V \in \mathbb{R}^{d_{\mathrm{model}} \times d_v}$ et $W^O \in \mathbb{R}^{hd_v \times d_{\mathrm{model}}}$, et $\sqrt{d_k}$ une constante de normalisation. La figure 2.12 illustre le module d'attention multitête.

Les Transformers ont montré de meilleures performances dans diverses tâches de modélisation séquentielle en traitement du langage naturel (la traduction automatique, la génération de texte et le résumé automatique) [Dev+19]. Leur capacité à modéliser les relations à longue distance et à capturer les structures hiérarchiques des données en fait un choix attrayant pour la modélisation d'écriture en-ligne. L'étude de [Aks+20] montre une supériorité du Transformer au LSTM pour la génération d'écriture manuscrite en-ligne de texte et de diagrammes. Les Transformer offrent également une alternative puissante aux LSTM pour la reconnaissance de texte manuscrit (HTR), un modèle CNN-Transformer bout-en-bout (sans segmentation en mots, lignes ou paragraphes) pour la reconnaissance de texte manuscrit d'une page entière de document hors-ligne est proposé par [SK21 ; CCP23].

L'approche [CCP23] emploi comme encoder un CNN plus léger ($1.7M$ de paramètres) comparé à l'encoder de [SK21] qui est un *ResNet* de $21.4M$ de paramètres. Un décodeur transformer est utilisé en suite pour transcrire le texte à partir la carte de caractéristique fourni par l'encoder. L'approche [CCP23] est capable de reconnaître la mise en page en plus de du texte à travers la prédiction des *tokens* spéciaux pour de mise en page. La figure 2.13 montre cette architecture CNN-Transformer. La carte de caractéristiques fournie par

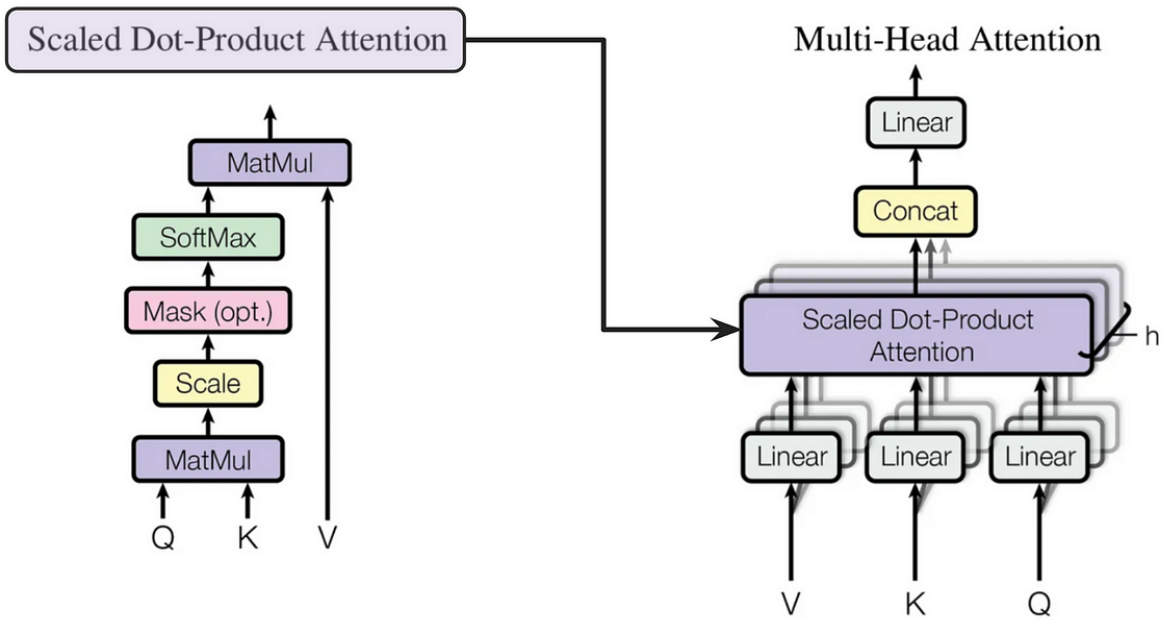


FIGURE 2.12 – (Gauche) Attention avec un produit scalaire normalisé. Un masque est appliqué aux scores d'attentions dans le décodeur pour garantir l'auto-régressivité. (Droite) Un module d'attention multitête consiste en un ensemble de têtes d'attention calculées indépendamment puis agrégées ensemble pour produire une sortie cachée. Illustration par [Vas+17].

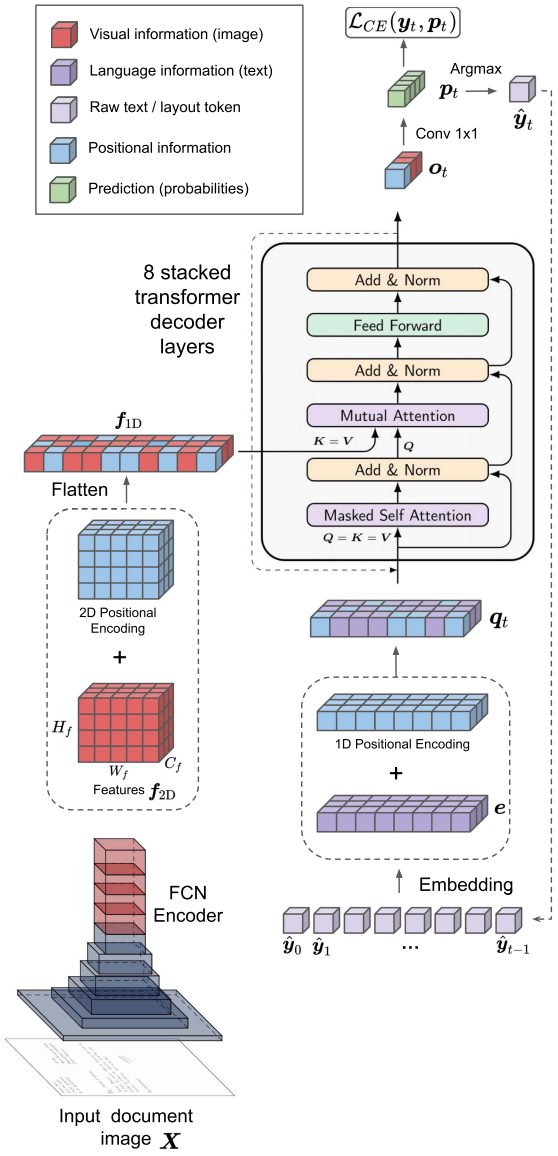


FIGURE 2.13 – Architecture CNN-Transformer pour la reconnaissance de texte manuscrit à l'échelle document. Illustration par [CCP23].

l'encodeur CNN est additionnée avec une carte d'encodages positionnels 2D, qui résulte de la concaténation de deux encodages positionnels 1D 2.4 distincts pour les coordonnées X et Y des pixels dans la carte de caractéristiques.

La prédiction de l'ordre de lecture (cf.2.14) désigne l'extraction de l'ordre des mots constituant un document hors-ligne. Il s'agit d'une tâche importante pour l'extraction de

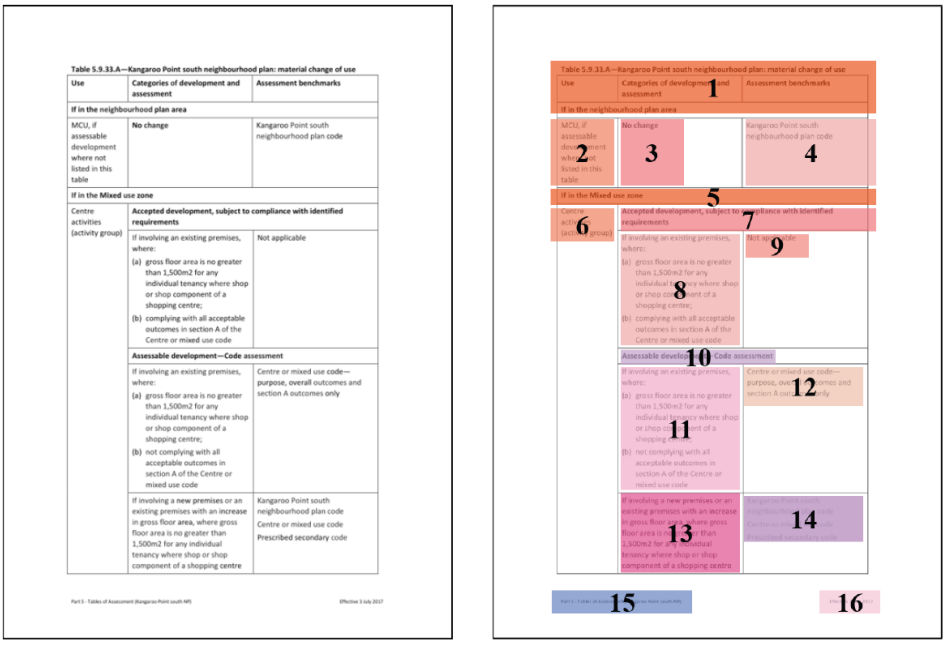


FIGURE 2.14 – Exemple d'ordre de lecture de la base de données [Wan+21].

connaissance dans les documents imprimés (ou PDF). Elle permet d'améliorer l'accessibilité des documents. Pour pouvoir être lu par des systèmes de synthèse vocale (text to speech) ou pour optimiser la lecture pour une variété de taille de résolution d'écran (smartphone, tablette, etc.).

Elle repose sur des techniques combinant du traitement du langage naturel et de la vision par ordinateur pour l'analyse de la mise en page pour améliorer l'ordre des mots d'un document transcrit avec un moteur d'OCR. *LayoutReader* [Wan+21] est un modèle Transformer séquence-à-séquence utilisant à la fois des informations textuelles et de mise en page, il emploi le réseau multi-modale *LayoutLM*[Xu+20] (100M de paramètres) de reconnaissance et de compréhension de document (lui même basé sur BERT) comme encodeur et effectue le re-ordonnancement des mots avec un décodeur produisant une permutation des mots à partir de des sortis cachés de l'encodeur et les embeddings des mots. La figure 2.15 montre l'architecture du réseaux LayoutReader.

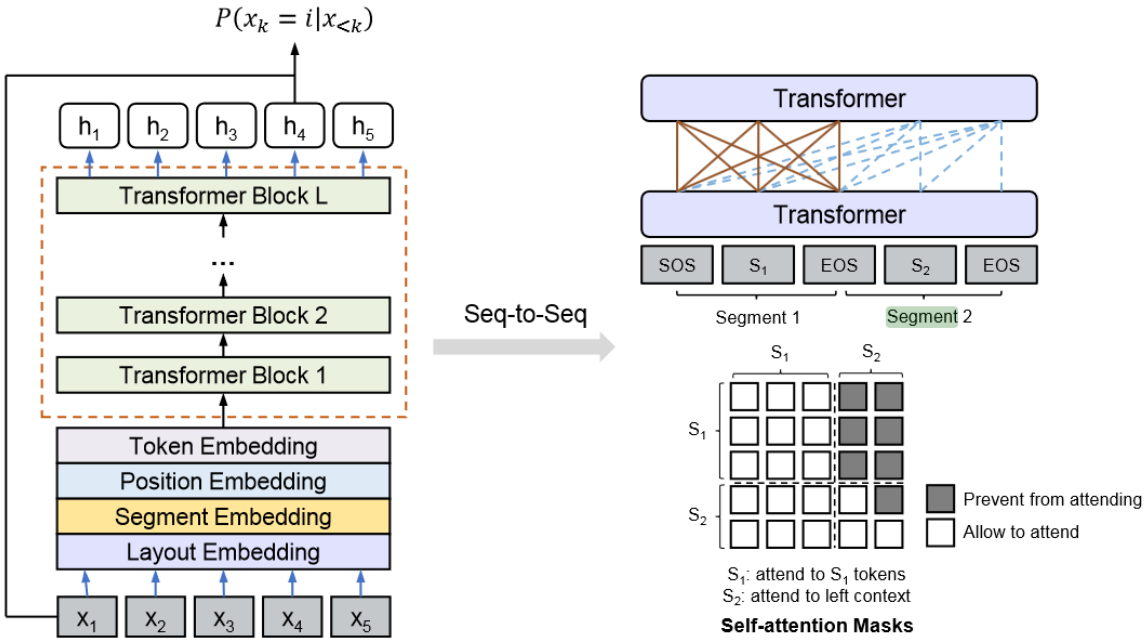


FIGURE 2.15 – Architecture du réseau LayoutReader. Le modèle prend en entrée la séquence de mots et leur mise en page (en segment) obtenu d'un système OCR. Le décodage s'effectue par segment de texte (ligne), ici segment 1 désigne la ligne source produite par un système d'OCR et segment 2 la ligne cible ré-ordonné. Les masques d'attentions habituellement sont employés pour assurer une auto-régressivité.

LayoutReader atteint un score BLEU de 0.9819 surpassant significativement les approches heuristiques de tris gauche à droite et de haut en bas. Bien que cette tâche ajoute une temporalité aux documents imprimés, sous forme de séquence ordonnée de mots, dans cette thèse, nous nous intéressons à l'extraction des traits et à leur ordonnancement dans les documents manuscrit hors-ligne.

2.3 Travaux existants

Dans cette section, nous explorerons l'état de l'art pour la reconstruction de trajectoire de stylo à partir d'une image d'écriture hors-ligne. Nous verrons d'abords que les différentes approches heuristiques peuvent se classer en deux familles : les méthodes globales d'optimisation de parcours de graphe sur des critères empiriques (section 2.3.1) et les méthodes locales de résolution des ambiguïtés (section 2.3.2). Ensuite nous verrons quelques applications récentes des réseaux de neurones à notre tâche dans la section 2.3.3.

2.3.1 Squelettisation et approche globale d'optimisation de parcours graphe

Dans la littérature, ils existent deux familles d'approches pour la reconstruction de la trajectoire du stylo à partir d'une image hors-ligne en niveau de gris. Ces deux familles d'approches divergent sur une question essentielle : faut-il exploiter directement les indices visuelles de l'image originale en niveau de gris, ou alors travailler sur un squelette de l'image hors-ligne binarisé ?

La première famille d'approches [DR95] propose d'analyser directement l'image en niveau de gris (scan/photo) afin extraire des caractéristiques visuelles fournissant des indices quant à la trajectoire du stylo (comme un changement d'épaisseurs, l'intensité de l'encre, etc.). Ces caractéristiques sont absentes ou altérées par les algorithmes de squelettisation.

La deuxième famille d'approches effectue une analyse au niveau du squelette, l'image est d'abord binarisée puis squelettisée. La squelettisation est une opération morphologique consistant à réduire l'épaisseur des traits d'une image d'écriture hors-ligne binaire en une épaisseur d'un pixel. Cela permet notamment de s'abstraire de la nature du support et d'outil d'inscription employé (papier-crayon, tableau blanc-feutre, etc) pour l'écriture et d'extraire une représentation en graphe. Cette représentation est la plus courante dans la

littérature, pour réduire le problème en une optimisation plus traditionnelle de graphes. Cette optimisation se base souvent sur des critères empiriques de pondération des arêtes tels que la longueur, la courbure, etc.

Cette deuxième famille d'approches [Boc+93 ; GWf92 ; Jag96] agit sous une présupposition de bonne continuité (angles, directions et épaisseurs) des segments du tracé chez les humains qui permet de maintenir une fluidité et une régularité de l'écriture. Cette présupposition est censée refléter la bio-mécanique humaine lors de l'écriture [VG91] qui est efficient en énergie. En règle générale, on doit éviter les changements abrupts de direction, orientation lors de l'écriture.

Par exemple, [Jag96] formule la reconstruction de la trajectoire du stylo à partir d'image de mot comme étant un problème du voyageur de commerce, en cherchant à déterminer le parcours de graphe ayant la courbure totale minimale. Les hypothèses de points de départ et de fin de mots sont basées sur le sens gauche à droite de l'écriture. Cette optimisation est effectuée au niveau des lettres, en découpant l'image de mot au niveau des ligatures, pour éviter une explosion combinatoire lié à cette optimisation NP-difficile.

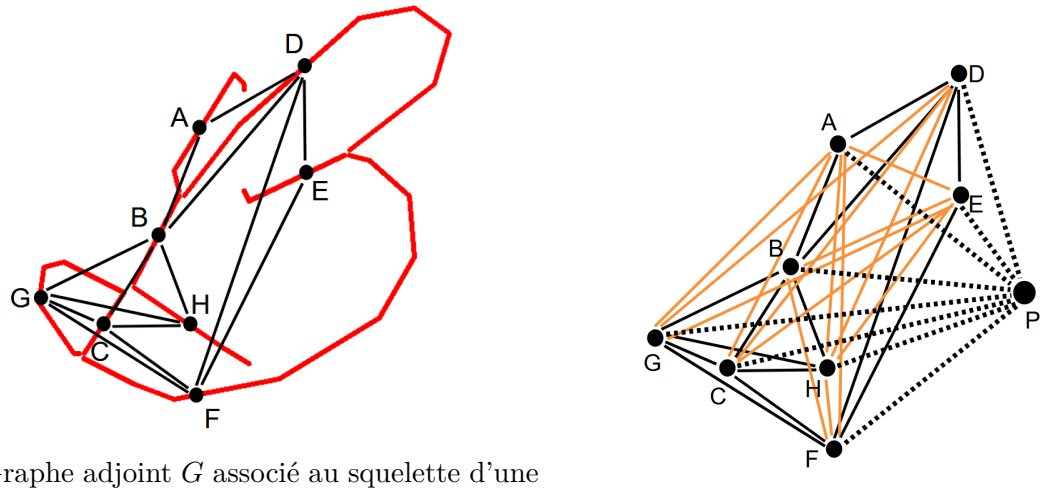
D'abord, le squelette de l'image hors-ligne du mot est obtenu à l'aide de l'algorithme d'amincissement par contour [Kwo88] : Cet algorithme supprime itérativement plusieurs couches de contour de l'écriture jusqu'à l'obtention d'un contour de largeur unitaire d'un seul pixel. Un graphe de squelette est ensuite construit : les arcs sont les zones d'intersections et les nœuds correspondent aux autres segments d'épaisseur un. Dans ce graphe pondéré G, les nœuds correspondent à une zone contiguë formée par l'ensemble des pixels d'une jonction et un arc existe entre deux nœuds si les deux jonctions associées forment les deux extrémités d'un segment du squelette. Soit A et B deux sommets du graphe G, l'arc AB porte une pondération associée à la fluidité de la transition du segment A à B, égale à l'angle entre les deux segments.

Notre problème d'estimation de la trajectoire du stylo peut être formulé comme suivant : sachant un sommet de départ et de fin, on cherche à déterminer le chemin de coût minimal traversant chaque sommet de G au moins une fois. Il s'agit presque d'un problème de recherche du chemin hamiltonien le plus court dans un graphe (voyageur du commerce).

Cependant, pour les rebroussements nécessitant deux parcours dans deux sens opposés du même sommet du graphe, les sommets associés doivent donc être traversés deux fois et donc le chemin à déterminer n'est donc plus un chemin hamiltonien. Une réduction de ce problème au problème du voyageur est montrée par [Jäg98]. Cette réduction s'appuie

sur la complétion du graphe G à un graphe complet G_K (cf. 2.16). Les nouveaux arcs sont associés au plus court chemin entre deux nœuds donnés dans le graphe G. Ainsi, il est possible de parcourir plusieurs fois un sommet du squelette par un chemin hamiltonien dans G_K.

Un algorithme par séparation et évaluation, méthode très utilisée pour résoudre les problèmes NP-complets, permet de déterminer le chemin hamiltonien de coût minimal dans G_K.



(a) Graphe adjoint G associé au squelette d'une lettre "B".

(b) Graphe complet G_K.

FIGURE 2.16 – (Gauche) Graphe construit pour une image de la lettre "B". (Droite) Complétion du graphe en graphe complet. Illustration par [Jäg98].

Pour éviter une explosion combinatoire, le graphe G est subdivisé en plusieurs composantes connexes indépendantes de taille modérée. La reconstruction de la trajectoire est opérée de manière séparée sur chaque composante. Cette décomposition est effectuée au niveau des ligatures à l'aide des cartes de contours [SW92]. Le calcul des angles et donc la pondération du graphe est effectuée avec des approximations linéaires des segments des squelettes combinés avec l'information de l'épaisseur des traits pour corriger les potentielles erreurs dans le squelette. Les différentes composantes du graphe sont ensuite triées horizontalement de gauche à droite.

Cette approche obtient un taux de reconnaissance des signaux en-ligne reconstruit de 73% sur une base de 6934 images de mots monotraits synthétiques générées à partir du signal en-ligne avec une épaisseur de 3 pixels. Cette approche présente cependant plusieurs

inconvénients :

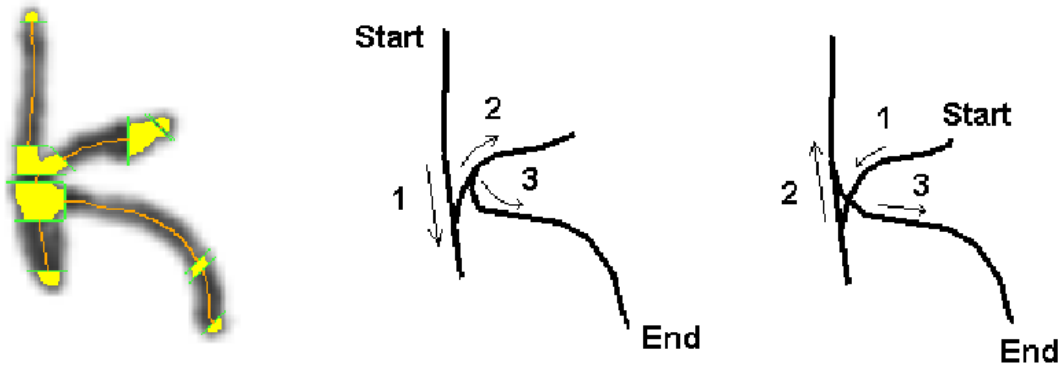


FIGURE 2.17 – La bonne trajectoire de stylo ne minimise pas toujours la courbure totale. (gauche) découpage en graphe de la lettre "k", (milieu) parcours correct et (droite) parcours minimisant la courbure totale. Illustration par [RAC05].

— Le lever et poser de stylo ne sont pas prédit et limite son application aux mots mono-traits.
— La complexité temporelle de l'optimisation est un frein à son application à des lettres à plusieurs traits.
— L'hypothèse de minimisation de courbature n'est pas toujours vraie, comme Illustré par la figure 2.17

La technique décrite dans [LAL99] repose également sur la réduction de la courbure globale. Elle est capable de traiter des images multitraits. En proposant plusieurs décompositions du graphe, un coût supplémentaire est associé aux levers de crayon pour prendre en compte le choix de décomposition du graphe dans le calcul du meilleur parcours. Sur les lettres isolées, les résultats de reconnaissance sont de $86,1\%$ sur 260 images de lettres et 89.8% pour 197 images de mots.

Les méthodes de [QNY06 ; KY00 ; QY04] reposent également sur la recherche du parcours présentant la courbure minimale dans un graphe. Ce parcours de courbure minimale est obtenu en définissant et en résolvant un problème d'optimisation de chemin eulérien. Leur approche s'applique exclusivement sur les tracés mono-trait. Dans les travaux de [RAC05] cette technique a été étendue sur des lettres isolées multitraits.

2.3.2 Résolutions locales des ambiguïtés

Dans les méthodes locales de reconstruction du signal en-ligne, le chemin à suivre par le stylo est décidé localement au niveau de chaque jonction à l'aide d'une analyse de la configuration locale la jonction et de l'historique des mouvements précédents du stylo [NB10 ; NK17]. L'avantage des méthodes locales réside dans leur faible coût de calcul comparé au coût parfais exponentiel des approches globales, il peut cependant être difficile de concevoir des règles heuristiques générales qui puissent être appliquées à différents styles d'écriture.

Chan et al. [Cha20] présentent une approche hybride combinant une analyse locale pour simplifier une optimisation globale de graphe. Elle consiste d'abord à diviser le squelette d'une image binarisée en jonctions et segments. Les étapes clés de leur système proposé pour l'extraction de traits à partir d'image d'expressions mathématiques manuscrites hors-ligne synthétiques sont les suivantes :

1. Binarisation et squelettisation.
2. Constructions du graphe de segments et jonctions.
3. Simplifications et corrections du graphe.
4. Fusion des segments et détection de la direction pour former des traits : Initialement chaque segment forme un candidat de trait. L'optimisation repose deux critères : la minimisation de l'angle entre deux segments ainsi que sur la minimisation du nombre total de traits.
5. Ordonnancement des traits : tout d'abord, les traits sont regroupés par découpage X-Y, les groupes sont ensuite ordonnés de gauche à droite, puis de haut en bas. L'ordre des traits intra-groupe est ensuite établi en effectuant un tri horizontal et vertical (voir figure 2.18).

Le critère de bonne continuité ne conduit pas toujours au bon choix d'appariements entre arcs, car les distorsions du squelette au niveau des intersections faussent la notion de bonne continuité. Cette approche tend aussi à produire beaucoup de traits, le critère de minimisation du nombre de traits n'est pas suffisant. Enfin, l'ordonnancement basique des traits de gauche à droite puis de haut en bas peut mener à des incohérences au niveau des traits intra-symboles.

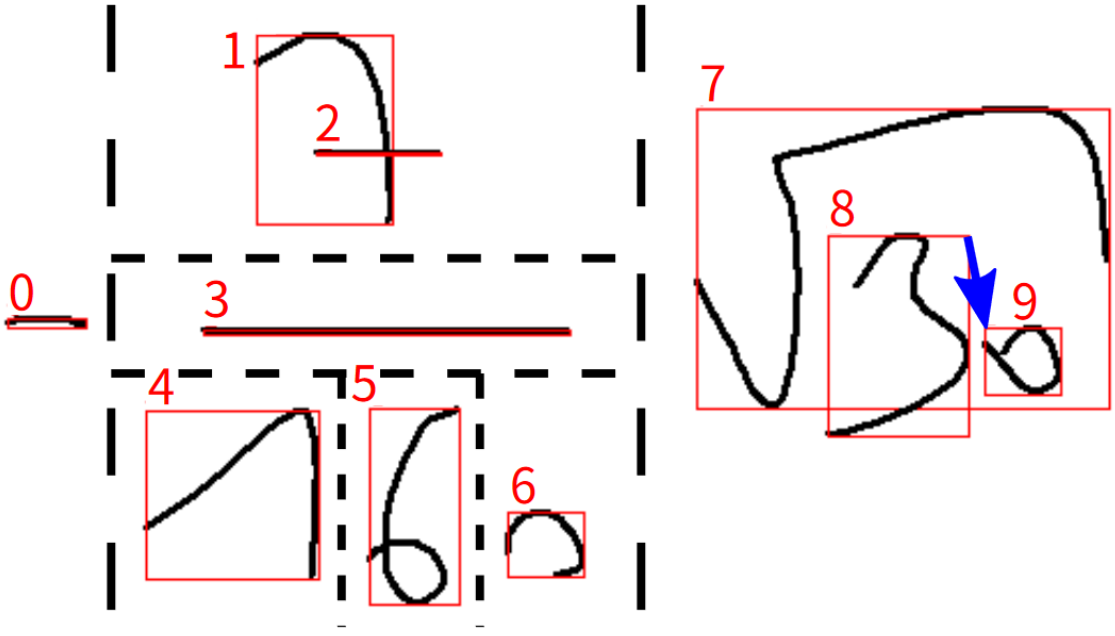


FIGURE 2.18 – Découpage X-Y des traits et ordonnancement des traits. Illustration par [Cha20].

2.3.3 Approche à réseaux de neurones

Avec l'émergence des méthodes d'apprentissage profond, de nouvelles approches neuronales pour la reconstruction de trajectoires de stylo ont récemment été proposées dans l'état de l'art. Alors que les méthodes heuristiques reposent sur des règles et des heuristiques conçues manuellement, les réseaux de neurones ont la capacité d'apprendre directement à partir des données afin de développer leur propre représentation et leurs propres critères pour modéliser les mouvements du stylo.

Modèle au niveau caractère

Bhunia et al [Bhu+18] propose une première approche bout-en-bout avec un modèle CNN-LSTM. Le modèle prédit de manière séquentielle les points de coordonnées de trajectoire du stylo à partir d'images de caractères hors-ligne. Il comporte deux étapes : l'extraction d'une séquence de vecteurs caractéristiques à partir de l'image hors-ligne à l'aide d'un réseau de neurones convolutifs. Ensuite, cette séquence est traitée par un réseau LSTM encodeur-décodeur qui régresse séquentiellement les points du signal d'écriture

en-ligne.

Les images de caractères manuscrits de taille 64×64 sont utilisées comme entrée du réseau CNN. Le réseau est composé de six couches de convolution avec un filtre de taille 3×3 empilées pour apprendre une représentation profonde des caractéristiques, suivi par un *maxpooling* de 1×2 pour obtenir la carte de caractéristiques. Ensuite, une séquence de vecteurs de caractéristiques est extraite à partir des cartes de caractéristiques produites par le CNN, qui constituent l'entrée des couches récurrentes. Chaque colonne, de largeur d'un pixel, en partant de la gauche à la droite de la carte de caractéristiques est associé à un vecteur de la séquence caractéristique. Chaque colonne de la carte de caractéristiques correspond à une région rectangulaire de l'image hors-ligne. Et donc chaque vecteur de la séquence de caractéristiques est associé à un vecteur descripteur de l'image pour cette région (cf. fig2.19).

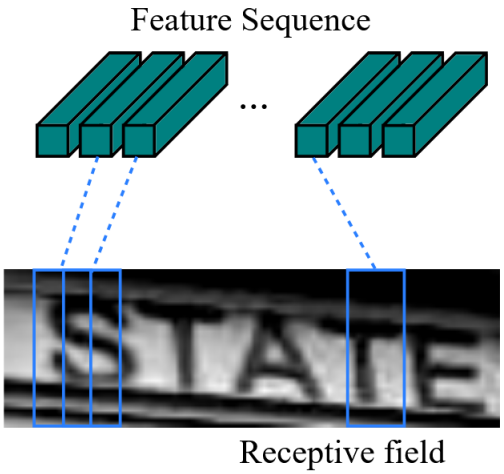


FIGURE 2.19 – Zone de l'image associée au champ récepteur d'un vecteur de la séquence de caractéristiques.

La séquence de vecteurs caractéristiques extraite est employée pour obtenir une séquence de point décrivant la trajectoire de stylo à l'aide d'un réseau encodeur-décodeur LSTM. Il est utile d'avoir une séquence caractéristique plus longue pour améliorer les performances du réseau LSTM encodeur-décodeur. En conséquence, la couche de *max-pooling* utilise une fenêtre rectangulaire de taille 1×2 au lieu d'une fenêtre carrée plus conventionnelle. Cette modification permet d'obtenir des cartes de caractéristiques avec une largeur plus grande, ce qui se traduit par une séquence de vecteurs caractéristiques plus longue.

L'encodeur du LSTM est alimenté avec la séquence de vecteurs caractéristiques et son

état caché obtenu à la fin de la propagation avant de cette séquence est utilisé comme un vecteur sorti, de taille fixe, représentant toute l'image. L'état caché du décodeur LSTM est initialisé avec ce vecteur descripteur de l'image sorti par l'encodeur. La sortie du décodeur, est conditionnée par les coordonnées du point précédent. Un MLP est employé pour générer les coordonnées x, y à partir des sorties. Le modèle est appris avec une fonction de coût basé sur une distance $L1$ entre la séquence de points prédit et la séquence vérité. Le modèle montre une meilleure performance sur des lettres mono-traits de script indiens comparé à l'approche heuristiques de [KY00]. Cependant, un travail important reste à surmonter pour modéliser du contenu multi-trait plus complexe.

Zhao et al[ZYT18] proposent de considérer le problème de reconstruction de trajectoire de stylo comme un problème de correspondance d'image à séquence guidé par un modèle CNN modélisant une transition du stylo. En inférence, un algorithme alimentant le CNN itérativement avec ses prédictions de mouvement de stylo permet de reconstruire le tracé du stylo (voir fig.2.20). L'approche est appliquée aux symboles chinois hors-ligne isolés. Le CNN prend en entrée une image de taille fixe 28×28 du squelette d'un sym-

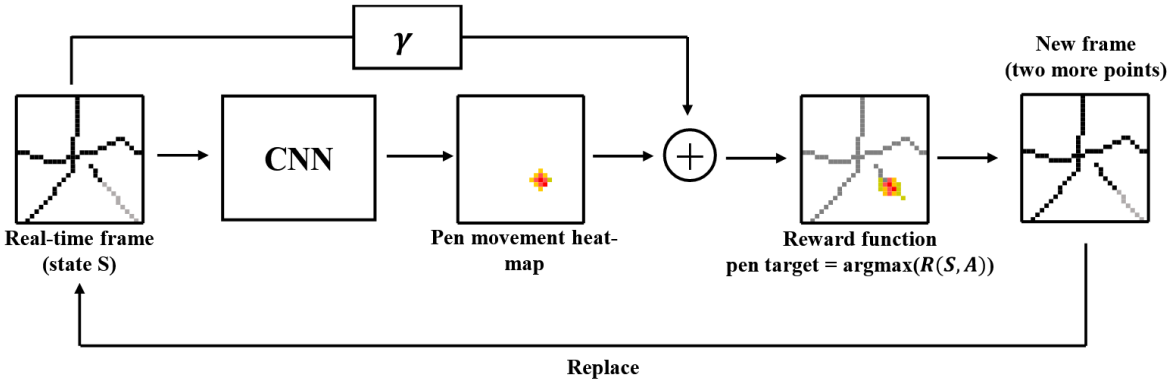


FIGURE 2.20 – Modèle CNN et algorithme d'inférence itérative.

bole hors-ligne ainsi que deux autres images contenant le tracé effectué à l'état courant et à l'état précédent. Le réseau produit en sortie une distribution de probabilité de la prochaine position du stylo sur une carte de 28×28. Le réseau comprend trois couches de convolution et quatre couches entièrement connectées. La couche de convolution initiale utilise 128 filtres de taille 2×2 avec un pas de 1, suivis d'une couche de *maxpooling* de 2×2 avec un pas de 1. La deuxième couche de convolution utilise 64 filtres de 2×2 avec un pas de 1, suivis d'une couche de *maxpooling* de 2×2 avec un pas de 1. La troisième couche de convolution utilise 32 filtres de 2×2 avec un pas de 1. Cette dernière couche de

convolution est suivie de trois couches entièrement connectées avec respectivement 4096, 2048 et 1024 neurones. La couche de sortie est également entièrement connectée avec 784 sorties correspond à une carte de 28×28.

L'apprentissage s'effectue avec la fonction de coût de l'erreur quadratique moyenne. Les images vérités des prochaines positions de stylo sont lissées avec une gaussienne centrée autour des coordonnées du stylo pour mieux capturer l'incertitude dans la régression. Cependant, ces méthodes ([Bhu+18 ; ZYT18]) présentent quelques inconvénients majeurs :

— la complexité du modèle dépend directement de la résolution de l'image hors-ligne. Une petite résolution (28×28) peut suffire pour des applications au niveau caractères, mais rendra des images illisibles dans le cas de contenus plus large tels que des équations mathématiques. Si l'on augmente la taille des images durant l'apprentissage cela conduira à une augmentation quadratique du nombre total de paramètres dans le modèle, due aux couches entièrement connectées.
— Les poser et lever de stylos ne sont pas prédits. Ils constituent une information indispensable pour définir une écriture en-ligne.
— L'approche suppose que le squelette vérité de l'image hors-ligne est connu. Cependant, les algorithmes de squelettisation créent souvent des distorsions et bruits non souhaitables. Le comportement du CNN sur des squelettes inférés bruités n'est pas discuté.

La méthode proposée par **Archibald et al.** [Arc+21] étend l'approche *CNN-LSTM* de [Bhu+18] en introduisant deux solutions supplémentaires :

— Une architecture modifiée pour prendre en entrée des lignes, des images de taille plus larges et variable, ainsi que la prédiction des posers et levers du stylo et une régression des coordonnées relatives du stylo au lieu des coordonnées absolus.
— Un apprentissage adaptatif avec une fonction de coût basé sur une distance DTW afin de mieux aligner les trajectoires prédites et vérités. Elle permet une invariance aux permutations de paire de traits consécutives, considéré ici subjective et problématique à l'apprentissage.

Les images hors-lignes sont redimensionnées pour avoir une hauteur de 60 pixels puis un CNN significativement plus profonds avec 11 couches de convolutions fournit une séquence de vecteurs caractéristiques. Les vecteurs sont de tailles 1024 ici et sont ensuite traité par une Bi-LSTM de deux couches suivies par une convolution 1D pour obtenir quatre sorties : les coordonnées relatives (dx, dy), une probabilité associée au début d'un trait (sos) et une probabilité de fin de la trajectoire (eos) (voir fig2.21). La distance de Manhattan

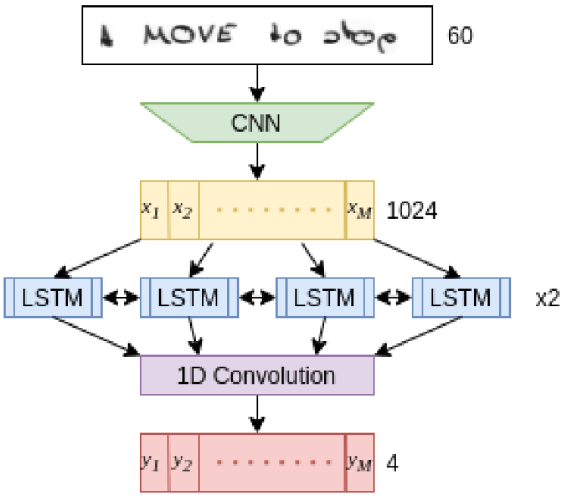


FIGURE 2.21 – Modèle CNN-BiLSTM

proposé par [Bhu+18] accorde une importante pénalisation à la différence de fréquence d'échantillonnage entre les points prédit et les points de référence même si le trait prédit couvre correctement l'image. De plus, le réseau LSTM permet une mise en correspondance de deux séquences entre deux séquences de longueurs différentes. D'un côté, en entrée la séquence de vecteurs caractéristiques, extraite par le CNN, de longueurs proportionnelles à la largeur de l'image. La séquence de points prédite par le réseau est donc de la même taille que la séquence de vecteurs caractéristiques. De l'autre côté, la séquence de points de la trajectoire de référence peut avoir des longueurs différentes pour une même résolution d'image, en fonction des symboles écrit dans l'image. En conséquence, une fonction de coût bijective point à point n'est pas toujours envisageable.

La distance DTW [SC78] est utilisée ici à la place de la distance de Manhattan pour aligner dans le temps les points prédits et les points de référence (GT) de manière plus précise et flexible en autorisant des décalages temporels. DTW est combiné avec un algorithme adaptatif de la vérité en fonction des prédictions du modèle. En effet, la variabilité naturelle dans l'ordre et direction des traits est parfois invisible dans l'image hors-ligne : c'est le cas pour certains traits avec un ordre interchangeable ("i"), les traits retardés (accents, points, etc.) et les traits de direction inversible (barre de t). Cette variabilité constitue un bruit qui pose un défi pour l'apprentissage des réseaux de neurones profonds, car la fonction de coût fourni un retour contradictoire dans le cas où l'ordre des traits est subjective.

Pour pallier à ce problème et mieux prendre en compte les traits aberrants (ordre ou direction inversé) dans l'écriture, les trajectoires vérités sont mises à jour avec une des deux opérations suivantes : inversion de la direction d'un trait ou la permutation de deux traits adjacents. La mise à jour ayant la distance DTW la plus faible avec la prédiction du modèle est considérée comme une nouvelle vérité sans traits aberrants et donc mieux adapté. Le jeu de données *IAM-OnDB* phrases [MB02] est utilisé pour l'apprentissage du réseau de neurones. L'apprentissage est enrichi avec 200000 lignes synthétiques supplémentaires générées à l'aide du réseau neuronal générateur de Graves et al. [Gra14].

Le système se démarque en tant que première méthode à aborder notre problème au niveau de la ligne de texte et montre des résultats globalement pertinents. Plus précisément, le système montre de bonne performance dans le cas d'une écriture bien soignée et avec des mots bien espacés. Cependant, il éprouve des difficultés à traiter la ponctuation, ainsi que les traits isolés (comme le point du "i" ou la barre dans du "t") ou autres traits retardés (comme les accents). De plus, certains traits très proches des bordures de l'image peuvent parfois être omis.

2.4 Conclusion

Dans ces sections, nous avons exploré l'état de l'art pour la reconstruction de signal d'écriture en-ligne à partir d'image d'écriture hors-ligne. Il existe deux familles d'approches heuristiques : les approches globales et les approches locales.

Les approches globales résolvent le problème à travers une optimisation de parcours de graphe de squelette de cout minimal sur des critères comme la courbure et la longueur. Malgré leur capacité à produire des résultats supérieurs aux approches locales, leur temps d'exécution très important, due à l'explosion combinatoire de leur exploration, constitue un frein à leur utilisation. Leurs critères empiriques d'optimisation ne sont pas toujours vérifiés et peuvent donc mener à une estimation grossière du parcours optimal. Les approches locales relient les arcs aux intersections en se basant uniquement sur des critères locaux. Ils sont moins couteux à mettre en œuvre. Néanmoins, ils ne sont pas toujours fiables par manque de contexte global.

Les approches neuronales basées sur des CNN ainsi que des réseaux récurrents LSTM ont montrés des meilleurs résultats et ont permis pour la première fois d'étendre la reconstruction de signal à des images de ligne entière de texte. Cependant, la reconstruction produite n'est pas toujours fidèle et ne couvre pas souvent entièrement le squelette de

l'écriture. Nous avons également observé des limites dans les prédictions de levers de stylo et des incohérences dans les résolutions de jonctions moins triviale.

Le tableau 2.1 montre une comparaison entre les différentes approches de l'état de l'art. L'absence d'un protocole d'évaluation standardisé rend difficile cette comparaison. Les différentes bases de données publiques et privées et les métriques d'évaluations utilisées dans la littérature rend compliqué l'établissement d'un comparatif.

Auteur	**Année**	**Méthode**	**Échelle**	**Expérimentation**	**Résultat**
Boccignon et al. [Boc+93]	1993	Local + Squelette	Lettre	10,000 lettres par 20 scripteurs	97.00%
Jager et al [Jäg98]	1998	Globale + Squelette	Numéro	NIST[Wil+92]	73.00% ♠
Plamondon and Privitera [PP99]	1999	Local + contour	Mono-trait	6934 mots mono-traits	89.00%
Lallican et al. [LAL99]	2000	Contour + Graphe	Mot	IRONOFF	80.00%
L'Homer [L'H00]	2000	Contour	Lettre	520 lettres de NIST	90.00%
Kato and Yasuhara [KY00]	2000	Squelette	Mono-trait	100 mots/lettre mono-traits.	91.60%
Doermann et al. [Doe+02]	2002	Contour	Boucle	1270 mots par 5 scripteurs	84.00%
El Baati et al. [EB+05]	2005	Contour	Mot	50 mots arabes par 2 scripteurs	83.71% ♠
Niels et al. [NDPH05]	2006	Squelette	Lettre	UNIPEN corpus trait d'épaisseur 1 pixel, 3370 images, les squelettes erronés sont supprimés	86.00%
Qiao et al. [QNY06]	2006	Squelette	Lettre	UNIPEN épaisseur de 3 pixels	96.00%
Rousseau et al. [RCA06]	2006	Squelette	Lettre	IRONOFF	59.00% ♠
Bhunua et al. [Bhu+18]	2018	CNN-BLSTM	Lettre	Lettres Devanagari LIPI 20K images d'apprentissages 8K de test	95.90%
Zhao et al. [ZYT18]	2018	CNN	Idéogramme	Idéogrammes chinois de CASIA et lettres isolées de Unipen	78.3%
Chan et al. [Cha20]	2020	Squelette	Expressions mathématiques	10K images de CROHME épaisseur de 1.	65% ♠
Archibald et al. [Arc+21]	2021	CNN-BLSTM	phrases	10K lignes de IAM et 200K lignes générées artificiellement (Graves et al.)	0.024 ♦

TABLE 2.1 – Différent travaux de reconstruction de trajectoire de stylo à partir d'image et leurs résultats[NB10]. Les résultats fournis sont basés sur le taux de reconstruction correcte, le taux de reconnaissance du signal en-ligne prédit ♠ ou alors une DTW ♦.

Chapitre 3

Contributions

Ce chapitre présente les différentes contributions de la thèse, dans la section 3.1 nous situerons notre contexte d'application, nous nous concentrerons sur la généralisation de l'approche neuronale aux échelles plus larges de phrases et d'expressions mathématiques. Nous présenterons les bases de données en-lignes considérées ainsi que le protocole d'évaluation proposé avec une métrique d'alignement entre un signal en-ligne reconstruit et le signal vérité, mieux adaptée aux variations de fréquences d'échantillonnage.

Dans la section 3.2, nous proposerons une première approche avec un réseau de neurones entièrement convolutif multitâche basé sur le modèle de [ZYT18], capable de traiter des images d'expressions hors-ligne de taille arbitrairement large. Nous montrerons l'avantage du modèle pour la squelettisation et les limites du paradigme basé sur la séquence de caractéristiques purement visuelles.

Dans la section 3.3, nous présenterons une méthode originale avec un paradigme alternatif basé sur l'information positionnelle des différents segments de l'image hors-ligne. Ce paradigme sera implémenté avec un réseau Transformer. Nous montrerons que cette méthode surpasse plusieurs limitations de l'état de l'art sur les phrases cursives et les expressions mathématiques.

3.1 Données et Métriques

Dans cette section nous présenterons les bases de données nécessaires pour l'apprentissage de deux approches à réseaux de neurones ainsi que le protocole de génération d'image d'écriture synthétique avec une épaisseur variable.

3.1.1 Données et pré-traitements

Dans cette thèse, nous portons notre travail sur la généralisation des approches à réseaux de neurones au-delà de l'échelle de lettre isolée [Bhu+18 ; ZYT18] vers les échelles

phrases et formules mathématiques. Pour cela, nous proposerons d'employer des bases de données d'écriture manuscrite en-ligne à différentes échelles (lettres, mots, phrases et expressions) pour générer synthétiquement des images hors-lignes.

Lettre isolée

La base de données UNIPEN [Guy+94] comprend essentiellement divers benchmarks d'entraînement d'évaluation de systèmes de reconnaissance de lettre en-lignes (avec quelques mots). Seuls les ensembles de lettres sont sélectionnés pour notre étude au niveau lettre. Les ensembles associés, utilisés sont 1a (16000 chiffres isolés), 1b (28000 lettres majuscules isolées) et 1c (61000 lettres minuscules isolés). Il s'agit de l'une des premières bases d'écriture en-ligne jamais collectée pour l'apprentissage de systèmes de reconnaissances en-ligne. Cette base a servi comme base d'évaluations de plusieurs méthodes de reconstruction de trajectoires de stylo, dont l'approche CNN de Zhao et al. [ZYT18]. Ils sélectionnent *10000* lettres et chiffres pour l'apprentissage et l'évaluation de leur modèle.

Mots et phrases

La base de données IRONOFF [VG+99] contient 32000 lettres isolées et 50000 mots cursifs collectés dans les deux modalités en-ligne et hors-lignes et écrits par des scripteurs français. Il s'agit de la seule base de données qui fournit des paires de vraies images d'écriture hors-ligne et le signal d'écriture en-ligne correspondant. Cependant, le désalignement inévitable et observé dans cet appariement freine tout apprentissage supervisé.

IAM-OnDB [LB05], est une large base de données publique de phrases manuscrites en-ligne. Elle a été acquise via une interface électronique à partir d'un tableau blanc. La base de données contient 13049 lignes, avec 86272 mots provenant d'un dictionnaire de 11059 mots, écrits par plus de 200 scripteurs. Les auteurs fournissent aussi la rastérisation des signaux en-ligne en image hors-ligne binaire avec une épaisseur de trait variée d'une image à l'autre.

Expressions mathématiques

La base de données *CROHME*[Mah+19], est le jeu de donnée de référence pour l'évaluation de systèmes de reconnaissance d'expressions mathématiques en-ligne et hors-ligne. La base de données inclut 12178 formules avec un vocabulaire de 101 symboles. En plus de la représentation *MathML* de l'expression, la correspondance entre les symboles et les

Base de données	**Quantité**	**Apprentissage**	**Validation**	**Test**
UNIPEN	traits	12k	722	2K
	lettres	8K	500	2K
IRONOFF	traits	153K	10K	69K
	lettres isolées	18K	1K	8K
	mots	25K	3K	11K
	total	44K	4K	19K
IAM-OnDB	traits	172K	32K	96K
	lettres	164K	31K	89K
	mots	37K	7K	20K
	phrases	6K	1K	3K
CROHME19	traits	137K	13K	17K
	symboles	180K	9K	12K
	formules	10K	0.9K	1K

TABLE 3.1 – Bases de données publiques considérées pour l'apprentissage et l'évaluation des deux approches neuronales proposées dans cette thèse.

traits du signal est également fournie. Les formules en-ligne ont été rastérisées en image hors-ligne synthétique en niveau de gris à une résolution de 1000×1000 avec une épaisseur d'un pixel pour la tâche de reconnaissance de formules manuscrites hors-ligne. [Cha20] applique son système de reconstruction de signal d'écriture en-ligne sur cette base hors-ligne et emploie un moteur de reconnaissance en-ligne pour obtenir un taux de reconnaissance d'expression de 65.22%. Notons que depuis ces travaux, la base CROHME2023 [Xie+23] contient des expressions enregistrées dans les deux modalités, mais souffrant des mêmes difficultés d'alignement que la base multimodale IRONOFF[VG+99].

Nous appliquons les mêmes étapes de prétraitement sur l'ensemble des jeux de données. Les étapes de prétraitement suivantes sont appliquées aux données en ligne :

— Normalisation spatiale de l'écriture : les signaux en-ligne des différentes bases sont acquis avec différentes résolutions avec des variations inter-scripteur dans la taille d'écriture. Pour réduire la variation de taille d'écriture, nous normalisons le signal en-ligne en le redimensionnant pour avoir une taille de trait moyenne fixe T_{moy}. La taille d'un trait est la diagonale de sa boîte englobante en pixel. Ensuite la taille moyenne des traits t_{moy} du signal est calculé, le ratio $\frac{t_{\text{moy}}}{T_{\text{moy}}}$ est utilisé pour redimensionner les points du signal. T_{moy} est fixé à 100 pour nos expériences.

— Rastérisation : Nous transformons les signaux en ligne en images en niveaux de

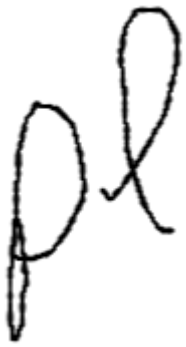


FIGURE 3.1 – Image hors-ligne synthétique de dimensions 100×180.

gris en le dessinant avec une épaisseur de trait aléatoirement choisie pour chacun des points entre 2 et 3 pixels.

La figure 3.1 présente le même mot en en-ligne après normalisation et rastérisation.

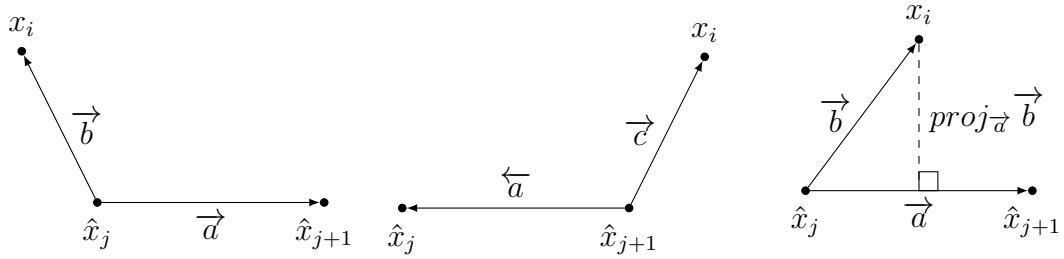
3.1.2 Métrique invariante à la fréquence d'échantillonnage avec DTW-seg

Dans la section 2.1.2, nous avons présenté deux métriques couramment utilisées pour évaluer la fidélité d'un signal en-ligne reconstruit à l'aide d'un système de conversion d'écriture hors-ligne en signal en-ligne, vis-à-vis du signal vérité produit par un humain : la *RMSE* et *DTW*. La DTW permet de trouver une correspondance optimale entre les points des deux séquences temporelles, même lorsqu'elles ont des longueurs différentes ou si elles sont décalées dans le temps. Cependant, les variations de fréquence d'échantillonnage peuvent introduire des coûts supplémentaires qui ne sont pas liés aux caractéristiques réelles de l'écriture, ce qui peut affecter la capacité du DTW à évaluer correctement la similarité entre deux signaux. Dans cette section, nous proposons d'implémenter une DTW modifiée en remplaçant la distance euclidienne point à point f par la distance segment à point g, dans l'objectif de minimiser l'impact de la fréquence d'échantillonnage. Ces travaux ont été publiés dans [MMLM23a]

Soit x_i un point de x et $[\hat{x}_j, \hat{x}_{j+1}]$ un segment formé par deux points consécutifs de $\hat{x}$,

la distance point à segment, illustrée par figure 3.2, est définie par :

$$\vec{a} = \overrightarrow{\hat{x}_j\hat{x}_{j+1}}, \vec{b} = \overrightarrow{\hat{x}_j x_i},\ \vec{c} = \overrightarrow{\hat{x}_{j+1}x_i}$$
$$g(x_i, [\hat{x}_j, \hat{x}_{j+1}]) = \begin{cases} \vec{a}\cdot\vec{b} < 0,\ f(x_i, \hat{x}_j) \\ \overleftarrow{a}\cdot\vec{c} < 0,\ f(x_i, \hat{x}_{j+1}) \\ sinon,\ \left\| proj_{\vec{a}}\vec{b} \right\|^2 \end{cases} \tag{3.1}$$



(a) $\overrightarrow{a}\cdot\overrightarrow{b} < 0$ (b) $\overleftarrow{a}\cdot\overrightarrow{c} < 0$ (c) $0 \leq \frac{\overrightarrow{a}\cdot\overrightarrow{b}}{\|a\|^2} \leq 1$

FIGURE 3.2 – Distance point à segment.

En remplaçant f par g en tant que fonction de coût dans les expressions 2.1 et 2.2, nous définissons DTW_{seg}. Cette distance accorde une plus grande importance à la proximité d'un point prédit au squelette du signal vérité. Ainsi, les différences d'échantillonnage entre les deux signaux sont moins impactantes dans le coût total de l'alignement. Les segments sont formés uniquement au sein des points d'un même trait. Les segments formés entre deux points appartenant à deux traits distincts (associé à la fin d'un trait et le début d'un autre) sont omis (cf. fig 3.3).

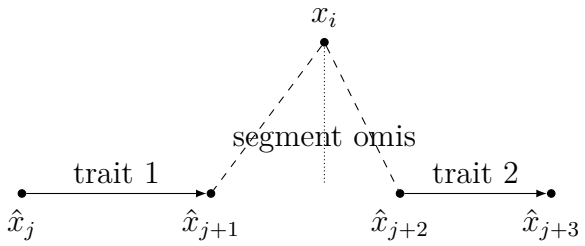


FIGURE 3.3 – Le point de x_j se situe ici entre les extrémités de deux traits consécutives $\hat{x}_{j+1}$ et $\hat{x}_{j+2}$. Cependant, ce segment est ignoré et x_j est aligné avec les segments $[\hat{x}_j, \hat{x}_{j+1}]$ ou $[\hat{x}_{j+2}, \hat{x}_{j+3}]$ uniquement.

Pour comparer la sensibilité à la fréquence d'échantillonnage des différentes métriques

$RMSE$, DTW et DTW_{seg}, nous effectuons des expériences avec les deux stratégies d'échantillonnages spatiales suivantes :

— Échantillonnage linéaire équidistant de distance d ;

— Moyenne mobile simple (MMS) entre deux points consécutifs :

$$x'_t = \frac{x_t + x_{t-1}}{2}$$

Nous avons utilisé le jeu de validation de *IRONOFF* [VG+99] contenant 19888 mots et le jeu de validation de *FLOWCHARTS* [Awa+11] comprenant 172 organigrammes. Le Tableau 3.2 montre que DTW_{seg} est comparativement plus faible après les transformations de sur-échantillonnage ($d = 2, 5$) et de moyenne mobile simple par rapport à DTW et $RMSE$. Ces transformations dégradent moins l'information spatiale des signaux par rapport au sous-échantillonnage ($d = 10$) qui est associé à un pic dans les différentes métriques dû à une perte importante d'information. On observe que DTW est la plus élevée des trois métriques lors du sous-échantillonnage de *IRONOFF* avec $d = 10$. DTW_{seg} est relativement 1.5 fois moins élevée. Pour *FLOWCHARTS*, lors du sous-échantillonnage avec $d = 15$, DTW_{seg} est 3 fois inférieure à DTW.

Jeu de donnée	**ré-échantillonnage**	**RMSE ↓**	**DTW ↓**	**DTW$_{\text{seg}}$ ↓**
IRONOFF	d=2	1.19	1.38	0.32
	d=5	2.43	1.64	0.76
	d=10	4.35	2.87	1.90
	MMS	3.01	1.89	0.78
FCs	d=2	29.03	8.04	0.25
	d=5	42.43	7.81	0.60
	d=10	58.46	7.66	1.29
	d=15	72.43	7.73	2.04
	MMS	64.99	6.03	1.82

TABLE 3.2 – Évaluation de différentes stratégies d'échantillonnage avec $RMSE$,DTW et DTW_{seg}.

De plus, nous utilisons notre métrique DTW_{seg} pour évaluer les systèmes de l'état de l'art pour la conversion d'écriture hors ligne en écriture ligne, à savoir [Dia+22 ; Arc+21 ; Cha20], en utilisant leurs implémentations officielles publiques. Nous incluons également une méthode interne basée sur un ensemble de règles et heuristiques portées sur un critère de continuité. Le Tableau 3.3 présente les résultats de leurs évaluations sur l'ensemble de validation de l'ensemble de données IRONOFF. Les images hors-ligne sont synthétique-

ment générées suivant la démarche détaillée dans 3.1.

Approche	**DTW** ↓	**DTW$_{\text{seg}}$** ↓	**RMSE** ↓
Interne [Privée]	5.00	4.40	11.94
Chan et al. [Cha20]	5.64	5.06	12.89
Archibald et al. [Arc+21]	8.10	7.45	15.77
Diaz et al. [Dia+22]	22.71	21.81	33.14

TABLE 3.3 – Évaluation des approches de l'état de l'art.

Nous constatons une concordance entre les trois métriques pour le classement des différentes approches. Toutes les approches mentionnées précédemment prédisent des signaux en ligne sur-échantillonnés par rapport à la vérité terrain, ce qui entraîne un coût d'alignement plus important pour la DTW classique par rapport à DTW_{seg}.

3.1.3 Conclusion

Nous avons présenté la métrique DTW_{seg}, une modification de DTW avec une fonction de coût point à segment dédiée à l'alignement d'écritures manuscrites en ligne. Nous avons montré que les métriques classiques, telles que la distance $RMSE$ et DTW, surestiment l'importance de la fréquence d'échantillonnage. En revanche, DTW_{seg} permet d'aligner de manière plus précise des signaux en minimisant l'impact de la fréquence d'échantillonnage. Nous avons également dressé une nouvelle comparaison de l'état de l'art de la conversion d'écriture hors ligne en écriture en-ligne avec la DTW_{seg}. DTW_{seg} peut être directement employée comme fonction de coût pour l'apprentissage d'un réseau de neurones pour les tâches de modélisation d'écriture en-ligne, de la même manière que DTW[CB18]. Nous émettons l'hypothèse que DTW_{seg} fournit des informations structurelles pertinentes, car son gradient encourage les prédictions du réseau à se rapprocher du signal dans son ensemble plutôt que d'un seul point du signal. Les variations naturelles dans le ductus constituent un bruit important qui est un obstacle majeur à l'apprentissage des réseaux de neurones. En éliminant une partie de ce bruit, celui lié aux variations de fréquence d'échantillonnage, DTW_{seg} peut permettre une meilleure modélisation de l'écriture en-ligne. Il s'agit d'une piste possible d'amélioration, mais que nous n'avons pas exploré dans cette thèse.

3.2 FCNN et généralisation aux images de taille variable

3.2.1 Modèle multitâche entièrement convolutif

Zhao et al. [ZYT18] ont proposé une approche CNN itérative pour générer une trajectoire de stylo (section 2.3.3) à partir d'une image hors-ligne. Ils ont obtenu des résultats satisfaisants sur les caractères hors-lignes chinois et anglais. Cependant, la complexité de leur modèle dépend directement de la résolution de l'image hors ligne à cause des dernières couches qui sont entièrement connectées. La basse résolution employée (28×28) est insuffisante pour des applications au-delà du niveau des caractères, tels que des expressions mathématiques ou lignes de textes. Cela limite en pratique la généralisation du modèle aux contenus plus larges. Leur méthode repose également sur le squelette pour guider la prédiction des mouvements du stylo et pour arrêter la génération. Comme mentionné dans la section 2.2.1 la squelettisation d'image hors ligne est une étape essentielle et complexe. Le squelette inféré est souvent incomplet et bruité. Notre approche est une extension de la méthode CNN de Zhao et al. [ZYT18] : un réseau multitâche entièrement convolutif pour la prédiction des mouvements et états du stylo ainsi que le squelette. Notre approche va au-delà des limitations de la méthode originale en étant applicable à n'importe quelle résolution d'image [MMLM21]. Le modèle est bout-en-bout, éliminant ainsi la nécessité d'un prétraitement de squelettisation de l'image hors-ligne. De plus la classification de l'état du stylo permet aussi la segmentation des traits. La figure 3.4 montre les entrées et les sorties de notre modèle. L'entrée de notre réseau se compose de cinq images, les images de la positon précédente et actuelle du stylo $F_{i-1}, F_i \in \{-2, -1, 0\}^{h \times w}$ et l'image hors ligne en niveaux de gris $I \in [0, 1]^{h \times w}$. Nous fournissons également les coordonnées des pixels sous forme de deux images $I_X, I_Y \in [0, 1]^{h \times w}$. Le réseau dispose d'un champ récepteur de taille 32×32, les gradients spatiaux que nous fournissons constitue des informations globales qui peuvent aider à améliorer les décisions du réseau dans les régions locales. Le réseau produit trois images en sortie, le squelette complet I_S, l'image des extrémités finales des traits I_E et la prochaine position du stylo I_{i+1}^{POS} et enfin la classification de l'état du stylo parmi trois états possibles {Poser, Lever, Fin}. La redondance dans les différentes sorties permet de guider l'apprentissage de la tâche principale (la position du stylo et l'état du stylo). Nous entraînons notre réseau sur des images hors lignes synthétiques avec une largeur de trait variable générée à partir des signaux en lignes.

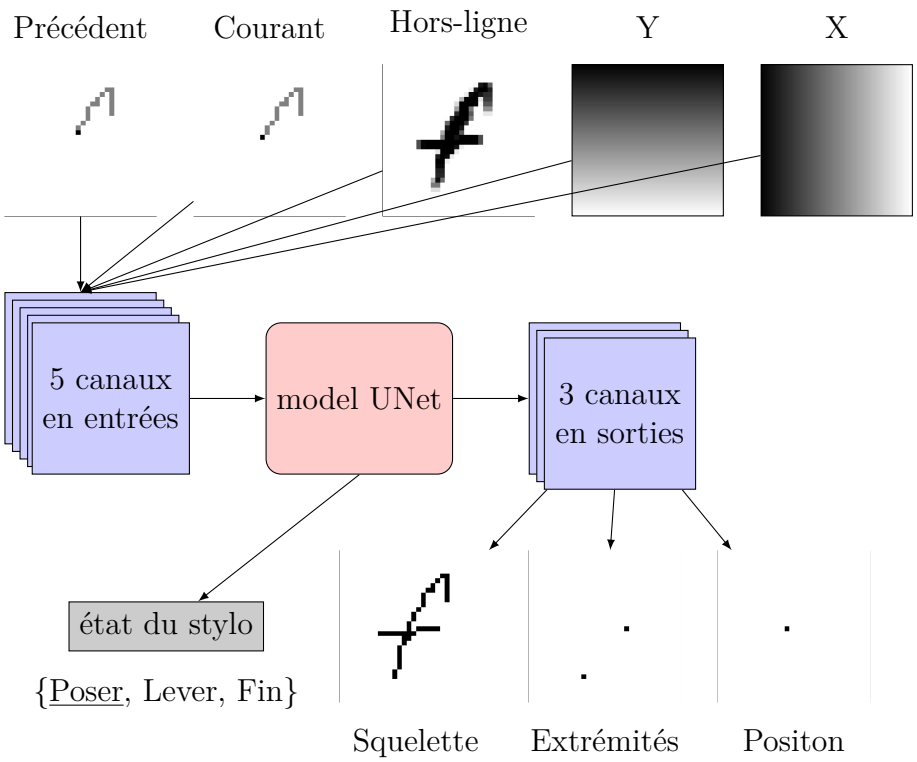


FIGURE 3.4 – Entrées et sorties de notre modèle CNN.

Motivés par l'application réussie des réseaux de neurones entièrement convolutifs (FCNN) dans [SS+16] (décrite dans la section 2.2.1) pour la simplification des croquis, nous adaptons notre modèle à partir de l'architecture U-Net [RFB15]. U-Net est un FCNN utilisé pour la segmentation d'images dans des applications biomédicales. UNet comprend un chemin de réduction (encodeur) et un chemin de restauration (décodeur) avec des connexions de raccourci pour réutiliser les informations à haute définition.

Le chemin de réduction (encodeur) est une succession de convolutions 3×3 suivies de la fonction d'activation $ReLU$ et de couches de max pooling 2×2. L'image en entrée est encodée dans une carte de caractéristiques basse résolution H. Le chemin de restauration décode la carte de caractéristiques H pour revenir à la résolution d'origine en utilisant des convolutions transposées (parfois appelées convolutions inverses ou déconvolutions) avec un pas de 2×2. Les informations à haute définition sont réutilisées grâce aux connexions de raccourci reliant les couches de réduction aux couches de restauration. Comme le montre la figure 3.5, les images des mouvements du stylo (F_{i-1}, F_i), l'image hors ligne I et les images des coordonnées I_X, I_Y sont encodées dans une carte de caractéristiques cachée de

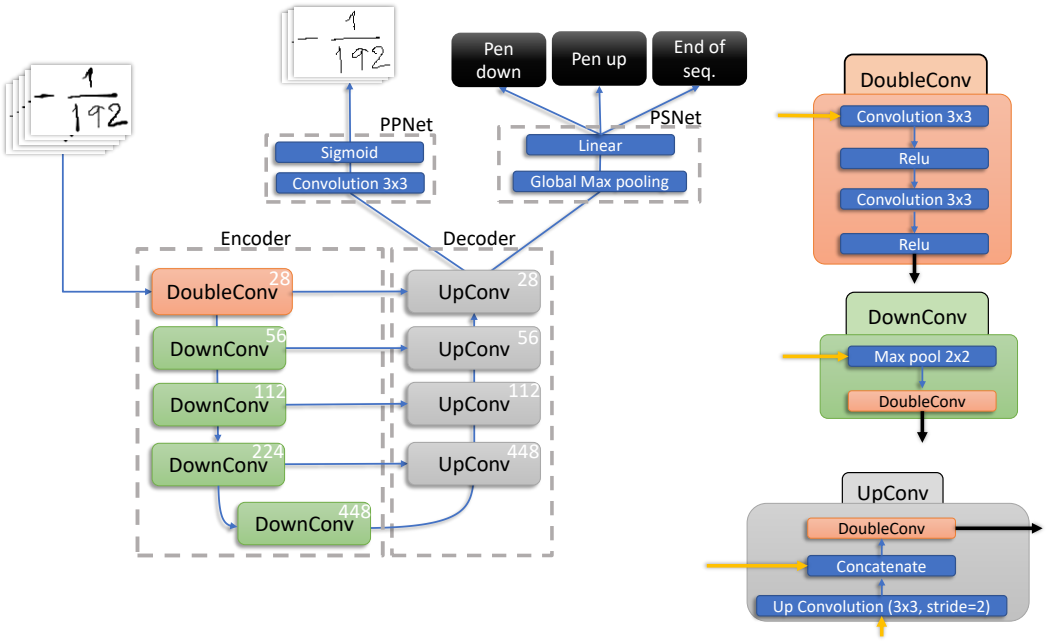


FIGURE 3.5 – Architecture du réseau. Le nombre de filtres utilisés dans chaque convolution est indiqué sur chaque bloc. L'encodeur et le décodeur sont partagés par PSNet et PPNet.

petite taille $H \in \mathbb{R}^{448}$:

$$H = Encoder(F_{i-1}, F_i, I, I_X, I_Y).$$

Le chemin de restauration décode la carte de caractéristiques H pour obtenir une carte $O \in R^{h \times w \times 28}$ à la résolution d'origine et produit l'image du squelette $\hat{I}S$, les positions de fin de trait $\hat{I}E$ et localise la prochaine position du stylo $\hat{I}_{pos}^{i+1}$. Les expressions (3.2) à (3.4) détaillent le calcul de ces sorties images.

$$O = \text{Decoder}(H) \tag{3.2}$$

$$\hat{I}_S, \hat{I}_E = \sigma(\text{conv}_2(O)) \tag{3.3}$$

$$\hat{I}_{pos}^{i+1} = \sigma(\text{conv}_1(O) \odot I_S) \tag{3.4}$$

Les fonctions conv_1 et conv_2 sont des couches de convolution classiques avec respectivement 1 et 2 noyaux. La fonction σ est la fonction sigmoïde. Le produit avec le squelette I_S avant la sortie de la prochaine position permet de contraindre la prochaine position à se situer sur le squelette prédit. Cela simplifie la tâche de la dernière opération conv_1. Une

opération similaire est réalisée dans [ZYT18], mais en tant qu'étape de post-traitement, en utilisant le squelette extrait préalablement.

L'encodeur et le décodeur définissent le réseau de prédiction de la position du stylo `PPNet`. Nous modifions U-Net en ajoutant un troisième chemin, un réseau de classification de l'état du stylo `PSNet` avec une couche entièrement connectée. Nous agrégeons la sortie du décodeur O en un vecteur de taille fixe avec un *global max pooling* et l'injectons dans la couche de classification :

$$\hat{P}_{i+1} = \text{PSNet}(O). \tag{3.5}$$

Cela permet au réseau de classification d'avoir une vue complète de l'image d'entrée, alors que le décodeur dispose d'un champ récepteur de taille fixe pour effectuer sa prédiction, même si les couches de *max pooling* augmentent de manière exponentielle le champ récepteur. Le stylo peut prendre trois valeurs d'état différentes. En plus des états standard *stylo baissé* et *stylo levé*, nous définissons un état *fin* indiquant que le scripteur a terminé l'écriture. Nous considérons qu'une *fin* est un *stylo levé*, ce qui implique une classification multi-étiquettes. L'état *fin* est nécessaire, car il s'agit de la condition d'arrêt pour le cadre itératif utilisé dans l'inférence. En effet, la condition employée par [ZYT18], le parcours de tous les pixels du squelette est insuffisante, car un scripteur peut tracer certains pixels plusieurs fois (pixels aux jonctions et segments à double tracé) et certains pixels du squelette associé à du bruit peuvent ne pas être traversée. Notre réseau est entraîné avec une fonction de coût multitâche composée d'une entropie croisée binaire pour la classification de l'état du stylo $\hat{P}$, d'une soft-F1 pour la squelettisation $\hat{I}_S$ et d'une *MSE* pour la prédiction des extrémités de traits $\hat{I}_E$ et la prédiction la position du stylo $\hat{I}_{pos}$.

$$\begin{aligned}
&\mathcal{L} = \mathcal{L}_P + \mathcal{L}_S + \mathcal{L}_E + \mathcal{L}_{POS} \\
&\mathcal{L}_P(\hat{P}, P) = \frac{1}{3} \sum_{y \in \{down, up, end\}} -(P_y \log(\hat{P}_y) + (1 - P_y) \log(1 - \hat{P}_y)) \\
&\mathcal{L}_S(\hat{I}_S, I_S) = \frac{1}{h \times w} \sum_{h,w} \text{softF1}(\hat{I}_S, I_S) \\
&\mathcal{L}_E(\hat{I}_E, I_E) = \frac{1}{h \times w} \sum_{h,w} \text{softF1}(\hat{I}_E, I_E) \\
&\mathcal{L}_{POS} = \frac{1}{h \times w} \sum_{h,w} \left\| I_{pos} - \hat{I}_{pos} \right\|_2^2
\end{aligned} \tag{3.6}$$

où $I_S, I_E, I_{pos} \in 0, 1^{h \times w}$ sont respectivement les images de vérité terrain du squelette,

de la prochaine position du stylo et des extrémités des traits. La sortie de `PPNet` prédit conjointement la position du prochain point, le squelette ainsi que les fins des traits, avec trois couches de sortie distinctes. Nous définissons la position du prochain point comme le pixel ayant l'activation la plus élevée dans la carte de chaleur en sortie I_{i+1}^{POS}.

Une analyse complète des performances de notre architecture de réseau est effectuée. Tout d'abord, nous évaluons les sous-tâches : la squelettisation, la prédiction de la position du stylo et la classification de l'état du stylo. Ensuite, nous évaluons l'extraction de traits à l'aide de la combinaison du modèle proposé avec un algorithme d'inférence itératif. Nous comparons également nos résultats avec ceux de [ZYT18] sur l'ensemble de données Unipen.

3.2.2 Évaluation sur les symboles isolés et expressions hors-ligne

Pour évaluer `PSNet`, nous calculons le score f1 par classe pour la classification de l'état du stylo, l'erreur $TOP-1$ est la métrique d'évaluation pour la prédiction de la position du stylo. Le score F1 est employé pour évaluer l'extraction du squelette et des points de fin de trait. Les résultats d'évaluation sur les ensembles de données UNIPEN et CROHME sont répertoriés dans le tableau 3.4.

En comparaison, [ZYT18] obtient un taux d'erreur de top-1 pour la position du stylo de 1.2% sur UNIPEN. Aucune implémentation publique de leur travail n'est disponible. Nous avons implémenté et entraîné leur réseau de neurones sur UNIPEN. Certains méta-paramètres non spécifiés ont été optimisés sur l'ensemble de validation. Nous avons obtenu un taux d'erreur de 4.6%. Ce résultat doit être comparé aux 15.5% d'erreur obtenue par notre réseau sur le même ensemble de données dans le tableau 3.4.

	Prochaine position du stylo	Squelette	Fin des traits	État du stylo		
				Poser	Lever	EOS
CROHME	0.100	0.993	0.754	0.990	0.770	0.870
UNIPEN	0.155	0.939	0.6691	0.990	0.682	0.789

TABLE 3.4 – Résultat d'évaluation de `PPNet` et `PSNet` sur les ensembles de données de test UNIPEN et CROHME. Le tableau montre l'erreur TOP-1 pour la position du stylo, le score F1 pour le squelette et les points de fin de trait ainsi que le score F1 par classe pour la classification de l'état du stylo.

Un algorithme d'inférence itératif adapté de [ZYT18] est employé pour extraire les traits avec notre réseau. À chaque itération, en plus de la prédiction du prochain point, la

sortie de l'état du stylo est utilisée pour déterminer la fin des traits. La position du stylo est contrainte au squelette en multipliant la carte de chaleur I_{POS}^{i+1} par le squelette prédit I_S. Nous évaluons l'algorithme d'extraction de traits proposé sur UNIPEN et CROHME. Nous utilisons l'intersection sur union des traits (`SIoU`) de [Guo+19], définie comme suit :

$$\texttt{SIoU} = \frac{1}{n} \sum_{i=1,\dots,n} \max_{j=1,\dots,m} \frac{P_i \cap \hat{P}_j}{P_i \cup \hat{P}_j}, \tag{3.7}$$

avec n étant le nombre de traits dans le signal en ligne vérité et m le nombre de traits dans celui prédit. Un trait de vérité terrain P_i est associé au trait prédit $\hat{P}_j$ avec la plus grande `IoU`. `SIoU 75%` correspond au taux de traits pour lesquels le `SIoU` est supérieur à 75%.

Le tableau 3.5 compare ces métriques pour les deux systèmes sur l'ensemble de données UNIPEN. La méthode de [ZYT18] ne fournit pas d'information sur l'état du stylo. Ainsi, nous considérons que le stylo est levé si la prochaine position du stylo n'est pas 8-connexe à la position actuelle. Nous pouvons observer que notre approche a un meilleur

Méthode	SIoU	SIoU 75%	Nombre de paramètres
Zhao et al. [ZYT18]	0.3587	0.4673	17.8 Millions
Ours	0.568	0.283	5.9 Millions

TABLE 3.5 – Évaluation et comparaison de l'extraction de traits sur UNIPEN.

`SIoU`. Cependant, les deux ont des résultats assez faibles ; dans le meilleur cas, il n'y a qu'une correspondance de 56.8% entre les traits prédits et les traits de vérité terrain. Le `SIoU75%` montre des comportements différents pour les deux systèmes. Il semble que notre approche sur-segmente les traits (seulement 28.3% des traits correspondent à plus de 75%), tandis que le système de Zhao et al. semble fusionner des traits (sous-segmentation). Nous pouvons également remarquer que notre système a trois fois moins de paramètres que l'autre. Pour mieux comprendre le comportement du système proposé sur de grandes images, nous étudions dans le tableau 3.6 ses performances pour différents sous-ensembles de l'ensemble de données CROHME en fonction du nombre de traits dans l'encre d'origine. Nous pouvons observer qu'il y a une légère diminution du `SIoU` pour l'encre contenant entre 10 et 20 traits, mais les résultats sont globalement stables autour de 53.2%. La faible valeur du `SIoU75%` montre que nous sur-segmentons la plupart des traits, car seulement 22% des traits correspondent à plus de 75% des traits d'origine. Cependant, nous notons de manière surprenante que le résultat le plus bas est pour les petites expressions (17.7%

pour moins de 5 traits). Nous pensons que ceci est dû au fait que le système réussit bien sur les traits longs et droits (comme la barre de fraction, l'intégrale, le signe égal, ...) qui sont rares dans les petites expressions.

	[1,5]	]5,10]	]10,15]	]15,20]	]20,25]	All
`SIoU`	0.510	0.511	0.498	0.498	0.535	0.532
`SIoU75%`	0.177	0.178	0.192	0.217	0.229	0.220

TABLE 3.6 – Résultat de notre approche pour différentes tailles d'expressions.

La Figure 3.6 présente les résultats de l'inférence complète sur deux échantillons de tailles différentes. Nous pouvons voir dans les images hors-lignes sur la première ligne. La deuxième ligne de la figure montre les traits vérité. Nous pouvons voir dans la dernière ligne que les traits prédits suivent correctement le squelette. Nous pouvons remarquer que les traits se terminent à l'extrémité du squelette et que tous les symboles ont étés couverts en entier. Cependant, nous observons une sur-segmentation des traits. Le système prédit bien les lignes droites longues (barres de fraction, parties égales) et les courbes lisses (parties des symboles α, θ ou t). Les difficultés surviennent dans les parties complexes où les traits se croisent (dans les symboles α, θ), aux points d'inflexion (dans les symboles 7, 1 ou tan) où le modèle décide souvent de lever le stylo.

3.2.3 Discussion

Dans cette section, nous avons présenté une approche bout-en-bout avec un réseau entièrement convolutif pour l'extraction de traits à partir d'images hors-ligne. Le réseau prédit simultanément la position du stylo, le squelette, les extrémités des traits et l'état du stylo à chaque itération. L'algorithme d'inférence d'itératif de [ZYT18] est complété par la prédiction de l'état du stylo afin d'extraire des traits. À notre connaissance, cette approche est la première à aborder l'extraction des traits à partir d'une résolution d'image arbitraire. La squelettisation et l'extraction des extrémités des traits montrent de bons résultats. Cependant, nous montrons les limites de l'approche convolutive sur ce type de problème : nos expériences ont montré une tendance du modèle à sur-segmenter les traits, en raison d'une part du contexte spatiale et temporaire insuffisant et d'autre part les limitations de modélisation temporelle de l'architecture CNN. Le biais inductif des CNN pour le traitement des informations spatiales en fait un choix naturel pour les tâches de traitement d'images et autres données en grille. Pour pallier à la limitation temporelle du CNN, due

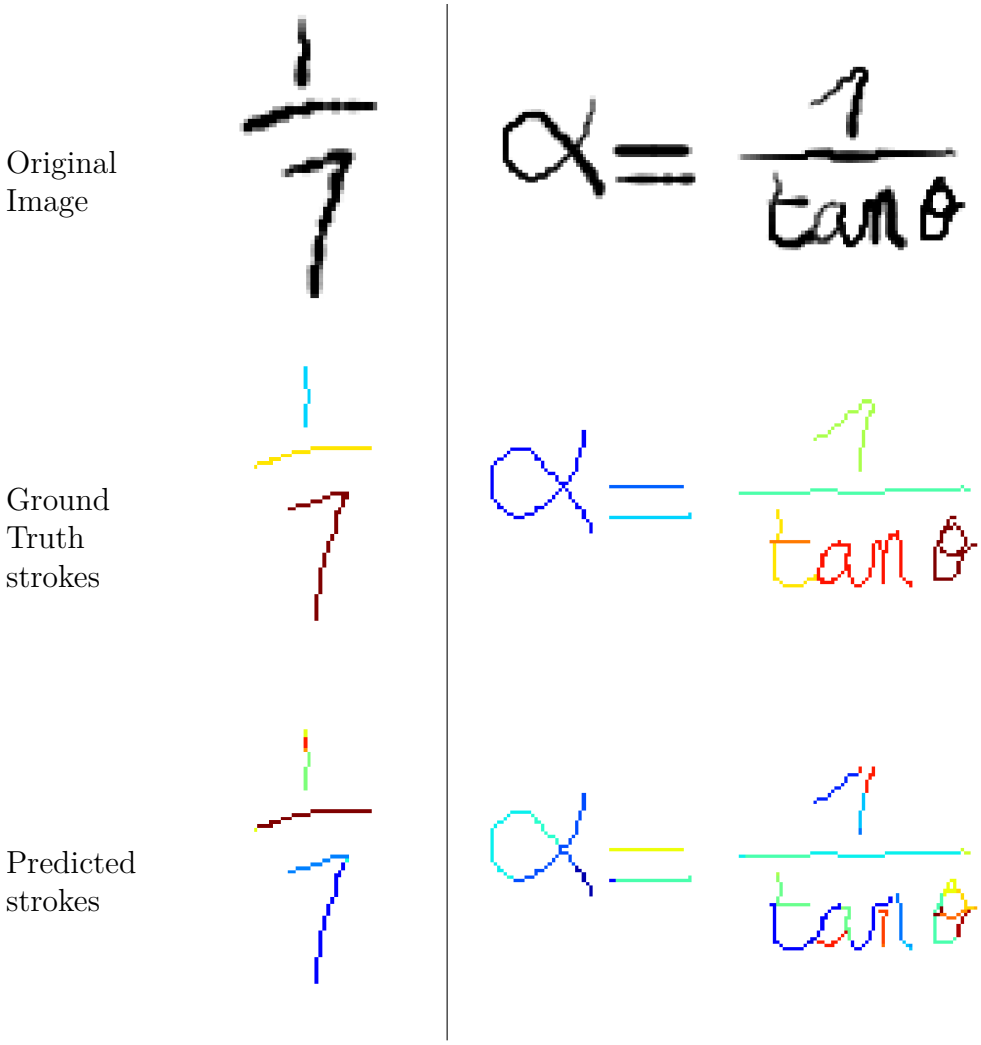


FIGURE 3.6 – Visualisation de l'extraction de traits pour une image simple et une image plus complexe. Chaque trait est dessiné avec une couleur différente.

à l'absence d'un mécanisme de mémoire, l'empilement de plusieurs trames consécutive est couramment employé. Le CNN peut capturer certaines informations temporelles à partir des images empilées. Cependant, cette méthode a une portée de mémoire assez courte et peut ne pas capturer pleinement les dépendances à long terme. Le nombre d'images à empiler est un hyper-paramètre qui doit être choisi en fonction du contexte de la tâche et de la longueur des dépendances temporelles que le modèle doit être capable de capturer. Comme [ZYT18], nous avons employé un empilement de deux images consécutives de mouvements du stylo. Il s'agit d'un empilement de longueurs minimal et qui doit être augmenté afin de permettre la capture de dépendances de plus longues portées. En général, un choix courant est de prendre entre 4 et 10 [Mni+13] images consécutives.

Une solution hybride CNN-RNN adoptée par [Arc+21] consiste à utiliser un réseau de neurones récurrent (en l'occurrence un BiLSTM) par-dessus d'un CNN. Le réseau récurrent est utilisé pour capturer les dépendances séquentielles entre les vecteurs caractéristiques extrait à partir du CNN. Il est capable de maintenir une mémoire des informations passées et de modéliser les dépendances temporelles de plus longues portées. Cette architecture offre une meilleure alternative à l'approche consistant à utiliser uniquement un CNN avec empilement d'images consécutives, comme le démontre leur performance élevée sur des séquences plus longues de lignes de texte.

Dans la section 3.3, nous explorons une approche séquentielle alternative basée sur les Transformers [Vas+17] pour modéliser de manière plus précise les mouvements du stylo dans l'écriture manuscrite [Aks+20], grâce à leur mécanisme d'attention. Nous adaptons cette architecture au traitement d'images hors-ligne afin de capturer les caractéristiques spatiales de l'écriture.

3.3 Transformer au niveau sous-trait

Comme mentionné dans la section précédente, un mécanisme de mémoire est indispensable pour modéliser des trajectoires dynamiques du stylo. L'approche à base de réseaux de neurones récurrents de [Arc+21], présentée dans la section 2.3.3, incorpore un mécanisme de mémoire et a permis une première généralisation des réseaux de neurones au-delà du niveau caractères vers les lignes de texte. Cette section présente une autre approche, avec un mécanisme de mémoire également, basé sur un réseau de neurone Transformer au niveau sous-trait. Elle fait partie de la famille des approches topologiques. On s'inspire des connaissances acquises par les experts humains : beaucoup d'approches heuristiques

(cf.2.3.3) ont convergé vers un algorithme basé sur un découpage en un ensemble de primitives géométriques et une optimisation de leur ordonnancement.

Dans cette partie, nous présentons une approche Transformer au niveau sous-traits se déclinant en deux modèles : un Transformer d'encodage de sous-trait (SET) et un Transformer d'ordonnancement de sous-trait (SORT). Nous évaluons notre méthode sur les mots cursifs, les phrases et les expressions mathématiques.

3.3.1 Vue d'ensemble

Dans cette section, nous présentons un nouveau système basé Transformer pour produire des reconstructions fidèles de la trajectoire du stylo à partir d'image hors-ligne (illustré dans la figure 3.7) nommé *SET SORT* et publié dans [MMLM23b ; MMLM].

Premièrement, un algorithme de découpage d'une image hors-ligne en sous-traits agnostique à la nature de l'écriture et quasiment sans heuristiques est utilisé. Les vecteurs descripteurs des sous-traits sont ensuite produits en encodant la séquence de points d'un sous-trait avec un Transformer d'encodage de sous-trait (SET). L'ensemble des vecteurs descripteurs de sous-traits est ensuite fournis en entrée à un Transformer d'ordonnancement de sous-traits (SORT). SORT infère un ordre à cet ensemble de sous-traits extraits précédemment à travers une prédiction discrète d'une (pseudo) permutation et prédit les levers de stylos associés.

3.3.2 Extraction des sous-traits

Extraction hybride des sous-traits

Nous commençons par utiliser le UNet (voir sec.3.2), entraîné au préalable pour extraire un squelette de l'image hors ligne en entrée. Pour éliminer les quelques ambiguïtés restantes dans le squelette, l'algorithme d'amincissement de [ZS84] est appliqué au squelette prédit. Un algorithme de découpage des sous-traits basé sur la détection des jonctions est ensuite appliqué au squelette. Un pixel de jonction est défini comme un pixel du squelette ayant plus de deux autres pixels de squelette dans son voisinage 8-connexe. Une fois les pixels de jonctions détectés, ils sont supprimés. Les composantes connexes résultantes sont des segments ayant deux extrémités (figure 3.8) à l'exception des rares occurrences de boucles parfaites n'ayant aucune extrémité. Deux sens de parcours (ou direction) d'une extrémité à l'autre sont envisageables pour un segment donné : un segment dirigé définit donc un sous-trait.

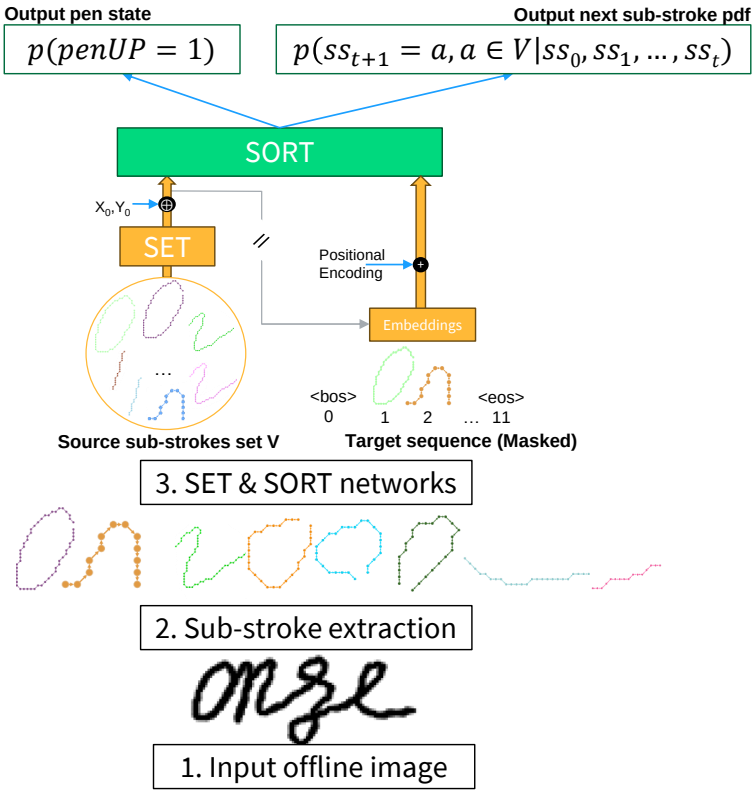


FIGURE 3.7 – Aperçu de l'approche proposée. Après l'extraction des sous-traits à partir du squelette du hors-ligne, le réseau SET permet d'obtenir un vecteur descripteur de sous-trait. Le réseau SORT utilise l'ensemble des vecteurs descripteurs V et l'historique pour prédire le prochain sous-trait ss_{t+1} et l'état du stylo.

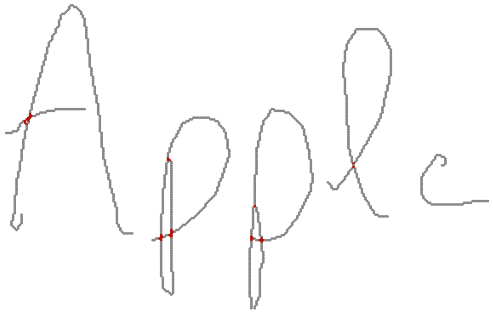
L'ensemble des sous-traits extraits est noté par $V = \{ss_0, \ldots, ss_i, \ldots, ss_n\}$ avec $ss_i = (x_k, y_k)_{k=1}^{m}$ un sous-trait (*i.e.* un segment dirigé) défini par une séquence de points de longueur m d'origine (x_1, y_1) et d'extrémité (x_m, y_m). L'objectif est de déterminer la permutation partielle π de l'ensemble des sous-traits V, indiquant l'ordre d'écriture des sous-traits $\pi(V) = S = (ss_{\pi(1)}, \ldots, ss_{\pi(i)}, \ldots, ss_{\pi(N')})$ avec $ss_{\pi(i)} \in V$.

Il s'agira de trouver le bon ordonnancement avec le modèle SORT (3.3.4), mais avant ça il faut définir la vérité que nous appelons oracle.

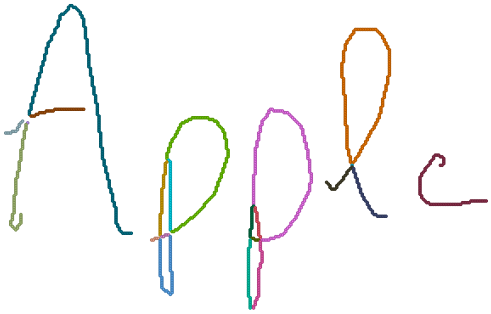
(a) Image d'écrire hors-ligne de la base [VG+99] de dimension 298×184.

(b) Squelette inféré par le FCN [MMLM21]. Ce squelette contient quelques artefacts.

(c) Amincissement via l'algorithme de [ZS84] pour obtenir un squelette de 1 pixel d'épaisseur.



(d) Détection des pixels de jonction, coloriés en rouge ici.



(e) Suppression des pixels de jonctions. Chaque composante connexe constitue un segment.

FIGURE 3.8 – Extraction des sous-traits de l'image hors-ligne.

Ordonnancement vérité des sous-traits : oracle

La permutation vérité des sous-traits π peut être dérivée à partir du signal en-ligne original avec l'algorithme suivant :

1. Alignement des sous-traits avec le signal d'écriture en-ligne : chaque sous-trait est aligné sur une sous-séquence du signal en-ligne avec une DTW. Il s'agit d'une variante de DTW [Tor+09] sans la contrainte de frontière (voir eq.2.1.2).
2. Rejet des sous-traits d'un segment à sens unique : une partie des sous-traits est ignorée car la plupart des segments sont parcouru une seule fois et dans un sens unique. Seuls les segments des zones de rebroussement ou les boucles peuvent être parcouru dans deux sens ou plusieurs fois respectivement, dans ce cas, les sous-traits correspondants seront associés à des sous-séquences différentes du signal en-ligne originale. Si les deux sous-traits d'un même segment sont associés à la même sous-séquence du signal en-ligne, le sous-trait ayant le coût d'alignement le plus élevé est rejeté.
3. Ordonnancement des sous-traits : une fois les sous-traits filtrés et alignés sur différentes sous-séquences du signal en-ligne, les sous-traits sont ordonnés suivant le même ordre d'apparition de leur sous-séquence attribuée dans le signal en-ligne originale.

Cette permutation π est désignée comme la solution de l'**oracle** à notre problème de d'ordonnancement des sous-traits d'une image hors-ligne. Cet ordonnancement est utilisé comme vérité pour l'apprentissage de notre réseau Transformer *SORT* (section 3.3.4). Il s'agit d'une approximation satisfaisante du signal en-ligne.

Notre choix de la modélisation de l'écriture au niveau sous-trait est motivé par les travaux de [Aks+20] sur la génération de diagramme et d'écriture en-ligne à l'aide de Transformer au niveau trait. Ils démontrent que les Transformers au niveau trait surpassent les approches RNN [Gra14]. Cependant, ils concluent que les traits de l'écriture cursive posent des défis, car les traits longs ne peuvent pas être correctement encodés dans un vecteur de taille fixe. Dans ce travail, notre modélisation s'effectue à une échelle plus fine, celle des sous-traits. Les sous-traits sont des formes de nature géométrique plus simple (ligne droite, courbes ouvertes courtes, etc.) qu'un trait entier, ils sont donc bien plus faciles à modéliser. L'objectif ici étant de prédire la séquence indiquant l'ordre d'écriture des différents sous-traits et dans quelle mesure ils doivent être fusionnés pour former des traits grâce aux prédictions de lever de stylo pour indiquer la fin du trait.

3.3.3 Vecteurs descripteurs de sous-traits avec SET

Dans la section 2.3.3, nous avons exploré diverses approches neuronales partageant un même paradigme : l'obtention d'une représentation visuelle globale de l'image hors-ligne à travers une séquence de vecteurs caractéristiques extraits par un CNN. Cette séquence de vecteurs caractéristiques est ensuite exploitée pour prédire l'information temporelle, soit directement ([ZYT18]), soit au moyen d'un réseau récurrent ([Bhu+18]). Cependant, cette méthode présente un inconvénient notable : le CNN agrège localement l'information en réduisant la dimensionnalité de l'image, ce qui peut entraîner une perte potentielle d'informations fines et locales présentes dans l'image d'origine. Cette compression de l'information peut conduire à des trajectoires simplifiées, voire incomplètes, du tracé du scripteur. Comme illustré dans la section 2.3.1, les approches heuristiques combinant une représentation spatio-temporelle locale au niveau du segment avec une analyse globale demeurent plus performantes que les approches neuronales, malgré leurs limites.

Dans cette section, nous allons explorer une représentation alternative plus riche de l'image hors-ligne en intégrant l'information positionnelle avec l'information visuelle. L'intégration de l'information positionnelle est cruciale pour extraire des caractéristiques temporelles locales, indispensables pour des critères de décision importants tels que la courbure et l'orientation.

Dans un premier temps, nous avons étudié une approche au niveau pixel : la carte de caractéristiques intermédiaire fournie par le FCN (section 3.2) est combinée avec les coordonnées des pixels du squelette (figure 3.9) pour obtenir un ensemble de vecteurs descripteurs de l'image hors-ligne. Cet ensemble de vecteurs descripteurs est ensuite fourni à un Transformer autorégressif pour prédire les coordonnées de la trajectoire du stylo.

Une étude préliminaire a été effectuée sur des images de symboles mono-trait des bases de données de mots et expressions ([VG+99 ; Mah+19]). Le modèle *UNet* proposé dans la section 3.2 est figé et utilisé pour obtenir les vecteurs de caractéristiques visuelles nécessaires pour l'apprentissage du modèle Transformer (figure 3.9). Ce modèle surpasse l'approche FCN proposée dans la section 3.2, sur cette base mono-trait avec un DTW de 0.33 contre 1.23 pour le modèle FCN. La table 3.7 montre que les caractéristiques positionnelles ont un impact plus important sur la précision de la reconstruction de la trajectoire du stylo. La dernière ligne du tableau montre que le nuage de vecteurs descripteurs positionnels des pixels du squelette à lui seul permet d'atteindre le meilleur résultat. Cela peut être dû aussi à une fusion sous-optimale à l'aide d'une somme des deux caractéristiques visuelle et positionnelle. Les images hors-lignes synthétiques sont

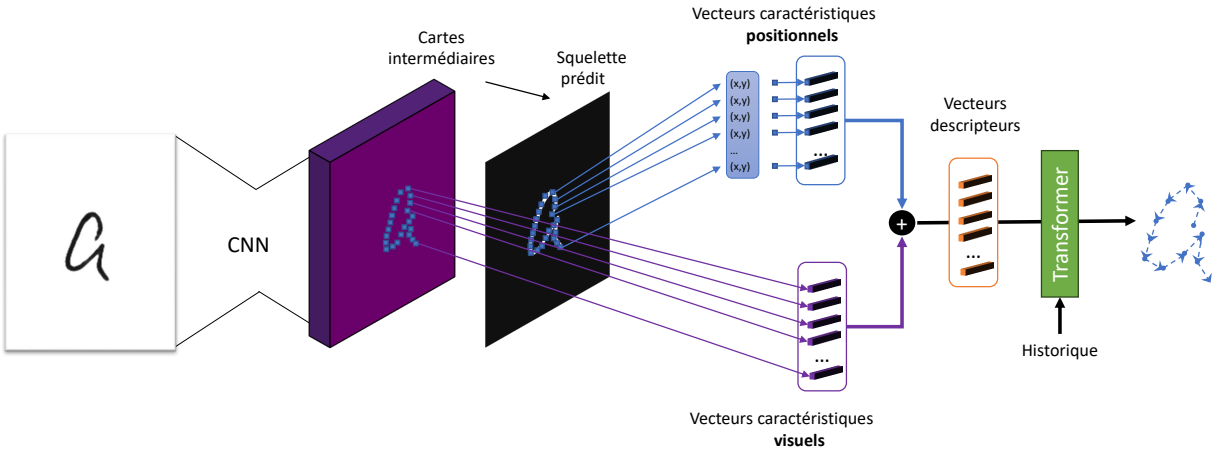


FIGURE 3.9 – vecteurs descripteurs basés sur l'information positionnelle et visuelle de l'image hors-ligne. La carte caractéristique intermédiaire désigne la sortie de l'avant-dernière couche du FCN. Les vecteurs caractéristiques visuelles du squelette sont sélectionnés, puis additionnés avec les coordonnées des pixels du squelette pour obtenir l'ensemble des vecteurs descripteur de l'image.

Caractéristiques visuelle	**Caractéristiques positionnelle**	**DTW↓**
✓	✓	0.33
✓	✗	0.57
✗	✓	**0.17**

TABLE 3.7 – Étude de l'impact des caractéristiques visuelles et positionnels sur les résultats du Transformer au niveau pixel. Bases d'apprentissage mots et expressions mono-traits.

des matrices très creuses, avec moins de 5% de pixels qui portent une information. En choisissant uniquement les pixels du squelette comme entrée du Transformer, on réduit considérablement son empreinte en mémoire et en temps de calcul tout en réduisant son espace de recherche et donc simplifier son optimisation. De plus, en conditionnant la sortie du Transformer par le squelette entier, en évitant une compression excessive de la représentation de l'image, le signal en-ligne sera reconstruit plus finement et couvrirait de plus près le trait de l'image.

Cette approche est limitée aux symboles isolés mono-traits et ne s'étend pas aux mots ou phrases. La taille du nuage de points du squelette augmente considérablement avec l'augmentation du nombre de traits. L'empreinte mémoire quadratique du Transformer rend impossible le passage aux échelles supérieurs. Il devient impératif de réduire intelligemment la dimensionnalité de notre représentation sans sacrifier la précision des

caractéristiques de l'image, comme cela était le cas dans notre approche antérieure.

En analogie avec la couche de *pooling* dans un réseau CNN qui réduit la résolution d'une carte de caractéristiques en séparant l'image en plusieurs petites régions indépendantes et en les agrégeant ensuite, nous proposons un *pooling* par segment. Un segment comme défini dans la section 2.3, est un morceau contigu, non orienté et non ambiguë du squelette. Les segments sont des entités disjointes du squelette qui peuvent être agrégées pour obtenir des vecteurs descripteurs de segment du squelette. Le nombre de segments dans une image hors-ligne phrase est relativement petit, ainsi un modèle Transformer peut être envisagé.

On rappelle qu'un segment est associé à deux sous-traits, un pour chaque sens de parcours du segment. Pour obtenir, les vecteurs caractéristiques des sous-traits, nous adoptons la méthode de [Aks+20]. Les sous-traits sont d'abord normalisés par les dimensions de l'image pour avoir leur coordonnée dans l'intervalle $[0, 1]$, ils sont ensuite translatés pour avoir comme point initial l'origine $(0, 0)$. Un réseau auto-encodeur (SET) est employé pour apprendre des vecteurs descripteurs de sous-trait. Il est constitué de deux modèles : un Transformer décodeur auto régressif pour encoder un sous-trait de longueur variable en un vecteur descripteur de dimension d_{model} et un réseau décodeur MLP pour reconstruire le sous-trait à partir de dernier vecteur descripteur :

— **Transformer encodeur autorégressif** avec N_l couches, de dimension d_{model}, N_h têtes d'attention et un réseau à propagation avant de dimension d_{ff} est employé pour **encoder** un sous-trait. Il prend un sous-trait (une séquence de point $ss_i = ((x, y)_{k=0}^{m})$ de longueur m) en entrée et fournit en sortie un vecteur descripteur $E_{ss_i} \in \mathbb{R}^{d_{model}}$. Les points $(x, y)_k$ sont d'abord projetés linéairement en vecteurs de dimension d_{model} puis additionnés avec un encodage sinusoïdal de leur position dans la séquence. Cette séquence de vecteurs est fournie au Transformer en entrée, la sortie du Transformer O_m pour le dernier point $(x, y)_m$ de la séquence est projeté linéairement pour obtenir le vecteur descripteur E_{ss_i} du sous-trait ss_i, de dimension D (comme illustré par la figure 3.10).

— **Un MLP** estime la courbe paramétrique d'un sous-trait en fonction de son vecteur descripteur $F(E_{ss_i}, t) \in \mathbb{R}^2$, $t \in [0, 1]$. F est donc ici modélisé par un MLP possédant deux couches de 512 neurones avec une fonction d'activation $ReLU$. Le réseau prédit les coordonnées $(\hat{x}_t, \hat{y}_t)$ en fonction du temps t et du vecteur descripteur du sous-trait E_{ss_i}. Cette composante est utile uniquement durant l'entraînement afin d'apprendre des vecteurs descripteurs capturant fidèlement la géométrie des sous-traits.

La distance euclidienne $L2$ est utilisée comme fonction de coût pour l'apprentissage du modèle SET :

$$\mathcal{L}_{SET} = \sum_{t \sim \mathcal{U}_{[0,1]}} \left\| ss_i^t - F(E_{ss_i}, t) \right\|_2 \tag{3.8}$$

Avec $ss_i^t = (x_t, y_t)$ les vraies coordonnées du sous-trait ss_i à la position t et $F(E_{ss_i}, t) = (\hat{x}_t, \hat{y}_t)$ l'estimation prédite par le MLP à t à partir de vecteur descripteur E_{ss_i} de sous-trait fourni par le Transformer. Durant l'apprentissage, quatre points sont échantillonnés aléatoirement $t \in [0, 1]$ pour le calcul de $\mathcal{L}_{SET}$.

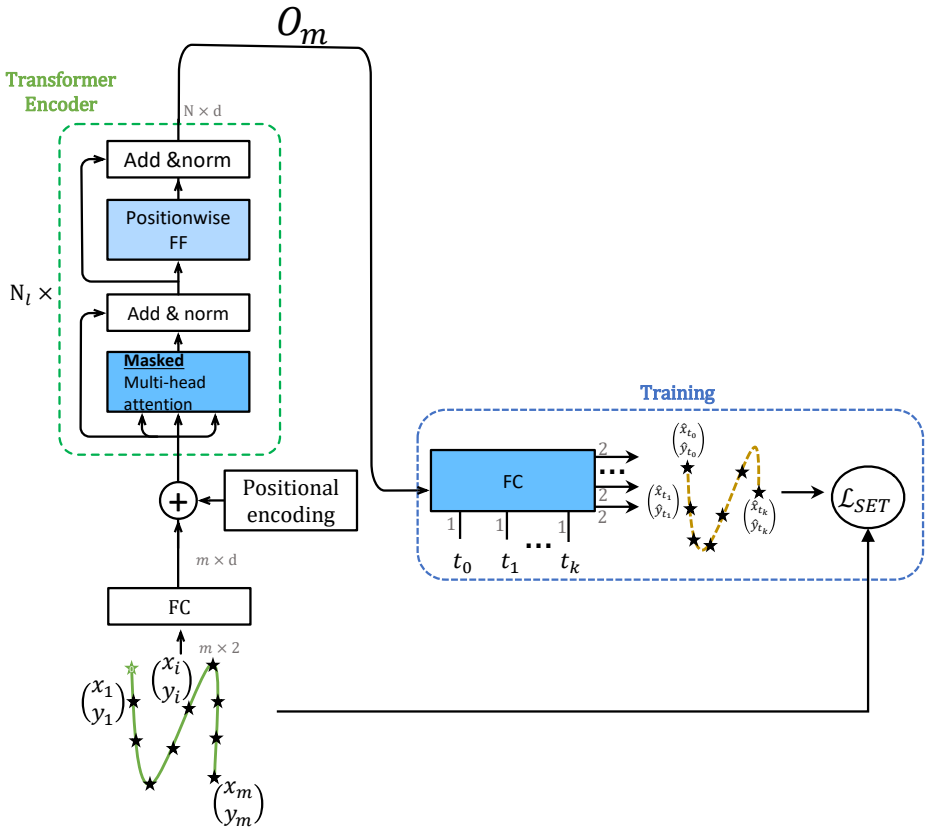


FIGURE 3.10 – Architecture du réseau SET.

Une fois les vecteurs descripteurs obtenus à l'aide de *SET*, ils sont re-localisés dans l'image en concaténant les coordonnées absolues de leur point de départ $E'_{ss_i} = [ss_i^0; E_{ss_i}] \in \mathbb{R}^{d_D+2}$. Rappelons que la position dans l'image était omise précédemment dans le calcul des vecteurs descripteurs de sous-trait. Il s'agit d'une information globale indispensable pour formuler et répondre à des requêtes sur l'ordre des sous-traits.

3.3.4 Ordonnancement de sous-traits avec SORT

Le modèle *SORT* est constitué d'un Transformer(encodeur-décodeur) avec N_l couches, des dimensions de d_{model} et d_{ff} avec N_h têtes d'attention, à l'exception de la dernière couche (sortie) du décodeur $L^{(N_l)}_{\text{decoder}}$ du Transformer possédant une seule tête d'attention ($N_h = 1$). Le Transformer *SORT* prend en entrée l'ensemble des vecteurs descripteurs des sous-traits (ensemble E'_V) et prédit en sortie la séquence $\pi(V) = S$ indiquant l'ordre des sous-traits ainsi que l'état du stylo, comme illustré par la figure 3.11. L'encodage positionnel habituellement additionné aux vecteurs descripteurs en entrée de l'encodeur Transformer est omis ici. En effet, comme il s'agit d'un ensemble, la notion d'ordre n'existe pas : la position des sous-traits dans la séquence fait partie des informations à inférer par notre modèle. Le décodeur prend en entrée la séquence des vecteurs descripteurs de sous-traits masquée E'_{SM}, initialisé par un token de début de séquence <*bos*> : $E'_{\text{SM}} = \{E'_{bos}; \bigcup_{i=1}^{|S|-1} E'_{\text{SM}_i}\}$. Il prédit en sortie les distributions de probabilité du prochain sous-trait $\hat{p}$ et l'état du stylo $\hat{q}$ à l'instant t sachant son historique $H_t = \left\{E'_{\text{SM}_k}\right\}_{k=0}^{t}$:

$$\begin{aligned} \hat{p}\left(\mathrm{X}_t = x \mid H_t; \theta\right) \\ \hat{q}\left(\mathrm{UP}_t = 1 \mid H_t; \theta\right) \end{aligned} \tag{3.9}$$

X_t est la variable aléatoire discrète associé au t-ème sous-trait de la séquence des sous-traits ordonnée parmi l'ensemble des sous-traits $x \in \{V; <\text{eos}>\}$ possible. Cette variable peut également prendre la valeur du token spécial <*eos*> pour indiquer la fin de la séquence. En pratique, les deux tokens <*bos*> et <*eos*> sont représentés par le même vecteur nul : $E'_{\text{bos}} = E'_{\text{eos}} = \mathbf{0}_{\mathbb{R}^{d_D+2}}$.

UP_t est la variable aléatoire binaire représentant une occurrence du lever du stylo après l'écriture du t-ème sous-trait et donc la fin du trait. La fin de la séquence <*eos*> est considérée aussi comme un lever du stylo.

La prédiction du prochain sous-trait est dérivée des scores d'attentions croisés (encodeur décodeur) de la couche de sortie du décodeur de *SORT*. Soit (K, V) la paire de clés valeurs obtenus à partir de la sortie de l'encodeur et Q la requête de la couche de sortie du décodeur, on a :

$$\begin{aligned} \hat{p}_{X_t} &= \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_{\text{model}}}}\right) \in \mathbb{R}^{|S|\times|V+1|} \\ \hat{q}_{\text{UP}_t} &= \mathbf{W}^T \hat{p}_{X_t} \mathbf{V} + \mathbf{b} \end{aligned} \tag{3.10}$$

Avec $\mathbf{W} \in \mathbb{R}^{d_{model}}$et $\mathbf{b} \in \mathbb{R}$ des paramètres de projection apprenables.

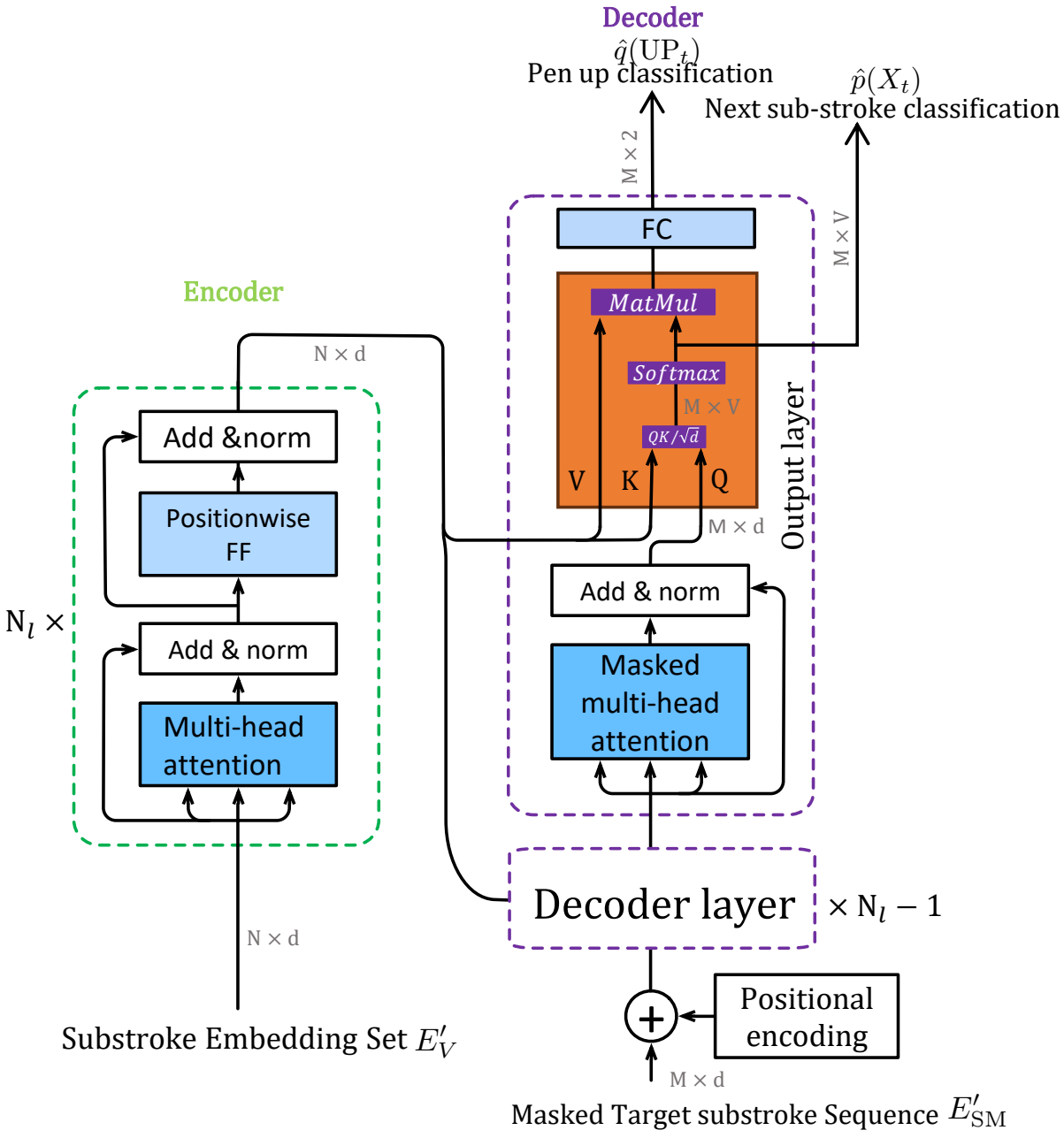


FIGURE 3.11 – Architecture du réseau SORT. La couche de sortie exprimée par l'expression 3.10 est ici représentée. Les dimensions des tenseurs en sortie des couches sont indiquées en gris. Illustration inspirée de [Zha+23].

Le modèle *SORT* est appris avec une fonction de coût multitâche $\mathcal{L}_{SORT}$ combinant une entropie croisée $\mathcal{L}_{\Pi}$ entre la distribution de probabilité du prochain sous-trait prédit $\hat{p}$ et la vérité p et une entropie croisée binaire $\mathcal{L}_{\text{UP}}$ pour la classification de l'état de stylo.

$$\begin{aligned}
\mathcal{L}_{\text{SORT}} &= \lambda_1 \mathcal{L}_{\Pi_t} + \lambda_2 \mathcal{L}_{\text{UP}_t} \\
\mathcal{L}_{\text{UP}_t} &= -\left(\mathbb{1}_{\text{UP}_t} \log(\hat{q}_{\text{UP}_t}) + (1 - \mathbb{1}_{\text{UP}_t})(\log(1 - \hat{q}_{\text{UP}_t}))\right) \\
\mathcal{L}_{\Pi_t} &= -\sum_{x \in V \cup \{eos\}} p_{X_t}(x) \log(\hat{p}_{X_t}(x)) \\
\mathcal{L}_{\Pi_t} &= -\log(\hat{p}_{X_t}(\pi(t)))
\end{aligned} \tag{3.11}$$

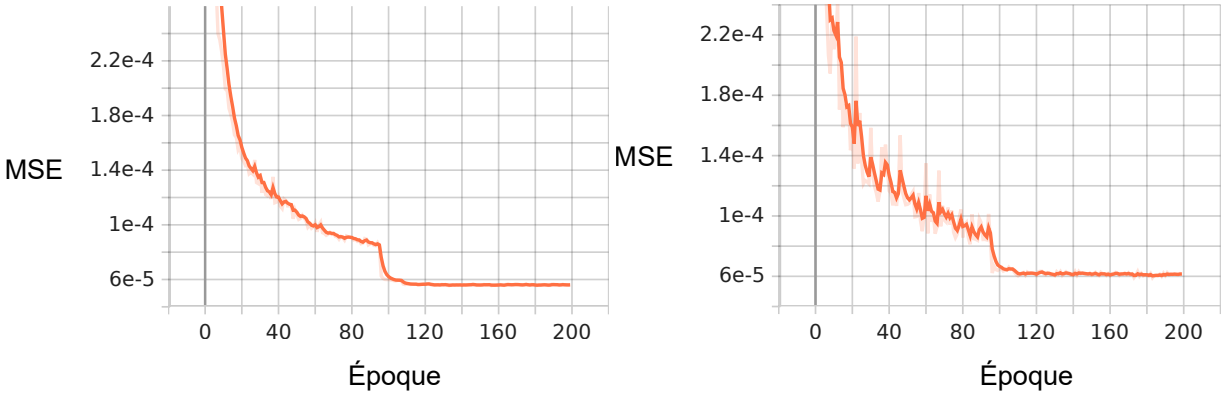
avec $(\lambda_1, \lambda_2) \in \mathbb{R}^2$, $\mathbb{1}_{condition} = 1$ si *condition* est vraie, 0 sinon. $\hat{(q)}, q$ sont les distributions de probabilités de lever du stylo prédite et vérités respectivement.

3.3.5 Apprentissage des modèles et inférence

Le réseau *SORT* est appris séparément de *SET*, l'ordre vérité des sous-traits $\pi(V)$ de l'image hors-ligne est dérivé à partir du signal en-ligne (section 3.3.2). Le Transformer encodeur de sous-trait *SET*, utilise six couches ($N_l = 6$), une dimension $d_{\mathbf{model}} = 64$ et $N_h = 4$ têtes d'attentions. Les vecteurs descripteurs de sous-traits sont de dimension $D = 8$. Les images hors-lignes sont générées synthétiquement à partir des bases en-lignes comme décrit dans la section 3.1.1, avec une épaisseur de trait aléatoire, entre deux et trois pixels.

Nous employons le réseau *UNet* [MMLM21] pour la squelettisation. Après l'extraction des sous-traits, le réseau SET est entrainé sur la base de lettre et mots IRONOFF pendant 200 époques, pour une durée totale de 8 heures. Le taux d'apprentissage est de 0.001 et la taille du lot d'apprentissage est fixée à 32. La figure 3.12 montre l'évolution de la fonction de coût $\mathcal{L}_{SET}$ durant l'apprentissage. Un sous-trait est une séquence de 39 points en moyenne.

Le modèle Transformer *SORT* utilise les mêmes meta-parametères $N_l, d_{\mathbf{model}}, N_h$ que *SET*. Plusieurs apprentissages dédiés sur les bases de mots IRONOFF, expressions CROHME19 et phrases IAM-OnDB sont effectués. La figure 3.13 montre les courbes d'apprentissages et validation pour 100 époques sur les bases ironoff, crohme et une combinaison de toutes les bases. L'apprentissage est effectué avec un lot de 10 images et dure 36 heures. La base mots et lettres isolées *IRONOFF* contient en moyenne 31 sous-traits par image. Les bases d'expressions *CROHME* et de phrases *IAM-OnDB* contiennent en moyenne 66 et



(a) Courbe d'apprentissage. (b) Courbe de validation.

FIGURE 3.12 – Courbes d'apprentissages et de validation du modèle *SET* sur la base IRONOFF. On rappelle que la fonction de coût $\mathcal{L}_{SET}$ employée ici est une MSE.

162 sous-traits par image respectivement.

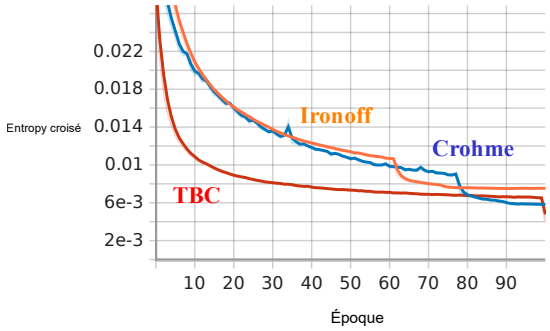
A l'inférence, nous suivons le même processus pour extraire les sous-traits de l'image hors ligne en entrée. Les vecteurs descripteurs des sous-traits sont ensuite produits à l'aide du réseau *SET*. Le réseau *SORT* prédit ensuite itérativement le prochain sous-trait et l'état du stylo correspondant. Nous sélectionnons le sous-trait avec la probabilité prédite la plus élevée comme le prochain qui sera utilisé comme entrée pour la prochaine itération. Lorsque le stylo est à l'état posé, le sous-trait courant et le sous-trait successeur sont fusionnés par un segment interpolé linéairement. Cela permet de remplir le vide laissé par la suppression de la jonction entre les deux sous-traits. L'inférence se termine lorsque le jeton spécial *eos* est prédit comme prochain sous-trait. Le résultat est une séquence de sous-traits fusionné en traits, il peut donc être considéré comme un signal en-ligne.

3.3.6 Évaluations sur les expressions hors-ligne et ligne de texte

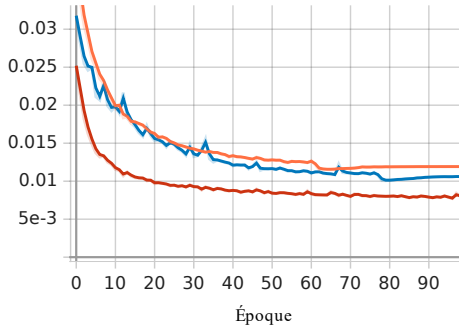
Le système est évalué à l'aide de deux métriques : le taux de reconnaissance en-ligne et la DTW (cf.3.1). Une étude comparative est réalisée entre notre approche et deux autres approches de l'état de l'art ([Cha20] et [Arc+21]) sur les jeux de données de mots et lignes cursives latins [VG+99 ; LB05] ainsi que les expressions mathématiques [Mah+19].

Comparaison quantitative

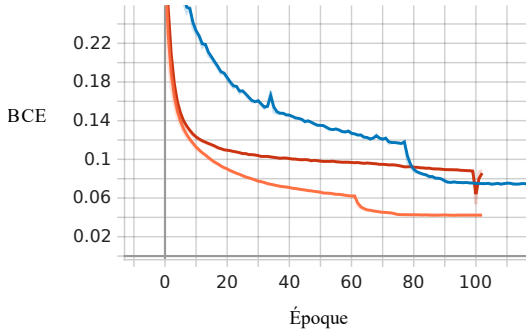
L'oracle ou l'ordre vérité des sous-traits (section 3.3.2), désigné par les lignes (d) des tables (3.8, 3.9 et 3.10), démontre une approximation très précise du signal en-ligne vérité



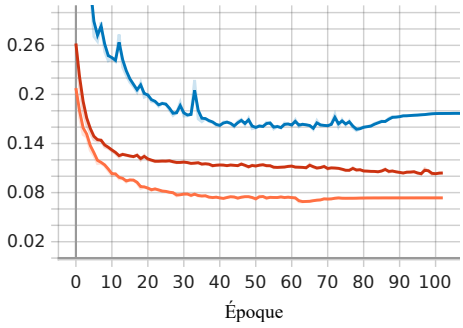
(a) Courbe d'apprentissage de l'ordonnancement des sous-traits $\mathcal{L}_{\Pi_t}$.



(b) Courbe de validation de l'ordonnancement des sous-traits $\mathcal{L}_{\Pi_t}$.



(c) Courbe d'apprentissage de la classification des levers de stylo $\mathcal{L}_{\mathrm{UP}_t}$.



(d) Courbe de validation de la classification des levers de stylo $\mathcal{L}_{\mathrm{UP}_t}$.

FIGURE 3.13 – Courbes d'apprentissages et de validation du modèle *SORT* sur les bases IRONOFF, CROHME et toutes les bases confondues (TBC). La fonction de coût $\mathcal{L}_{SORT}$ employé est une combinaison de deux fonctions de coût de classification $\mathcal{L}_{\Pi_t}$ et $\mathcal{L}_{\mathrm{UP}_t}$.

(lignes e).

La disparité entre les deux reste très faible avec un écart de DTW qui reste inférieur à 0.5 sur l'ensemble des jeux de données considérés. Cette disparité est attribuée aux perturbations induites par l'extraction des sous-traits comme mentionné précédemment.

Les résultats obtenus sur ***IRONOFF*** 3.8, le jeu de données de lettres et mots français, montre que notre méthode (c) surpasse les approches (a) et (b) de l'état de l'art sur les métriques DTW et de taux de reconnaissance 3.8.

La ligne (c) montre que notre réseau de neurones est assez léger et compte seulement 2 millions de paramètres (en incluant le CNN de squelettisation) soit 3.5 fois moins de paramètres par rapport au *CNN-LSTM* ligne (b). Bien que notre approche atteigne les

Méthode	Paramètres	DTW ↓	DTW_{seg} ↓	CRR↑	WRR↑
(a) Chan et al. [Cha20]	-	5.75	5.06	73.45	60.00
(b) CNN-BiLSTM [Arc+21]	7M	7.09	7.45	59.22	41.43
(c) SET SORT	2M	**3.25**	**2.72**	**90.85**	**81.06**
(d) Oracle	-	0.33	0.32	92.56	83.45
(e) Signal en-ligne vérité	-			93.03	83.81

TABLE 3.8 – Résultats obtenus sur le jeu de test de *IRONOFF*.

meilleures performances sur *IRONOFF*, la comparaison entre les résultats obtenus par notre approche (c) et les résultats de l'oracle (d) (cf.table 3.8), montre qu'il existe encore une marge de progression pour notre approche. Cette marge est de l'ordre de 2.5% environ pour les taux de reconnaissances (CRR et WRR) et 3 pixels environ pour DTW.

Sur **CROHME**, une comparaison quantitative entre notre approche (c) et le système à base de règle (a) de [Cha20] est réalisée sur le jeu de validation de CROHME2019 (table 3.9). L'approche *CNN-LSTM* précédemment mentionnée est exclue car elle n'a pas été prévue pour traiter des images d'expressions. La table 3.9 montre une chute drastique de performance sur les expressions mathématiques pour le modèle précèdent SET SORT (c) , appris uniquement sur les mots cursifs de *IRONOFF*. En effet, la complexité de la tâche augmente considérablement en passant sur des expressions mathématiques. Un apprentissage par transfert sous forme d'affinage du modèle pré-entraîné sur les mots (c) permet de mieux exploiter les connaissances déjà apprises pour généraliser correctement aux expressions. Par conséquence, les résultats obtenus (ligne f) sont légèrement meilleurs que l'état de l'art (a) avec 53.75% des expressions mathématiques correctement reconnues et un coût d'alignement DTW plus faible.

Méthode	DTW↓	DTW_{seg} ↓	SDTW ↓	ExpRate↑
(a) Chan et al.	16.30	16.13	6.54	52.43
(c) SET SORT	24.54	24.37	12.29	29.37
(f) affinage(*fine-tuning*) (c) sur *CROHME*	**13.75**	**13.59**	4.43	53.75
(g) affinage (f) sur *MyScript-MathDB*	13.93	13.80	**2.93**	**59.31**
(d) Oracle	0.24	0.22	0.50	69.41
(e) Signal en-ligne vérité				69.77

TABLE 3.9 – Résultats obtenus sur *CROHME2014* test set.

Nous pouvons observer que le taux de reconnaissance d'expressions est un indicateur moins fiable que le taux de reconnaissance précédemment utilisé pour les mots. Les expressions mathématiques peuvent présenter une grande variabilité dans l'ordre des symboles ce qui peut augment artificiellement la valeur de la DTW. Nous n'avons donc pas de métrique très fiable pour comparer les approches. C'est pourquoi nous proposons une modification de la métrique DTW au niveau trait ($SDTW$) qui sera indépendante de l'ordre des traits. Elle permet de mesurer la précision des prédictions des levers de stylo. Cette métrique est définie d'une manière similaire que la $SIoU$ (équation 3.7) en cherchant pour chaque trait vérité le meilleur trait prédit (équivalent à un rappel). On voit dans la table 3.9 que la valeur est nettement plus faible que la DTW. Cela signifie que les traits sont plutôt correctement extraits mais peut être pas dans l'ordre de la vérité.

Les Transformers sont connus pour nécessiter beaucoup de données d'apprentissage. Comparé aux réseaux de neurones récurrents, ils exigent des quantités de données plus volumineuses pour optimiser leur entraînement [Kap+20]. Le jeu d'entraînement de *CROHME* d'environ $10,000$ expressions mathématiques est complété avec $15,000$ expressions supplémentaires de la base de données propriétaire de MyScript (dénoté par MyScript-MathDB). On obtient ainsi des résultats (ligne g) significativement meilleurs par rapport à l'état de l'art (a), avec une amélioration de 6.88 points sur ExpRate et une extraction de traits ($SDTW$) deux fois plus précise (2.93 contre 6.54). Les mêmes conclusions sont observables sur le jeu de test de *CROHME2019*, comme montré par la table 3.10. Comme indiqué dans les lignes (g) des tables 3.10 et 3.9, une meilleure DTW au niveau signal n'est pas toujours synonyme d'une extraction de trait plus précise ou d'un meilleur taux de reconnaissance. Même si (a) obtient une DTW légèrement meilleure à moins d'un pixel près que notre modèle (g), cela ne se traduit pas par une segmentation de trait plus précise ou un meilleur taux de reconnaissance. La DTW est une métrique stricte qui impose un

Method	DTW↓	DTW_{seg} ↓	SDTW↓	ExpRate↑
(a) Chan et al.	**14.29**	**14.14**	7.19	57.01
(g) SET SORT affiné sur *MyScript-MathDB*	14.98	14.80	**3.85**	**62.00**
(d) Oracle	0.26	0.24	0.69	72.29
(e) GT online				73.13

TABLE 3.10 – Résultats sur l'ensemble de test CROHME 2019.

ordre unique des traits. Elle reste donc très sensible aux variations humaines dans l'ordre des traits. Le modèle (g) a été exposé à plus de données d'entraînement et donc apprend d'autres manières d'ordonner les traits d'un symbole mathématique.

Sur la base de phrases **IAM-onDB**, la table 3.11 montre une comparaison entre notre approche et l'état de l'art. L'apprentissage est effectué sur toutes les bases confondues (les

Méthode	**WRR**	**LineRate**
Chan et al	70.64	22.30
CNN-BiLSTM	74.36	11.08
SET SORT	91.16	60.25
Oracle	94.15	70.65
Signal original	95.32	71.83

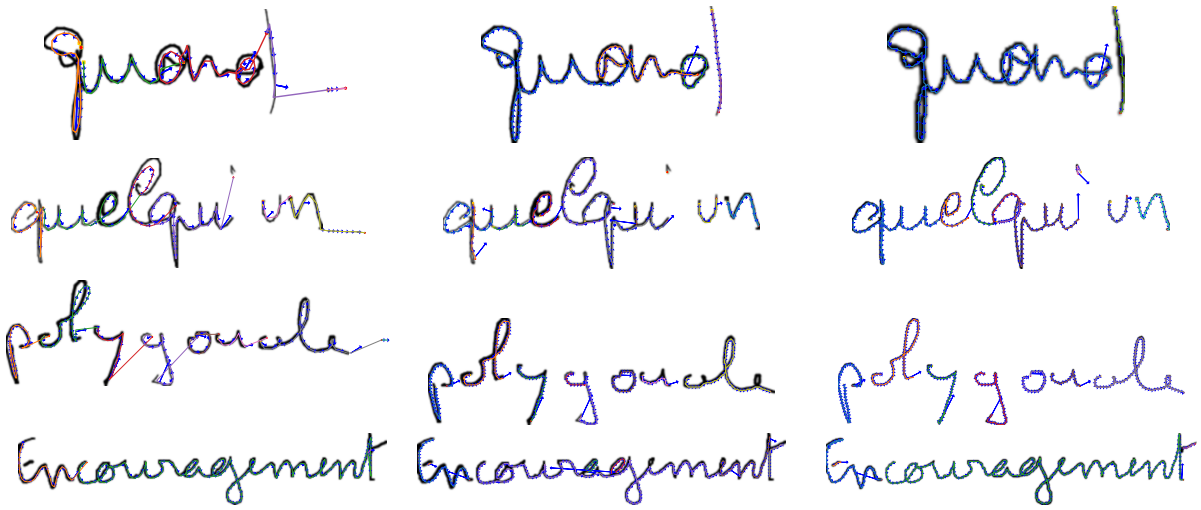
TABLE 3.11 – Résultats sur l'ensemble de test de IAM-OnDB.

bases de mots, phrases et maths). Notre réseau se distingue par sa meilleure reconnaissance des mots d'une phrase, lui permettant d'atteindre 60.25% de taux de reconnaissance de phrases (*LineRate*). En moyenne, les phrases du jeu de test contiennent plus de 6 mots, les mauvaises reconnaissances de l'approche neuronale et heuristiques contribue à leur taux de reconnaissance faible au niveau phrase. On note que la marge de progression du modèle SORT reste assez large, elle atteint 10% sur les bases d'expressions et phrases.

Comparaison qualitative

La figure 3.14 montre une évaluation qualitative entre les résultats de notre approche et l'état de l'art pour quatre échantillons aléatoires de **IRONOFF**. Cette figure illustre bien les difficultés surmontées grâce à notre approche. La prédiction des levers de stylo avec une meilleure précision permet de résoudre les erreurs de sous/sur-segmentation de trait.

Sur l'exemple de la première ligne de figure 3.14 (mot "quand"), quatre traits différents sont prédits par (a) et (b) contre deux traits attendus et correctement prédits par (c).



(a) Archibald et al. (b) Chan et al. (c) SET,SORT

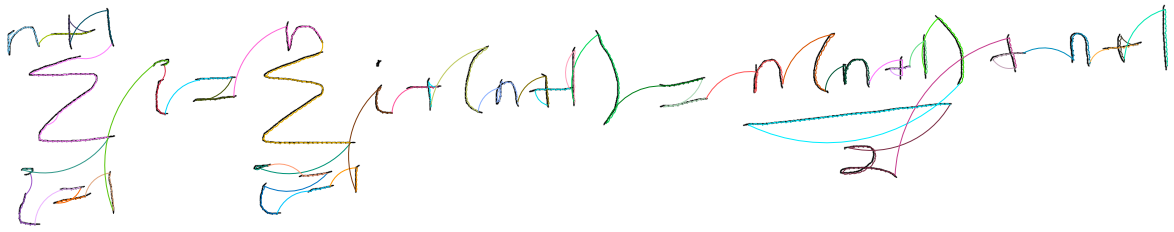
FIGURE 3.14 – Comparaison de notre approche (c) avec [Arc+21] (a) et [Cha20] (b) sur des échantillons de IRONOFF. Chaque trait est dessiné avec une couleur distincte. Les flèches bleues indiquent la direction des mouvements du stylo. Les premiers et derniers points des traits sont respectivement en jaune et en rouge.

Le problème de mauvaise couverture du squelette est atténué : certains détails ou partie de traits sont souvent absents dans la reconstruction produite par les systèmes de l'état de l'art. C'est le cas par exemple, sur la figure 3.14 où les petites boucles des caractères "e" sont omises dans les résultats de (b) (lignes 3 et 4). (a) a tendance à omettre certains traits aussi, comme l'apostrophe (ligne 2) et la barre horizontale du "E" (ligne 4). Dans certains cas, l'approche neuronale (a) prédit un trait qui dévie du squelette. Ceci est visible dans le "q" et "a" de la première ligne ainsi que la lettre "l" de la deuxième ligne.

Enfin, une meilleure résolution des jonctions est observée : les décisions aux embranchements à l'entrée des jonctions complexes ainsi que les rebroussements présentent une difficulté considérable pour l'état de l'art. Cela est visible sur le "a" de la première ligne. Le modèle (a) produit une résolution grossière en simplifiant les deux boucles par un seul trait. Le système (b) quant à lui écrit la lettre en deux traits, avec un deuxième trait moins plausible. La dernière ligne montre un autre exemple avec deux difficultés : la partie haute du "t" est écrite avec un trait tracé en double. Contrairement à (b), les deux approches neuronales (a) et (c) effectuent correctement les rebroussements. Cependant (a) produit

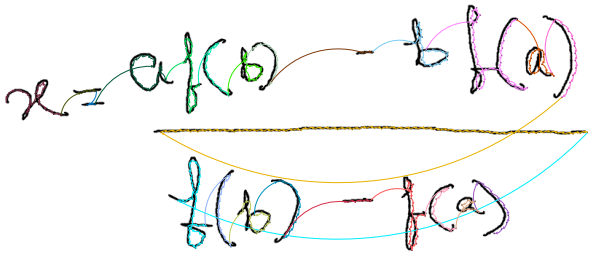
un trait de croisement incomplet au "t" et (b) prend un embranchement incorrect à la jonction entre le trait du "t" et son trait de croisement horizontale.

La figure 3.14 illustre également que le modèle *CNN-LSTM* (a) éprouve fréquemment des difficultés avec les prédictions de fin de séquence comme montré les trois premières lignes de la figure. La figure 3.15 montre une comparaison visuelle entre notre approche et l'état de l'art pour deux expressions mathématiques du jeu de test de **CROHME**. Globalement, on observe que notre méthode propose un ordre plus cohérent pour les traits

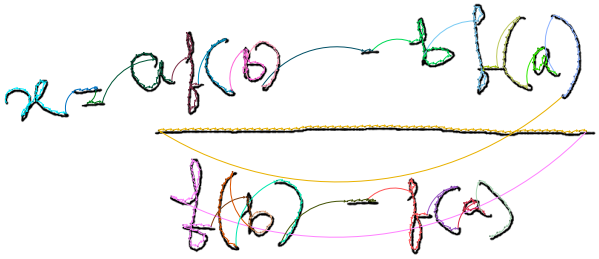


(a) Chan et al. [Cha20]

(b) SET,SORT



(c) Chan et al.



(d) SET,SORT

FIGURE 3.15 – Exemples d'inférences de [Cha20] et de notre approche sur deux échantillons de CROHME. Les rectangles rouges en pointillés montrent des erreurs de reconstruction du signal.

inter et intra symboles. En examinant la figure 3.15a, nous pouvons repérer plusieurs erreurs d'ordonnancement de traits dans l'approche (a). Le point du "i" est omis, les traits des deux symboles "i" et "=" sont mélangés de manière désordonnée et enfin le

rebroussement du premier trait du dernier "n" n'est pas présent. Le résultat de notre modèle (c) présenté dans la figure 3.15b montre une grande fiabilité en ce qui concerne les erreurs précédentes. Cependant, une erreur différente est relevée : notre réseau redessine à tort le trait vertical du premier signe "+", au lieu de dessiner le chiffre "1" après avoir tracé le Σ. Néanmoins, le modèle parvient à se rectifier et procède ensuite à un enchaînement correct les traits restants. La similarité des formes entre les deux traits combinés à leur proximité spatiale dans l'image, pourrait expliquer cette erreur.

La figure 3.15 présente également un autre exemple avec une expression rationnelle. Une extraction de traits sous-optimale dans le cas (c) provoque la fusion de la barre horizontale du "f" avec la parenthèse ouvrante. Par conséquent, le moteur de reconnaissance en ligne est induit en erreur, le faisant interpréter ce symbole comme un "H" au lieu d'un "f". Grâce à notre prédiction plus précise de l'état du stylo, nous pouvons extraire correctement les traits des deux symboles.

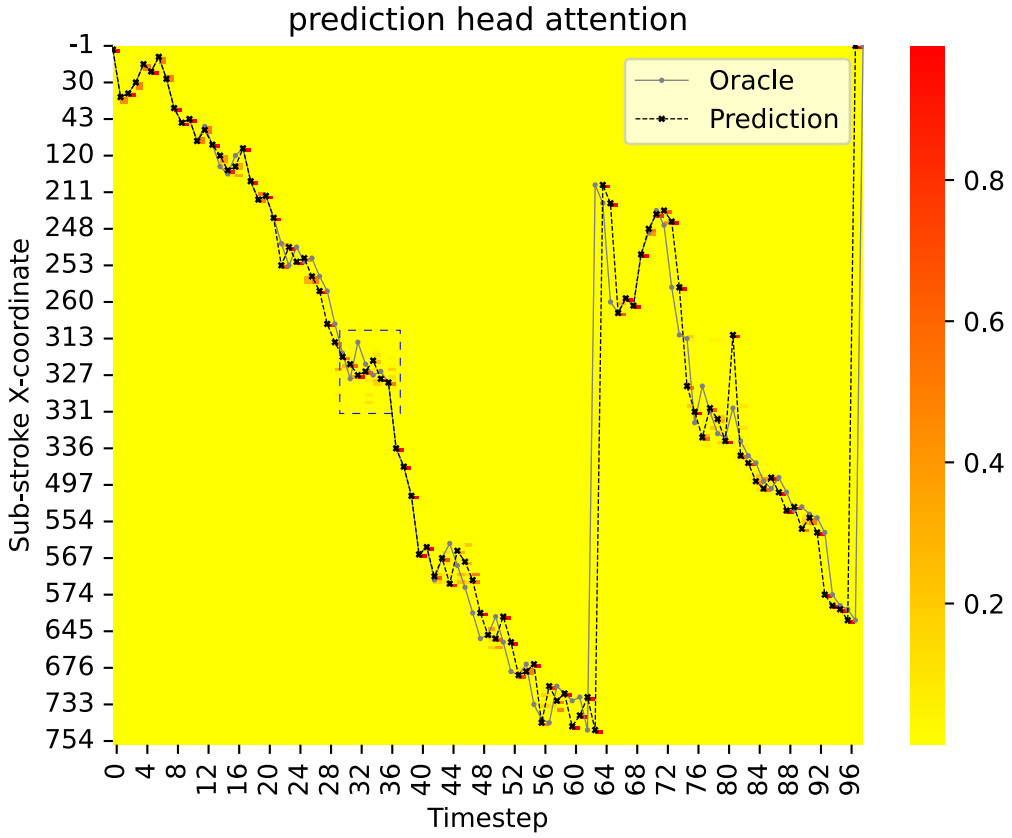
Notre modèle génère plusieurs différents styles d'ordonnancement de traits pour des situations comparables par opposition au système (c) qui propose un style souvent monotone. Par exemple, dans la figure 3.15b les indices sont prédits en dernier après les exposants et les opérateurs. C'est une manière moins courante d'écrire, mais elle qui reste tout autant plausible. Dans la figure 3.15d, notre modèle génère deux styles différents d'écriture pour la même sous-expressions ((b) et (a)) : les parenthèses et leur contenu ne sont pas toujours dans le même ordre.

La figure 3.16 montre le déroulement de l'inférence pour l'expression de la figure 3.15d. La matrice montre la distribution de probabilité prédite à chaque étape. Nous observons que le réseau a une grande confiance dans ses prédictions ce qui se manifeste par des distributions avec des petites variances, centrées autour de sous-traits adjacents. Le signal reconstruit par le réseau reflète globalement les mêmes dynamiques temporelles que le signal en ligne réel.

Étude d'ablation

La table 3.12 montre les résultats d'apprentissage pour chaque base séparément et les apprentissages sur la combinaison des bases.

Le modèle appris sur toutes les bases confondues (ligne 7) permet d'atteindre le meilleur résultat sur la base de phrases, néanmoins une perte de performance sur de plus de 2% est observée sur les formules mathématiques par rapport au modèle appris sur les bases mots et expressions (ligne 6). Cela montre une sur-spécialisation du modèle sur



(a) Carte d'attention obtenue lors de l'inférence.

(b) Ordre de l'oracle

(c) Ordre inféré par *SORT*

FIGURE 3.16 – (a) Carte de chaleur de l'attention de la couche de sortie du décodeur SORT pour l'expression de la figure 3.15d. L'axe des ordonnées représente l'ensemble des sous-traits triés de gauche à droite (avec le premier point) à des fins d'illustration uniquement. Les jetons *bos* et *eos* sont ici indiqués par un -1. Les prédictions du réseau ainsi que l'ordre de l'oracle sont dessinées au-dessus de la carte de chaleur. Un exemple de divergence entre l'ordre inféré par *SORT* (c) et l'ordre de l'oracle (b) est observable sur (a) entre les timesteps 24 et 40.

	Taux de reconnaissances	
Méthode	**Ironoff** (lettres + mots)	**Crohme** (expressions)
(1) Chan et al. 2020	60.00%	57.01%
(2)**Ironoff**	79.67%	32.74%
(3)**CROHME**	71.15%	51.04%
(4)**MS-MathDB**	74.14%	55.58%
(5)**Crohme + MS-MathDB**	74.98%	60.08%
(6)**Ironoff + Crohme + MS-MathDB**	**80.01%**	**62.21%**
(7)**Ironoff + Crohme + MS-MathDB + IAM-onDB**	79.35%	59.44%

TABLE 3.12 – Apprentissage à partir d'une combinaison des différentes bases de données et taux de reconnaissances correcte du signal en-ligne prédit, au niveau mot et expression.

la base des mots et phrases qui altère ses capacités à généraliser au mieux sur les formules mathématiques également.

La table 3.13 montre les résultats de plusieurs apprentissages avec des valeurs de pondération entre les deux fonctions de coût du modèle *SORT*. La première ligne désigne

λ_1	λ_2	ExpRate ↑
0	1	51.13
1	1	31.48
1	5	49.95
1	30	60.95
1	50	62.88
1	80	**63.89**
1	100	62.21

TABLE 3.13 – Pondération entre les deux fonctions de coût de *SORT*. λ_1 et λ_2 correspondent à la fonction de coût de classification de lever de stylo et d'ordonnancement de sous-traits respectivement. Bases d'apprentissages mots et expressions.

un modèle SORT qui ne prédit pas les levers du stylo mais uniquement l'ordonnancement des sous-traits. À l'inférence on utilise une règle simple pour prédire les levers de stylo : si deux sous-traits consécutifs ne sont pas reliés à la même jonction, alors un lever de stylo est prédit. La première ligne montre les limites de cette approche : elle ne prend pas en compte les traits qui se chevauchent et les sauts entre sous-traits singuliers (sous-trait constitué d'un point). Le meilleur résultat est obtenu avec $\lambda_2 = 80$. Ce qui montre qu'il faut donner beaucoup plus d'importance à la prédiction de l'ordre mais que la prédiction des levers ne doit pas être négligée.

Nous expérimentons aussi avec une architecture alternative à [Aks+20], dans le but de tester une simplification de l'apprentissage des vecteurs descripteurs de sous-traits et

un apprentissage simultané de *SET* et *SORT*. Le modèle modifié SET′ est obtenu avec les modifications suivante de *SET* :

— Les sous-traits ne sont pas translatés à l'origine.
— Chaque sortie du Transformer encodeur régressif est directement décodée pour produire la coordonnée du prochain point. La fonction de coût est alors la somme des erreurs *MSE*. L'embedding est ici la sortie de Transformer encodeur de taille d_{model}.
— Il n'est plus nécessaire de concaténer les coordonnées des premiers points des sous-traits au vecteur descripteurs pour construire l'entrée de *SORT*.

La table 3.14 montre l'intérêt de l'approche de [Aks+20] pour l'apprentissage de vecteurs descripteurs, particulièrement sur les expressions mathématiques qui présentent des sous-traits plus longs et plus complexes à ordonner.

	IRONOFF	CROHME
SET [Aks+20] + apprentissage séparé	**80.01**	**62.21**
SET′ + apprentissage simultané	79.99	58.85

TABLE 3.14 – Comparaison entre un apprentissage simultané avec SET′ et separé des Transformers. Bases d'apprentissages : mots et expressions mathématiques.

Analyse d'erreurs

L'analyse des erreurs sur la base d'expressions mathématiques a mis en avant deux types d'erreurs qui présentent le plus de difficulté pour notre approche :

— Les erreurs problématiques d'omissions de traits : parfois, le modèle SORT omet les sous-traits d'un trait. Il s'agit du problème le plus fréquent et problématique car il induit des erreurs de reconnaissance ou des différences visuelles fortes.
— Les erreurs de traits imaginaires : Le modèle SORT prédit dans certain cas plusieurs fois le même sous-trait ou alors des traits imaginaires à cause d'une prédiction incorrecte de poser de stylo. La table 3.15 montre un comparatif entre les reconstructions prédites par *SORT* et les signaux en-ligne orignaux sur la base de test de CROHME2019.

142 expressions en-lignes parmi les 861 expression dont le signal orignal est reconnue avec le moteur de reconnaissance en-ligne ont une reconstruction en-ligne erronée par notre approche SET et SORT. Le nombre moyen de symboles par image sur ces 142 expressions

		Avec signal original		Total
		Reconnu	Incorrect	
Avec signal reconstruit	Reconnu	719	22	741
	Incorrect	**142**	307	449
Total		861	329	1190

TABLE 3.15 – Tableau comparatif des résultats, sur la base de test de CROHME2019, de reconnaissances des expressions mathématiques en-ligne originale et reconstruits avec notre modèle *SORT*.

est de 10 et le nombre de symbole omis en moyenne est de 1.7 soit moins du sixième du nombre moyen de symboles par image. La figure 3.17 montre un exemple d'inférence erroné de notre modèle. Ici, parmi les douzes symboles vérités, deux sont impossibles à reconnaître à causes des traits omis. Sur ces 142 expressions, la figure 3.18 montre le

FIGURE 3.17 – Exemple d'inférence incorrecte de notre approche. En rouge les traits omis et en violet un trait en trop à cause d'une mauvaise prédiction de poser de stylo.

nombre moyen de symboles omis par le système de reconstruction parmi les expressions totalisant un nombre donné de symboles. On remarque une tendance à la hause pour le nombre d'omission de symboles lorsque les expressions sont plus longues, particulièrement au-delà de 20 symboles. Il est difficile de conclure quant à la prévalence de ces erreurs d'omission sur des contextes d'écritures plus large, comme par exemple les expressions multilignes [Gao+23].

3.3.7 Conclusion

Dans cette section, nous avons présenté une nouvelle approche avec Transformer au niveau des sous-traits pour reconstruire les mouvements du stylo à partir d'une écriture hors-ligne. Nous abandonnons ici le paradigme centré sur l'utilisation des caractéristiques visuelles obtenues de l'image hors-ligne à l'aide d'un CNN. Nous modélisons l'information

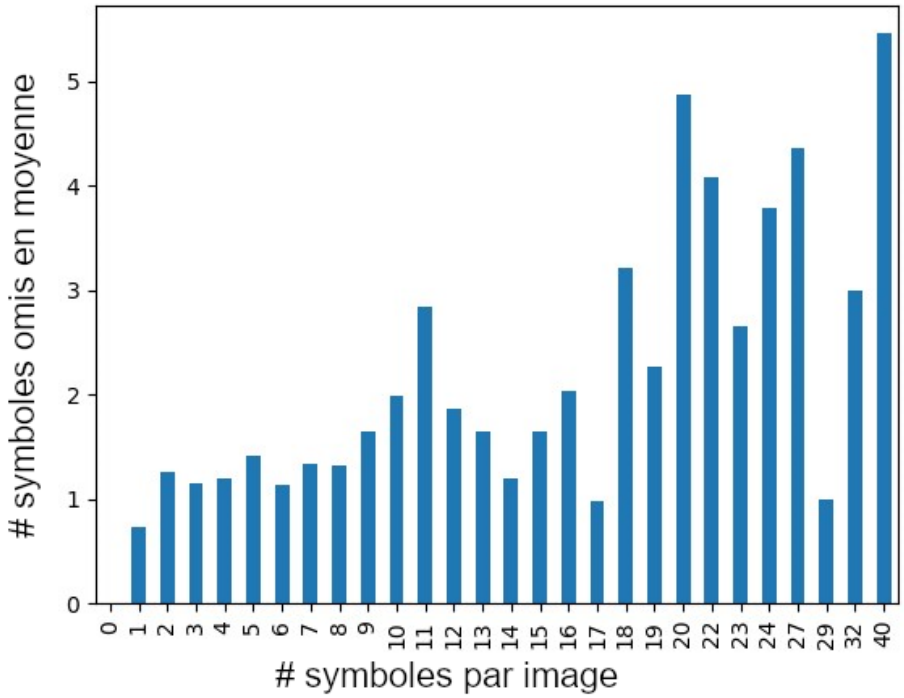


FIGURE 3.18 – Nombre moyen de symboles omis en moyenne en fonction du nombre de symboles par image des 142 expressions erronées mentionnées précédemment.

portée par les pixels de l'image à un plus haut niveau, celui du niveau sous-trait. Un sous-trait est un segment orienté non ambiguë du squelette reliant deux jonctions. Notre méthode repose uniquement sur l'information positionnelle et temporelle portée par les sous-traits sans caractéristiques visuelles. Notre approche Transformer se compose de deux modules :

— SET, un Transformer fournissant un vecteurs descripteurs pour chaque sous-trait, de taille fixe quel que soit la longueur du sous-trait. Ce vecteur descripteur paramétrise fidèlement la courbe du sous-trait.

— SORT, un Transformer prédisant la permutation correcte de l'ensemble des sous-traits grâce à leur vecteurs descripteurs. Il prédit également les levers du stylo.

Notre approche surpasse l'état de l'art sur les mots latins et les expressions mathématiques.

Chapitre 4

Conclusion

Dans cette thèse, nous avons proposé deux approches différentes de réseaux de neurones visant à reconstruire la trajectoire du stylo à partir d'une image d'écriture hors-ligne. L'objectif était de montrer leur capacité à généraliser au-delà des petites images de symboles isolés à des contenus manuscrits plus larges tels que les mots, les lignes ou encore du contenu bidimensionnel comme les expressions mathématiques. Il était également essentiel que les reconstructions produites par notre système soient aussi fidèles que possible, ce qui faciliterait une éventuelle utilisation de notes hors-ligne converties en notes en-ligne sur le logiciel de prise de notes sur tablette Nebo.

La section 3.2 présente l'approche bout-en-bout avec un réseau de neurones entièrement convolutif pour l'extraction de traits à partir d'images hors-ligne d'expressions mathématiques de taille variable [MMLM21]. Ce travail a exploré l'extension de l'approche CNN décrite dans [ZYT18] au-delà des symboles isolés. Le modèle proposé remplace le précédent CNN par un *UNet* (FCNN) multitâche, capable de prédire simultanément la position du stylo, le squelette, les extrémités des traits et les levers de stylo à chaque itération. Cette approche était la première approche neuronale dans la littérature à aborder l'extraction des traits à partir d'une image de résolution variable, elle a démontré de bons résultats de squelettisation mais en revanche une extraction de trait sous-optimale. Les expériences réalisées sur les expressions mathématiques ont montré une tendance du modèle à sur-segmenter les traits, en raison d'une part du contexte spatiale et temporel insuffisant et d'autres parts les limitations de modélisation temporelle intrinsèque à l'architecture CNN.

Le travail précédent a été exploité dans le chapitre 3.3 pour extraire des sous-traits automatiquement avec un squelette inféré par le *UNet*. Une modélisation au niveau sous-traits avec des Transformers constituée d'un encodeur de sous-traits pour extraire leurs vecteurs caractéristiques (SET) et d'un décodeur (SORT) pour ordonner les différents sous-traits ainsi que la prédiction de lever de stylo. Cette approche s'est montrée meilleure que l'état de l'art sur les bases de données de lignes de textes et d'expressions mathéma-

tiques. Elle a surmonté plusieurs limitations relevées dans la littérature : grâce à une prédiction précise des moments où le stylo est levé, une segmentation des traits très précise a été obtenue. De plus, une meilleure cohérence dans l'ordonnancement a été réalisée, tout en utilisant moins de paramètres par rapport à d'autres approches neuronales. Le Transformer a clairement démontré sa capacité à capturer la grande diversité et variabilité dans l'ordre et les directions des traits parmi différents individus. De plus la complexité quadratique du Transformer est maîtrisée grâce à l'utilisation réduite aux sous-traits du squelette.

Perspectives à courts termes

L'approche Transformer au niveau des sous-traits a été appliquée exclusivement aux langues latines (français, anglais) ainsi qu'aux symboles mathématiques ayant un sens d'écriture de gauche à droite et un ductus de lettre similaire. Si les expressions mathématiques ont montré plus de variabilité dans l'ordre des tracés, il serait pertinent d'évaluer la capacité du modèle à traiter d'autres types de systèmes d'écriture. Par exemple, l'arabe, caractérisé par un sens d'écriture opposé et un ductus de lettre plus complexe (quatre formes différentes par lettre), ou les idéogrammes chinois, comprenant un très grand nombre de symboles aux tracés complexes avec un nombre de traits important (12 en moyenne par symbole). On pourrait également envisager d'explorer des systèmes d'écriture anciens, tels que le grec boustrophédon. La reconstruction du ductus pour le grec ancien pourrait aider à la reconnaissance du scripteur des documents anciens.

La sortie du modèle Transformer se présente sous la forme d'une séquence de sous-traits extraits de l'image, pouvant potentiellement donner lieu à une écriture saccadée (due à la nature discrète des sous-traits). Il ne s'agit pas d'un signal en ligne continue, il peut s'avérer nécessaire d'effectuer une étape de post-traitement incluant un lissage et un embellissement afin d'obtenir une écriture plus fluide et esthétique.

Perspectives à longs termes

Notre approche Transformer s'appuie sur une étape de squelettisation correcte pour pouvoir récupérer tous les sous-traits potentiellement suivis par les mouvements du stylo. Le Transformer prédit ensuite une permutation de cet ensemble de traits. Cette sortie intermédiaire peut ensuite être transformée en un signal en-ligne avec un post-traitement.

Une perspective intéressante pour l'amélioration de cette approche serait d'explorer des modèles bout en bout capable de prédire la trajectoire du stylo directement à partir de l'image d'écriture hors-ligne sans nécessiter des étapes de pré-traitement pour la squelettisation ni de post-traitement pour avoir un signal. La solution actuelle sépare le FCNN de squelettisation de l'image hors-ligne du reste du modèle de reconstruction du signal en-ligne. Cette séparation est principalement due à l'étape de découpage en sous-traits du squelette, qui est réalisée de manière algorithmique à l'aide de calculs non-différentiables. En conséquence, il est impossible de concevoir une approche bout-en-bout pour cette solution. Pour avancer vers une solution bout-en-bout, il serait nécessaire de repenser cette étape intermédiaire d'extraction de sous-traits pour qu'elle fasse partie de l'apprentissage du réseau de neurones. Il sera aussi important d'avoir en sortie directement un signal en-ligne (et éviter les post-traitements d'embellissement) et non une représentation intermédiaire sous forme de séquence de sous-traits d'images.

L'objectif ultime est sans aucun doute d'évoluer vers un système capable de convertir une page entière d'un document hors-ligne en un document en-ligne. Cependant, même si l'approche Transformer ne présente pas de contrainte structurelle pour traiter des images de grande taille et riches en contenu, les limitations déjà observées concernant les séquences longues de lignes de texte, ainsi que la complexité quadratique en temps et en espace du modèle Transformer, freine encore l'utilisation au niveau du document de ce modèle efficace au niveau phrase.

TABLE DES FIGURES

1.1 Exemple du ductus d'écriture de lettres de l'alphabet latin avec un feutre. 7
1.2 Ductus de l'écriture boustrophédon et exemple d'ancien Sabéen. 8
1.3 systèmes d'écriture cursive D'Nealian et Zaner-Bloser. 9
1.4 Chaîne de traitements pour la conversion hors-ligne vers en-ligne. 14

2.1 Pipeline de Transfert de style d'écriture hors-ligne à hors-ligne. 18
2.2 Variantes de ductus générant la même image hors-ligne. 20
2.3 Alignement linéaire $RMSE$ et DTW. 24
2.4 Collecte et alignement de donnée hors-ligne et en-ligne. 25
2.5 Rastérisation avec une épaisseur de trait variable. 26
2.6 Architecture FCNN encodeur-décodeur de [SS+16]. 27
2.7 Architecture FCNN encodeur-décodeur de [SS+16]. 28
2.8 Illustration des portes du LSTM. Source de l'image : Wikipédia. 29
2.9 Reconnaissance et Synthèse d'écriture avec les LSTMs. 30
2.10 Modèle LSTM de synthèses d'écriture en-ligne. 30
2.11 Architecture Transformer encodeur-décodeur [Dev+19]. 32
2.12 Module d'attention multitêtes. 34
2.13 Architecture CNN-Transformer pour la reconnaissance d'écriture. 35
2.14 Exemple d'ordre de lecture de la base de données [Wan+21]. 36
2.15 Architecture du réseau LayoutReader. 37
2.16 Exemple de construction de graphe de segment. 40
2.17 La bonne trajectoire de stylo ne minimise pas toujours la courbure totale. . 41
2.18 Découpage X-Y des traits et ordonnancement des traits. 43
2.19 Champ récepteur d'un vecteur de la séquence de caractéristiques. 44
2.20 Modèle CNN et algorithme d'inférence itérative. 45
2.21 Modèle CNN-BiLSTM 47

3.1 Image hors-ligne synthétique de dimensions 100×180. 54
3.2 Distance point à segment. 55

3.3 Cas particulier des segments entre deux traits consécutifs. 55
3.4 Entrées et sorties de notre modèle CNN. 59
3.5 Architecture du réseau FCN pour l'exatraction de traits. 60
3.6 Visualisation des résultats du modèle FCN. 65
3.7 Vue d'ensemble de l'approche SET SORT. 68
3.8 Extraction des sous-traits de l'image hors-ligne. 69
3.9 Transformer au niveau pixel. 72
3.10 Architecture du réseau SET. 74
3.11 Architecture du réseau SORT. 76
3.12 Courbes d'apprentissages et de validation du modèle *SET*. 78
3.13 Courbes d'apprentissages et de validation du modèle *SORT*. 79
3.14 Résultats sur IRONOFF de l'état de l'art et notre approche. 83
3.15 Exemples d'inférences de [Cha20] et de notre approche sur CROHME. . . . 84
3.16 Carte de chaleur de l'attention de la couche de sortie de SORT. 86
3.17 Exemple d'inférence incorrecte de notre approche. 89
3.18 Nombre moyen de symboles omis en moyenne en fonction du nombre de symboles pour 142 expressions avec une reconstruction erronées. 90

LISTE DES TABLEAUX

2.1 Résultats des travaux de reconstruction de trajectoire de stylo à partir d'image. 49

3.1 Bases de données publiques considérées. 53
3.2 Évaluation de différentes stratégies d'échantillonnage. 56
3.3 Évaluation des approches de l'état de l'art. 57
3.4 Résultat d'évaluation de `PPNet` et `PSNet`. 62
3.5 Évaluation et comparaison de l'extraction de traits sur UNIPEN. 63
3.6 Résultat de notre approche pour différentes tailles d'expressions. 64
3.7 Impact des caractéristiques visuelles et positionnels sur les résultats du Transformer. 72
3.8 Résultats obtenus sur le jeu de test de *IRONOFF*. 80
3.9 Résultats obtenus sur *CROHME2014* test set. 81
3.10 Résultats sur l'ensemble de test CROHME 2019. 82
3.11 Résultats sur l'ensemble de test de IAM-OnDB. 82
3.12 Résultats de différentes combinaisons de bases d'apprentissages. 87
3.13 Résultats de différentes valeurs de pondérations des fonctions de coût de SORT. 87
3.14 Comparaison entre un apprentissage simultané avec SET′ et séparé des Transformers. 88
3.15 Comparatif entre les reconstructions prédites par *SORT* et les signaux en-ligne orignaux sur la base de test de CROHME2019. 89

Publications de l'auteur

[MMLM21] Elmokhtar Mohamed Moussa, Thibault Lelore et Harold Mouchère, « Applying End-to-End Trainable Approach on Stroke Extraction in Handwritten Math Expressions Images », in : *Document Analysis and Recognition – ICDAR 2021*, sous la dir. de Josep Lladós, Daniel Lopresti et Seiichi Uchida, Lecture Notes in Computer Science, Cham : Springer International Publishing, 2021, p. 445-458, isbn : 978-3-030-86334-0, doi : 10.1007/978-3-030-86334-0_29.

[MMLM23a] Elmokhtar Mohamed Moussa, Thibault Lelore et Harold Mouchère, « Point to Segment Distance DTW for Online Handwriting Signals Matching : » in : *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods*, 12th International Conference on Pattern Recognition Applications and Methods, Lisbon, Portugal : SCITEPRESS - Science and Technology Publications, 2023, p. 850-855, isbn : 978-989-758-626-2, doi : 10.5220/0011672600003411, url : https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0011672600003411 (visité le 07/09/2023).

[MMLM23b] Elmokhtar Mohamed Moussa, Thibault Lelore et Harold Mouchère, « SET, SORT ! A Novel Sub-stroke Level Transformers for Offline Handwriting to Online Conversion », in : *Document Analysis and Recognition - ICDAR 2023*, sous la dir. de Gernot A. Fink et al., Lecture Notes in Computer Science, Cham : Springer Nature Switzerland, 2023, p. 81-97, isbn : 978-3-031-41676-7, doi : 10.1007/978-3-031-41676-7_5.

À paraître

[MMLM] Elmokhtar Mohamed Moussa, Thibault Lelore et Harold Mouchere, « RECONSTRUCTING INK POINT SEQUENCES », demande de brev. europ. 23305923.7, MYSCRIPT SAS, filed.

Bibliographie

[MO14] Pam A. Mueller et Daniel M. Oppenheimer, « The Pen Is Mightier Than the Keyboard : Advantages of Longhand Over Laptop Note Taking », in : *Psychological Science* 25.*6* (1er juin 2014), p. 1159-1168, issn : 0956-7976, doi : 10.1177/0956797614524581, url : https://doi.org/10.1177/0956797614524581 (visité le 15/10/2023).

[MDR19] Kayla Morehead, John Dunlosky et Katherine A. Rawson, « How Much Mightier Is the Pen than the Keyboard for Note-Taking ? A Replication and Extension of Mueller and Oppenheimer (2014) », in : *Educational Psychology Review* 31.*3* (1er sept. 2019), p. 753-780, issn : 1573-336X, doi : 10.1007/s10648-019-09468-2, url : https://doi.org/10.1007/s10648-019-09468-2 (visité le 16/10/2023).

[Urr+21] Heather L. Urry et al., « Don't Ditch the Laptop Just Yet : A Direct Replication of Mueller and Oppenheimer's (2014) Study 1 Plus Mini Meta-Analyses Across Similar Studies », in : *Psychological Science* 32.*3* (1er mars 2021), p. 326-339, issn : 0956-7976, doi : 10.1177/0956797620965541, url : https://doi.org/10.1177/0956797620965541 (visité le 16/10/2023).

[Leb22] Marie Lebrisse, « Rediger et Prendre Des Notes : : Impacts de Nebo® et MyScript Iink® », Thèse soutenue, Poitiers, 2022, url : https://www.theses.fr/s221871 (visité le 19/09/2023).

[Luo+18] Linlin Luo et al., « Laptop versus Longhand Note Taking : Effects on Lecture Notes and Achievement », in : *Instructional Science* 46.*6* (1er déc. 2018), p. 947-971, issn : 1573-1952, doi : 10.1007/s11251-018-9458-0, url : https://doi.org/10.1007/s11251-018-9458-0 (visité le 16/10/2023).

[Fer+11] Björn Fermgård et al., « Digital Pen », brev. amér. 20110310066A1, Anoto AB, 22 déc. 2011, url : https://patents.google.com/patent/US20110310066A1/en (visité le 17/10/2023).

[PP99] R. Plamondon et C.M. Privitera, « The Segmentation of Cursive Handwriting : An Approach Based on off-Line Recovery of the Motor-Temporal Information », in : *IEEE Transactions on Image Processing* 8.*1* (jan. 1999), p. 80-91, issn : 1941-0042, doi : 10.1109/83.736691.

[May+20] Martin Mayr et al., « Spatio-Temporal Handwriting Imitation », 23 mars 2020, arXiv : 2003.10593 [cs], URL : http://arxiv.org/abs/2003.10593 (visité le 19/03/2021).

[VG+99] C. Viard-Gaudin et al., « The IRESTE On/Off (IRONOFF) Dual Handwriting Database », in : *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99 (Cat. No.PR00318)*, Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99 (Cat. No.PR00318), sept. 1999, p. 455-458, DOI : 10.1109/ICDAR.1999.791823.

[Xie+23] Yejing Xie et al., « ICDAR 2023 CROHME : Competition on Recognition of Handwritten Mathematical Expressions », in : *Document Analysis and Recognition - ICDAR 2023*, sous la dir. de Gernot A. Fink et al., Lecture Notes in Computer Science, Cham : Springer Nature Switzerland, 2023, p. 553-565, ISBN : 978-3-031-41679-8, DOI : 10.1007/978-3-031-41679-8_33.

[Rou07] Laëtitia Rousseau, « Reconnaissance d'écriture manuscrite hors-ligne par reconstruction de l'ordre du tracé en vue de l'indexation de documents d'archives », INSA Rennes, 18 juin 2007.

[NDPH05] Emli-Mari Nel, J.A. Du Preez et B.M. Herbst, « Estimating the Pen Trajectories of Static Signatures Using Hidden Markov Models », in : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.*11* (nov. 2005), p. 1733-1746, ISSN : 1939-3539, DOI : 10.1109/TPAMI.2005.221.

[QNY06] Yu Qiao, M. Nishiara et M. Yasuhara, « A Framework Toward Restoration of Writing Order from Single-Stroked Handwriting Image », in : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.*11* (nov. 2006), p. 1724-1737, ISSN : 1939-3539, DOI : 10.1109/TPAMI.2006.216.

[Boc+93] G. Boccignone et al., « Recovering Dynamic Information from Static Handwriting », in : *Pattern Recognition* 26.*3* (mars 1993), p. 409-418, ISSN : 00313203, DOI : 10.1016/0031-3203(93)90168-V, URL : https://linkinghub.elsevier.com/retrieve/pii/003132039390168V (visité le 06/06/2023).

[DR95] David S. Doermann et Azriel Rosenfeld, « Recovery of Temporal Information from Static Images of Handwriting », in : *International Journal of Computer Vision* 15.*1* (1er juin 1995), p. 143-164, ISSN : 1573-1405, DOI : 10.1007/BF01450853, URL : https://doi.org/10.1007/BF01450853 (visité le 15/06/2023).

[KY00] Y. Kato et M. Yasuhara, « Recovery of Drawing Order from Single-Stroke Handwriting Images », in : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.*9* (sept. 2000), p. 938-949, ISSN : 1939-3539, DOI : 10.1109/34.877517.

[Jäg98] Stefan Jäger, *Recovering dynamic information from static, handwritten word images :[bridging the gap between on-line and off-line handwriting recognition]*, Fölbach, 1998, URL : https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=c86aca2884847347c5bcb2e04294a493758f721a.

[HAMB13] Abdelâali Hassaïne, Somaya Al Maadeed et Ahmed Bouridane, « ICDAR 2013 Competition on Handwriting Stroke Recovery from Offline Data », in : *2013 12th International Conference on Document Analysis and Recognition*, 2013 12th International Conference on Document Analysis and Recognition, août 2013, p. 1412-1416, DOI : 10.1109/ICDAR.2013.285.

[Tri] « Triangulation Based Skeletonization and Trajectory Recovery for Handwritten Character Patterns », in : *KSII Transactions on Internet and Information Systems* 9.*1* (31 jan. 2015), ISSN : 19767277, DOI : 10.3837/tiis.2015.01.022, URL : http://www.itiis.org/digital-library/manuscript/940 (visité le 29/06/2023).

[Dia+22] Moises Diaz et al., « Writing Order Recovery in Complex and Long Static Handwriting », in : *International Journal of Interactive Multimedia and Artificial Intelligence* 7.*4* (2022), p. 171, ISSN : 1989-1660, DOI : 10.9781/ijimai.2021.04.003, URL : https://www.ijimai.org/journal/sites/default/files/2022-05/ijimai_7_4_15.pdf (visité le 29/06/2023).

[SC78] H. Sakoe et S. Chiba, « Dynamic Programming Algorithm Optimization for Spoken Word Recognition », in : *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26.*1* (fév. 1978), p. 43-49, ISSN : 0096-3518, DOI : 10.1109/TASSP.1978.1163055.

[NV06] Ralph Niels et Louis Vuurpijl, « Automatic trajectory extraction and validation of scanned handwritten characters », in : *Tenth International Workshop on Frontiers in Handwriting Recognition*, Suvisoft, 2006.

[Bhu+18] A. Kumar Bhunia et al., « Handwriting Trajectory Recovery Using End-to-End Deep Encoder-Decoder Network », in : *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018 24th International Conference on Pattern Recognition (ICPR), août 2018, p. 3639-3644, doi : 10.1109/ICPR.2018.8546093.

[Arc+21] Taylor Archibald et al., « TRACE : A Differentiable Approach to Line-Level Stroke Recovery for Offline Handwritten Text », in : *Document Analysis and Recognition – ICDAR 2021*, sous la dir. de Josep Lladós, Daniel Lopresti et Seiichi Uchida, t. 12823, Cham : Springer International Publishing, 2021, p. 414-429, isbn : 978-3-030-86333-3 978-3-030-86334-0, doi : 10.1007/978-3-030-86334-0_27, url : https://link.springer.com/10.1007/978-3-030-86334-0_27 (visité le 19/09/2022).

[ZYT18] Bocheng Zhao, Minghao Yang et Jianhua Tao, « Pen Tip Motion Prediction for Handwriting Drawing Order Recovery Using Deep Neural Network », in : *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018 24th International Conference on Pattern Recognition (ICPR), Beijing : IEEE, août 2018, p. 704-709, isbn : 978-1-5386-3788-3, doi : 10.1109/ICPR.2018.8546086, url : https://ieeexplore.ieee.org/document/8546086/ (visité le 18/11/2020).

[Cha20] Chungkwong Chan, « Stroke Extraction for Offline Handwritten Mathematical Expression Recognition », in : *IEEE Access* 8 (2020), p. 61565-61575, issn : 2169-3536, doi : 10.1109/ACCESS.2020.2984627, url : https://ieeexplore.ieee.org/document/9051736/ (visité le 21/06/2023).

[HE17] David Ha et Douglas Eck, *A Neural Representation of Sketch Drawings*, 19 mai 2017, arXiv : 1704.03477 [cs, stat], url : http://arxiv.org/abs/1704.03477 (visité le 03/07/2023), preprint.

[Guo+19] Yi Guo et al., « Deep Line Drawing Vectorization via Line Subdivision and Topology Reconstruction », in : *Computer Graphics Forum* 38.7 (oct. 2019), p. 81-90, issn : 0167-7055, 1467-8659, doi : 10.1111/cgf.13818,

URL : https://onlinelibrary.wiley.com/doi/10.1111/cgf.13818 (visité le 28/06/2023).

[Kov07] Bence KOVARI, « Time-Efficient Stroke Extraction Method for Handwritten Signatures », in : *Applied Categorical Structures - ACS* (1er jan. 2007), p. 157-161, ISSN : 978-960-6766-18-3.

[Gra14] Alex GRAVES, *Generating Sequences With Recurrent Neural Networks*, 5 juin 2014, arXiv : 1308.0850 [cs], URL : http://arxiv.org/abs/1308.0850 (visité le 27/06/2023), preprint.

[Mad+22] Boraq MADI et al., « HST-GAN : Historical Style Transfer GAN for Generating Historical Text Images », in : *Document Analysis Systems*, sous la dir. de Seiichi UCHIDA, Elisa BARNEY et Véronique EGLIN, Lecture Notes in Computer Science, Cham : Springer International Publishing, 2022, p. 523-537, ISBN : 978-3-031-06555-2, DOI : 10.1007/978-3-031-06555-2_35.

[GBC16] Ian GOODFELLOW, Yoshua BENGIO et Aaron COURVILLE, *Deep Learning*, MIT Press, 2016.

[Zha+23] Aston ZHANG et al., *Dive into Deep Learning*, 22 août 2023, DOI : 10.48550/arXiv.2106.11342, arXiv : 2106.11342 [cs], URL : http://arxiv.org/abs/2106.11342 (visité le 20/09/2023), preprint.

[SS+16] Edgar SIMO-SERRA et al., « Learning to Simplify : Fully Convolutional Networks for Rough Sketch Cleanup », in : *ACM Transactions on Graphics* 35.*4* (11 juill. 2016), p. 1-11, ISSN : 0730-0301, 1557-7368, DOI : 10.1145/2897824.2925972, URL : https://dl.acm.org/doi/10.1145/2897824.2925972 (visité le 03/07/2023).

[Gra+09] Alex GRAVES et al., « A Novel Connectionist System for Unconstrained Handwriting Recognition », in : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.*5* (mai 2009), p. 855-868, ISSN : 1939-3539, DOI : 10.1109/TPAMI.2008.137, URL : https://ieeexplore.ieee.org/abstract/document/4531750 (visité le 17/11/2023).

[Bis94] Christopher M. BISHOP, *Mixture Density Networks*, 1994, URL : https://publications.aston.ac.uk/id/eprint/373/ (visité le 17/11/2023).

[Vas+17] Ashish VASWANI et al., *Attention Is All You Need*, 5 déc. 2017, arXiv : 1706.03762 [cs], URL : http://arxiv.org/abs/1706.03762 (visité le 03/07/2023), preprint.

[Dev+19] Jacob DEVLIN et al., « BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding », in : *Proceedings of the 2019 Conference of the North*, Proceedings of the 2019 Conference of the North, Minneapolis, Minnesota : Association for Computational Linguistics, 2019, p. 4171-4186, DOI : 10.18653/v1/N19-1423, URL : http://aclweb.org/anthology/N19-1423 (visité le 05/11/2023).

[Aks+20] Emre AKSAN et al., « CoSE : Compositional Stroke Embeddings », in : *Advances in Neural Information Processing Systems* (2020), URL : https://proceedings.neurips.cc/paper/2020/file/723e8f97fde15f7a8d5ff8d558ea3f16Paper.pdf (visité le 18/11/2023).

[SK21] Sumeet S. SINGH et Sergey KARAYEV, « Full Page Handwriting Recognition via Image to Sequence Extraction », in : *Document Analysis and Recognition – ICDAR 2021*, sous la dir. de Josep LLADÓS, Daniel LOPRESTI et Seiichi UCHIDA, Lecture Notes in Computer Science, Cham : Springer International Publishing, 2021, p. 55-69, ISBN : 978-3-030-86334-0, DOI : 10.1007/978-3-030-86334-0_4.

[CCP23] Denis COQUENET, Clement CHATELAIN et Thierry PAQUET, « DAN : A Segmentation-free Document Attention Network for Handwritten Document Recognition », in : *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023), p. 1-17, ISSN : 0162-8828, 2160-9292, 1939-3539, DOI : 10.1109/TPAMI.2023.3235826, URL : https://ieeexplore.ieee.org/document/10013687/ (visité le 06/11/2023).

[Wan+21] Zilong WANG et al., « LayoutReader : Pre-training of Text and Layout for Reading Order Detection », in : *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic : Association for Computational Linguistics, 2021, p. 4735-4744, DOI : 10.18653/v1/2021.emnlp-main.389, URL : https://aclanthology.org/2021.emnlp-main.389 (visité le 06/11/2023).

[Xu+20] Yiheng Xu et al., « LayoutLM : Pre-training of Text and Layout for Document Image Understanding », in : *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, New York, NY, USA : Association for Computing Machinery, 20 août 2020, p. 1192-1200, ISBN : 978-1-4503-7998-4, DOI : 10.1145/3394486.3403172, URL : https://doi.org/10.1145/3394486.3403172 (visité le 07/11/2023).

[GWf92] V Govindaraju, D Wang et given-i=SN family=Srihari given=SN, « Using Temporal Information in Off-Line Word Recognition », in : *Advanced Technology Conference*, t. 1, 1992, p. 529.

[Jag96] S. Jager, « Recovering Writing Traces in Off-Line Handwriting Recognition : Using a Global Optimization Technique », in : *Proceedings of 13th International Conference on Pattern Recognition*, Proceedings of 13th International Conference on Pattern Recognition, t. 3, août 1996, 150-154 vol.3, DOI : 10.1109/ICPR.1996.546812.

[VG91] Gerard P. Van Galen, « Handwriting : Issues for a Psychomotor Theory », in : *Human Movement Science* 10.*2-3* (mai 1991), p. 165-191, ISSN : 01679457, DOI : 10.1016/0167-9457(91)90003-G, URL : https://linkinghub.elsevier.com/retrieve/pii/016794579190003G (visité le 20/06/2023).

[Kwo88] Paul Kwok, « A Thinning Algorithm by Contour Generation », in : *Communications of the ACM* 31.*11* (nov. 1988), p. 1314-1324, ISSN : 0001-0782, 1557-7317, DOI : 10.1145/50087.50092, URL : https://dl.acm.org/doi/10.1145/50087.50092 (visité le 15/06/2023).

[SW92] Frank Y. Shih et Wai-Tak Wong, « A New Single-Pass Algorithm for Extracting the Mid-Crack Codes of Multiple Regions », in : *Journal of Visual Communication and Image Representation* 3.*3* (sept. 1992), p. 217-224, ISSN : 10473203, DOI : 10.1016/1047-3203(92)90018-O, URL : https://linkinghub.elsevier.com/retrieve/pii/1047320392900180 (visité le 19/06/2023).

[RAC05] L. Rousseau, E. Anquetil et J. Camillerapp, « Recovery of a Drawing Order from Off-Line Isolated Letters Dedicated to on-Line Recognition », in : *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, Eighth International Conference on Document

Analysis and Recognition (ICDAR'05), août 2005, 1121-1125 Vol. 2, DOI : 10.1109/ICDAR.2005.199.

[LAL99] PIERRE-MICHEL LALLICAN, « Reconnaissance de l'ecriture Manuscrite Hors-Ligne : Utilisation de La Chronologie Restauree Du Trace », These de doctorat, Nantes, 1er jan. 1999, URL : https://www.theses.fr/1999NANT2046 (visité le 07/11/2023).

[QY04] Yu QIAO et M. YASUHARA, « Recovering Dynamic Information from Static Handwritten Images », in : *Ninth International Workshop on Frontiers in Handwriting Recognition*, Ninth International Workshop on Frontiers in Handwriting Recognition, oct. 2004, p. 118-123, DOI : 10.1109/IWFHR.2004.87.

[NB10] Vu NGUYEN et Michael BLUMENSTEIN, « Techniques for Static Handwriting Trajectory Recovery : A Survey », in : *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, New York, NY, USA : Association for Computing Machinery, 9 juin 2010, p. 463-470, ISBN : 978-1-60558-773-8, DOI : 10.1145/1815330.1815390, URL : https://doi.org/10.1145/1815330.1815390 (visité le 27/06/2023).

[NK17] Zouhaira NOUBIGH et Monji KHERALLAH, « A Survey on Handwriting Recognition Based on the Trajectory Recovery Technique », in : *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), avr. 2017, p. 69-73, DOI : 10.1109/ASAR.2017.8067762.

[MB02] U.-V. MARTI et H. BUNKE, « The IAM-database : An English Sentence Database for Offline Handwriting Recognition », in : *International Journal on Document Analysis and Recognition* 5.*1* (1er nov. 2002), p. 39-46, ISSN : 1433-2833, DOI : 10.1007/s100320200071, URL : https://doi.org/10.1007/s100320200071 (visité le 27/06/2023).

[Wil+92] R Allen WILKINSON et al., *The first census optical character recognition system conference*, t. 184, US Department of Commerce, National Institute of Standards et Technology, 1992.

[L'H00] Eric L'Homer, « Extraction of Strokes in Handwritten Characters », in : *Pattern Recognition* 33.7 (1er juill. 2000), p. 1147-1160, ISSN : 0031-3203, DOI : 10.1016/S0031-3203(99)00103-X, URL : https://www.sciencedirect.com/science/article/pii/S003132039900103X (visité le 08/11/2023).

[Doe+02] D. Doermann et al., « Hidden Loop Recovery for Handwriting Recognition », in : *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, août 2002, p. 375-380, DOI : 10.1109/IWFHR.2002.1030939, URL : https://ieeexplore.ieee.org/document/1030939 (visité le 08/11/2023).

[EB+05] A. El Baati et al., « Recovery of Temporal Information from Off-Line Arabic Handwritten », in : *The 3rd ACS/IEEE International Conference onComputer Systems and Applications, 2005.* The 3rd ACS/IEEE International Conference onComputer Systems and Applications, 2005. Jan. 2005, p. 127-, DOI : 10.1109/AICCSA.2005.1387116, URL : https://ieeexplore.ieee.org/document/1387116 (visité le 08/11/2023).

[RCA06] Laëtitia Rousseau, Jean Camillerapp et Eric Anquetil, « What Knowledge about Handwritten Letters Can Be Used to Recover Their Drawing Order ? », in : Tenth International Workshop on Frontiers in Handwriting Recognition, Suvisoft, 23 oct. 2006, URL : https://inria.hal.science/inria-00104373 (visité le 15/06/2023).

[Guy+94] I. Guyon et al., « UNIPEN Project of On-Line Data Exchange and Recognizer Benchmarks », in : *Proceedings of the 12th IAPR International Conference on Pattern Recognition (Cat. No.94CH3440-5)*, 12th International Conference on Pattern Recognition, t. 2, Jerusalem, Israel : IEEE Comput. Soc. Press, 1994, p. 29-33, ISBN : 978-0-8186-6270-6, DOI : 10.1109/ICPR.1994.576870, URL : http://ieeexplore.ieee.org/document/576870/ (visité le 09/11/2023).

[LB05] Marcus Liwicki et Horst Bunke, « IAM-OnDB - an On-Line English Sentence Database Acquired from Handwritten Text on a Whiteboard », in : *Proceedings of the Eighth International Conference on Document Analysis and Recognition*, ICDAR '05, USA : IEEE Computer Society, 31 août 2005,

p. 956-961, ISBN : 978-0-7695-2420-7, DOI : 10.1109/ICDAR.2005.132, URL : https://doi.org/10.1109/ICDAR.2005.132 (visité le 27/06/2023).

[Mah+19] Mahshad MAHDAVI et al., « ICDAR 2019 CROHME + TFD : Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection », in : *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019 International Conference on Document Analysis and Recognition (ICDAR), sept. 2019, p. 1533-1538, DOI : 10.1109/ICDAR.2019.00247.

[Awa+11] Ahmad-Montaser AWAL et al., « First Experiments on a New Online Handwritten Flowchart Database », in : IS&T/SPIE Electronic Imaging, sous la dir. de Gady AGAM et Christian VIARD-GAUDIN, San Francisco Airport, California, USA, 23 jan. 2011, 78740A, DOI : 10.1117/12.876624, URL : http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.876624 (visité le 03/10/2022).

[CB18] Marco CUTURI et Mathieu BLONDEL, *Soft-DTW : A Differentiable Loss Function for Time-Series*, 20 fév. 2018, arXiv : 1703.01541 [stat], URL : http://arxiv.org/abs/1703.01541 (visité le 04/08/2023), preprint.

[RFB15] Olaf RONNEBERGER, Philipp FISCHER et Thomas BROX, *U-Net : Convolutional Networks for Biomedical Image Segmentation*, 18 mai 2015, arXiv : 1505.04597 [cs], URL : http://arxiv.org/abs/1505.04597 (visité le 27/07/2023), preprint.

[Mni+13] Volodymyr MNIH et al., *Playing Atari with Deep Reinforcement Learning*, 19 déc. 2013, arXiv : 1312.5602 [cs], URL : http://arxiv.org/abs/1312.5602 (visité le 02/08/2023), preprint.

[ZS84] T. Y. ZHANG et C. Y. SUEN, « A Fast Parallel Algorithm for Thinning Digital Patterns », in : *Communications of the ACM* 27.*3* (1er mars 1984), p. 236-239, ISSN : 0001-0782, DOI : 10.1145/357994.358023, URL : https://dl.acm.org/doi/10.1145/357994.358023 (visité le 07/09/2023).

[Tor+09] Paolo TORMENE et al., « Matching Incomplete Time Series with Dynamic Time Warping : An Algorithm and an Application to Post-Stroke Rehabilitation », in : *Artificial Intelligence in Medicine* 45.*1* (jan. 2009), p. 11-34, ISSN : 09333657, DOI : 10.1016/j.artmed.2008.11.007, URL : https:

//linkinghub.elsevier.com/retrieve/pii/S0933365708001772 (visité le 13/11/2023).

[Kap+20] Jared KAPLAN et al., *Scaling Laws for Neural Language Models*, 22 jan. 2020, arXiv : 2001.08361 [cs, stat], URL : http://arxiv.org/abs/2001.08361 (visité le 14/11/2023), preprint.

[Gao+23] Chenyang GAO et al., « ICDAR 2023 Competition on Recognition of Multi-line Handwritten Mathematical Expressions », in : *Document Analysis and Recognition - ICDAR 2023*, sous la dir. de Gernot A. FINK et al., Lecture Notes in Computer Science, Cham : Springer Nature Switzerland, 2023, p. 566-576, ISBN : 978-3-031-41679-8, DOI : 10.1007/978-3-031-41679-8_34.

COLLEGE ED MATHEMATIQUES
DOCTORAL STIC
PAYS DE LA LOIRE

Nantes Université

Titre : Conversion d'écriture hors-ligne en écriture en-ligne et réseaux de neurones profonds

Mot clés : écriture manuscrite, CNN, Transformer

Résumé : Cette thèse se focalise sur la conversion d'images statiques d'écriture hors-ligne en signaux temporels d'écriture en-ligne. L'objectif est d'étendre l'approche à réseau de neurone au-delà des images de lettres isolées ainsi que de les généraliser à d'autres types de contenus plus complexes.

La thèse explore deux approches neuronales distinctes, la première approche est un réseau de neurones convolutif entièrement convolutif multitâche *UNet* basé sur la méthode de [ZYT18]. Cette approche a démontré des bons résultats de squelettisation mais en revanche une extraction de trait problématique. En raison des limitations de modélisation temporelle intrinsèque à l'architecture CNN. La deuxième approche s'appuie sur le modèle de squelettisation précédent pour extraire les sous-traits et propose une modélisation au niveau sous-traits avec deux Transformers : un encodeur de sous-trait (SET) et un décodeur pour ordonner les sous-traits (SORT) à l'aide de leur vecteur descripteur ainsi que la prédiction de lever de stylo. Cette approche surpasse l'état de l'art sur les bases de données de mots, phrases et d'équations mathématiques et a permis de surmonter plusieurs limitations relevées dans la littérature.

Ces avancées ont permis d'étendre la portée de la conversion d'image d'écriture hors-ligne vers l'écriture en-ligne pour inclure des phrases entières de texte et d'aborder un type de contenu complexe tel que les équations mathématiques.

Title: Offline handwritting conversion to online and neural networks

Keywords: handwritting, CNN, Transformer

Abstract: This thesis focuses on the conversion of static images of offline handwriting into temporal signals of online handwriting. Our goal is to extend neural networks beyond the scale of images of isolated letters and as well to generalize to other complex types of content. The thesis explores two distinct neural network-based approaches, the first approach is a fully convolutional multitask UNet-based network, inspired by the method of [ZYT18]. This approach demonstrated good results for skeletonization but suboptimal stroke extraction. Partly due to the inherent temporal modeling limitations of CNN architecture.

The second approach builds on the previous skeletonization model to extract sub-strokes and proposes a sub-stroke level modeling with Transformers, consisting of a sub-stroke embedding transformer (SET) and a sub-stroke ordering transformer (SORT) to order the different sub-strokes as well as pen up predictions. This approach outperformed the state of the art on text lines and mathematical equations databases and addressed several limitations identified in the literature.

These advancements have expanded the scope of offline-to-online conversion to include entire text lines and generalize to bidimensional content, such as mathematical equations.