Document Attention Network (DAN) vs Document Understanding Transformer (DONUT)

Mulapi Tita Ketsia[1], Thomas Constum[2], Thierry Paquet[3]

[(1,2,3)]Université de Rouen Normandie

6 février 2023

1 Résumé

Les systèmes d'archivage, avec l'avènement des nouvelles technologies et le stockage des documents numériques, ont fait de la reconnaissance des documents l'un des domaines le plus prisé en matière d'extraction d'informations. De l'extraction simple de l'information à la compréhension des documents, l'état de l'art des différentes architectures de reconnaissances a beaucoup évolué et a atteint des performances élevées. Les architectures traditionnelles basées sur l'usage des moteurs de reconnaissance optique de caractères (OCR), ont présenté des inconvénients tels que des calculs volumineux et la propagation des erreurs. Dans ce travaille nous présentons un parallélisme assez complet entre deux architectures de reconnaissances des documents approuvées dans la littérature qui sont DAN [1] et DONUT [2]. Ils désignent, respectivement, un réseau d'attention aux documents pour la reconnaissance de documents manuscrits et, un Transformateur de compréhension de document, tous deux sans OCR. Les résultats obtenus à l'état de l'art par les deux approches montrent en effet que la tâche de reconnaissance des images de documents peut être effective sans qu'aucune segmentation ne soit effectuée en amont. Enfin, nous verrons comment nous avons généré des données synthétiques basées sur des équations mathématiques en latex issues du concours organisé par l'ICDAR [7] pour tester la capacité du DAN à reconnaître des équations mathématiques multilignes, cette dernière faisant l'objet d'un autre travail.

Mots-Clés :
Optical Character Recognition (OCR), Document Attention Network (DAN), Document Layout Analysis (DLA), Document Understanding Transformer (DONUT), Visual Document Understanding (VDU), Handwritten Document Recognition (HDR), Natural Language Processing (NLP) Bidirectional Auto-Regressive Transformers (BART), Fully Convolutional Network (FCN), Convolutional Neural Network (CNN), Swin-Transformer (ST), Multi Layer Perceptron (MLP), Character Error Rate (CER), Intersection over Union (IoU), Mean Average Precision (mAP), End-to-End (E2E), Synthetic Data (SD), Layout Ordering Error Rate (LOER), Post-Processing Edition Rate (PPER), Graph Edit Distance (GED), Tree Edit Distance (TED), Teacher Forcing, Curriculum learning, Visual Question Answering (VQA), Connectionist Temporal Classification (CTC).

2 Introduction

Ce résumé s'adresse aux personnes qui utilisent la science des données, l'intelligence artificielle, les sciences économiques, les sciences mathématiques ou les sciences de l'ingénieur dans la gestion de la production. Il est nécessaire d'avoir des connaissances en apprentissage automatique, en traitement d'image, en optimisation non linéaire et en statistique pour comprendre les notions abordées.
Réalisé dans le cadre d'un travail d'étude et de recherche, il est principalement axé sur la comparaison entre les méthodes de reconnaissance de documents DAN et DONUT.

Le DAN, est une proposition du laboratoire Litis de l'Université de Rouen Normandie qui vient en rupture à des sytèmes de reconnaissances de caractères et de documents. Il a spécialement été conçu pour traiter des images de documents manuscrits, cependant, les expériences ont démontrés qu'il est également compatible avec des images de documents imprimés du fait qu'il est pré-entrainé avec ces derniers. Dans la suite de ce travaille, on s'intéressera à la première version du DAN qui est apparue le 1^{er} Août 2022.

Le DONUT quant à lui, est proposé par le laboratoire NaverLab de Google, il s'agit d'un modèle qui vient en rupture à des sytèmes d'extraction d'entités nommées et de compréhension des doc-

uments. Contrairement à la première version du DAN il est conçu pour comprendre des images de documents. Ici, nous nous intéressons à sa version du 23 Août 2022.

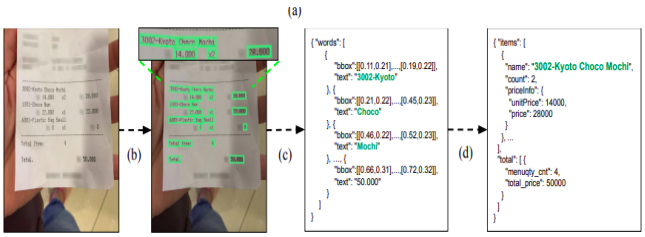


FIGURE.1 Schéma de l'extraction conventionnelle d'informations sur des images de documents semi-structurée :
(a)Pipeline, (b)la détection de texte est effectuée pour obtenir les emplacement de texte et (c)Chaque boîte est transmise au module de reconnaissance pour comprendre les caractères. (d)Enfin, les textes reconnus et leurs emplacements sont passés au module suivant à traiter pour la fomre structurée souhaitée des informations. [2]

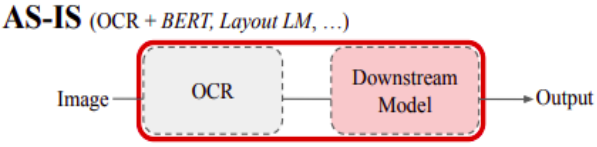


FIGURE.2 Aperçu du Pipeline traditionnel [2]

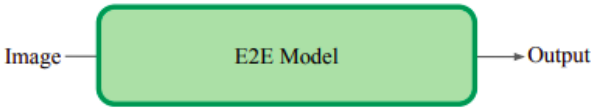


FIGURE.3 Aperçu du Pipeline des modèles de bout en bout (DAN, DONUT) [2]

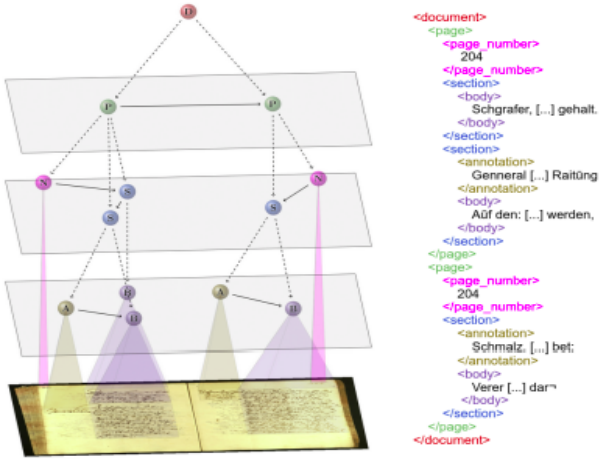


FIGURE.4 Un aperçu du fonctionnement du DAN [1]

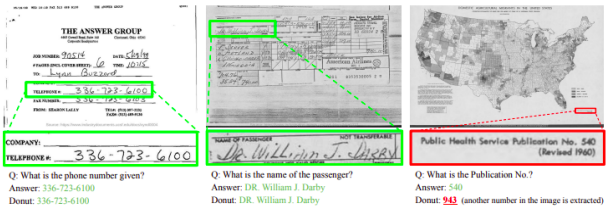


FIGURE.5 Un aperçu du fonctionnement du DONUT [2]

Outre le résumé, l'introduction, la conclusion et les références, ce document décrit de façon synthétique les travaux connexes, les architectures, les données, l'apprentissage des modèles, les étapes de prédictions, les métriques ainsi que les évaluations telles que mentionnées dans les papiers respectifs, tout juste avant d'aboutir à un bref aperçu sur la génération des images synthétiques que nous avons annexé.

3 Travaux Connexes

Traditionnellement, les pipelines à double phase intégraient à la fois une partie pour l'extraction d'information textuelle avec un OCR et une partie pour l'extraction d'information de mise en page des images de documents avec des modèles en aval (DLA) de l'extraction textuelle(voir figure 2), de façon non-exhaustive voici une liste de ces modèles : Bert, LayoutLM, Roberta, Bros, SPADE, WYVERN [1,2].

De ce fait, afin d'évaluer ce type d'architecture, les cherheurs se sont appuyés sur l'usage de 2 types de métriques, l'une pour le texte extrait à l'aide de l'OCR et l'autre en vue d'évaluer la capacité des downstream modèles à extraire les informations structurelles (DLA) des images de documents. Des métriques classiques telles que la F1-score, la CER ont longtemps été utilisés comme mesure de per-fomances de la première phase et, l'IoU, la mAP, ZoneMap pour la seconde phase.

Nous verrons plus loin qu'il est important pour ces architectures de procéder par un ensemble de techniques d'augmentations des données, du pre-processing au post-processing en passant par l'apprentissage(Segmentation) et la Reconnaissance.

Enfin retenons que c'est suite à des inconvénients tels que les processus coûteux, les problèmes de généralisations, la propagation de l'erreur et le mécanisme de post-processing dit post-OCR que les auteurs du DAN et du DONUT proposent des modèles end-2-end.

4 Les architectures DAN et DONUT

Les deux modèles sont composés d'un encodeur et d'un décodeur. Pour le DONUT ils sont entièrement basés sur des Transformers, tandis que le DAN possède un encodeur à base d'un FCN et un décodeur de type Transformer. L'idée est d'encoder et de décoder les données de bout en bout.

4.1 Les encodeurs

On sait que les réseaux de types CNN ou les Transformers peuvent être utilisés comme réseaux d'encodeur. Pour répondre à la question de savoir comment représenter une image de document, on se sert de l'encodeur comme outil d'extraction des caractéristiques et c'est aussi à ce niveau là qu'on peut configurer certaines variables comme la taille de l'image, le nombre de channels ou encore la dimension des données attendues par le décodeur(plongement ou embedding).

4.1.1 Le Fully Convolutionnal Network du DAN et le Swin-Transformer du DONUT

Dans la figure 6, le FCN du DAN prend en entrée une image de document qu'il convertit ensuite en une Features Map à 2-dimensions, laquelle est additionnée à un encodage positionelle de même dimension qui permet de conserver les informations spatiale, avant d'être applatit en une séquence de caractéristqiues à 1-dimension (l'embedding), cette phase se déroule en une seule fois tout juste avant d'envoyer au décodeur la représentation vectorielle de l'image en 1-dimension. Il procède aussi par élimination des mélanges diffus et à la normalisation des instances dans le but d'éviter le sur-ajustement et d'améliorer les performances. [1].

Pour le DONUT qui est assez implicite dans la figure 7, l'encodeur visuel ST divise l'image d'entrée en patches qui ne se chevauchent pas, pour chaque patches on applique un module de multi-head attention basé sur une fenêtre décalée et un MLP à deux couches, ce qui permet de faciliter l'application des couches de fusions de patches aux jetons de patch afin de représenter l'image en un ensemble d'embedding [2].

4.2 Les décodeurs

Ici les décodeurs ont plusieurs fonctions mais servent principalement à faire la prédiction c'est à dire à décoder la représentation en embedding de l'image afin d'extraire les informations textuelles pour le DAN, de comprendre et d'extraire les informations textuelles pour le DONUT. Le Décodeur du DAN est un Transformer classique qui travaille sur des séquences de caractères , ce qui implique qu'il a la capacité de prédire une centaine de caractères (classes), Tandis que celui du DONUT est un modèle à base de Transformer bien connu dans la littérature et pré-entrainé sur plusieurs langues qui s'appel BART ce qui lui confère une capacité supplémentaire à décoder non pas des caractères mais des n-gram de caractères, on dit que le décodeur du DONUT est un Swin-BART (Swin-B).

4.2.1 Les Transformers DAN et DONUT pour le décodeur

Le décodeur du DAN suit un processus récurrent qui en quelque sorte est source de sa lourdeur, il prend en entrée un embedding à 1-dimension, les prédictions(tokens) précédentes et renvoi en sortie (à chaque instant t), le jeton final prédit qui a la plus grande probabilité. Dans [1], les auteurs ont décidés que le mécanisme de multi-head attention modélise la capacité du DAN à dire quel est le prochain caractère grace à une requête (Query:Q) tout en s'intéressant aux parties importantes et en se basant sur les derniers caractères, autrement, il s'intéresse à la fois au texte et à l'image afin de faire une séléction correcte à partir de la position actuelle. En plus de produire en sortie des symboles de caractères et des jetons, le DAN renvoie des jetons spéciaux qui aident à représenter les informations de mise en pages telles que les sauts à la ligne dans un format de type balisage, etc.

Du coté du DONUT, le BART est le modèle officiel, c'est-à-dire que les auteurs en [2], initialisent les poids du modèle de décodeur avec ceux du modèle BART multilingue qui est pré-entrainée et accessible au publique. En entrée de ce modèle (décodeur) des nouveaux jetons spéciaux sont ajoutés pour le prompt (une invite) de chaque tâche afin d'assurer la phase de compréhension, en aval des expériences. Contrairement au DAN, le décodeur du DONUT va mapper les caractéristiques dérivées dans une séquences de jetons de "sous-mots" (n-gram) pour construire un format souhaité de type JSON, plus simplement, le DONUT interprète toutes les tâches en aval comme un problème de prédiction de JSON. Nous verrons plus loin comment il est formé afin de mieux appréhender son fonctionnement.

4.3 Détails techniques des architectures et configurations

4.3.1 Détails techniques

Pour le DAN, Une fois qu'on a produit les embeddings à 1-dimension par l'encodeur, le décodeur

du DAN utilise des Query(Q), Key(K) et values(V) provenant de la même entrée, afin de modéliser les dépendances entre les séquences prédites (c'est la fonction de self-attention) et l'extraction des informations visuelles à l'encodeur (K et V qui proviennent de la 1-dimension) sur la base de Q qui provient des prédictions précédentes permettant d'indiquer où le modèle doit porter son attention pour prédire le jetons suivant (c'est la fonction de la Mutual-Attention) [1]. Afin de comprendre la figure 6, il convient de prendre connaisssances des annotations suivantes, nous notons donc par :

1. X : l'image de document à l'entrée de l'encodeur
2. f_{2D} : la Features Map à 2-dimensions
3. PE_{2D} : L'encodage positionnelle à 2-dimensions
4. $f_{1D} = flatten(f_{2D_{x,y}} + PE_{2D}(x, y)$: l'embedding fournit par l'encodeur ou la représentation applatit de l'image
5. $(\hat{y}_0, ..., \hat{y}_{t-1})$: les prédicitons (tokens) précédentes à la position actuelle.
6. d_{model} : la dimension de l'embedding au niveau du transformer, elle est obtenue après avoir convertie en vecteur les tokens d'entrées $e_{\hat{y}i}$ pour ensuite les additionnées avec un encodage positionnel à 1-dimension, à la position
7. ensuite, on parvient à obtenir la nouvelle Query notée $q_{t,i} = PE_{1D}(i) + e_{\hat{y}i}$
8. le décodeur utilise une fenêtre de longueur 100 pour l'auto-attention, dont pour une séquence donnée "s" de longueur L_s, la trame de sortie $t^{th}O_t$ est calculée sur l'intervalle $[S_a, S_{t-1}]$ avec $a = max(0, t - 100$
9. $\hat{y}_t$: le jeton final prédit et ayant la plus grande probabilité P_t
10. le calcule de la perte de l'entropie croisée entre la prédiction et la véritée terrain : $L_{CE}(y_t, p_t)$

Pour le DONUT,on essaye de répondre à la question de savoir comment comprendre une image de document, on rappel que son encodeur mappe une image de document donnée en embedding et, avec les embedding (plongements) codées, le décodeur se charge de générer une séquence de jetons qui peut être convertis en un type cible d'informations sous une forme structurée (JSON)[2], comme dans la figure 7, il se représente en quatre niveaux :

1. les données d'entrées : l'image de document au niveau de l'encodeur, l'invite (prompt) pour tester les capacités du DAN à comprendre les documents à l'aide de leur embedding grace à l'intégration de la VQA
2. Encoder-Decoder : les deux transformers du DONUT
3. Output Sequences : les réponses aux questions qui représentent l'extraction et la compréhension des images de documents, représentés par un langage de balisage
4. Le parsing des réponses au format balisage en JSON.

4.3.2 Configurations techniques

On peut retenir comme configurations techniques du DAN les éléments suivants:

1. Encoder : FCN
2. Decoder : Transformer
3. Zones d'évaluations : au niveau des paragraphes et des lignes pour faciliter la comparaison
4. Puissance de calcul : 1 GPU Tesla V100 de 32Gb (ressources CRIANN)
5. Précision mixte automatique : avec Pytorch

Tandis que pour le DONUT nous avons :

1. Encoder : ST avec modification des couches initailes et définition de la taille de la fenêtre comme résolution d'entrée 2560x1920
2. Decoder : BART pour le compromis vitesse-temps, utilisation de ses quates premières couches, modification de la longueur maximale à 1536
3. Pre-training et Training : 64 A100 GPU, mini-batch de taille 196, optimiser Adam, laerning rate programmé et, initial rate sélectionné
4. Vitesse du DONUT mesurée sur un GPU P40 (plus lent que le A100)

5 Architectures

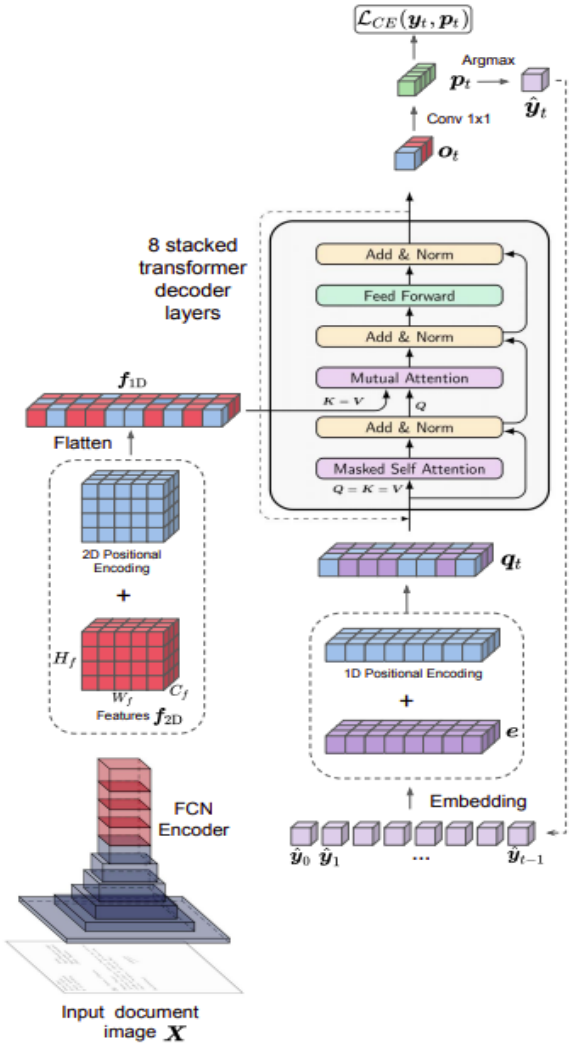


FIGURE.6 L'architecture du DAN [1]

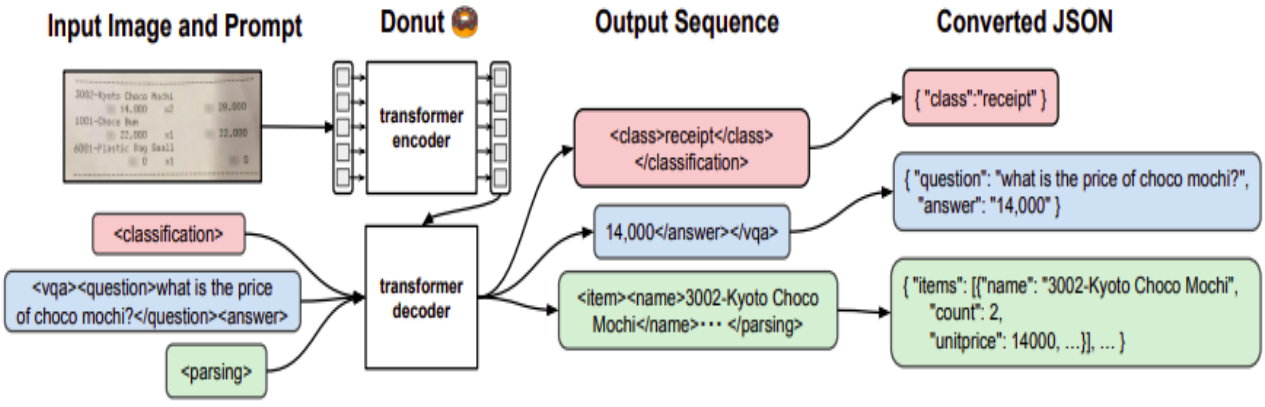


FIGURE.7 L'architecture du DONUT [2]

Comme nous pouvons le constater, les auteurs en [2] n'ont pas été assez implicite sur le DONUT qu'on peut qualifier de boite noire.
A l'inverse, on peut affirmer que les auteurs en [1] ont mieux détaillés l'architecture du DAN ce qui simplifie sa compréhension.

Dans le point suivant, nous abordons les données (datasets) et nous verrons comment pour chaque modèle, les auteurs ont essayé d'une manière ou d'une autre de garantire la diversité au moment de l'apprentissage afin d'être robuste face aux nouvelles données qui ne sont rien d'autres que des données qui émanent du monde réel.

6 Les datasets

Plusieurs datasets du monde réel ont été utilisés afin de garantir l'efficacité des deux modèles à reconnaitre des documents variés en langue et en police. Et pour se faire, les auteurs ont généré des données synthétiques afin d'avoir suffisament de données pour le pré-apprentissage et l'apprentissage. Dans l'idéal pour que les données synthétiques ressemblent aux données réelles, ils ont utilisés des images de documents réels pour le DAN (voir figure 8) et des informations provenant de Wikipedia et de ImageNET pour le DONUT (voir figure 9).

6.1 La génération des données synthétiques

6.1.1 Les données synthétiques du DAN

Les auteurs en [1] ont implémenté un outil de génération d'image de document synthétique à l'aide du langage python, outil rendu disponible sur github **https://github.com/FactoDeepLearning/DAN**.

Les auteurs ont généré des lignes imprimés synthétiques pour le pré-apprentissage du modèle et des documents imprimés synthétiques pour l'apprentissage, en appliquant une politique 90/10 qui consiste à commencer l'apprentissage avec 90% de données synthétiques et 10% de données réelles puis inverser la tendance petit à petit jusqu'à obtenir 20% de données synthétiques et 80% de données réelles. Pour créer les documents synthétiques (D_{line}), le logiciel applique des techniques d'ingénierie simple, à base de règle, il extrait des documents originaux (D_{doc}), des transcriptions de lignes de textes isolées y_i associées à une classe de mise en page C_i, les lignes synthétiques sont généré à la volée pendant le pré-apprentissage, en séléctionnant aléatoirement une transcription de ligne de texte parmi les D_{line}.

6.1.2 Les données synthétiques du DONUT

Ici, les auteurs ont utilisé une application existante du nom **SynthDoG** dont le laboratoire Naver est propriétaire, il se base sur une heuristique qui permet de génerer aléatoirement les données, en utilisant des techniques de rendu d'image pour imiter les documents réels. Pour y parvenir, ils ont combinés des bases de données telles que imageNET, afin de varier les arrières plans, de créer des documents à page vierge et des textures de documents lesquelles se sont succéder par l'intégration des mots et des phrases provenant de plusieurs sources Wikipedia en anglais, en japonais, en chinois, etc.

6.2 Les données réelles

6.2.1 Les données réelles du DAN

Trois jeux de données ont été utilisés, un mélange de document manuscrit et imprimé.

1. RIMES 2009 et RIMES 2011 : ce sont deux datasets qui regroupent des documents de scénarios courriers qui sont en français, au format Niveau de Gris avec une résolution de 300dpi avec un total de 7 classes (Expéditeur, Recepteur, Lieu, Objet, Ouvert, Contenu ou Body, pièces jointes). la version 2009 a été utilisé au niveau des pages et, celle de 2011 a permit d'évaluer le DAN au niveau des lignes et des paragraphes.

2. READ 2016 : est un dataset très structuré qui regroupe des documents à doubles pages avec des annotations sur les marges et qui est d'origine allemande. Il contient également sept classes et permet d'évaluer le DAN au niveau des lignes, des paragraphes et surotut des doubles pages et fonctionne très bien sur les marges.

3. MAURDOR 2 : est un dataset qui contient des images de document multilingue(Français, Anglais, Arabes). il possède quatre catégries (C1 : Formulaire, C2 : Documents commerciaux, C3 : Correspondances Manuscrites privés, C4 : Correspondances privés-professionnelle, C5 : Diagrammes), mais seule les catégories C3 et C4 ont été utilisées du faits que ce sont des scénarios courriers manuscrit et dactylographique, il a uniquement était utilisé dans le cadre de la reconnaissance du texte.

6.2.2 Les données réelles du DONUT

A leur disposition, les auteurs du DONUT ont utilisé cinq datasets à la fois publiques et surtout privés aux nombreux services, produits ou applications actif du laboratoire Naver à travers le monde. On retient qu'il y a un dataset du nom de DocVQA, deux datasets publiques et deux datasets privés qui sont respectivement : CORDE, BILLET et BUSINES CARD, RECEIPT.

1. CORDE : ce sont des reçus en langue Latine, avec 30 champs uniques (menu, price, etc), ayant une structure complexe et repartie en 0.8K en Train, 0.1K en Validation et 0.1 comme données de Test.

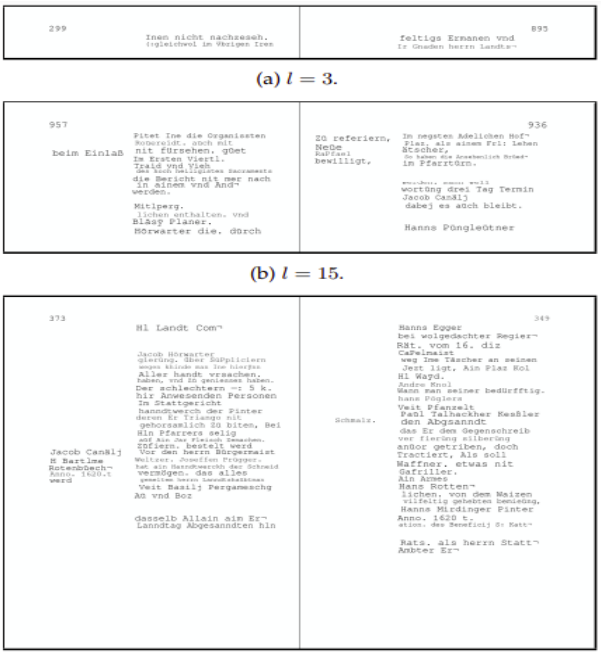
2. BILLET : ce sont des tickets de train en langue chinoise qui ont une structure simple de huit champs (ticket number, starting station, etc.), avec 1.5K en Train, 0.5K en Test et les données de validations qui sont générées à partir des 10% des données de Train.

3. BUSINESS CARD : ce sont des cartes de visite d'origine japonaise, qui ont une structure simple

de 11 champs (name, company, adress, etc.) avec 20K en Train, 0.3K en validation et 0.3K en Test.

4. RECEIPT : ce sont des images de reçu d'origine Coréenne avec 80 champs(Info magasins, etc.) et contenant 40K en Train, 1K en Validation et 1K en Test.

5. DocVQA : c'est un dataset qui permet de tester les capacités supplémentaires du DONUT à comprendre des documents, pour une image de DocVQA, on donne une paire de questions et DONUT prédit la réponse en capturant des infos visuelles et textuelles dans l'image. Il contient 50K questions définies sur plus de 12K documents, 40K en Train, 5K en Validation et 5K en test.



(a) $l = 3$.

(b) $l = 15$.

(c) $l = l_{\max} = 30$ (end of curriculum stage, no crop).

FIGURE.8 Image de document synthétique DAN [1]

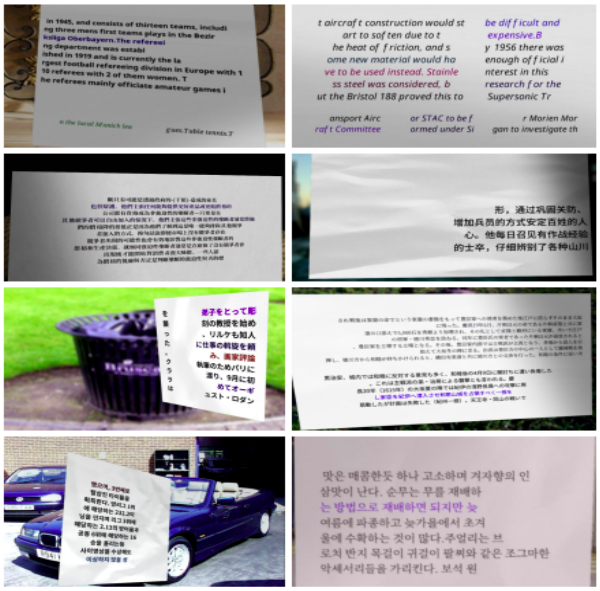


FIGURE.9 Image synthétique DONUT source SynthDoG [2]

FIGURE.10 Image sythnétique DONUT source SynthDoG, ImageNET, Wikipédia anglais, chinois,

japonais et coréen[2]

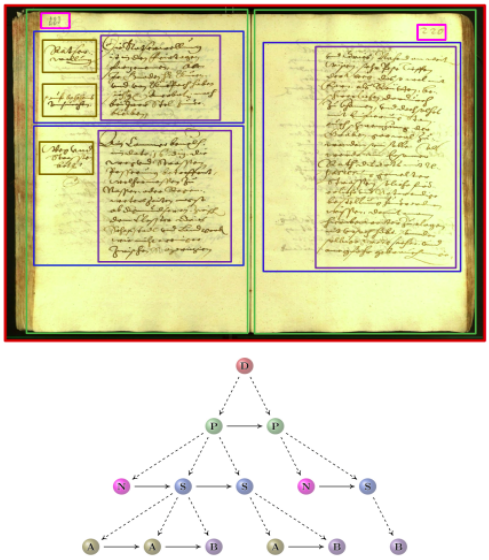


FIGURE.11 Dataset READ 2016 [1]

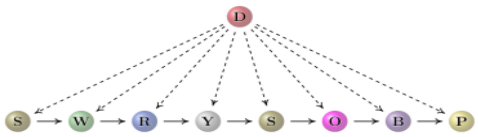


FIGURE.12 Dataset RIMES 2009[1]

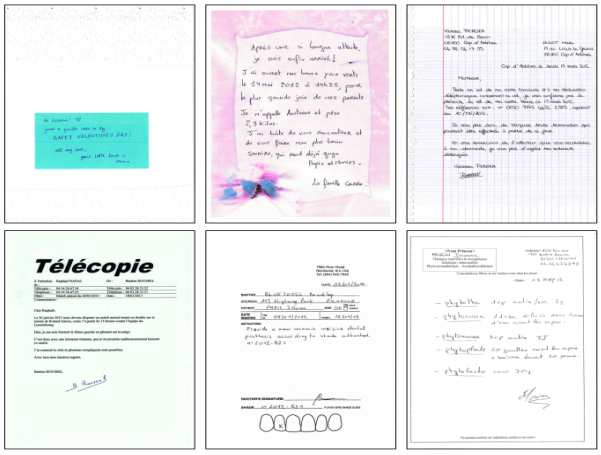


FIGURE.13 Dataset MAURDOR[1]

FIGURE.14 Dataset BUSINESS CARD au dessus et RECEIPT en dessous[2]

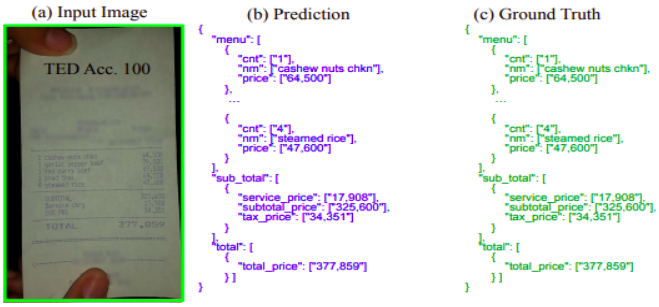


FIGURE.15 Dataset CORD[2]

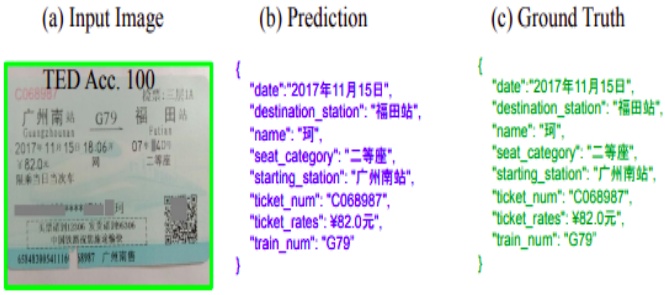


FIGURE.16 Dataset TICKET[2]

6.3 Pre-processing et Post-processing des données

Dans les deux cas, il est question de bien gérer la mémoire et sa consommation grace à la configuration de la mise en échelle, de la résolution et de la normalisation des images, c'est la phase de pre-processing.

Ensuite, au niveau du décodeur, quelques tâches supplémentaires dites de post-processing sont effectuées, au niveau du DAN on se base sur des heuristiques permettant d'appliquer un ensemble de règles afin de corriger les tokens de layouts prédits non appariés, par exemple fermer une balise ouverte mais non fermée ou supprimer les jetons isolés, veiller au respect de la grammaire des jetons sur les relations hierarchiques du langage et, les caractères de trop comme les espaces duppliqués qu'on retrouve dans la prédiction du texte sont supprimés de la prédiction grace à la CTC.

Du coté du DONUT, on prend une orientation complètement différente, si la sortie est mal structurée, on considère que le champ est perdu et on utilise des expressions régulières pour gérer la situation. Enfin, de même que pour la tâche classique de Question-Answering, si la réponse à la question posée ne figure pas dans l'image, DONUT est incapable de la générer.

6.4 Data Augmentation

Les auteurs en [1,2] ont fait de l'augmentation des données sur les images réelles et synthétiques en modifiant la résolution, en tranformant les perspectives, en faisant de la distorsion élastique, en appliquant des opérations morpho-math comme l'érosion et la dilation, en faisant de l'aléatoire sur les couelurs en appliquant des flou gaussien pour le DAN afin d'obtenir une collections de documents aux aspects anciens, en appliquant du bruit gaussien et en gérant l'accentuation.

7 Apprentissage et prédiction

Afin d'avoir un réseau de neuronnes profond qui apprend correctement sur des attentions lorsque les données sont volumineuses, des stratégies pour l'amélioration de la convergence ont été utilisées, c'est le cas du Curriculum Learning et du Teacher Forcing.

7.1 Curriculum learning

C'est une technique qui permet d'apprendre sur une image entière de façon indirecte, c'est à dire, en augmentant progressivement la longueur de la séquence cible au file des itérations. En plus simple, plutot que de traiter un document entier, le modèle apprend d'abord sur une ligne, ensuite un peu plus, jusqu'à attendre l_{max} (nombre de ligne maximale présent ou fixé dans le document).

7.2 Teacher Forcing

C'est le Forçage de l'apprentissage, il s'agit d'une technique qui permet au modèle de corriger ses erreurs durant la phase de l'apprentissage, quelle que soit la sortie du modèle à chaque pas de temps, il obtient la vraie valeur en entrée pour le pas de temps suivant. Il s'agit d'un moyen simple et efficace d'entraîner un modèle de génération de texte. C'est efficace car vous n'avez pas besoin d'exécuter le modèle de manière séquentielle, les sorties aux différents emplacements de la séquence peuvent être calculées en parallèle. C'est donc une sorte d'apprentissage supervisé.

7.3 L'apprentissage du DAN et du DONUT

7.3.1 Le pré-entrainement

Durant deux jours , le DAN a apprit à apprend l'extraction des caractéristiques sur des Mini-batch de taille 16, les auteurs du DAN ont entraîné un modèle d'OCR au niveau de la ligne, sur des lignes synthétiques imprimées en utilisant la perte CTC pour faire ensuite du transfert learning.

Du coté du DONUT, c'est la lecture du texte par l'encodeur, DONUT apprend à comprendre l'ensemble du document, il est pré-entrainé avec des images de documents et leurs annotations textuelles, il apprend donc à lire les textes en prédisant les mots suivants, On dit aussi qu'il apprend comme un modèle de langage visuel sur l'image.

7.3.2 L'entrainement

Il a fallu deux jours de plus pour entrainer le DAN par transfert learning. Cette fois-ci sans mini-batch, on se base sur le forçage de l'apprenant avec des documents réels et synthétiques, on s'intéresse à l'ordre de lecture (grâce au mécanisme d'attention) et une fois que le

modèle comprend l'ordre de lecture, il est affiné en étant adapté aux images du monde réel.

L'ajustement du DONUT consiste à faire du Fine-Tuning ou de la mise au point pour que le décodeur génère des séquences de tokens qui peuvent être convertie en un JSON. On fait une évaluation approfondie pour interpréter toutes les tâches en aval comme un problème de prédiction JSON, le forçage de l'enseignant aide à répondre à la question de savoir comment apprendre l'image document.

7.4 Extraction et compréhension des images de document pour la prédiction

Une fois que l'apprentissage est réalisé, on peut tester le DAN et le DONUT en les fournissant des images de documents. Afin d'effectuer une prédiciton correcte, chaque modèle possède des jetons de début et de fin. Le décodeur du DAN commence par un jeton initial $\hat{y}_0 =< sot >$ qui désigne le début de la transcription et se termine par la prédiction d'un jeton spéciale $\hat{y}_{L_y+1} =< eot >$. De l'autre coté, le décodeur du DONUT a apprit à générer une séquence de Tokens qui peut être convertie en un JSON représentant les informations de sorties souhaités, mais avant cela, il génère aussi deux jetons spéciaux qui sont $[START\_Class]$ et $[END\_Class]$, le contenu est représenté par une classe appelée "memo" $[memo]$ il est introduit à condition qu'il existe, ce qui permet ensuite d'obtenir une séquence inversible un à un en JSON {"$Class$" : "$memo$"}.

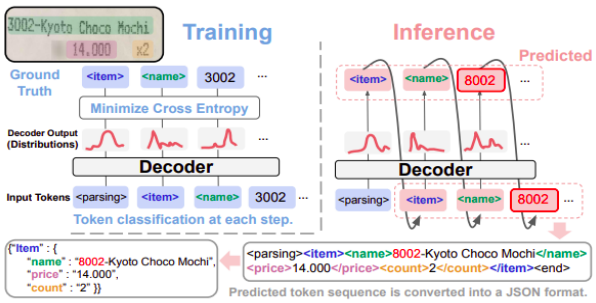


FIGURE.17 Entrainement et prédiction avec DONUT[2]

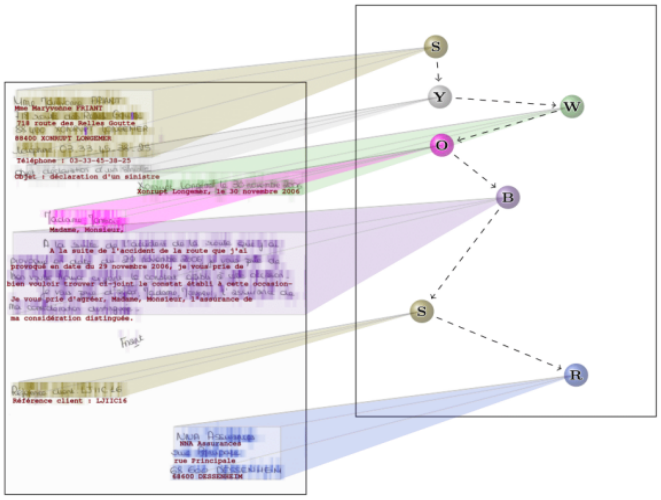


FIGURE.18 Prédiction du DAN sur RIMES 2009[1]

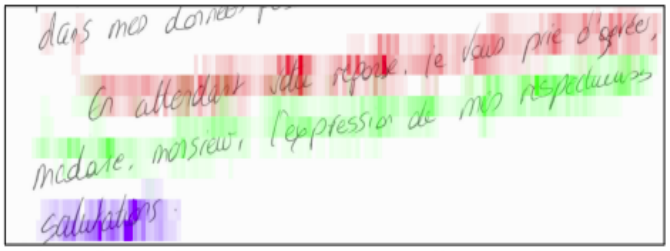


FIGURE.19 Mécanisme d'attention avec DAN[1]

8 Les métriques d'évalutations

Pour évaluer ce genre d'architecture on s'intéresse aux deux types d'extractions : le texte et la mise en page. Dans leur contribution, les auteurs du DAN ont eu l'ingéniosité de proposer une métrique capable d'évaluer conjointement le texte et le layout, une première à notre connaisance c'est la métrique mAP_{CER}, on divise en sous séquences les prédictions et les vérités terrains, pour considérer une prédiction comme True ou False et pour chaque classe, on ordonne les séquences par leur score de confiance et une sous séquences est True Positif si le CER entre la prédiction et la vérité terrain est inférieur à un seuil donné, les sous séquences sont ensuite supprimés jusqu'à ce qu'il n'y en ait plus. En plus de cela, il y'a deux autres métriques assez intéressante qui sont la LOER et la PPER, la première permet à l'aide de la distance d'édition des graphes (GED) de se baser sur les balises comme des caractères afin de calculer sous la forme d'un graphe les balises entre la vérité terrain et les prédictions, quant à la PPER, elle permet de vérifier que la phase de post-processing ne pousse pas le DAN à faire du sur ou du sous apprentissage, le taux s'est avéré faible ce qui prouve à suffisance que cette phase n'implique aucun incident.

Du coté du DONUT, on fonctionne à la façon

traditionnelle, c'est à dire qu'on a une métrique pour l'évaluation du texte et une métrique pour l'évaluation de la mise en page matérialisé par la F1-score et la précision globale. Avec la F1-score, on vérifie si les infos extraites sont dans la vérité terrain ou non, cette métrique est simple et facile mais ne peut pas mesurer la structure prédite (Hiérarchie imbriquée, Groupes, ...) d'où l'importance de la précision globale basée sur la distance d'édition d'arbre (TED) pour tous documents représentés sous forme d'arbre.

TABLE.1 Les métriques du DAN

METRIQUES	DAN
CTC	Yes
CrossEntropy	L=$\sum_{t=1}^{L_y+1} L_{CE}(y_t, p_t)$
LOER	$\frac{\sum_{K}^{i=1} GED(y_i^{graph}, \hat{y}_i^{graph})}{\sum_{i=1)}^{K} n_{e_i}+n_{n_i}}$
mAP_{CER}	$\frac{\sum_{cS} AP_{CER_c}^{5:50:5} * len_c}{\sum_{cS} len_c}$
PPER	$\frac{\sum_{i=1}^{K} n_{pped_i}}{\sum_{i=1}^{K} y_{len_i}^{layout}}$

TABLE.2 Les métriques du DONUT

METRIQUES	DONUT
F1-score	$\frac{TP}{TP+\frac{1}{2}FP+FN}$
Précision	max(0, 1-TED(pr,gt)/TED(ϕ, gt))

9 L'évaluation du DAN et du DONUT

Evaluer le DAN et le DONUT dans un même environnement technique et avec un ou des datasets requiert énormément de ressources en puissance de calculs. Nous avons donc jugé bon de critiquer les différentes évaluations réalisées par les auteurs des deux méthodes. Par ailleurs, nous allons tester le DAN sur un jeux de données qui reprend des équations mathématiques dans un autre travail.

9.1 Evaluations à l'état de l'art

D'abord nous pouvons affirmer que les études d'abblations réalisés sur les deux méthodes sont satisfaisantes, mais au-delà de cela nous ne pourrions pas comparer l'évaluation de ces deux modèles du fait qu'ils n'ont ni utilisés les mêmes données ni utilisés les mêmes métriques et encore ils ne se sont pas comparés au même modèle existant à l'état de l'art, le DAN s'intéresse plus dans sa version actuelle aux modèles d'extraction d'infromation alors que le DONUT en plus de cela, s'intéresse essentiellement aux modèles de compréhension de document.

9.1.1 Evaluation du DAN

Dataset	Approach	CER ↓	WER ↓	LOER ↓	mAP_{CER} ↑	PPER ↓
RIMES 2011 [41]	**Line level**					
	[4] FCN	3.04%	8.32%	✗	✗	✗
	[51] CNN+BLSTM[a]	**2.3%**	9.6%	✗	✗	✗
	Ours (DAN)[c]	2.63%	**6.78%**	✗	✗	✗
	Paragraph level					
	[3] FCN	4.17%	15.61%	✗	✗	✗
	[1] CNN+MDLSTM[b]	2.9%	12.6%	✗	✗	✗
	[4] FCN+LSTM[b]	1.91%	6.72%	✗	✗	✗
	Ours (DAN)[c]	**1.82%**	**5.03%**	✗	✗	✗
RIMES 2009 [49]	**Paragraph level**					
	Ours (DAN)[c]	5.46%	13.04%	✗	✗	✗
	Page level					
	Ours (DAN)[c]	4.54%	11.85%	3.82%	93.74%	1.45%

[a] This work uses a different split (10,203 for training, 1,130 for validation and 778 for test).
[b] with line-level attention.
[c] with character-level attention.

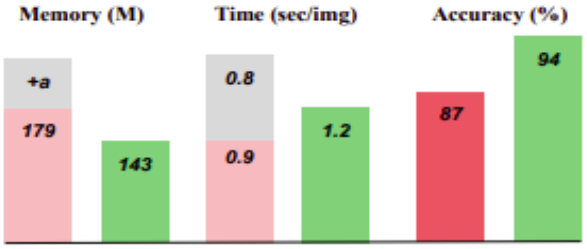
TABLE.3 Ici les auteurs ont évalués les préédictions des caractères et des mots sur RIMES 2011 et RIMES 2009 au niveau des lignes et des paragraphess, on constate que contrairement aux autres modèles le DAN se débrouille bien au niveau des paragraphes alors qu'au niveau des lignes la combinaison du CNN et du BLSTM s'en sort mieux à quelques différences près. [1]

Approach	CER ↓	WER ↓	LOER ↓	mAP_{CER} ↑	PPER ↓
Line level					
[36] CNN+BLSTM[a]	4.66%	✗	✗	✗	✗
[43] CNN+RNN	5.1%	21.1%	✗	✗	✗
[4] FCN+LSTM[b]	**4.10%**	**16.29%**	✗	✗	✗
Ours (DAN)[a]	**4.10%**	17.64%	✗	✗	✗
Paragraph level					
[3] FCN	6.20%	25.69%	✗	✗	✗
[4] FCN+LSTM[b]	3.59%	13.94%	✗	✗	✗
Ours (DAN)[a]	**3.22%**	**13.63%**	✗	✗	✗
Single-page level					
Ours (DAN)[a]	3.43%	13.05%	5.17%	93.32%	0.14%
Double-page level					
Ours (DAN)[a]	3.70%	14.15%	4.98%	93.09%	0.15%

[a] with character-level attention.
[b] with line-level attention.

TABLE.4 Ici on se concentre sur READ 2016, le DAN a des résultats impressionnants à tous les niveaux[1]

9.1.2 Evaluation du DONUT



(b) System Benchmarks.

FIGURE.18 l'architecture E2E du DONUT contrairement à l'architecture traditionnelle présente des avantages intéressant, on gagne en Mémoire, plus au moins en temps et, on a de bonne performance.[2]

	OCR	#Params	Time (ms)	Accuracy (%)
BERT	✓	110M + $\alpha^{\dagger}$	1392	89.81
RoBERTa	✓	125M + $\alpha^{\dagger}$	1392	90.06
LayoutLM	✓	113M + $\alpha^{\dagger}$	1396	91.78
LayoutLM (w/ image)	✓	160M + $\alpha^{\dagger}$	1426	94.42
LayoutLMv2	✓	200M + $\alpha^{\dagger}$	1489	95.25
Donut (Proposed)		143M	**752**	**95.30**

TABLE.5 Alors que le DAN possède un encodeur qui utilise 1.7M de paramètres, le DONUT lui contrairement à ses concurrents n'est pas le modèle qui requiert le plus ou le moins de paramètes il est au pif des extrémités tel que vous pouvez le constater.[2]

			CORD [45]			Ticket [12]			Business Card			Receipt		
	OCR	#Params	Time (s)	F1	Acc.	Time (s)	F1	Acc.	Time (s)	F1	Acc.	Time (s)	F1	Acc.
BERT* [22]	✓	$86^{\dagger}_{M} + \alpha^{\ddagger}$	1.6	73.0	78.2	1.7	74.3	91.7	1.5	40.8	87.0	2.5	70.3	77.3
BROS [18]	✓	$86^{\dagger}_{M} + \alpha^{\ddagger}$	1.7	74.7	80.3									
LayoutLM [65]	✓	$89^{\dagger}_{M} + \alpha^{\ddagger}$	1.7	78.4	87.3									
LayoutLMv2* [64,66]	✓	$179^{\dagger}_{M} + \alpha^{\ddagger}$	1.7	78.9	87.0	1.8	87.2	90.1	1.6	52.2	92.9	2.6	72.9	89.0
Donut		$143^{\dagger}_{M}$	**1.2**	**84.0**	**93.5**	**0.6**	**94.1**	**98.8**	**1.4**	**58.7**	**95.1**	**1.9**	**78.6**	**94.4**
SPADE* [25]	✓	$93^{\dagger}_{M} + \alpha^{\ddagger}$	4.0	74.0	84.5	4.5	11.9	60.6	4.3	32.3	88.3	7.3	64.1	79.9
WYVERN* [21]	✓	$106^{\dagger}_{M} + \alpha^{\ddagger}$	1.2	43.3	70.2	1.5	41.8	76.2	1.7	29.9	88.8	3.4	71.5	92.0

TABLE.6 Enfin dans ce tableau on peut conclure que par rapport aux modèles qui existent à l'état de l'art, le DONUT comme modèle de compréhension des documents présente des résultats concluant, lui conférant ainsi le mérite de modèle efficace pour la compréhension des images de documents à l'ère actuel [2]

10 Conclusion

Ce document décrit un parallélisme assez détaillé du DAN et du DONUT, nous pouvons retenir que le modèle DAN est fait pour extraire des informations textuelles à partir de documents et structurer ces informations. Le DAN gère mieux les informations difficiles à extraire, et a été évalué avec des pages uniques et doubles, au niveau des lignes et des paragraphes. Cependant, le DAN présente encore des faiblesses et il serait intéressant de proposer une amélioration sur la réduction du temps de prédiction et d'améliorer la reconnaissance de documents multisectoriels ainsi que de lui donner des capacités de compréhension des images de documents grace à la VQA. DONUT le conccurent est beacoup plus flexible aux types de documents et aux langues, mais possède aussi des limites, comme son incapacité à identifier les textes minuscules, et échoue lorsqu'il ne peut pas extraire les informations. Les auteurs pourraient explorer des méthodes d'amélioration du pre-training, ainsi que des méthodes basées sur la génération de textes pour gérer les situations où la réponse ne figure pas dans le document. Nous osons donc dire que ces deux modèles qui permettent de faire de la reconnaissance de document de bout en bout, ont été réalisé par deux entités différentes mais partagent à peu près la mêmes démarche de résolution.

References

[1] D. Coquenet, C. Chatelain, T. Paquet, "a Segmentation-free Document Attention Network for Handwritten Document Recognition" , 1 Aug 2022.

[2] Geewook Kim , Teakgyu Hong , Moonbin Yim, Jeongyeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han , and Seunghyun Park, NAVER CLOVA, NAVER Search, Tmax Google, NAVER AI Lab, "OCR-free Document Understanding Transformer", 23 Aug 2022

[3] Thierry Paquet, "Cours de Compression des données Master 1 Université de Rouen", Master 1, Année académique 2021-2022

[4] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol , Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, Devi Parikh, "VQA : Visual QUestion Answering", 27 Oct 2016

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, "Attention Is All You Need", 6 Dec 2017

[6] Thierry Paquet, "Machine Learning on Sequences Rouen Normandy University", Master 2, Année académique 2022-2023

[7] Internationale Confernece on Document Analysis and Recognition https://ai.100tal.com/icdar

11 ANNEXES

L'ICDAR pour International Conference on Document Analysis and Recognition organise chaque année des compétitions sur l'analyse et la reconnaissance des documents. Pour l'année 2023, elle propose parmis ses compétitions le challenge de la reconnaisasnce des équations mathématiques manuscrits.

Nous voulons donc utiliser le DAN pour reconnaitre des équations mathématiques manuscrites et pour se faire, plusieurs modificaitions que nous détaillons dans la suite sont à prévoir.

11.1 Les données et les métriques de la compétition

ICDAR 2023 prévoit 38 000 images d'équations mathétématiques avec comme répartition 9931 images à équations multilignes et 28 069 images à équations monolignes.

Au départ, 30 000 données d'entrainement sont rendues disponibles sur la plateforme, les 8000 autres constituent les données de tests qui sont disponibles en deux étapes, les données Test_A permettant de sélectionner les dix premier modèles sont disponibles à partir du 1^{er} Fevrier 2023 et le délai de soumission de la première évaluation est fixé au 9 Mars 2023 ensuite, le 10 Mars la deuxième étape avec le Test_B est rendu disponible et son délai de soumission est fixé au même jour.

Une image d'équations est identifiée par un nom de fichier le quel est associé à un code, tous deux présents dans le train_images.csv. Le code de l'image est sa vérité terrain qui n'est rien d'autre que une formule ou l'équation mathématique écrite LaTeX. Comprenons par là, que l'objectif d'utiliser un modèle comme le DAN pour un tel problème serait de prédire le code LaTeX (équation) associé à une image (d'équation).

Pour l'évaluation, les performances en terme de classement sont données par l'expression du rappel pour le classement du Test_A et, la combinaison du rappel sur l'expression et sur le caractère pour le Test_B. On note :

1. Expression recall : $S_{recall} = \frac{S_{right}}{S_{sum}}$ il désigne le pourcentage de séquences de formules LaTeX prédites correspondant à la vérité terrain
2. Character recall : $C_{rappel} = 1 - \frac{C_{difference}}{C_{somme}}$, $C_{difference}$ c'est la somme des distances d'édition pour toutes les images et C_{somme} est le nombre de caractèrs pour toutes les étiquettes.

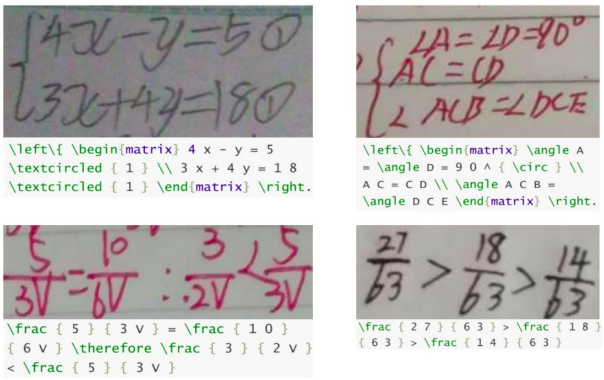


FIGURE.19 Exemples du jeu de données[7]

filename	LaTeX string
train_0.jpg	\begin{matrix} 9 x - y = a + 3 \\ 2 x + y = 5 a \end{matrix}
train_1.jpg	\left\{ \begin{matrix} \\ x ^ { 2 } - 5 x y + 6 y ^ { 2 } = 0 \end{matrix} \right.

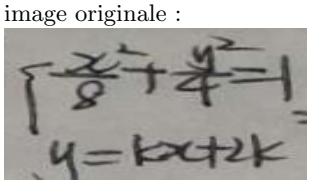
FIGURE.20 Exemple de deux lignes dans le fichier train_images.csv [7]

11.2 Génération des données synthétiques et organisations des données

Le DAN dans sa configuration de base nécessite énormément d'images. Pour se faire nous avons générés des images d'équations imprimés synthétiques afin d'avoir en notre possesion des données supplémentaires pour le pré-apprentissage du modèle, ces données ont été générés à partrir des données d'entrainement fournies, et sur les 30 000 Train, nous avons génerés 25 329 images synthétiques, en combinant plusieurs méthodes pour le pré-traitement telle que l'usage des expressions régulières pour ne citer que cela.

Voici un exemple :

1. nom du fichier : train_28838.jpg
2. image originale :



3. image synthétique générée :

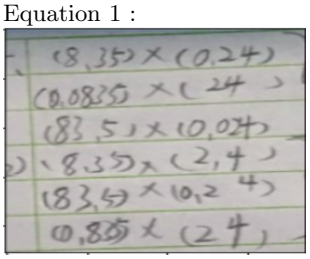
$$\begin{cases} \frac{x^2}{8} + \frac{y^2}{4} = 1 \\ y = kx + 2k \end{cases}$$

4. syntaxe LaTeX :

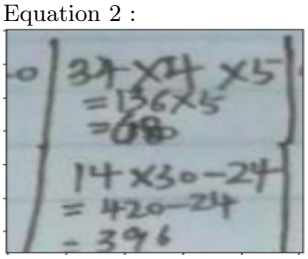
```
\left\{ \begin{matrix} \frac { x ^ { 2 } } { 8 } + \frac { y ^ { 2 } } { 4 } = 1 \\ y = k x + 2 k \end{matrix} \right.
```

Un autre exemple, il s'agit des deux équations à six lignes :

1. Equation 1 :



2. Equation 2 :



```
'Z',
'\\rightarrow',
'0',
'1',
'2',
'3',
'4',
'5',
'6',
'7',
'8',
'9',
'\\pi',
'\\alpha',
'\\beta',
'\\sum',
'\\sigma',
'a',
'b',
```

FIGURE.21 Un extrait des symboles [7]

Enfin, voici quelques chiffres :

TABLE.7 données originales et synthétiques

Nombre de lignes	Images Originales	Train_set	Images Synthétiques
1	28069	22560	18769
2	8579	6302	5944
3	1243	1040	525
4	81	70	63
5	26	26	26
6	2	2	2
TOTAUX	38.000	30.000	25.329

11.3 Perspectives

Maintenant que les données sont préparées, la prochaine étape consiste à configurer le DAN de sorte à ce qu'il puisse reconnaître les symboles LaTeX, pour se faire, des annotations sont prévu pour simplifier les syntaxes et par conséquent, faciliter l'apprentissage. Le DAN sera pré-entrainé sur les données synthétiques, entrainé sur les données réelles et testé sur différents jeux de tests, quel sque soit leur origine dès lors où ce sont des images d'équations mathématiques manuscrites.