Optimisation d'un modèle DocVQA sans OCR: Encodage hiérarchique et structurel à faible coût de documents dans un espace multimodal commun

par

Rayane BENCHAREF

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN GÉNIE DE LA PRODUCTION AUTOMATISÉE
M. Sc. A.

MONTRÉAL, LE 24 NOVEMBRE 2025

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Mohamed Cheriet, directeur de mémoire
Département de génie des systèmes à l'École de technologie supérieure

M. Christian Desrosiers, président du jury
Département de génie logiciel et TI à l'École de technologie supérieure

M. Clément Playout, examinateur externe
Département de génie informatique et génie logiciel à Polytechnique Montréal

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 19 NOVEMBRE 2025

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Ce mémoire est à la fois l'aboutissement de mon séjour de deux ans à l'école de technologie supérieure mais également la fin de mon parcours universitaire à l'école d'ingénieur ISIS Castres que j'ai intégrée en sortie de lycée. Il faudrait ainsi plus qu'un mémoire pour remercier toutes les personnes qui m'ont accompagné tout au long de ces six années.

Je vais tout de même commencer par la personne sans qui ce mémoire ne serait pas, Prof. Mohamed Cheriet, qui au fil de ces deux années, a su encadrer et orienter ma recherche à travers ces domaines complexes que sont l'apprentissage profond et la compréhension de documents. Je garde en souvenir les premières présentations chaotiques que je vous ai proposées, mais qui avec du temps et de bons conseils, ont vu leur clarté se relever.

Je ne peux oublier Hamdan, Saïda, Ben, Mohamed, Eya, Baha et les autres membres du laboratoire Synchromedia qui m'ont tendu la main au cours de ces deux années. Un grand merci également au Dr. Clément Playout (membre du jury externe) et au Prof. Christian Desrosiers (président du jury), qui ont accepté de prendre le temps d'évaluer mon travail.

Je tiens également à remercier Thomas, ancien élève de l'ISIS Castres et qui, alors que quatre promotions nous séparent, a pris le temps depuis plusieurs années de me conseiller et m'orienter au sujet de l'IA. Ce parcours aurait été bien différent sans toi.

Ensuite, j'aimerais rendre hommage à Kilian, qui a rejoint le laboratoire en même temps que moi, ces deux années n'auraient pas été les mêmes sans nos débats et péripéties interminables. Je ne peux oublier Maé, sans qui les pauses cafés auraient été bien différentes. Je me dois également de te remercier Raphaël, un coloc sans qui ce rude hiver aurait été encore plus long. Une mention spéciale à Maxence que j'ai retrouvé à Montréal après que nos chemins se soient séparés à la sortie du lycée et sans qui nos pique-niques auraient été bien moins animés. Je tiens également à remercier Raymond, Margaux, Étienne, Eliaz et tous les autres amis qui ont été là au cours de ces deux années.

En repensant à ces six années, je ne peux oublier les personnes que j'ai rencontrées à Castres, lors de ma sortie du lycée. Mes premières pensées vont à Hugo, Arnaud, Axel et Kilian, avec qui j'ai eu le plaisir (partagé j'espère) de vivre en colocation et dont les anecdotes sont bien trop nombreuses pour être listées ici. Je tiens également à remercier Colin, Ben et les autres deuxièmes années qui nous ont intégrés à l'école et ont su organiser des soirées mémorables. Une dédicace spéciale à Marion, dont les gâteaux savaient comment nous remonter le moral en période de partiels. Je ne peux oublier Cléa, dont les soirées films et plats de lasagnes étaient de dignes récompenses en fin de semestre. Je pourrais continuer pendant plusieurs paragraphes encore, mais un immense merci à Mélia, Lucas, Paul, Léo, Wiame, Ana et tous les autres avec qui je garde des souvenirs inoubliables. Enfin, je ne peux finir ce paragraphe sans remercier Lisa, avec qui j'ai vécu l'entièreté de ce parcours universitaire. Depuis les bancs de l'école d'ingénieur, en passant par notre Erasmus à Chypre et jusqu'au Canada, ces années pleines d'aventures n'auraient pas été les mêmes sans toi.

Pour conclure, je tiens à remercier ma famille, Chaya et Richard, qui ont suivi ce parcours de près. Mes cousins Samy, Ismael et Lyna que j'ai hâte de retrouver une fois rentré. Une pensée va naturellement à ma famille à l'île Maurice, qui m'a soutenu malgré la distance. Enfin, j'adresse mes derniers mots à mes parents, qui ont toujours été là pour moi et m'ont accompagné dans chaque décision que j'ai prise. J'espère que l'aboutissement de ce long chemin vous rendra fiers.

Optimisation d'un modèle DocVQA sans OCR: Encodage hiérarchique et structurel à faible coût de documents dans un espace multimodal commun

Rayane BENCHAREF

RÉSUMÉ

Le nombre de documents numériques a connu une forte augmentation au cours de la dernière décennie, et ce dans différents secteurs, que ce soit industriel, médical, académique et bien d'autres. Bon nombre de ces documents proviennent de numérisations (images de documents), permettant de construire des banques de données partagées au sein d'entreprises, institutions ou même sur internet. Ces grandes bases de données peuvent directement contenir les documents numérisés ou encore être tabulaires, contenant les informations provenant de ces derniers. Cependant, l'extraction manuelle d'informations contenues sur des documents numérisés est chronophage dans un contexte où le nombre de ces derniers ne cesse d'augmenter. Ainsi, automatiser l'extraction d'informations à grande échelle devient un besoin vital, comme par exemple dans des secteurs industriels où le temps est une ressource précieuse. Cette automatisation exige cependant des systèmes rapides, précis et peu coûteux afin qu'ils puissent être efficaces sur de grandes bases de documents.

L'avènement des grands modèles de langues (LLM) a montré de bonnes performances pour l'extraction d'information sur les tâches de réponse à des questions sur des données de texte (QA). Cependant, les images de documents sont des données variées, comportant différents types d'entités (photo, tableau, texte manuscrit, etc.) et pouvant avoir différentes structures (lettre, articles, etc.). Ainsi, elles sont différentes des données que les LLM prennent en entrée, et ne sont donc pas directement utilisables par ces derniers. Par conséquent, la tâche de réponse à des questions sur des images de documents (DocVQA) nécessite de représenter les images de documents afin que les modèles de langues puissent les utiliser afin de répondre à des questions. Dans ce contexte, les approches fondées sur des outils de reconnaissance de caractères optiques (OCR) nécessitent un entraînement supplémentaire, ajoutent de la complexité au système (détection, reconnaissance) et peuvent conduire à des erreurs de transcription. À l'inverse, les méthodes bout-en-bout (OCR-free), composées d'un encodeur visuel et d'un modèle de langue, bénéficient d'une architecture unifiée permettant à la fois de représenter le document et de répondre à la question. Ce type de méthodes regroupe des modèles de petite taille, peu coûteux en termes de calcul, mais limités en qualité de réponses, ainsi que des modèles à grande échelle (LVLM), performants en termes de résultats mais trop lourds pour des déploiements industriels.

Ce mémoire présente ainsi un système DocVQA OCR-free qui apprend un espace de représentation multimodal (image-texte), composé d'un encodeur visuel hiérarchique de petite taille, d'un projecteur multimodal et d'un modèle de langue à grande échelle. L'encodeur visuel transforme l'image de document en jetons (token) projetés sur l'espace du modèle de langue via le projecteur multimodal. Cet encodeur intègre également un encodage positionnel explicite de la mise en page, préservant l'ordre de lecture et la structure des éléments (tableaux, graphiques, zones textuelles) dans l'espace commun. Le décodeur linguistique à grande échelle met directement

ces représentations alignées en relation avec la question afin de générer la réponse sans outils additionnels tels que l'OCR. Ce système a été construit en distillant l'encodeur visuel de fondation d'un LVLM dans une architecture hiérarchique plus petite tout en gardant le LLM décodeur afin de réduire le coût de calcul tout en conservant des résultats proches du modèle initial. Afin d'assurer l'alignement image-texte de la représentation, l'encodeur distillé a été supervisé de bout-en-bout avec le LLM décodeur. Suite à cela, un module d'encodage spatial décomposant la position de chaque token en caractéristiques de Fourier a été ajouté afin d'enrichir les jetons par leur position d'origine sur le document. Ces approches ont été évaluées expérimentalement sur le jeu de données DocVQA, contenant des images de documents industriels de différents types (formulaires, lettres, articles, etc.). En utilisant le LVLM Paligemma qui a une performance de 84.77% ANLS, la distillation vers une architecture hiérarchique plus petite a permis de réduire la taille de son encodeur visuel par un facteur de 5, divisant de moitié sa latence (896ms → 446ms) tout en conduisant à un gap de seulement 2.1 points d'ANLS avec une performance de 82.67% ANLS. De plus, l'ajout de l'encodage positionnel a permis d'améliorer les résultats sur la qualité d'extraction des informations du document, réduisant ce gap à 1.31 points avec une performance de 83.46% ANLS. Ainsi, le système proposé surpasse en termes de performance les modèles OCR-free de petites tailles tels que Donut qui a une performance de 66.26% ANLS, et reste compétitif avec les LVLM tels que Paligemma ainsi qu'avec les méthodes se basant sur l'OCR telles que UDOP (84.70% d'ANLS).
Des analyses complémentaires sur la classification (RVL-CDIP) et l'analyse de structure (DocLayNet) montrent que l'encodeur capture la structure globale, tandis que le LLM traite cette dernière de manière plus approfondie à un niveau sémantique.
Enfin, le modèle a été adapté aux documents multi-pages via un sélecteur de page réutilisant les premières couches du LLM, sans paramètres supplémentaires. Cette approche limite le coût de calcul en maintenant le modèle à 2.6B paramètres tout en atteignant 71.73% ANLS, concurrençant les autres modèles de l'état de l'art tels que ScreenAI (72.9% ANLS/5B) ou encore DocOwl2 (69.42% ANLS/8B), démontrant une mise à l'échelle efficace pour des scénarios industriels complexes.

En résumé, ce mémoire démontre qu'un alignement image-texte guidé par une méthode OCR-free intégrant la géométrie spatiale permet de représenter des documents de structures variées contenant différents types d'entités. De plus, il souligne qu'une architecture hierarchique permet de réduire la complexité du système tout en maintenant une qualité de réponse compétitive. Enfin, l'adaptation du modèle aux documents multi-page sans paramètres supplémentaires montre l'extension du système à des cas d'utilisation plus complexes. Cette approche présente donc un DocVQA plus efficient et compétitif pour l'automatisation de l'extraction d'information.

Mots-clés: DocVQA, Image de Documents, OCR-Free, Espace de représentation

Optimization of an OCR-Free DocVQA model : Hierarchical and structural encoding at low cost of documents in a common multimodal space

Rayane BENCHAREF

ABSTRACT

The number of digital documents has seen a high increase during the last decade in several sectors such as industry, medicine, academia and others. A lot of those documents come from digitalization (document images), allowing to build shared databases inside enterprises, institutions or even across the internet. These high-scale databases may directly contain numerical documents or be tabular, having extracted information from documents. However, the manual extraction of this information can be time-consuming in a context where the number of digital documents continues to grow. Thus, automating the extraction of these information at a high scale becomes a vital need, as in industrial sectors where time is a precious resource. However, such automation requires fast, accurate and low-cost systems in order to be efficient and effective in high-scale document databases.

The advent of large language models (LLM) has shown good performance for information extraction on question-answering tasks (QA) with text data. However, document images are varied data, containing several entity types (picture, table, handwriting, text, etc.), and may have different structures (letter, article, etc.). Thus, these images are different from the data that LLM usually take as input, and therefore are not directly usable by them. Consequently, the task of visual question-answering on document images (DocVQA) needs to represent the document images in order to allow the LLM to answer the questions. In this context, methods based on optical character recognition tools (OCR) require additional training while adding complexity into the system (detection, recognition), and may lead to recognition errors. On the other hand, end-to-end methods (OCR-free), composed of a visual encoder and a language model decoder, have a unified architecture, allowing both to represent the document and answer the question. This type of methods can be divided into two groups. Firstly the lightweight methods, efficient with a small computational cost, but limited in performance. Then, there are the large visual language models (LVLM), which are accurate in performance but have a high computational cost that can lead to difficulties for industrial deployments.

Thus, this thesis presents an OCR-free DocVQA system that learns a multimodal representation space (image-text), composed of a small hierarchical visual encoder, a multimodal projector, and a LLM. The visual encoder transforms the document image into visual tokens, projected to the language model's representation space (embedding), through the multimodal projector. This encoder also integrates an explicit positional encoding of the document structure, preserving the reading order and element structures (table, graphics, text, etc.) in the multimodal space. The language model decoder directly uses these representations with the question to generate the answer without additional tools such as OCR. This system has been built by distilling the foundational visual encoder of an LVLM into a smaller hierarchical architecture, while keeping the LLM decoder, in order to reduce computational cost while conserving close results with the

initial model. To ensure the image-text alignment of the representation, the distilled encoder has been end-to-end supervised with the LLM decoder. Then, a spatial encoding module decomposes the position of each token on the document into Fourier features has been added in order to enrich the visual tokens by their original position. These approaches have been evaluated on the DocVQA dataset, which contains industrial document images of different types (forms, letters, articles, etc.). By using the LVLM Paligemma that has a performance of 84.77% ANLS, the distillation into a smaller hierarchical architecture has reduced the visual encoder size by a factor of five, halving its latency (896ms → 446ms) while leading to a gap of 2.1 points of ANLS with a performance of 82.67% ANLS. Moreover, the addition of the positional encoding has improved the extraction quality of information, reducing the gap to 1.31 points with a performance of 83.46% ANLS. Thus, the proposed system outperforms the results of lightweight OCR-free methods such as Donut, which has a performance of 66.26% ANLS, and stays competitive with LVLM as Paligemma and with OCR-based models such as UDOP (84.70% ANLS).
Additional analysis on classification (RVL-CDIP) and layout analysis (DocLayNet) show that the encoder captures the global structure, where the LLM handles deeper layout reasoning at a semantic level.
Finally, the model has been adapted to multi-page documents with a page selector, sharing the LLM's first layers. This approach limits the computational cost by keeping the model to 2.6B parameters while reaching 71.73% ANLS, competing with other state-of-the-art models such as ScreenAI (72.9% ANLS/5B) and DocOwl2 (69.42% ANLS/8B), showing an efficient scaling for complex industrial contexts.

In summary, this thesis shows that an image-text alignment led by an OCR-free method, which integrates the spatial geometry, enables the representation of document images of various structures and containing different entity types. Moreover, it underlines that a small hierarchical architecture reduces the system complexity while keeping a competitive response quality. Finally, the adaptation of the model to multi-page documents without additional parameters shows the extension of the system to more complex use cases. Thus, this approach presents a DocVQA more efficient and competitive for the automation of information extraction.

Keywords: DocVQA Document Images, OCR-Free, Embedding space

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

CE	Entropie croisée (Cross Entropy)
CNN	Réseau neuronal convolutif (Convolutional Neural Network)
DocVQA	Réponse à des questions visuelles sur des documents (Document Visual Question Answering)
ETS	École de Technologie Supérieure
GPU	Processeur graphique (Graphics Processing Unit)
IA	Intelligence Artificielle
LLM	Grand modèle de langue (Large Language Model)
LVLM	Grand modèle de vision-langage (Large Visual Language Model)
MLP	Perceptron Multicouche (Multi-Layer Perceptron)
MSE	Erreur moyenne quadratique (Mean Squared Error)
MLP	Perceptron Multicouche (Multi-Layer Perceptron)
PE	Encodage positionnel (Positional Encoding)
ViT	Transformeur visuel (Vision Transformer)
VRAM	Mémoire vidéo (Video Random Access Memory)

LISTE DES SYMBOLES ET UNITÉS DE MESURE

α	Taux d'apprentissage (learning rate)
D	Tailles des vecteurs d'embeddings
I	Une image
ms	Milliseconde
MiB	Mébioctet (Mebibyte), unité de mesure binaire pour la mémoire (RAM) valant 1 048 576 octets
$O()$	Notation "Big-O", complexité en terme de temps d'exécution ou d'espace mémoire requis par un algorithme
N	Nombre de vecteurs dans v et z_l
σ	Une fonction d'activation
v	Un ensemble de représentations abstraites (embeddings)
W	Une matrice de paramètres
x	Entrée d'un modèle
x_{pe}	Positions des éléments de l'entrée x
z_l	Représentations abstraites interne à un ViT à la couche l

INTRODUCTION

0.1 Contexte et motivation

"Just as electricity transformed almost everything 100 years ago, today I actually have a hard time thinking of an industry that I don't think AI will transform in the next several years" - Andrew NG

Les documents ont depuis toujours constitué une source d'information essentielle pour la civilisation humaine. Des tablettes d'argile datant de -5000 avant J.C, utilisées par les Sumériens pour enregistrer des transactions commerciales, en passant par les archives de papyrus de la Grèce antique, qui contenaient à la fois histoires mythiques et textes scientifiques (voir figure 0.1), à la création de l'imprimerie en 1450, facilitant la production de textes à grande échelles, les documents sont un moyen efficace de stocker et archiver de l'information (Guardian, 2025; of Encyclopaedia Britannica, 2025). De nos jours, avec l'arrivée d'internet, les documents passent au format numérique, ce qui permet de les rendre plus accessibles. Cette digitalisation est entraînée par la numérisation des documents, convertissant le format papier en image. En effet, une augmentation significative du nombre de documents numérisés a pu être observée lors de la dernière décennie à travers différents secteurs (Johanne Roy, 2015; Angela Tudico, 2022). En 2015, le Centre Hospitalier Universitaire (CHU) de Québec a numérisé environ 75 millions de pages de dossiers patients, ce qui a permis de libérer près de 2000 m^2 d'espace de stockage physique (Johanne Roy, 2015). À l'heure actuelle, certains documents médicaux sont encore remplis à la main sur papier tels que les ordonnances, formulaires de consentements... avant d'être scannés et numériquement archivés. Il en est de même dans le secteur administratif, où beaucoup de documents comme les formulaires, testaments, lettres de correspondance et autres sont remplis sur papier puis numérisés sous forme d'image. Cette tendance est par exemple démontrée par le gouvernement canadien qui met en avant la numérisation et la centralisation des

a) Tablette d'argile sumérienne, -5000 av. JC b) Papyrus, extrait du livre des morts, -1600 av. JC

Figure 0.1 Photos de documents anciens, prise au musé Redpath, de l'université McGill, Montréal

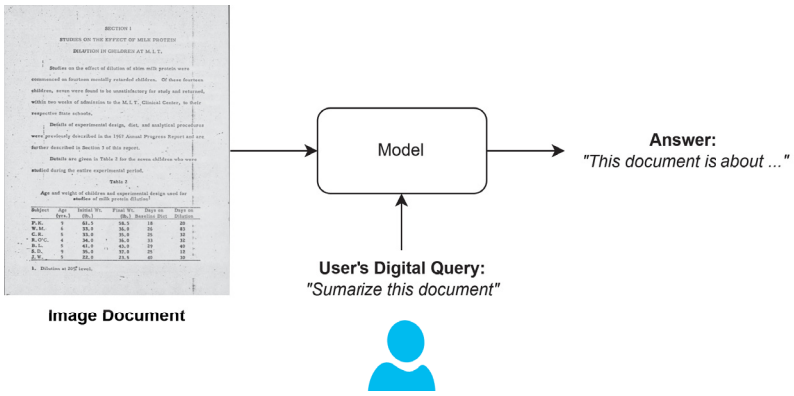
documents administratifs à travers différentes normes et directives (du Canada, 2025). Le secteur académique a lui aussi bénéficié de cette numérisation, rendant plus accessibles les anciens journaux et articles de recherche (Sainte-Anne, 2025). D'autres secteurs tels que la banque, l'assurance, le commerce de détail, les transports, l'industrie manufacturière se sont mis à numériser leurs documents, permettant de réduire les espaces de stockage physiques et de faciliter le partage de ces documents. En parallèle de cette augmentation, les informations contenues dans les documents sont de plus en plus stockées de manière structurée dans des bases de données afin de faciliter leur intégration dans des systèmes d'information à des fins d'analyse et de prise de décisions. Par exemple, l'augmentation du nombre de Dossier Médical Électronique (DME) nécessite l'extraction et l'enregistrement de données présentes dans les documents archivés vers des bases de données tabulaires (ASTP, 2021). Cependant, renseigner manuellement les informations provenant d'images de documents peut s'avérer être une tâche laborieuse, en raison du nombre important de documents numérisés. De plus, une fois numérisé, retrouver une information précise dans une collection de documents peut également être chronophage.

Ainsi, l'exploitation automatique des documents numérisés devient un besoin vital dans les

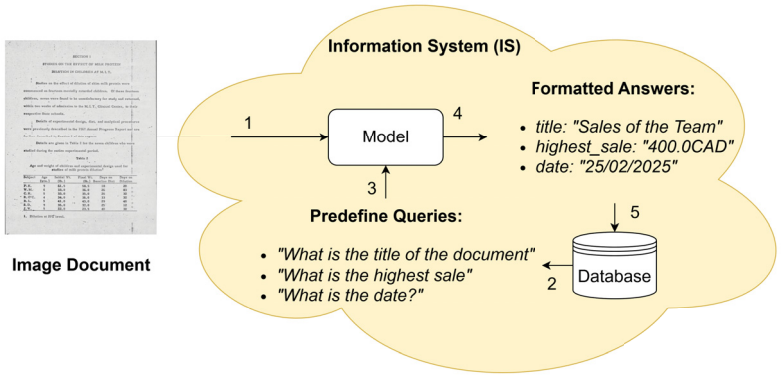
secteurs ayant un grand nombre de documents à analyser. En effet, pour que ces documents puissent être intégrés dans des systèmes d'information modernes, notamment des bases de données structurées, il est nécessaire d'en extraire le contenu pertinent. C'est dans ce contexte que s'inscrit la tâche de réponse à des questions visuelles sur des documents (Document Visual Question Answering, DocVQA), qui vise à interroger un modèle d'intelligence artificielle (IA) au sujet d'un document sous forme d'image à partir d'une question.

La figure 0.2 illustre trois cas d'usage de cette tâche. Le premier cas est la lecture de documents par l'interaction avec un assistant conversationnel (figure 0.2a), dans lequel un utilisateur pose des questions à propos d'un document. Cela peut permettre de résumer le document ou encore de retrouver rapidement une information spécifique, réduisant ainsi le temps de recherche. Ce cas d'utilisation peut par exemple être utilisé dans des contextes académiques pour étudiants et chercheurs, afin de faciliter leur travail.

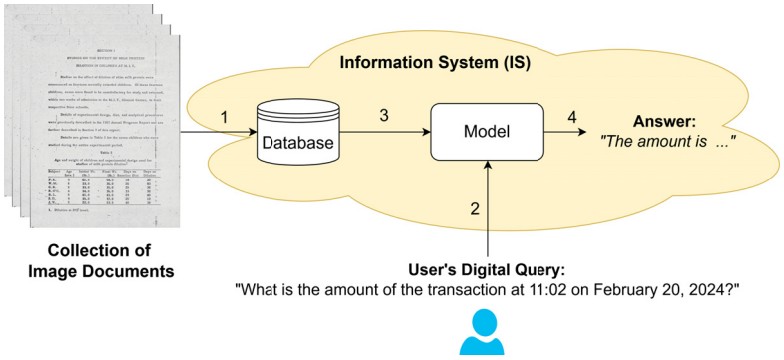
La figure 0.2b illustre un autre cas d'utilisation relié aux secteurs industriels. Comme énoncé précédemment, beaucoup d'industries enregistrent leurs données sous forme tabulaire dans des bases de données, cependant, certaines de ces informations proviennent de documents numérisés sous forme d'images. Ainsi, extraire à la main chacune de ces informations peut s'avérer chronophage et ne pas être optimal. La tâche de DocVQA permet d'automatiser cela en prédéfinissant un ensemble de questions faisant référence aux informations que l'on souhaite extraire. Ainsi, pour chaque nouveau document digitalisé entrant dans le système d'information, celui-ci est directement analysé avec l'ensemble des questions prédéfinies. À partir de cela, les informations du document sont extraites en fonction des questions et automatiquement sauvegardées dans la base de données. Cette automatisation permet donc aux employés d'optimiser leur temps de travail, en se consacrant à d'autres tâches plus importantes.



a) Assistant conversationnel de documents visuels



b) Analyse et extraction automatique d'informations de documents



c) Recherche et extraction d'une information dans une collection de documents numérisés

Figure 0.2 Cas d'utilisations de la tâche de réponse à des questions visuelles sur des documents (DocVQA)

Enfin, la figure 0.2c met en avant un dernier cas d'utilisation. Dans le cas où des documents sont stockés sous forme d'archives, leur nombre peut devenir conséquent. Ainsi, retrouver une information spécifique sans savoir dans quel document elle se situe peut s'avérer compliqué. La tâche de DocVQA permet également de rendre ce processus plus rapide en donnant la base d'image accessible à un modèle d'IA, qui est capable de prendre en entrée la question de l'utilisateur et de retourner la réponse à partir des documents contenus dans la base d'archives.

Cependant, pour chacun des cas d'utilisation cités ci-dessus, cette automatisation doit être fiable en extrayant uniquement les informations pertinentes et relatives aux questions posées, ce qui nécessite un modèle d'IA fiable. Au cours de la dernière décennie, l'augmentation de la capacité de calcul des ordinateurs a permis des avancées majeures dans le domaine de l'apprentissage profond (deep learning). En 2012, les processeurs graphiques (Graphics Processing Unit, GPU) NVIDIA ont été adoptés pour entraîner un réseau de neurones profond par Krizhevsky, Sutskever & Hinton (2012) lors de la compétition annuelle de classification ImageNet (2012). Les performances obtenues ont permis de démocratiser l'utilisation des GPU, entraînant le développement d'architectures plus complexes et profondes. L'une de ces dernières a été les Transformers (Vaswani *et al.*, 2017), qui sont à l'origine des grands modèles de langage (Large Language Model, LLM), tels que GPT-3 par Brown *et al.* (2020). Ces modèles de génération de textes se sont montrés performants pour l'extraction d'informations digitales (sous forme de texte) en fonction d'une instruction/question (Question Answering, QA).

0.2 Problématique, questions de recherche et focus de la thèse

Les LLMs se sont montrés particulièrement efficaces pour répondre à des questions sur du texte digital, en utilisant des représentations vectorielles (embeddings) du vocabulaire pour analyser à la fois un contexte (ressource digitale) en fonction d'une question et générer une réponse. Cependant, la tâche de DocVQA introduit la modalité d'image dont les LLM ne sont pas entraînés à prendre en entrée. Cette modalité entraîne de nouveaux défis en comparaison avec le texte.

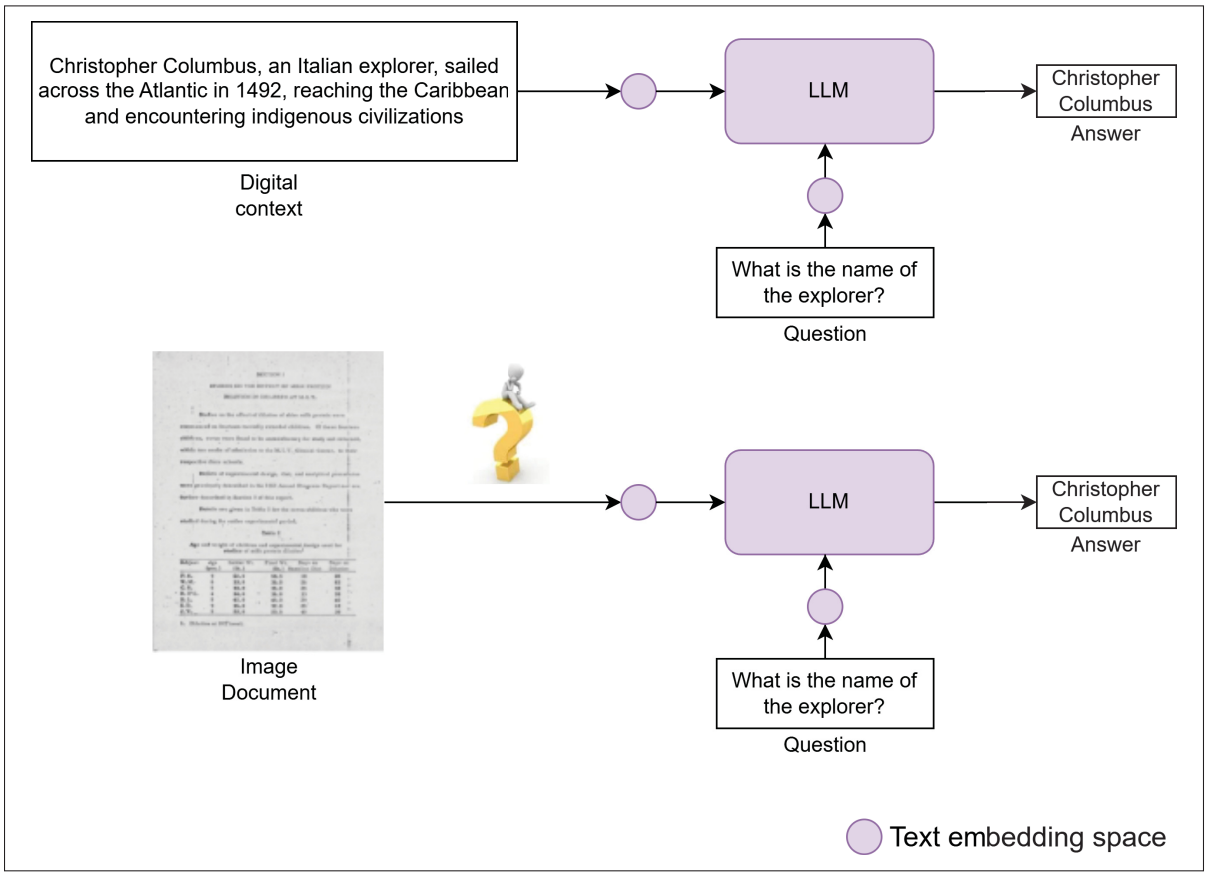


Figure 0.3 Entrée/sortie des LLMs, entraînés à representer et traiter du texte digital

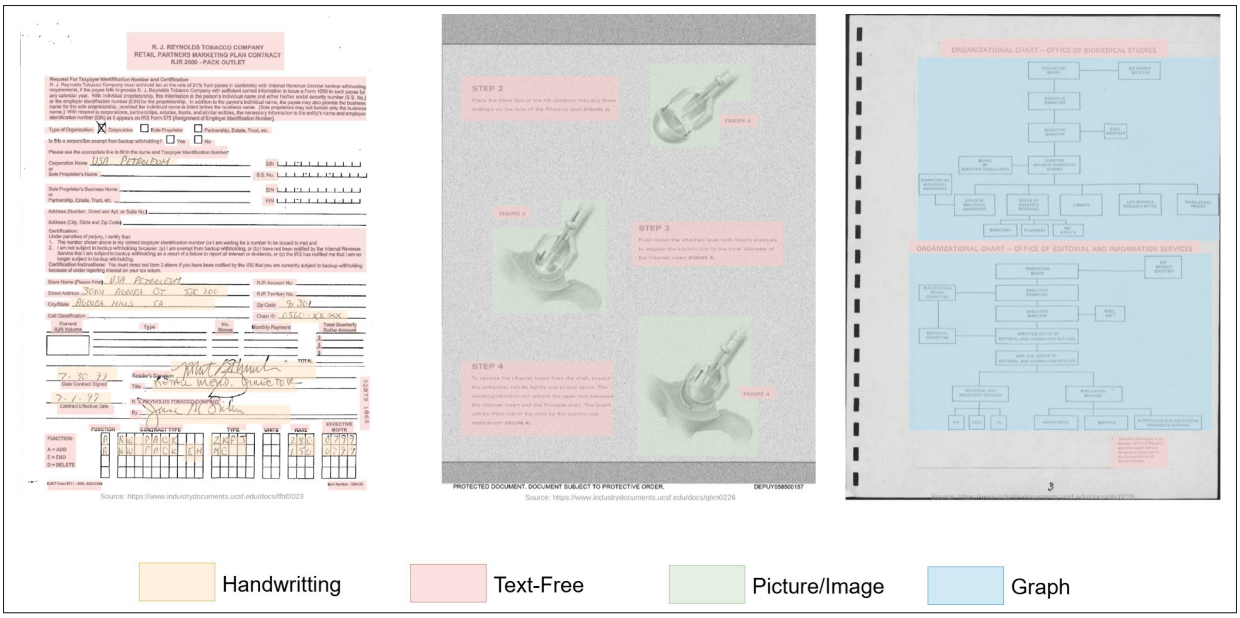
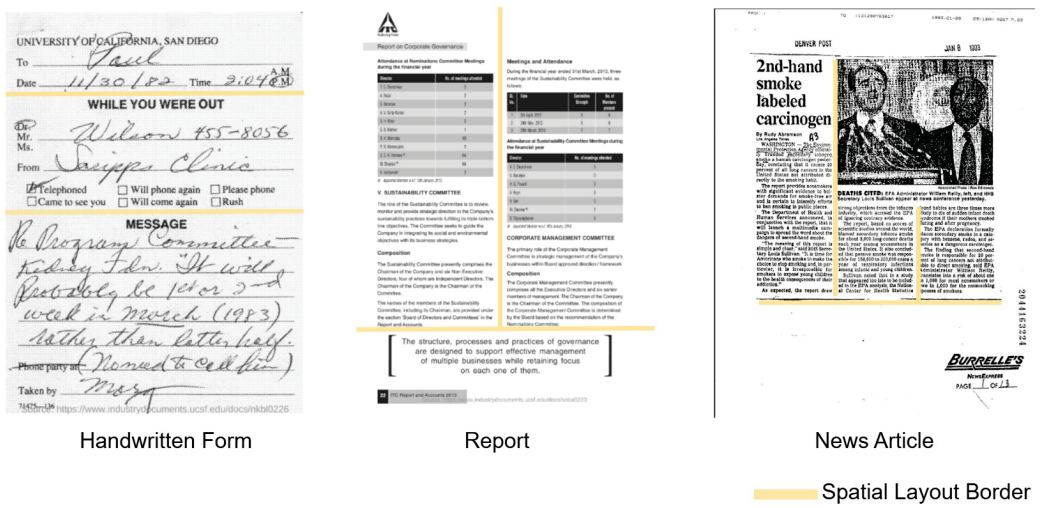


Figure 0.4 Variabilité des types d'information sur les images de documents

Les images de documents sont des données complexes ayant différentes composantes et structures, les rendant chacune uniques et variées par rapport aux autres. La figure 0.4 illustre différents types d'informations qui peuvent être retrouvées sur des documents. Ces types d'informations incluent les écritures manuscrites (handwriting), les textes écrits au clavier (text-free), les images (photo) et les graphiques. Ces types d'entités ont des caractéristiques différentes (formes, couleurs, tailles, etc.), ce qui peut complexifier leur extraction et leur représentation (**inter-variabilité**). D'autre part, une même entité peut avoir des variations d'un document à l'autre. Par exemple, pour les écritures manuscrites, un même mot peut être écrit différemment selon deux individus. Dans le cas d'une image, deux photos représentant le même objet peuvent avoir différentes caractéristiques selon l'angle et la luminosité de la photo. Ainsi, cette **intra-variabilité** est également un défi à prendre en compte pour représenter une image de document.

De plus, les images de documents ont des structures (layout) différentes en fonction de leurs types (lettre, article, formulaire, etc.). Ainsi, pour un document donné, l'ordre de lecture de ce dernier ne sera pas forcément le même que pour un autre. Cependant, il est essentiel de donner au



a) Structures spatiales d'images de documents, RVL-CDIP par Harley *et al.* (2015)

b) Structure sémantiques d'images de documents, DocLayNet par Pfitzmann *et al.* (2022)

Figure 0.5 Exemples de structures (layout) d'images de documents

modèle un ordre de lecture correct afin qu'il puisse retrouver les informations dans le document de manière cohérente. Le layout sémantique est également une composante importante, à savoir où se trouve le titre, quelle partie du document est un tableau, un graphique ou autre... Ainsi, il est essentiel d'introduire la position de chaque élément dans la représentation du document afin que le modèle puisse apprendre à induire la structure de ce dernier.

Les images de documents sont donc des données complexes, nécessitant un traitement différent d'un texte numérique pour la tâche de DocVQA. Cependant, elles doivent tout de même être représentées dans le même espace vectoriel que la question, afin que le LLM puisse générer une réponse correcte à partir de cette dernière.

Cela amène la problématique de ce mémoire ; **Comment représenter les images de documents, comportant différents types d'entités et des structures variées, dans le même espace de représentation qu'une question afin de résoudre la tâche de DocVQA ?**

Cette problématique est décomposée en plusieurs questions de recherches (QR), découlant de la complexité des images de documents et de leurs différences avec le texte :

- **QR 1** Comment représenter des images de documents contenant différents types d'entités ?
- **QR 2** Comment intégrer les positions spatiales des composantes du document afin d'en induire la structure de ce dernier ?
- **QR 3** Comment aligner la représentation d'une image de document avec une question représentée dans l'espace d'un modèle de langue (alignement image-texte) ?

Ainsi, la tâche de DocVQA est délicate avec des composantes variées dont le traitement d'image, le traitement du langage naturel, les modèles de langue et autres. Ce mémoire a pour but de se concentrer sur la partie représentation visuelle de cette tâche, essentielle à sa réalisation. Les images de documents pouvant également être variées, ici, le travail abordera la représentation d'images de documents industriels afin de répondre à l'augmentation de leur nombre dans les entreprises sur un large spectre de types de documents.

0.3 Structure du mémoire

Ce mémoire est organisé en huit chapitres. L'introduction présente le contexte, les motivations ainsi que les problématiques. Le prochain chapitre se concentre sur l'état de l'art des méthodes traitant ce sujet, afin de fournir au lecteur une compréhension des solutions existantes et de leurs limites. Il se conclut par la présentation de l'objectif du mémoire, divisé en sous-objectifs.

Le troisième chapitre est consacré à la théorie et aux fondements scientifiques des modèles utilisés. Les chapitres quatre et cinq présentent respectivement les méthodologies et approches mises en œuvre pour répondre aux deux sous-objectifs. Le chapitre six propose une extension du modèle aux documents multi-page. Le chapitre sept expose et discute les résultats obtenus. Enfin, le chapitre huit conclut ce mémoire et propose des pistes de recherche pour de futurs axes à explorer.

CHAPITRE 1

REVUE DE LA LITTÉRATURE ET OBJECTIFS

1.1 État de l'art

Cette section rassemble les différentes approches de l'état de l'art pour l'encodage d'images de documents dans le contexte de DocVQA.

1.1.1 Architectures d'apprentissage de représentation d'image de documents pour la tâche de réponse à des questions visuels de documents

Pour répondre à la tâche de DocVQA, il est essentiel de représenter les images de documents dans un espace de représentation qui contient les caractéristiques nécessaires des différents types d'entités pour répondre à la question.

En 2020, lors de la sortie du dataset DocVQA par Mathew, Karatzas & Jawahar (2021), les premières méthodes utilisaient des outils de reconnaissance de texte optique (Optical Character Recognition, OCR) qui permettent d'extraire et de convertir le texte des documents en texte digital afin de l'encoder avec la question (Mathew *et al.*, 2021). Cela permet d'avoir les deux entrées (contexte et question) dans la modalité de texte, directement exploitable par les modèles de langue basés sur l'architecture Transformer (Vaswani *et al.*, 2017). Cependant, cette technique rend le modèle « aveugle » aux informations du document non textuel comme les photos. Ainsi, les méthodes suivantes ont incorporé les caractéristiques visuelles des documents (Xu *et al.*, 2020; Huang, Lv, Cui, Lu & Wei, 2022). LayoutLMv2 par Xu *et al.* (2020) utilise en amont un réseau de convolution (Convolutional Neural Network, CNN) tel que ResNet par He, Zhang, Ren & Sun (2016). Ce dernier extrait des caractéristiques visuelles simples de l'image de manière locale par des fenêtres de convolution. Le Transformer encode par la suite ces caractéristiques de façon globale avec le texte extrait par OCR ainsi que la question, devenant de cette manière multimodal (voir figure 1.1b). Suite à cela, une version améliorée a été proposée par Huang *et al.* (2022), substituant le CNN par une simple couche linéaire, puis entraînant le modèle

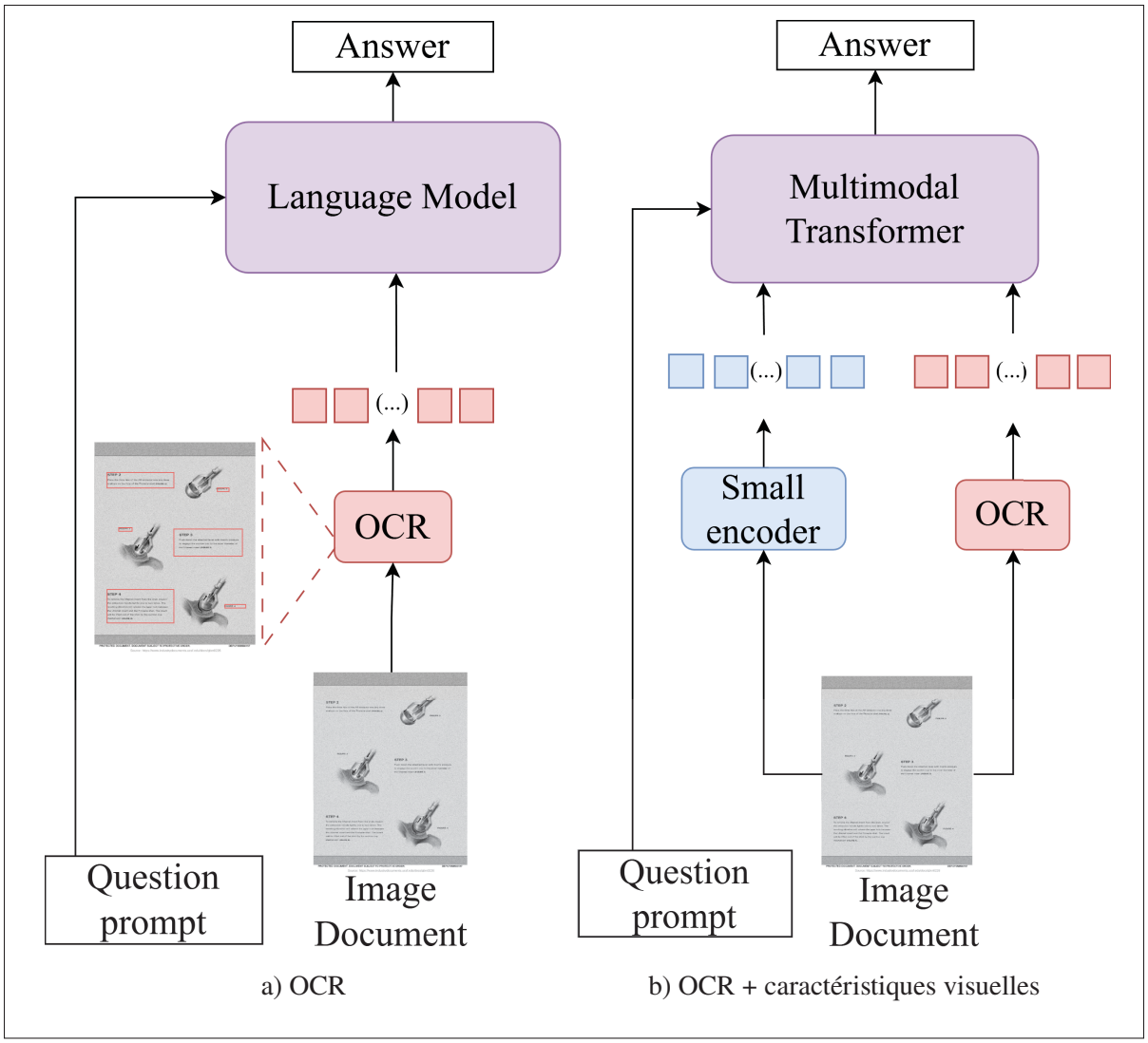


Figure 1.1 Premières méthodes utilisées pour representer un document pour la tâche de DocVQA

Transformer à encoder à la fois le texte extrait par OCR et l'image du document avec la question pour générer la réponse. Cependant, les méthodes se basant sur l'OCR nécessitent un coût de calcul supplémentaire pour détecter et reconnaître les composantes textuelles sur les documents. De plus, les erreurs de détection et de reconnaissance peuvent se propager dans le modèle et détériorer la qualité des réponses comme souligné par Kim *et al.* (2022).

En 2022, les architectures bout-à-bout (end-to-end) sont adoptées (Davis *et al.*, 2022; Kim *et al.*, 2022), se basant uniquement sur un encodeur visuel pour extraire et représenter les caractéristiques d'un document, sans appel à des outils externes tels que l'OCR. L'encodeur visuel de ces méthodes est dérivé de l'architecture initiale des Transformers, divisant le document en patchs afin de le représenter. Ces encodeurs basés sur les Swin Transformers par Liu *et al.* (2021) ont une architecture hiérarchique avec un coût de calcul linéaire, permettant d'avoir des résolutions d'images plus grandes et de capturer des détails à différentes granularités, renforçant la qualité de la représentation (voir section 2.3). Néanmoins, ces modèles end-to-end ayant peu de paramètres (lightweight), leurs performances restent assez faibles, ce qui peut rendre leur utilisation peu fiable dans un contexte industriel.

Suite à cela, les modèles de fondation en vision par ordinateur, basés sur l'architecture des Transformers Visuels (Vision Transformer, ViT) par Dosovitskiy *et al.* (2020) ont commencé à être développés. En augmentant leur taille avec plus de paramètres et, par extension, leurs capacités d'apprentissage (Radford *et al.*, 2021; Zhai, Mustafa, Kolesnikov & Beyer, 2023), ces modèles pré-entraînés à aligner les images avec le texte se sont montrés très efficaces en tant qu'encodeur visuel sur la tâche DocVQA. Cela a conduit à l'utilisation de modèles de vision-langage à grande échelle (Large Visual Language Model, LVLM) (Beyer *et al.*, 2024; Chen *et al.*, 2024; Gao *et al.*, 2024; Wu *et al.*, 2024), ayant un encodeur visuel large de fondation avec un modèle de langue à grande échelle (voir figure 1.2b). Ces modèles également end-to-end ont permis de repousser la limite de performances sur la tâche de DocVQA, améliorant les résultats des modèles à plus petite échelle cités précédemment. Ils peuvent être divisés en deux catégories : les modèles à résolution d'entrée fixe qui prennent toujours la même résolution en entrée et les modèles qui divisent les images en sous-images à résolution fixe, permettant d'avoir en entrée une image de résolution plus grande (tilling), (Chen *et al.*, 2024; Gao *et al.*, 2024; Wu *et al.*, 2024). Les encodeurs visuels de fondation étant non-hiérarchiques, leur coût de calcul est quadratique tel que $O(n^2)$ avec n le nombre de patchs de l'image (voir section 2.2.2.1), ce qui limite la résolution de l'image d'entrée pour la première catégorie. Ainsi, le tilling résout cela en incluant de la linéarité dans le coût de calcul tel que $O(n_t^2 \times t)$, avec t le nombre de tiles et n_t le

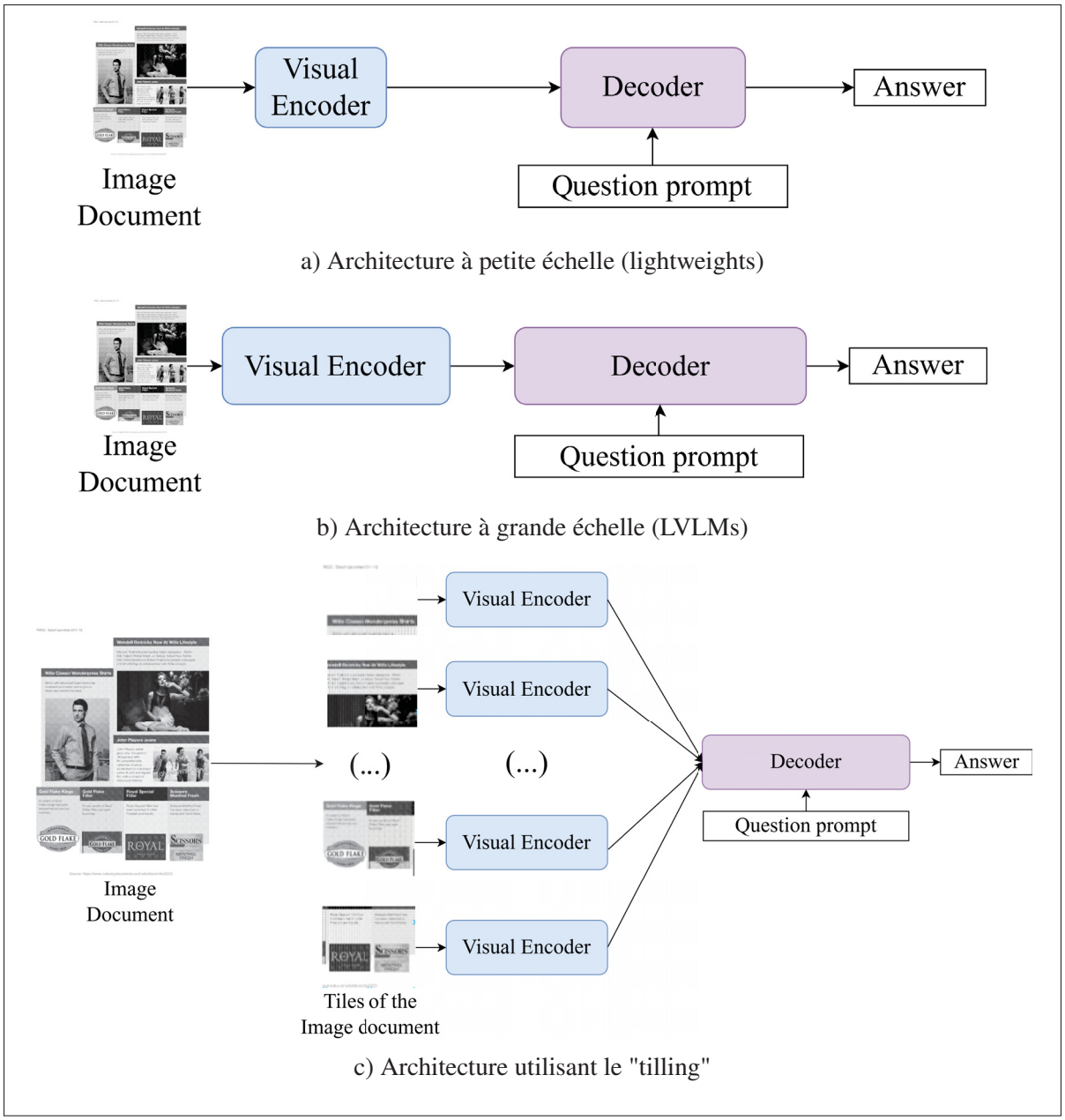


Figure 1.2 Méthodes bout-à-bout de l'état de l'art de DocVQA

nombre de patchs de chaque tile. Toutefois, cette méthode entraîne une étape de preprocessing supplémentaire et une parallélisation de l'encodage du document augmentant la complexité de l'architecture.

D'autre part, les LVLM ayant plus de paramètres, leur utilisation nécessite des infrastructures avec une plus grande capacité de calcul, pouvant entraîner des coûts supplémentaires.

Tableau 1.1 État de l'art de DocVQA

Méthode	Encodeur Visuel Spécificités			Modèle Transformer Spécificités			
	Architecture	Tilling	# Param (B)	OCR	Type	# Param (B)	ANLS (%) ↑
LVLMs							
InternVL, Chen *et al.* (2024)	CLIP	×	6		Décodeur Textuel	7	95.0
MiniInternVL, Gao *et al.* (2024)	CLIP	×	0.3		Décodeur Textuel	1.8	86.9
DeepseekVL2-Tiny, Wu *et al.* (2024)	SigLIP	×	0.4		Décodeur Textuel	3	88.9
Paligemma, Beyer *et al.* (2024)	SigLIP		0.4		Décodeur Textuel	2.6	84.77
Modèle à petite échelle bout-en-bout							
Donut, Kim *et al.* (2022)	Swin		0.074		Décodeur Textuel	0.126	66.26
Dessurt, Davis *et al.* (2022)	CNN				Encodeur-Décodeur Multimodal	0.127	63.2
Modèle se basant sur de l'OCR							
LayoutLMv3, Huang *et al.* (2022)	Patch embedding			×	Encodeur Multimodal	0.133	78.76
LayoutLMv2, Xu *et al.* (2020)	CNN			×	Encodeur Multimodal	0.2	78.08
Bert, Mathew *et al.* (2021)	-	-		×	Encodeur Textuel	0.110	63.5

1.1.2 L'encodage de position pour la tâche de DocVQA des modèles end-to-end

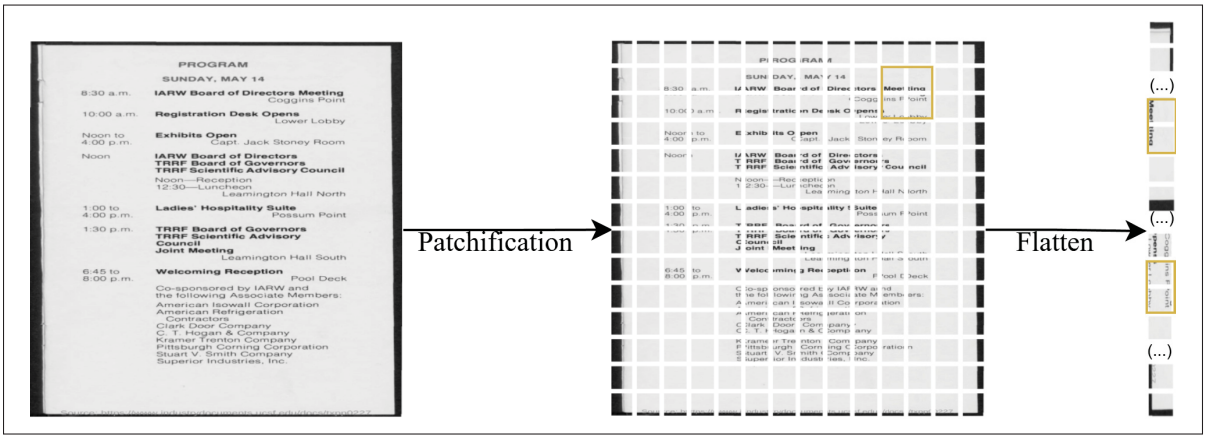


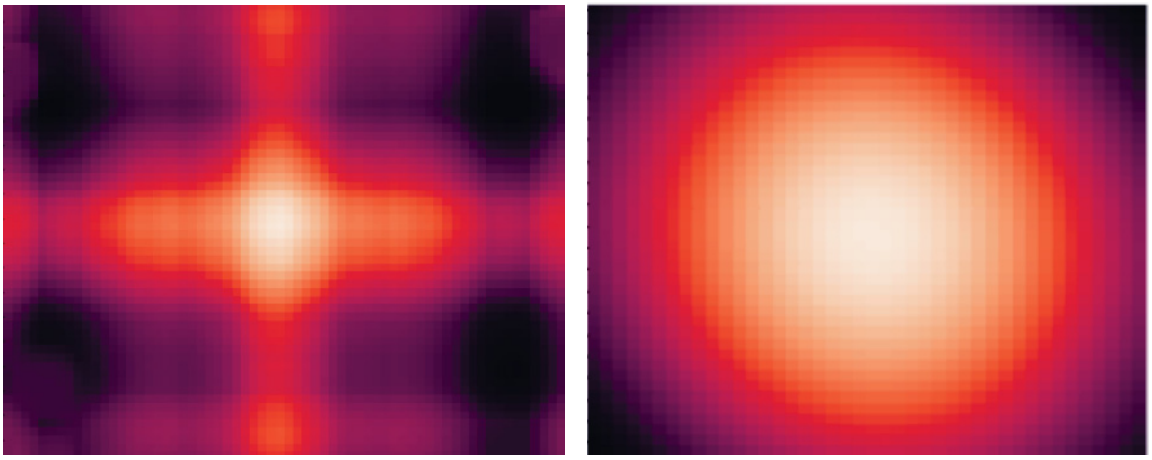
Figure 1.3 Division d'une image de document en patchs

Une image de document est spatialement représentée en deux dimensions (hauteur et longueur). La représentation par patchs des ViTs (voir section 2.2.1) nécessite de les traiter comme des séquences à une dimension en les aplatissant. La figure 1.3 représente cette étape. Cependant, l'aplatissement entraîne un changement d'ordre des patchs (voir encadré orange sur la figure). Ainsi, des patchs proches sur l'image originelle peuvent être éloignés dans la séquence, ce qui perturbe la structure du document traité. De plus, l'opération d'attention qui est au cœur des Transformers encodent tous les patchs de manière parallèle (voir section 2.2.2.1) et n'intègrent

pas la notion de position dans le calcul. Intégrer la position des patchs est donc essentielle pour représenter correctement le document.

Les Swin Transformers de l'état de l'art de DocVQA (Davis *et al.*, 2022; Kim *et al.*, 2022) sont basés sur le papier initial de Liu *et al.* (2021) et reprennent ainsi le même encodage de position. Cette architecture utilise un biais positionnel relatif pour les patchs contenus dans une fenêtre (voir section 2.3.2). Ainsi, cette approche limite localement l'information apportée par la position dans la représentation des patchs, ne procurant pas directement une information globale.

Les encodeurs visuels des LVLM sont quant à eux basés sur les modèles CLIP et SigLIP eux-



a) Similarités de position de l'encodeur de PaliGEMMA (LVLM) avec le patch central

b) Similarités de position spatialement correctes avec le patch central

Figure 1.4 Similarités de position du patch central d'une image de document

mêmes construits sur l'architecture initiale des ViTs (non-hiérarchiques). Ces méthodes utilisent un module de position à l'entrée du modèle, qui apprend pour chaque patch une représentation vectorielle de sa position sur le document, puis l'ajoute à la représentation du patch correspondant (voir section 2.2.1). Ainsi, ce module encode la position de manière absolue, chaque position encodée ne dépendant pas des autres. Cette approche, bien qu'apportant de bons résultats sur la classification d'images naturelles, ne permet pas d'encoder avec précision la position des patchs, rendant les positions orthogonales plus proches dans l'espace de représentations que les positions

réellement proches spatialement (voir figure 1.4). Ainsi, la position induite dans la représentation de chaque patch n'est pas corrélée avec la position spatiale réelle. Ces architectures étant construites et évaluées initialement sur des images naturelles, ce type d'encodage positionnel pourrait ne pas être le plus adéquat dans le contexte de DocVQA.

Tableau 1.2 État de l'art des encodages de positions (PE) des patchs pour la tâche DocVQA

Méthode	Information sur l'encodeur visuel et le module de position utilisé			
	Architecture/Papier fondation	Données d'évaluation	Type PE	Représentation des Positions
InternVL, Chen *et al.* (2024)	CLIP, Radford *et al.* (2021)	Image Naturelle	Absolue/Supervisé	Similarités orthogonales
MiniInternVL, Gao *et al.* (2024)	CLIP, Radford *et al.* (2021)	Image Naturelle	Absolue/Supervisé	Similarités orthogonales
DeepseekVL2-Tiny, Wu *et al.* (2024)	SigLIP, Zhai *et al.* (2023)	Image Naturelle	Absolue/Supervisé	Similarités orthogonales
Paligemma, Beyer *et al.* (2024)	SigLIP, Zhai *et al.* (2023)	Image Naturelle	Absolue/Supervisé	Similarités orthogonales
Donut, Kim *et al.* (2022)	Swin, Liu *et al.* (2021)	Image Naturelle	Biais relatif	Représentations locales
Dessurt, Davis *et al.* (2022)	Swin, Liu *et al.* (2021)	Image Naturelle	Biais relatif	Représentations locales

1.1.3 L'alignement des images de documents dans l'espace de représentation du modèle de langue

Les méthodes reposant uniquement sur l'OCR pour représenter un document peuvent directement encoder la question avec le texte de ce dernier, étant de la même modalité. Cependant, les méthodes utilisant des patchs visuels pour représenter le document doivent unifier l'espace de ces derniers avec le modèle de langue afin qu'il puisse les utiliser. De plus, les patchs représentent des fragments du document, ne contenant donc pas d'information complète à eux seuls, mais plutôt des morceaux d'entités (texte, images, tableaux, etc.). Pour permettre au modèle de langage de les traiter conjointement avec la question et de reconstruire l'information, il est nécessaire d'aligner les représentations visuelles et textuelles.

Dans le contexte de DocVQA, il existe deux paradigmes de fusion pour aligner les patchs avec la question :

Premièrement, la fusion en amont (early fusion) où l'image et la question sont traitées par un unique module de fusion qui peut être un encodeur multimodal ou une série de modules d'attention-croisée entre les patchs et la question (Huang *et al.*, 2022). Cette approche fusionne l'information

de manière précoce et potentiellement plus profonde. Dans ce cas, les représentations des patchs sont spécifiques à la question et doivent être recalculées pour chaque nouvelle requête.

Deuxièmement, la fusion intermédiaire (intermediate fusion) où l'encodage visuel est séparé du reste du modèle. Un encodeur visuel génère en premier lieu une représentation générale des patchs, qui peut être réutilisée pour différentes questions. Pour aligner cet espace de représentation avec celui du modèle de langue, un projecteur multimodal est ajouté, faisant le pont entre l'encodeur visuel et le modèle de langue (Kim *et al.*, 2022; Beyer *et al.*, 2024; Wu *et al.*, 2024; Chen *et al.*, 2024). Ce dernier est souvent constitué d'une simple couche linéaire ou d'un perceptron multicouche (MLP), projetant les patchs dans le même espace de représentation que la question suivant l'équation :

$$v_{lm} = v.W^{P} \tag{1.1}$$

dans le cas d'une simple couche linéaire et suivant les équations :

$$MLP(x) = W^{P2}\sigma(W^{P1}x) \tag{1.2a}$$

$$v_{lm} = MLP(v) \tag{1.2b}$$

dans le cas d'une projection par un MLP tel que $v \in \mathbb{R}^{N\times D}$ et $v_{lm} \in \mathbb{R}^{N\times D_{lm}}$ avec N le nombre de patchs, D la dimension de l'espace de représentation de l'encodeur visuel et D_{lm} la dimension de l'espace du modèle de langue.

De plus, il est nécessaire que le modèle de langue apprenne à analyser les patchs projetés conjointement avec la question afin de retrouver l'information correspondante dispatchée sur plusieurs patchs pour ensuite générer la réponse. Pour cela, les méthodes entraînent de bout en bout l'architecture complète sur des tâches de question-réponses afin que l'encodeur apprenne à encoder les patchs de manière à simplifier l'alignement par le projecteur multimodal et que le modèle de langue apprenne à retrouver des informations dispersées sur plusieurs patchs en fonction de la question afin de générer une réponse correcte.

Les documents pouvant être encodés pour extraire différentes informations, l'approche de la fusion intermédiaire semble préférable dans le contexte de ce mémoire afin de ne pas encoder un document indépendamment pour chaque question entrante par le système d'information (voir figure 0.2b).

1.2 Analyse de gap de la littérature

Les méthodes modernes de l'état de l'art de DocVQA ont démontré une progression accrue en comparaison aux premiers modèles. Les approches récentes montrent ainsi qu'il n'est plus nécessaire de se baser sur des outils externes tels que l'OCR pour représenter un document. En effet, les approches bout-en-bout proposent des systèmes unifiés basés sur les Transformer, capables de représenter des documents dans un espace multimodal utilisable par les modèles de langue pour répondre à des questions. Cependant, ces méthodes composées d'un encodeur visuel et d'un modèle de langue imposent un choix entre l'efficience et la performance. Les approches lightweights permettent de maintenir un coup de calcul faible en restreignant le nombre de paramètres (inférieur à 1B). Cependant, la taille restreinte de ces méthodes entraîne de faibles résultats sur la tâche de DocVQA, les rendant peu fiables dans des contextes d'utilisation industriels. D'autre part, les modèles de type LVLM, qui sont composés d'un grand encodeur visuel et d'un LLM, offrent des résultats solides mais nécessitent des infrastructures plus coûteuses du fait de leur nombre de paramètres (supérieur à 2B).
Ainsi, ces méthodes bout-en-bout imposent un choix entre efficience (taille des modèles) et performance (qualité des réponses), ce qui rend leur utilisation compliquée dans des contextes où ces deux métriques sont indispensables.

D'autre part, alors que la position des éléments des documents est une notion essentielle afin d'assurer un ordre de lecture correct et de faciliter la génération de la réponse, les méthodes bout-en-bout actuelles ne semblent pas avoir étudié l'encodage de position pour cette tâche. En effet, les encodeurs visuels dans ce contexte se basent sur les architectures initialement entraînées et évaluées sur des images naturelles. Les modèles lightweights utilisent un biais relatif par fenêtre, ne permettant pas de représenter la position globale de chaque élément

sur le document. Les LVLM utilisent des encodeurs visuels fondationnels (CLIP et SigLIP) composés d'un module d'encodage de position absolue à l'entrée du modèle, ne permettant pas de représenter la position de manière précise dans l'espace. De plus, les zones d'insertion de la position dans l'architecture de ces encodeurs visuels ne semblent pas plus étudiées, les modèles à petite échelle l'intégrant dans chaque opération d'attention (voir section 2.3.2) et les modèles de fondation à l'entrée de l'architecture (voir section 2.2.1).
Ainsi, là où les méthodes récentes de l'état de l'art se concentrent sur des approches bout-en-bout sans OCR, elles reprennent les architectures de fondation des encodeurs visuels en limitant les changements apportés à ces dernières. Cependant, ces modèles étant initialement entraînés sur des images de scènes naturelles, leurs modules de positions ne semblent pas optimaux pour la tâche de DocVQA. Ainsi, l'encodage de position est un aspect peu étudié dans l'état de l'art de DocVQA.

1.3 Objectifs du mémoire

L'objectif de ce mémoire est de **développer un modèle d'encodeur visuel capable de représenter un document afin de répondre à la tâche de DocVQA**.
Ce dernier est divisé en deux sous-objectifs :

SO1 - Réduire le coût de calcul de l'encodeur visuel d'un LVLM

Les modèles actuels ont démontré une capacité de répondre à la tâche de DocVQA sans OCR. Là où les méthodes légères sont efficientes mais peu robustes en termes de résultats, les LVLM montrent des résultats performants, surpassant même les méthodes utilisant de l'OCR. Cependant, ces modèles ayant beaucoup de paramètres, leur utilisation dans des environnements avec peu de puissance de calcul peut être compromise. Leurs encodeurs sont des modèles de fondation, dont la taille a permis de les rendre efficaces sur des données de différents types, pour des tâches variées, comme illustré par Chen *et al.* (2024). Cependant, les documents sont une sous-catégorie d'image bien spécifique, ne nécessitant pas forcément d'encodeur généraliste. Ainsi, cet objectif consiste à réduire la taille de l'encodeur visuel d'un LVLM afin de réduire son coût de calcul tout en conservant une représentation performante des images de documents.

SO2 - Intégrer un module de position spatial précis dans l'encodeur visuel

Les encodeurs visuels des méthodes bout-en-bout représentent le document en le divisant en patchs. Intégrer la position de ces derniers est donc un aspect important pour représenter un document. Cela aidera le modèle de langue à avoir un ordre de lecture correct des informations pour retrouver et générer la réponse à la question. Cependant, les méthodes actuelles se basent sur les papiers de fondation des architectures d'encodeurs visuels, évaluées sur des images naturelles, différentes des documents. Les modèles hiérarchiques ont un encodage relatif local ne permettant pas de représenter la position du patch sur l'ensemble du document de manière globale. Les encodeurs des modèles de fondations utilisent un encodage unidimensionnel, résultant en des positions de patchs dans l'espace de représentation qui ne reflètent pas leur position sur le document originel. D'autre part, l'insertion de la position dans l'architecture ne semble pas être plus étudiée. Cet objectif consiste donc à développer un module de position plus précis que ceux appliqués actuellement dans les modèles de DocVQA et à étudier la position d'insertion optimale dans l'architecture de l'encodeur visuel.

Par conséquent, ce mémoire se concentre sur l'apprentissage de représentation d'image de documents. Les deux sous-objectifs sont réalisés sur des documents d'une seule page afin de limiter les besoins en puissance de calcul lors des entraînements. Cependant, certains cas d'usage utilisent des documents comportant plusieurs pages, voire même des collections de documents (voir figure 0.2c). Ainsi, une dernière expérimentation sera de **construire un système se basant sur les résultats des sous-objectifs, capable de répondre à des questions sur des documents multi-pages**.

La figure 1.5 présente l'organisation des chapitres suivants, découlant des sous-objectifs présentés.

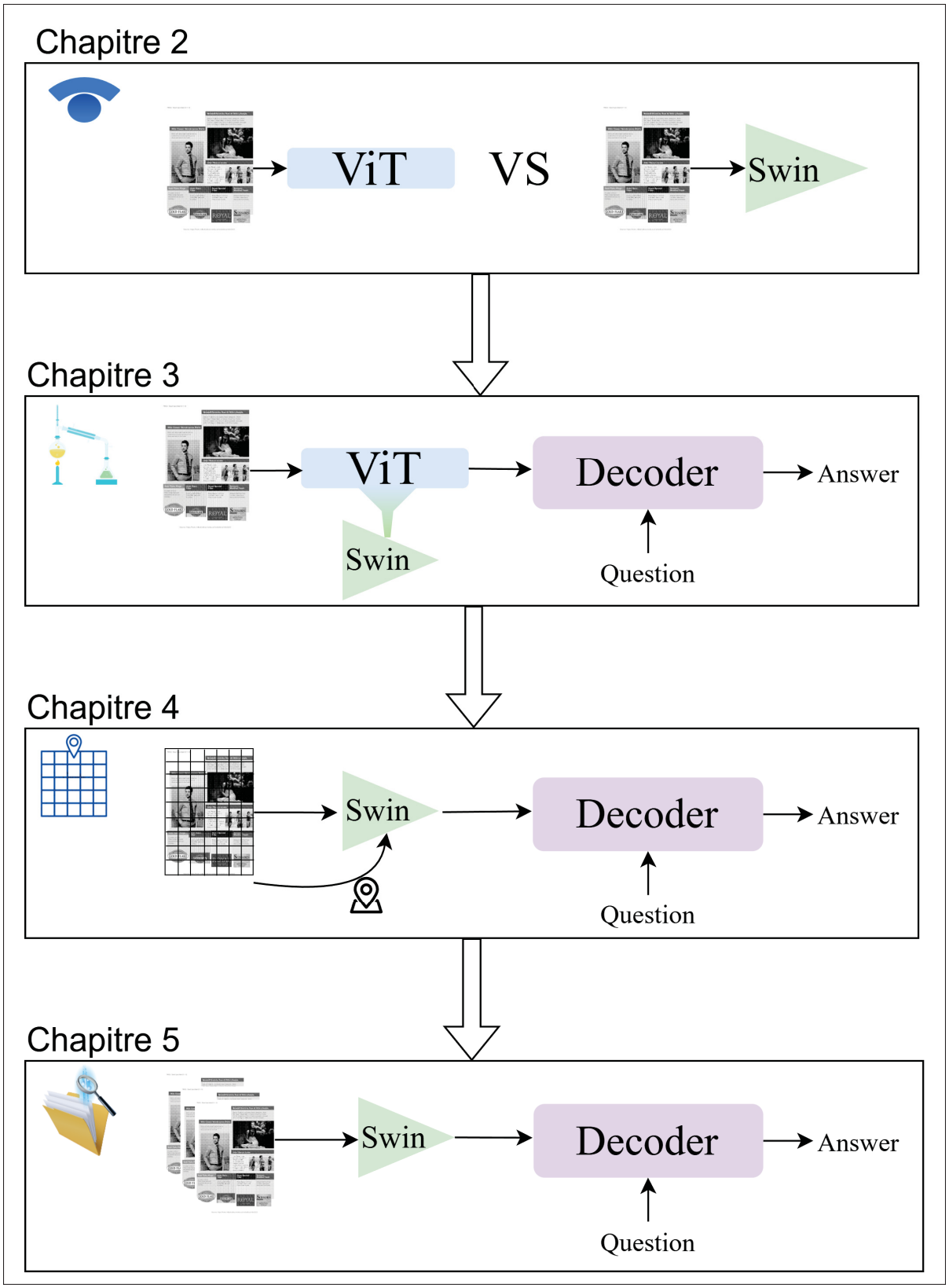


Figure 1.5 Organisations des chapitres suivant les sous objectifs présentés

CHAPITRE 2

L'APPRENTISSAGE DE REPRÉSENTATIONS DE DOCUMENTS AVEC LES TRANSFORMERS VISUELS

Ce chapitre aborde l'apprentissage de représentations abstraites d'images par des ViTs, architectures utilisées dans la suite de ce mémoire.

2.1 Les représentations abstraites d'images (embeddings)

Une image est un signal bidimensionnel, représenté par une matrice. Ce dernier est composé de pixels représentant l'intensité lumineuse en chaque point de l'image. Ces valeurs sont continues entre 0 et 255 et peuvent être représentées sur un seul canal si elles sont en noir et blanc ou trois canaux si elles sont en couleurs. Ainsi, une image peut se noter $I \in \mathbb{R}^{H \times W \times C}$, tel que H est le nombre de pixels sur la hauteur, W le nombre de pixels sur la largeur, et C le nombre de canaux. Selon la tâche, tous les pixels d'une image ne sont pas forcément pertinents, par exemple certaines zones comme le fond (background) d'un document sont moins importantes que les entités (textes, photos, etc.) qui le composent. Une entité sur une image est composée de plusieurs pixels. Pour des entités similaires, ces groupes de pixels ont souvent des variations locales d'intensité (formes/motifs visuels) qui se répètent.

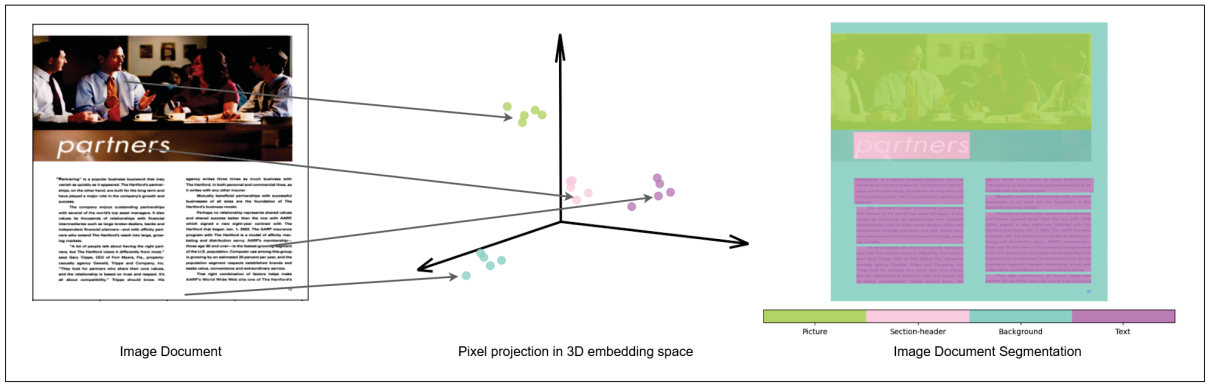


Figure 2.1 Schématisation d'un espace de représentation vectoriel pour la tâche de segmentation de documents

Une représentation abstraite ou embedding d'une image peut être définie comme un ou plusieurs vecteurs non compréhensibles pour l'œil humain mais contenant les caractéristiques et motifs visuels pertinents sur l'image pour répondre à une tâche. L'ensemble de ces vecteurs peut se noter $v \in \mathbb{R}^{N \times D}$ avec N le nombre de vecteurs pour représenter l'image et D la dimension de l'espace vectoriel. Ainsi, les vecteurs v représentant des entités similaires seront proches dans l'espace de représentations et inversement. Dans le cas d'apprentissage supervisé pour une tâche donnée, la représentation est apprise pour répondre au mieux à cette dernière. Par exemple, dans le cas d'une segmentation de document (Document Layout Analysis), l'objectif est de classifier chaque pixel avec le type d'entité auquel il appartient (titre, image, tableau, etc.). Ainsi, les formes/caractéristiques extraites de l'image et représentées dans l'embedding se concentrerons par exemple sur la position, la couleur et l'intensité des pixels, discriminant les éléments différents et rapprochant les éléments similaires dans l'espace vectoriel de représentation, permettant une classification correcte de chaque pixel (voir figure 2.1).

Les parties suivantes de ce chapitre se concentrent sur l'extraction et la représentation des caractéristiques visuelles afin de construire les embeddings d'images, en utilisant des ViTs.

2.2 Les Transformers Visuels

Les Transformers Visuels (Vision Transformers,ViTs) par Dosovitskiy *et al.* (2020), sont une variante appliquée à l'image de l'architecture Transformer (Vaswani *et al.*, 2017), initialement conçue pour le traitement du langage naturel. Cette section fournira la théorie derrière cette architecture qui s'est imposée comme fondation de la vision par ordinateur ces dernières années. La figure 2.2 illustre l'architecture des ViTs, qui sera une référence durant cette section.

2.2.1 Des pixels aux patchs

Du fait que les images de documents peuvent contenir des éléments de petite taille comme des écritures en bas de page (footnote), leur résolution peut être élevée, afin d'être capable de

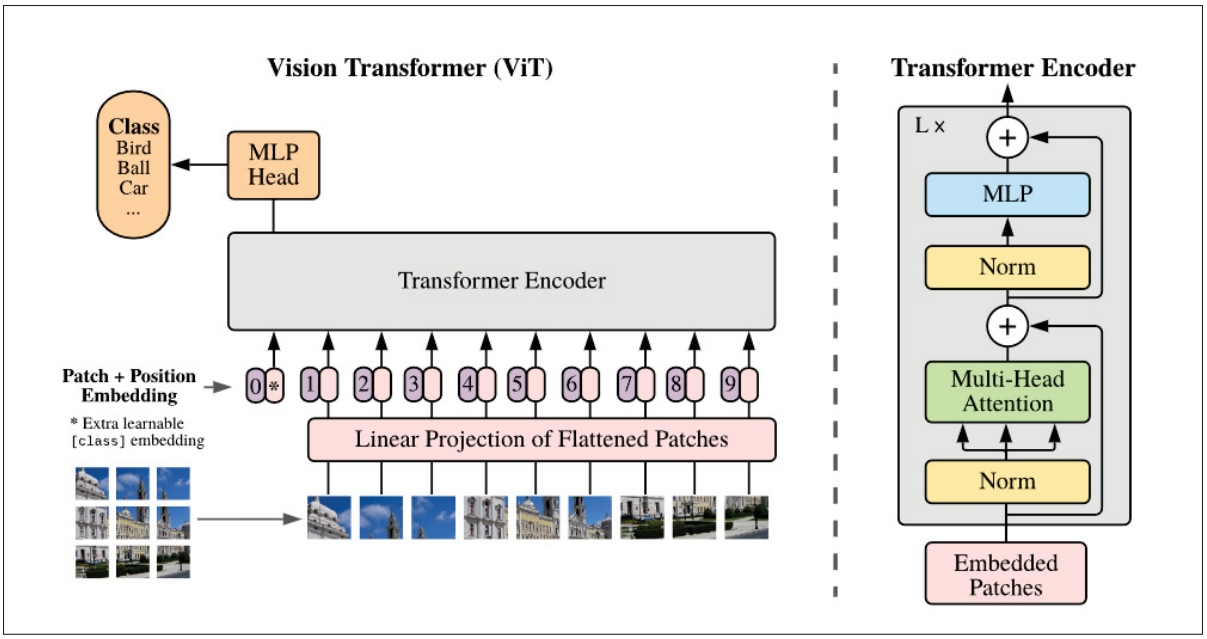


Figure 2.2 Architecture originel des Transformer Visuel, tirée de Dosovitskiy *et al.* (2020)

représenter chacun de ces éléments. Cependant, pour une image de document de haute résolution comme par exemple $I \in \mathbb{R}^{2560\times1920\times3}$, extraire caractéristiques visuels pertinentes parmi les 2560×1920 pixels peut s'avérer compliqué. De plus, traiter l'ensemble de ces pixels augmente rapidement le coût de calcul avec la résolution qui est quadratique chez les ViTs (voir section 2.2.2.1).

En vue de résoudre cela, pour une image d'entrée $x \in \mathbb{R}^{H\times W\times C}$, cette dernière va être divisée en N groupements de pixels non superposés, appelés patchs $x_p \in \mathbb{R}^{N\times P\times P\times C}$, de résolution $P \times P$ et C le nombre de canaux. Ces patchs sont ensuite aplatis et projetés linéairement dans un espace de représentation de dimension D. Ainsi, la dimension de l'image devient $x'_p \in \mathbb{R}^{N\times D}$. Là où la division de l'image en patch permet de réduire la résolution à traiter, la couche linaire permet de projeter chaque patch dans un espace de représentation vectoriel.

L'équation de cette transformation peut être notée telle que :

$$x'_p = W^{Emb}.flat(x_p) \tag{2.1}$$

avec $W^{Emb} \in \mathbb{R}^{D\times(P^2.C)}$ la projection linéaire et $flat$ la fonction d'aplatissement, retournant ici un vecteur de dimension $\mathbb{R}^{N\times(P^2.C)}$.

Les ViTs ne sont pas séquentiels, ils encodent chaque patch avec la même projection sans notion d'ordre, tout comme leur opération d'attention (voir section 2.2.2.1) qui encode chaque patch parallèlement. Ainsi, ils sont de base équivariants par permutation, sans notion de position dans la représentation ou dans la manière d'encoder. Les auteurs ont donc ajouté la position $x_{pe} \in \mathbb{R}^{N\times D}$ dans la représentation, qui est une matrice de poids apprise lors de l'entraînement. Cela permet d'inférer la position originelle de chaque patch sur l'image dans la séquence x'_p. Ainsi, la représentation de l'image enrichie par les positions spatiales peut être notée $z_0 \in \mathbb{R}^{N\times D}$ telle que.

$$z_0 = x'_p + x_{pe} \tag{2.2}$$

Dans le papier original, un patch supplémentaire est ajouté (classification token) et est utilisé pour des tâches après l'encodage de l'image (eg. classification). Comme cela n'est pas utilisé pour l'encodage dans le contexte de DocVQA, ce patch sera écarté pour la suite de cette thèse. Les ViTs étant composés de plusieurs couches, l'entrée de chacune d'entre elles sera notée z_{l-1}, avec l le numéro de la couche courante.

2.2.2 Le mécanisme d'attention

La projection de chaque patch dans l'espace de représentation se fait de manière locale et est ainsi indépendante des autres patchs. Or pour capturer des caractéristiques du document, un encodage global est nécessaire.

2.2.2.1 L'attention

L'attention (self attention) permet d'encoder les patchs de manière globale en fonction des caractéristiques visuelles qu'ils contiennent.

Dans le contexte de DocVQA, sur une image de document, le texte aura des propriétés différentes d'une photo/illustration (eg, couleur, forme, etc.), qui sont importantes à extraire. Ainsi, l'encodage a pour objectif de renforcer ces caractéristiques dans la représentation, en renforçant les patchs qui contiennent des caractéristiques pertinentes. Pour cela, l'ensemble des vecteurs de patchs z_{l-1} est encodé afin d'obtenir une requête (query) Q, une clé (key) K et une valeur (value) V telle que :

$$Q = z_{l-1}W_q \tag{2.3}$$

$$K = z_{l-1}W_k \tag{2.4}$$

$$V = z_{l-1}W_v \tag{2.5}$$

avec W_q, W_k et W_v des projections linéaires de dimension $\mathbb{R}^{D\times d_h}$. W_q projette chaque patch en fonction de la caractéristique recherchée, tandis que la W_k projette chaque patch dans le même espace d_h que la requête. Les projections W_q et W_k sont ainsi apprises conjointement afin de mettre en évidence les relations entre les patchs. Suite à cela, les scores d'attention sont calculés en faisant le produit scalaire (dot product) de Q et K, résultant en un coût opérationnel de $O(N^2)$. Les patchs qui auront un score élevé sont ceux dont la clé est fortement alignée/similaire à la requête, c'est-à-dire ceux qui se rapprochent le plus de la caractéristique recherchée. Ces scores sont normalisés et passés dans une fonction softmax afin que la somme des scores d'attention qu'un patch distribue à tous les autres patchs soit égale à 1. Ces scores servent ensuite de poids pour la matrice $V \in \mathbb{R}^{N\times d_h}$. Les patchs ayant des scores proches de 0 auront ainsi une magnitude faible dans l'espace de représentation, minimisant l'importance de l'information qu'ils contiennent. Les patchs ayant des scores élevés (proches de 1) auront aussi une magnitude plus grande dans la représentation et donc une influence plus élevée. Finalement, la formule

générale de l'attention peut se noter comme l'équation suivante :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_h}}\right)V \tag{2.6}$$

Ce mécanisme, en pondérant la combinaison des vecteurs de V, rapproche dans l'espace de représentation les patchs partageant des caractéristiques similaires et pertinentes pour la tâche.

2.2.2.2 L'attention multi-têtes

Comme décrit précédemment, le mécanisme d'attention permet de renforcer et rapprocher les patchs de caractéristiques pertinentes pour la tâche via les matrices Q et K. Par ailleurs, les caractéristiques pertinentes d'une image sont souvent multiples et peuvent nécessiter plusieurs scores pour être bien discriminées.

L'attention multi-têtes (multi-head attention) utilise le même mécanisme d'attention mais avec une requête Q^h, une clé K^h et une valeur V^h par tête h. Cela permet de rechercher dans chaque patch différentes caractéristiques (texture, forme, couleur, etc.) chacune associée à des paires de requêtes et de clés. Le nombre de têtes d'attention peut se noter H. Pour chaque tête $h \in \{0, \ldots, H-1\}$, les matrices de projection W_q^h, W_k^h et W_v^h sont définies, de dimensions $\mathbb{R}^{D \times d_h}$ correspondant respectivement aux requêtes, clés et valeurs, avec $d_h = \frac{D}{H}$. L'attention pour une tête h peut ainsi s'écrire :

$$\text{head}_h = \text{Attention}(Q^h, K^h, V^h) \tag{2.7}$$

Suite à cela, les résultats sont de nouveau concaténés, résultant en une dimension $\mathbb{R}^{N \times D_h.H}$. Afin d'unifier chaque patch avec sa nouvelle représentation, ces derniers passent à travers une dernière projection linéaire $W^O \in \mathbb{R}^{d_h.H \times D}$. L'attention multi-têtes peut donc se formuler telle que :

$$z'_l = \text{MultiHeadAttention}(z_{l-1}) = \text{Concat}(head_0, ..., head_{H-1})W^O \tag{2.8}$$

avec $z'_l \in \mathbb{R}^{N \times D}$ Ainsi ce mécanisme permet plus de flexibilité dans l'encodage en tenant compte de diverses caractéristiques.

2.2.3 Les couches de ViT

L'architecture ViT encadre le mécanisme d'attention par deux couches de normalisation, restreignant l'intervalle des valeurs dans l'espace de représentation et évitant des gradients trop importants. De plus, la représentation à l'entrée du bloc est ajoutée à la sortie du mécanisme d'attention permettant de garder en mémoire la représentation précédente tout en la modifiant uniquement en fonction des scores d'attention (voir figure 2.2). De plus, cet ajout résiduel permet de faciliter la retro-propagation des gradients et d'éviter le "vanishing gradient problem" comme expliqué dans le papier de ResNet (He *et al.*, 2016).

Enfin, après l'ajout de la couche résiduelle et la normalisation, chaque patch est projeté indépendamment des autres dans un petit réseau à deux couches (Feed-Forward) afin d'ajouter de la non-linéarité et d'apprendre des relations plus complexes entre les dimensions. Ce dernier est composé d'une première projection de dimension $W^0_{MLP} \in \mathbb{R}^{D \times DFF}$ avec $D << DFF$. L'espace de représentation DFF étant plus grand, il permet d'apprendre des relations plus complexes. Suite à cela, la représentation est passée dans une fonction d'activation GELU voir figure 2.3. Cela permet d'ajouter de la non-linéarité tout en régularisant les valeurs négatives sans pour autant désactiver celles proches de 0. Enfin, une dernière couche linéaire $W^1_{MLP} \in \mathbb{R}^{DFF \times D}$ est utilisée pour projeter à nouveau chaque patch dans l'espace de représentation initial.

$$\text{MLP}(x) = W^1_{MLP}(\text{GELU}(W^0_{MLP}(x))) \tag{2.9}$$

Une seconde couche résiduelle est ajoutée en sortie du MLP, pour les mêmes raisons que la première. Ainsi, une couche de ViT l peut être représentée par la suite d'équations suivantes :

$$z'_l = \text{MultiHeadAttention}(LN(z_{l-1})) + z_{l-1} \tag{2.10}$$

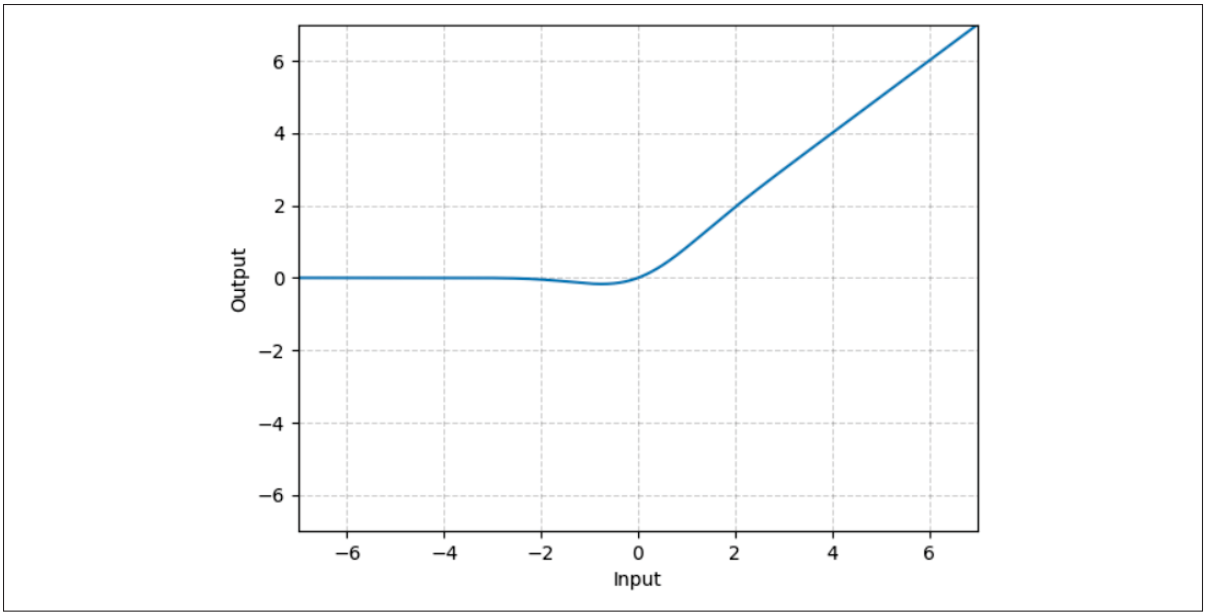


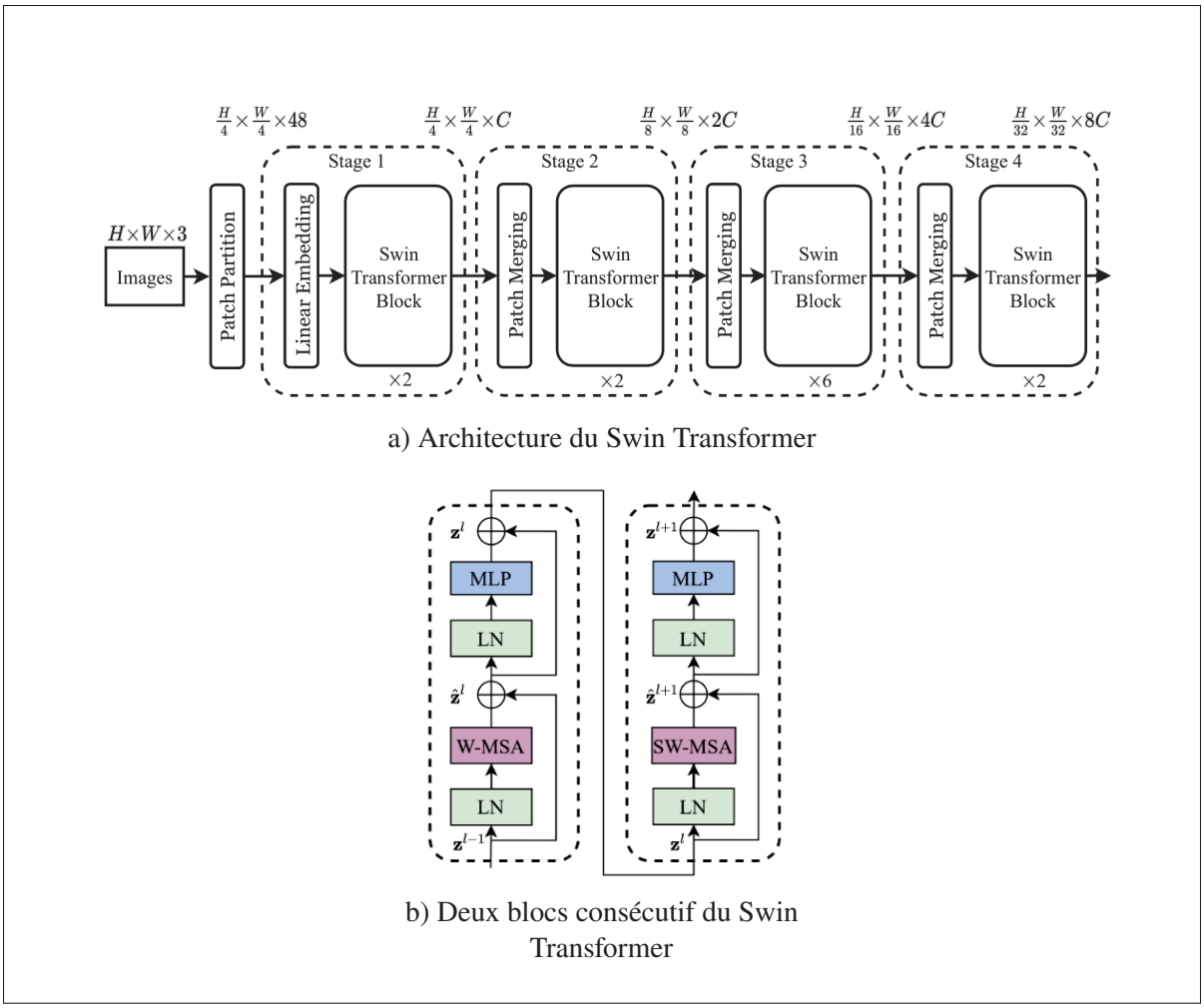
Figure 2.3 Fonction d'activation GELU

$$z_l = \text{MLP}(LN(z'_l)) + z'_l \tag{2.11}$$

Ainsi, pour chaque couche successive, les patchs ayant des caractéristiques importantes sont de plus en plus enrichis par les autres patchs ayant des caractéristiques similaires, permettant d'enrichir chaque patch avec des informations globales.

2.3 Les Transformers Visuels Hiérarchiques

Comme vu précédemment, les ViTs offrent une solide capacité d'apprentissage de représentations visuelles. Cependant, l'opération d'attention conduit à un coût de calcul quadratique ($O(N^2)$, limitant les résolutions d'image en entrée. Les Transformers visuels hiérarchiques comme les Swin Transformers par Liu *et al.* (2021) permettent de résoudre ce problème. Cette section a pour but d'expliquer le fonctionnement de ces derniers et leurs intérêts dans le contexte de DocVQA.



a) Architecture du Swin Transformer

b) Deux blocs consécutif du Swin Transformer

Figure 2.4 Schémas du Swin Transformer, tirée de Liu *et al.* (2021)

2.3.1 L'architecture hiérarchique

Les architectures hiérarchiques prennent en entrée une image $I \in \mathbb{R}^{H\times W\times C}$ et réduisent sa résolution au fur et à mesure qu'elle avance dans les couches profondes du réseau. Cela permet de capturer des caractéristiques à différentes échelles/niveaux, et ainsi de mieux représenter des éléments de différentes tailles. S'inspirant des architectures hiérarchiques de CNNs, le Swin Transformer (figure 2.5) allie l'opération d'attention des transformers avec une architecture hiérarchique. Les CNNs utilisent une opération de downsampling telle que le

pooling bidimensionnel, à l'échelle de pixels. Les Swin Transformers introduisent les couches de fusion de patchs (**patchs merging**).

Ce mécanisme divise la résolution par 2×2 en concaténant les patchs voisins sur la profondeur puis applique une couche linéaire $W_{merge} \in \mathbb{R}^{4C \times 2C}$ sur la concaténation. Ainsi, pour une résolution de $\frac{H}{4} \times \frac{W}{4} \times C$, la dimension après la concaténation sera de $\frac{H}{8} \times \frac{W}{8} \times 4C$ et $\frac{H}{8} \times \frac{W}{8} \times 2C$ après la projection linéaire. Après chaque couche de patchs merging, la résolution des patchs est donc divisée par 2 et la taille de l'espace de représentation est quand à elle multipliée par 2.

Les Swin Transformers sont composés d'une entrée qui, comme les ViTs, divise l'image en patchs et les projette dans l'espace de représentation (voir la section 2.2.1). Suite à cela, l'architecture est divisée en niveaux (**stage**) qui contiennent chacune une couche de patch merging (excepté pour la première). Puis chaque niveau est composé d'une succession de couches (block, voir figure 2.4b) similaires aux couches de ViTs (voir Section 2.2.3).
Ainsi, l'architecture hiérarchique du Swin Transformer permet de prendre en entrée des images de haute résolution, ce qui est utile dans la compréhension de documents pour représenter des éléments de petite taille tels que le texte. Cependant, l'opération d'attention des ViTs étant quadratique, son utilisation est mal adaptée pour des images de haute résolution, surtout sur les premiers niveaux où la résolution est encore élevée.

2.3.2 Les fenêtres d'attentions

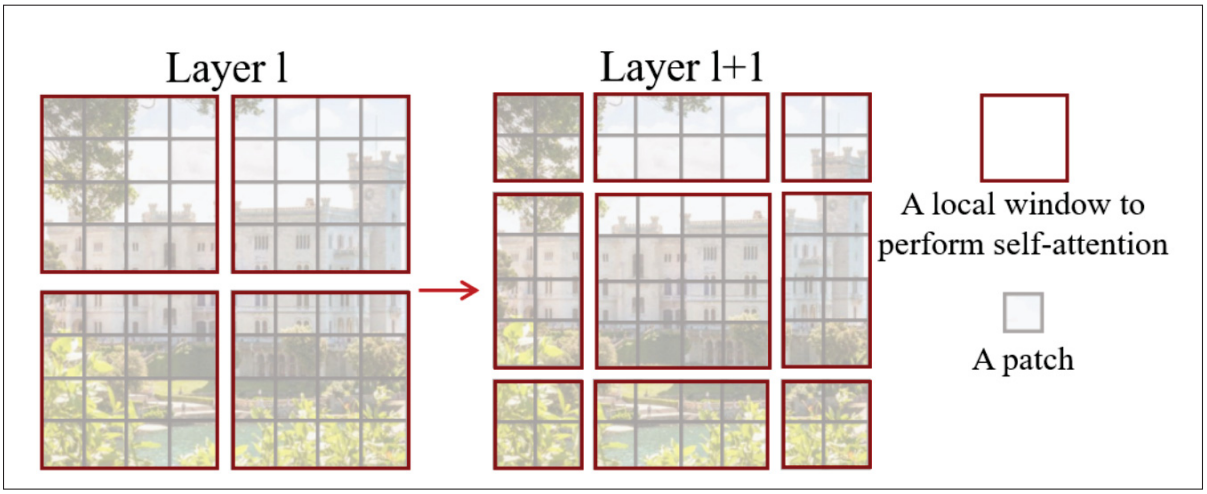


Figure 2.5 Illustration des fenêtres et fenêtres décalées d'attention tirée de Liu *et al.* (2021)

Afin de résoudre le coût de calcul quadratique de l'attention dans les ViTs, les auteurs de l'architecture Swin Transformer proposent une approche par fenêtre (window-multihead self-attention, W-MSA). Dans ce mécanisme, les patchs sont regroupés par des fenêtres contenant M patchs. Ainsi, l'attention est uniquement calculée entre les patchs d'une même fenêtre. Le coût de calcul de l'opération d'attention sera donc de $O(M^2 w)$ avec $w = \frac{N}{M}$ le nombre de fenêtres et $M^2 << N^2$. Ainsi, ce mécanisme permet de réduire le coût de calcul de l'attention passant de $O(N^2)$ pour les ViTs classiques à $O(NM)$. Cependant, cette attention étant locale, les connexions entre chaque patch sont restreintes. Ainsi, les auteurs proposent d'alterner entre deux répartitions de fenêtres différentes (shifted-windows, SW-MSA) afin de permettre des connexions croisées entre les patchs voisins de différentes fenêtres.

De plus, les auteurs ont introduis un biais spatial $B \in \mathbb{R}^{M \times M}$ qui est ajouté dans l'opération d'attention tel que

$$\text{W-Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_h}} + B\right)V \tag{2.12}$$

avec $Q, K, V \in \mathbb{R}^{M \times d_h}$. Ce biais remplace l'encodage de position des ViTs (voir Section 2.2.1) et démontre de meilleures performances sur les tâches de vision sur des images de scènes naturelles.

Finalement, les fenêtres d'attention permettent de restreindre le coût de l'opération d'attention, permettant à cette architecture de prendre en entrée des images de plus haute résolution tout en limitant l'augmentation du coût de calcul, même dans les premières couches.

2.4 Conclusion sur les ViTs

Ainsi, les ViTs sont une architecture efficace pour extraire et représenter les caractéristiques d'une image. Ayant fait leurs preuves sur les images naturelles, ils sont également utilisés dans l'état de l'art de DocVQA pour encoder une image de document afin qu'elle puisse être utilisée par un modèle de langue dans le but de répondre à une question. Comme expliqué dans le chapitre 1, les architectures classiques (section 2.2) sont utilisées dans les modèles à grande échelle (LVLMs), composés d'un encodeur visuel et d'un modèle de langue ayant beaucoup de couches, entraînant de bonnes performances mais nécessitant plus de ressources de calcul. Les Swin Transformer sont quant à eux, utilisés dans les modèles de petites tailles, requérant moins de ressources, mais dégradant les résultats. D'autre part, les méthodes d'encodage de position de ces modèles ont été évaluées sur les images naturelles dans les papiers originels, sans investigation supplémentaire pour la tâche de DocVQA. Pour poursuivre, les chapitres suivants présentent les méthodologies et approches expérimentales afin de résoudre ces gaps dans l'état de l'art.

CHAPITRE 3

ALLÉGER SANS OUBLIER : TRANSFÉRER LES CAPACITÉS D'UN MODÈLE FONDATION VERS UN ENCODEUR LÉGER

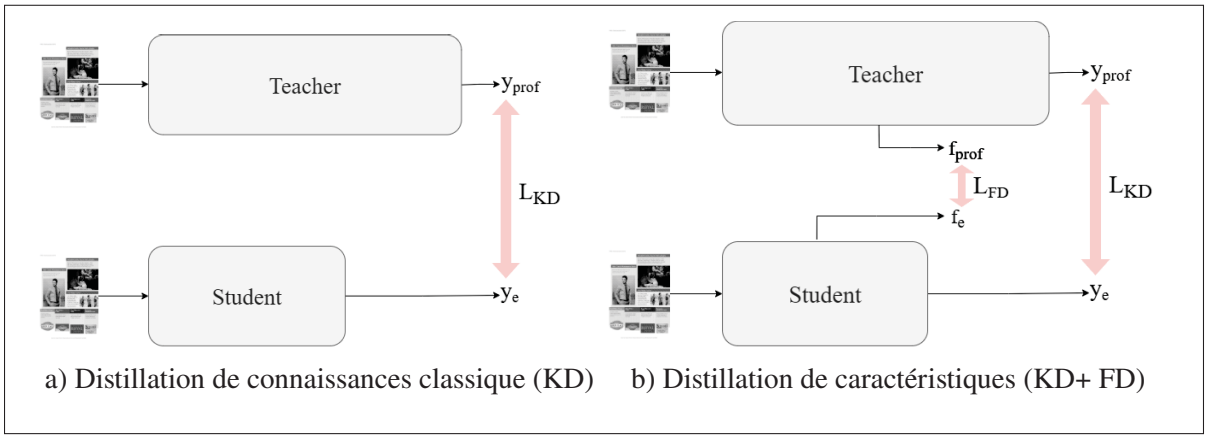
Ce chapitre décrit la méthodologie utilisée pour résoudre le premier objectif de ce mémoire qui est la réduction de l'encodeur visuel d'un LVLM. Comme indiqué dans le chapitre 1, les modèles de la tâche de DocVQA sont soit constitués de petits modules, requérant peu de puissance de calcul mais résultant en une performance limitée, soit de larges modèles de fondation qui repoussent les limites des petits modèles mais avec un coût de calcul plus excessif. Ainsi, l'état de l'art impose un choix entre performance et efficience. L'objectif de ce chapitre est donc de pallier à cela en réduisant la taille de l'encodeur visuel d'un modèle de fondation tout en gardant un LLM en décodeur, afin de conserver des résultats compétitifs, tout en réduisant le coût de calcul.

Les encodeurs visuels de fondation sont basés sur des architectures classiques telles que CLIP Radford *et al.* (2021) et SigLIP Zhai *et al.* (2023), entraînant une complexité d'attention qui évolue de manière quadratique avec le nombre de patchs (voir section 2.2). Les images de documents pouvant être de haute résolution, utiliser un modèle hiérarchique tel que le Swin transformer (voir Section 2.3.1), qui a une complexité linéaire, permettrait de prendre en entrée des images plus grandes tout en limitant le coût de calcul. L'approche choisie pour cet objectif a donc été la distillation, qui permet un transfert de connaissances entre architectures hétérogènes.

3.1 La réduction de modèles

Les premières méthodes de réduction de modèles se basent sur la suppression de paramètres non utilisés pour résoudre la tâche (pruning) introduite par LeCun, Denker & Solla (1989). Avec l'avènement des modèles profonds (AlexNet par Krizhevsky *et al.* (2012)), les méthodes de pruning ont été développées pour réduire la taille et le coût de calcul de ces méthodes (Han, Pool, Tran & Dally, 2015). Plus récemment, avec l'arrivée des ViTs comme modèles de référence pour le traitement d'image, ces méthodes ont été adaptées à leurs différents types de couches (*eg.*, têtes d'attention, MLP, etc.) comme par Yang *et al.* (2023), permettant de réduire le coût de

calcul de ces modèles. Cependant, même si le pruning a démontré une grande efficacité pour réduire la taille des modèles tout en conservant une bonne performance, cette technique ne permet pas de changer l'architecture du modèle initial.



a) Distillation de connaissances classique (KD) b) Distillation de caractéristiques (KD+ FD)

Figure 3.1 Schématisation de la distillation

La distillation de connaissances (Knowledge Distillation, KD) par Hinton, Vinyals & Dean (2015) vise à transférer les connaissances d'un modèle professeur vers un modèle plus petit appelé étudiant (voir figure 3.1a). Pour cela, la supervision de l'étudiant se fait par la sortie du professeur tel que :

$$L_{KD} = L(y_e, y_{prof}) \tag{3.1}$$

$$Loss = \lambda_1 L(y_e, y_{gt}) + \lambda_2 L_{KD} \tag{3.2}$$

avec y_e, y_{prof} et y_{gt} les valeurs prédites respectives de l'étudiant, du professeur, et la réponse correcte (ground truth) avec L une fonction de coût. λ_1 et λ_2 sont des coefficients induisant l'importance de chaque erreur dans la mise à jour des poids. Ainsi, l'étudiant apprend à imiter les prédictions d'un modèle plus gros, le permettant de se rapprocher des performances de ce dernier pour une tâche spécifique tout en ayant un coût de calcul plus faible. Dans ce contexte, le professeur est déjà entraîné et n'est donc pas supervisé lors de la distillation (offline distillation). Certaines méthodes ont été développées pour entraîner le professeur en même temps que l'étudiant (online distillation) permettant de superviser un ou plusieurs étudiants avec un ou

plusieurs professeurs (Zhang, Xiang, Hospedales & Lu, 2018). D'autres approches utilisent un seul modèle qui renforce son propre apprentissage en étant à la fois le professeur et l'étudiant (self-distillation) introduite par Zhang, Bao & Ma (2021), permettant aux couches profondes de superviser des couches précédentes. Dans le cas de ce mémoire, le modèle professeur est déjà entraîné, l'objectif étant de le réduire. La première catégorie de méthodes est choisie.

La distillation de connaissances classique consiste à superviser l'étudiant à l'aide des sorties du professeur. Cependant, cette approche laisse à l'étudiant la liberté de construire ses propres représentations internes, en cherchant uniquement à reproduire les sorties finales du professeur. La distillation de caractéristiques (Feature Distillation, FD) introduite par Ba & Caruana (2014) étend cette idée en guidant l'étudiant à travers une supervision supplémentaire sur une couche intermédiaire (hint layer) de l'enseignant, transférant ainsi une partie de ses connaissances internes vers le modèle plus léger (voir figure 3.1b). L'équation de cette méthode peut s'écrire :

$$L_{FD} = \lambda_3 L_f(f_e^l, f_{prof}^h) \tag{3.3}$$

$$Loss = \lambda_1 L(y_e, y_{gt}) + \lambda_2 L_{KD} + \lambda_3 L_{FD} \tag{3.4}$$

avec f_e^l la sortie de l'étudiant sur la couche intermédiaire choisie, f_{prof}^h la hint layer, L_f la fonction de perte et λ_3, le coefficient d'importance de cette supervision dans l'apprentissage.
La distillation de caractéristiques nécessite tout de même d'avoir un espace de représentation de même dimension pour le professeur et l'étudiant, et dans le cas d'une image, d'avoir des cartes de caractéristiques (feature maps) de même taille afin de permettre le calcul de l'erreur. Différents travaux ont été menés pour distiller des réseaux neuronaux convolutifs (CNNs) (Kim, Park & Kwak, 2018; Chen, Choi, Yu, Han & Chandraker, 2017; Lin *et al.*, 2022; Chen, Liu, Zhao & Jia, 2021; Chen *et al.*, 2022a), lesquels, en raison de leur structure hiérarchique, ont des dimensions de feature maps qui varient entre l'enseignant et l'étudiant. Pour permettre la distillation des caractéristiques, des projecteurs convolutionnels (petites couches de convolution) sont utilisés afin d'aligner les dimensions des cartes de caractéristiques de l'enseignant et

de l'étudiant. Les méthodes initiales (Kim *et al.*, 2018; Chen *et al.*, 2017; Lin *et al.*, 2022; Chen *et al.*, 2021) consistent à entraîner l'ensemble du modèle sur la tâche, ce qui réduit l'interprétabilité de la distillation et ajoute des étapes dans l'entraînement. Pour y remédier, SIMKD (Chen *et al.*, 2022a) propose de ne distiller que les cartes de caractéristiques de la dernière couche convolutionnelle, tout en réutilisant la tête de classification de l'enseignant. Ainsi, ils n'entraînent que l'encodeur de l'étudiant en supervisant sa représentation interne par le professeur.

Le coût élevé en temps d'entraînement et nombre de données requis par les ViTs a conduit à l'introduction de la distillation pour ces derniers. DearKD par Chen *et al.* (2022b) utilise un CNN comme enseignant pour entraîner un étudiant ViT. Afin de permettre la distillation, les dimensions de sortie du ViT sont alignées avec les cartes de caractéristiques du CNN grâce à une interpolation bilinéaire. Les auteurs Liu *et al.* (2022) améliorent les performances d'un CNN en utilisant un ViT comme enseignant. Ils alignent les cartes de caractéristiques du CNN avec les dimensions de sortie du ViT à l'aide d'un petit MLP. Ces méthodes ont montré qu'il est possible de distiller des architectures hétérogènes en alignant les dimensions de leurs représentations internes. Cependant, elles prennent la même résolution d'entrée et se concentrent uniquement sur la distillation inter-architecture entre des paires CNN/ViT.

Plus récemment, d'autres études ont utilisé la distillation entre des paires de ViTs. Yang *et al.* (2024a) étudient comment utiliser la distillation entre ViTs avec la même résolution d'entrée et le même nombre de patchs en sortie. En appliquant une simple projection linéaire pour aligner les dimensions d'embedding de l'étudiant avec celles de l'enseignant, ils ont montré que la distillation entre ViT est efficace pour réduire leur taille en conservant la même architecture. Yang *et al.* (2024b) ont quant à eux distillé des ViTs d'architecture hétérogène tels que CLIP en professeur et un Swin en étudiant. Leurs résultats montrent que la distillation entre architectures hétérogènes conduit à de meilleures performances que l'approche précédente. Là où ces méthodes se concentrent uniquement sur des tâches simples comme la classification d'image, les avancées récentes en DocVQA ont conduit à certains travaux utilisant la distillation de caractéristiques

afin de réduire la taille de l'encodeur visuel (Gao *et al.*, 2024; Van Landeghem *et al.*, 2024). MiniInternVL par Gao *et al.* (2024) réduit la taille de l'encodeur visuel InternViT (Chen *et al.*, 2024), en le faisant passer de 6B paramètres à 300M, ce qui reste une taille conséquente. DistillDoc par Van Landeghem *et al.* (2024) adapte la méthode de SIMKD Chen *et al.* (2022a), en ne distillant que l'encodeur visuel, mais en utilisant l'OCR pour aider le décodeur sur la tâche. De plus, ces méthodes supposent que les paires étudiant/enseignant prennent la même résolution d'image et retournent le même nombre de patchs en sortie. Par conséquent, elles ne traitent pas le problème d'alignement du nombre de patchs.

En conclusion, la distillation de caractéristiques est une méthode efficace pour transférer un apprentissage d'un modèle professeur vers un modèle plus petit. Des études ont montré l'efficacité de la distillation entre différents types de modèles (CNN et ViT). Cependant, les méthodes de distillation de caractéristiques entre ViTs contraignent la résolution d'entrée de l'étudiant afin que ses dimensions en sortie soient alignées avec celles du professeur. Cependant, les images de documents pouvant être de haute résolution, étudier différentes tailles de ces dernières pourrait s'avérer important pour faire varier les coûts de calcul et les performances pour un même étudiant.

3.2 Méthodologie et Architecture

Les LVLMs peuvent être séparés en deux groupes : les méthodes utilisant des résolutions d'entrée variables (Wu *et al.*, 2024; Chen *et al.*, 2024; Gao *et al.*, 2024) et celles ayant une résolution fixe (Beyer *et al.*, 2024). La première catégorie repose sur la séparation d'une image en sous-parties (tiles) qui sont envoyées parallèlement au même encodeur. Chaque tile a la résolution attendue par l'encodeur visuel, ainsi pour un modèle prenant en entrée une image de résolution $I \in \mathbb{R}^{H \times W \times C}$, une image de haute résolution $I_{HD} \in \mathbb{R}^{H_l \times W_l \times C}$ sera divisée en t tiles avec $t_i \in \mathbb{R}^{H \times W \times C}$. Cette méthode permet aux modèles de prendre en entrée des images de plus haute résolution tout en limitant le coût de l'attention. Néanmoins, cela entraîne un coût de prétraitement plus élevé (Wu *et al.*, 2024) et résulte en une représentation du document

non globale en sortie de l'encodeur. Le choix du modèle professeur s'est donc porté sur la deuxième catégorie. PaliGEMMA par Beyer *et al.* (2024) est un LVLM prenant une résolution fixe en entrée, composé d'un encodeur visuel SigLIP-SO400M (Zhai *et al.*, 2023) et d'un LLM décodeur Gemma-2B (Mesnard *et al.*, 2024). La taille de son décodeur qui n'est pas excessive et ses performances compétitives sur la tâche de DocVQA en font un professeur idéal à distiller.

L'encodeur visuel étudiant est un Swin Transformer (Liu *et al.*, 2021; Kim *et al.*, 2022), choisi pour ses propriétés hiérarchiques, son coût d'attention linéaire ainsi que sa forte représentation dans l'état de l'art avec plusieurs modèles pré-entraînés open-source. Ainsi, ses poids sont initialisés avec ceux du modèle Donut par Kim *et al.* (2022), déjà entraîné sur la tâche de DocVQA, ce qui permet de limiter les ressources nécessaires à l'entraînement. Inspiré par SIMKD (Chen *et al.*, 2022a), la projection multimodale et le décodeur de PaliGEMMA sont réutilisés dans l'architecture pour éviter un entraînement partant de poids non initialisé. Le modèle résultant a été nommé "Downscaling Image Visual Encoder for DocVQA" (DIVE-Doc).

3.2.1 Transfert de connaissance et alignement

La stratégie d'entraînement est divisée en deux étapes, permettant le transfert de connaissances et l'alignement entre le nouvel encodeur et le LLM décodeur.

La première phase est la distillation de l'encodeur de PaliGEMMA dans l'étudiant Swin. Les sorties de l'étudiant et du professeur sont respectivement notées $v^S \in \mathbb{R}^{N_S \times D_S}$ et $v^T \in \mathbb{R}^{N_T \times D_T}$. N_S et N_T représentent le nombre de patchs, tandis que D_S et D_T sont les dimensions des embeddings. Afin d'étudier l'impact de différentes résolutions d'entrées tout en permettant la distillation, deux stratégies de distillation sont explorées :

1.*Distillation à Résolution Fixe (Fixed-Resolution Distillation, FRD)*

Dans ce cas, une approche classique de distillation de caractéristiques est adoptée, où la résolution d'entrée est ajustée de manière à contraindre $N_S = N_T$ en sortie de l'étudiant. Les patchs produits par ce dernier sont alors simplement projetés par une couche linéaire dans

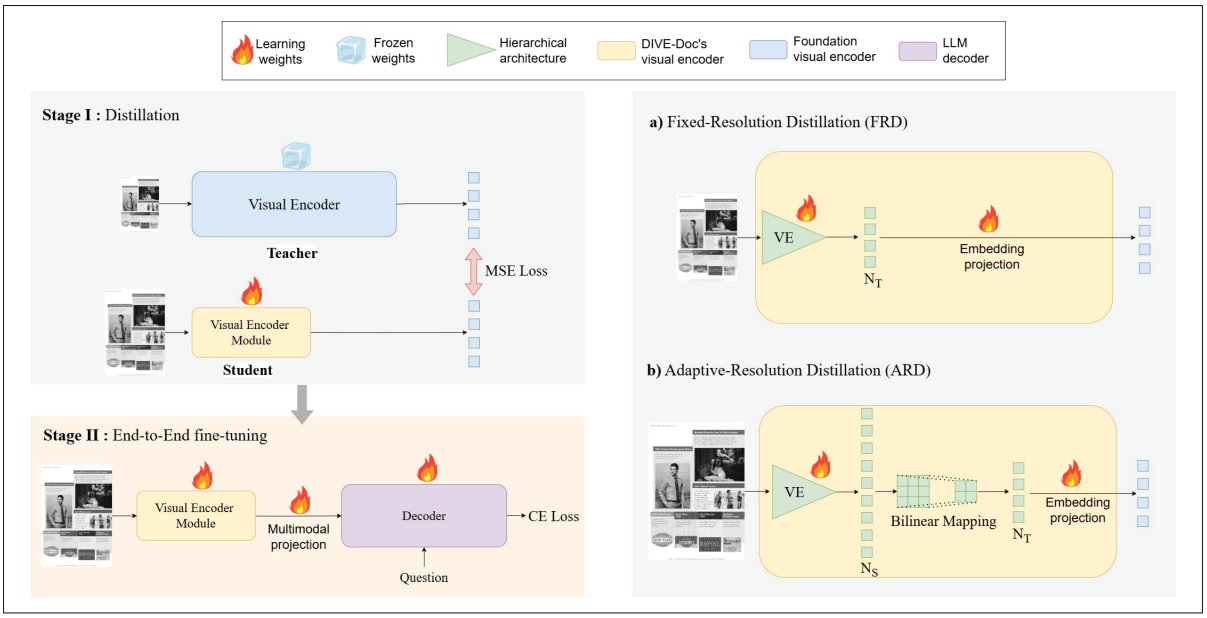


Figure 3.2 Stratégie d'entraînement de DIVE-Doc

l'espace de représentation du professeur D_T. Cette méthode, bien que simple sur le plan architectural, impose toutefois une contrainte directe sur la résolution d'entrée afin d'assurer la condition $N_S = N_T$ en sortie.

2. *Distillation à Résolution Adaptative (Adaptive-Resolution Distillation, ARD)*

Cette approche permet des résolutions d'entrée flexibles en redimensionnant la sortie de l'étudiant lorsque $N_S < N_T$ ou $N_S > N_T$ sans ajout de paramètre, afin que cette dernière soit alignée avec le professeur. Pour cela, la sortie de l'étudiant $v^S \in \mathbb{R}^{N_S \times D_S}$ est restructurée en cartes de caractéristiques $F^S_{\text{maps}} \in \mathbb{R}^{h_S \times w_S \times D_S}$. Ces dernières sont ensuite redimensionnées de taille $\mathbb{R}^{h_T \times w_T \times D_S}$, avec $h_T \times w_T = N_T$. Enfin, ces cartes sont aplaties et projetées dans l'espace de représentation du professeur D_T par une couche linéaire. Afin d'éviter des paramètres supplémentaires, une couche d'interpolation bilinéaire sans paramètres lorsque $N_S > N_T$ et une couche bicubique lorsque $N_S < N_T$ sont utilisées pour redimensionner les cartes de caractéristiques de l'élève. Cette approche aligne donc le nombre de patchs N_S en sortie de l'étudiant avec N_T, enlevant la contrainte sur la résolution d'entrée de la première approche.

Pour les deux approches de distillation, inspirées par Yang *et al.* (2024b), l'erreur quadratique moyenne (Mean Squared Error, MSE) est choisie comme fonction de coût :

$$MSE = \frac{1}{N_T \times D_T} \sum_{n}^{N_T} \sum_{d}^{D_T} (v_{n,d}^{S'} - v_{n,d}^{T})^2 \tag{3.5}$$

avec $v^{S'} \in \mathbb{R}^{N_T \times D_T}$ les vecteurs en sortie de l'étudiant, alignés avec le professeur. L'élève peut donc apprendre à imiter la position exacte de chaque patch dans l'espace de représentation du professeur.

La distillation implique donc de générer les embeddings du professeur v^T pour chaque image afin de calculer l'erreur MSE. Cependant, générer à chaque itération ces embeddings prend des ressources de calcul supplémentaires (empreinte sur la mémoire, VRAM). Afin de réduire l'empreinte VRAM de l'entraînement, les vecteurs d'embedding du professeur sont générés et sauvegardés dans une base de données avant l'entraînement, ce qui permet de ne pas charger le modèle lors de la supervision de l'étudiant. Une *hmap* est également générée avec comme clé, l'identifiant (id) de l'image dans le dataset originel et en valeur la ligne correspondante aux vecteurs d'embedding du professeur associés à cette image, ce qui permet de retrouver rapidement et facilement les embeddings v^T pour une image I.

La deuxième étape de cet entraînement est le finetuning bout-en-bout du modèle. En effet, même si l'encodeur visuel de l'étudiant a appris à imiter la représentation du professeur, ce dernier, du fait de sa composition différente, a pu apprendre des détails plus affinés ou différents du professeur, pouvant nécessiter un affinage supplémentaire avec le LLM décodeur. Ainsi, cette étape consiste à entraîner le modèle entièrement, afin d'aligner le nouvel encodeur visuel avec le décodeur. Pour cela, l'approche QLoRA par Dettmers, Pagnoni, Holtzman & Zettlemoyer (2023) a été adoptée afin de limiter les ressources nécessaires à cet entraînement. Les adaptateurs LoRA ont été initiés par Hu *et al.* (2022) et permettent d'affiner des modèles, avec un faible coût de

calcul. Pour cela, au lieu de mettre à jour les paramètres initiaux du modèle, une couche de nouveaux paramètres s'ajoutent au dessus de ces derniers, préservant les connaissance actuelles du modèle tel que :

$$h = Wx + W_{lora}x \tag{3.6}$$

avec $W \in \mathbb{R}^{d \times k}$ les paramètres pré-entraînés du modèle, gelés pour l'entraînement et $W_{lora} \in \mathbb{R}^{d \times k}$, les paramètres LoRA ajoutés pour le finetuning. Pour limiter le nombre de nouveaux paramètres, les adaptateurs LoRA sont composés de deux matrices à faibles rangs. On peut ainsi noter :

$$W_{lora} = BAx \tag{3.7}$$

avec $B \in \mathbb{R}^{d \times r}$ et $A \in \mathbb{R}^{r \times k}$ les décompositions de rang r des paramètres LoRA, avec $r << min(d, k)$. Ainsi, lors de l'apprentissage, seules les matrices A et B sont mises à jour, réduisant considérablement le nombre de paramètres à entraîner. En plus des adaptateurs LoRA ajoutés sur chaque couche de l'encodeur visuel, du projecteur multimodal et du décodeur, les paramètres pre-entraînés sont chargés sur 4 bit au lieu de 32 bit (QLoRA), permettant de réduire leur empreinte sur la VRAM.

Ainsi, en adoptant cette technique, le modèle peut être affiné de bout-en-bout tout en limitant la VRAM requise. Cette étape utilise la fonction d'entropie croisée comme fonction de coût, comparant la prédiction en sortie du décodeur $\hat{y}$ avec la réponse correcte y, présente dans l'ensemble de données tel que :

$$L_{CE} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{|A|} y_{t,i} \log\left(\frac{\exp(\hat{y}_{t,i})}{\sum_{j=1}^{|A|}\exp(\hat{y}_{t,j})}\right) \tag{3.8}$$

avec T la longueur de la réponse et $|A|$ la taille du vocabulaire du modèle de langue.

3.2.2 Évaluation et interprétation des connaissances de l'encodeur visuel

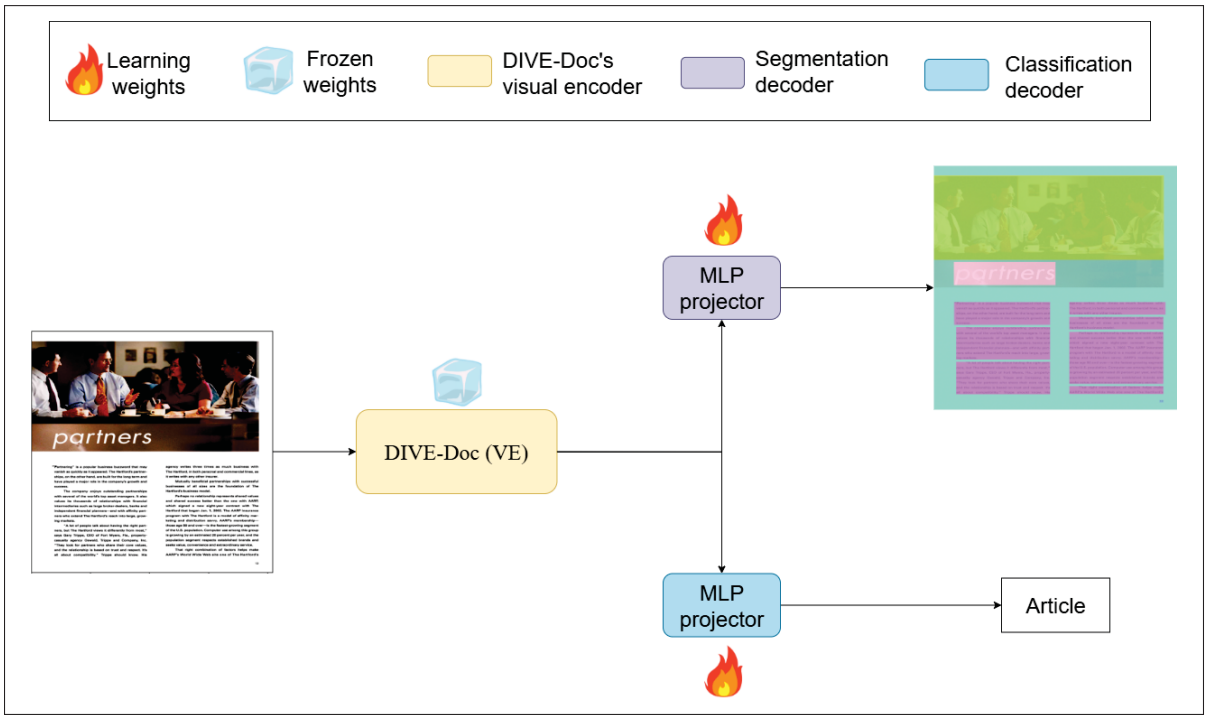


Figure 3.3 Stratégie d'évaluation de l'encoder visuel

Suite à l'évaluation du modèle sur la tâche de DocVQA, une question peut se poser : *Qu'est ce que l'encodeur visuel a réellement appris et quelles fonctions remplit-il dans la résolution de la tâche ?*. Comme énoncé en introduction et au long des chapitres, l'encodeur visuel permet d'extraire les caractéristiques du document afin que le LLM puisse ensuite les interpréter afin de répondre à la question. Cependant, les images de documents ont différents types d'éléments (texte, image, etc.) et des structures (layout) variées. La tâche de DocVQA nécessite à la fois d'extraire et de représenter les informations de différentes modalités, mais aussi d'avoir une compréhension de la structure du document (layout) à la fois générale (position spatiale des éléments) et sémantique (titre, description, tableau, etc.).

Dans le but de mieux étudier et comprendre le rôle de l'encodeur visuel dans le contexte de DocVQA, ce dernier a été évalué sur deux tâches de compréhension de documents. Premièrement, afin d'évaluer si l'encodeur visuel a une bonne représentation de la structure générale des

documents en étant capable de discriminer des structures différentes (lettres, articles, etc.), la tâche de classification de documents (Document Classification, DocCLS) a été choisie. Pour cela, un petit MLP est ajouté en sortie de l'encodeur visuel afin de réduire le nombre de patchs de N_T à 1, puis un deuxième MLP sert à classifier ce patch, passant de D_T à C_{CLS} avec C_{CLS} le nombre de classes possibles. La fonction de coût utilisée est l'entropie croisé, telle que :

$$L_{CE}^{cls} = -\sum_{i=1}^{C_{CLS}} y_i^{cls} \log\left(\frac{\exp(\hat{y}_i^{cls})}{\sum_{j=1}^{C_{CLS}} \exp(\hat{y}_j^{cls})}\right) \tag{3.9}$$

La deuxième évaluation a lieu sur la tâche d'analyse de la structure du document (Document Layout Analysis, DLA), consistant à classifier chaque patch ou pixel du document avec une classe de structure sémantique (titre, tableau, image, texte, etc.). Ainsi, ces classes peuvent être divisées en deux sous-groupes, les classes de modalités (image et texte) et les classes d'intra-modalité (tableau, formulaire, titre, etc.) qui sont toutes de la modalité de texte. Cette évaluation permet donc de déterminer si l'encodeur visuel est capable de représenter et de discriminer les modalités différentes, et d'avoir une compréhension sémantique du layout dans sa représentation.
Pour résoudre cette tâche, un petit MLP est ajouté à la sortie de l'encodeur afin de classifier chaque patch, projetant ainsi $v^{S'} \in \mathbb{R}^{N_T \times D_T}$ à $\mathbb{R}^{N_T \times C_{DLA}}$, avec C_{DLA} le nombre de classes. La fonction de coût utilisée est l'entropie croisé, tel que :

$$L_{CE}^{dla} = -\frac{1}{N_T}\sum_{n=1}^{N_T}\sum_{i=1}^{C_{DLA}} y_{n,i}^{dla} \log\left(\frac{\exp(\hat{y}_{n,i}^{dla})}{\sum_{j=1}^{C_{DLA}} \exp(\hat{y}_{n,j}^{dla})}\right) \tag{3.10}$$

Afin d'évaluer correctement ce que l'encodeur visuel a appris pour la tâche de DocVQA, les connaissances de ce dernier ne doivent pas être modifiées lors de l'entraînement des MLP pour les deux sous-tâches. Ainsi, les paramètres de l'encodeur visuel sont gelés comme illustré sur la figure 3.3, afin de ne pas modifier ce qu'il a appris pour la tâche de DocVQA, et de réutiliser ses connaissances sur les tâches de classification et d'analyse de structure de documents.

3.3 Notes finales et ouverture

La méthodologie de ce premier objectif peut ainsi être décomposée en deux parties. Premièrement, la réduction et l'évaluation de l'encodeur visuel d'un LVLM sur la tâche de DocVQA. Cette étape permet à la fois de réduire la taille de l'encodeur visuel d'un modèle de fondation et d'évaluer l'impact de différentes résolutions en entrée sur la performance de cette tâche. De plus, l'ajout de deux sous tâches de compréhension de documents, réutilisant l'encodeur visuel réduit de la première étape sans modifier ses poids, permet d'évaluer et d'interpréter les connaissances acquises par ce dernier pour la tâche de DocVQA.

Cependant, cette architecture ne permet pas d'enrichir le module visuel avec une position spatiale précise, qui est un des gaps de la littérature (voir chapitre 1) et pourrait avoir un impact non négligeable sur la représentation des structures de documents.

CHAPITRE 4

AU-DELÀ DU CONTENU : ENRICHIR LES REPRÉSENTATIONS VISUELLES PAR LA GÉOMÉTRIE SPATIALE DES DOCUMENTS

L'opération d'attention est le cœur des ViTs, permettant de mettre en avant les caractéristiques pertinentes des patchs, ainsi qu'un encodage global entre tous les patchs. Cependant, cette opération est équivariante par permutation, telle que si l'ordre des patchs change, leur embedding respectif sera le même en sortie. Dans les tâches de compréhension de document comme DocVQA, la réponse est souvent dispatchée sur plusieurs patchs, ainsi induire la position spatiale dans la représentation est essentielle pour que le modèle puisse avoir une compréhension de la structure du document (ordre des patchs) et retrouver la réponse à la question. Comme vu dans l'état de l'art, les encodeurs visuels dans la tâche de DocVQA se basent sur des papiers fondateurs, étudiés sur des images naturelles. Cependant, l'encodage positionnel qu'ils utilisent ne semble pas être adéquat pour représenter correctement un document (voir section 1). Ainsi, ce chapitre est dédié au deuxième objectif de ce mémoire, qui consiste à intégrer un encodage positionnel plus précis dans le modèle de vision afin d'améliorer la compréhension de la structure des documents dans ce dernier.

4.1 L'encodage des positions spatiales

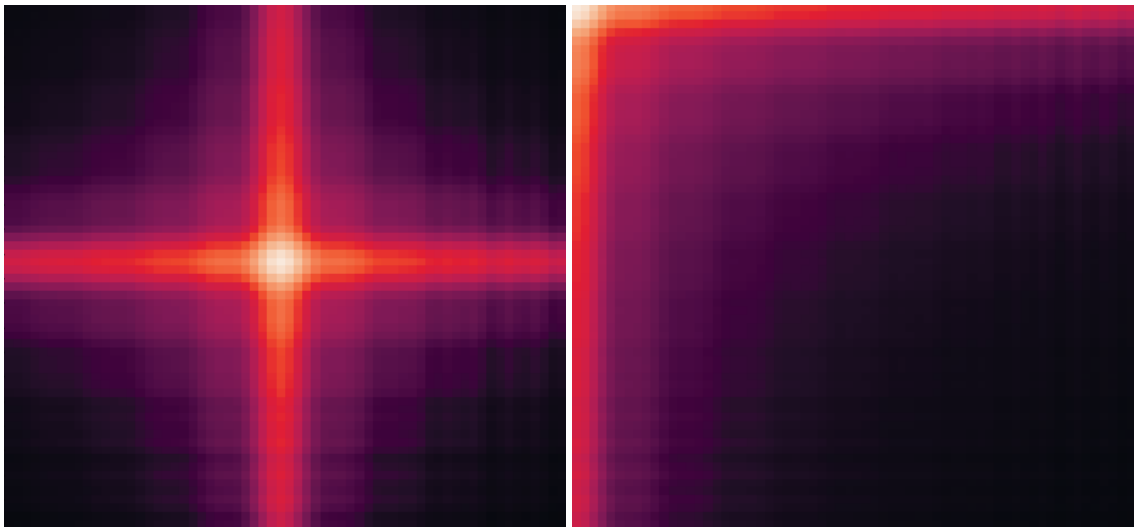
L'encodage de position est un aspect fondamental de l'architecture des Transformers depuis leur début. Ces dernières étant introduites sur des tâches de langage naturel (Vaswani *et al.*, 2017), les premières séquences étaient d'une seule dimension telle que des phrases. Ainsi, l'encodage de position prenait uniquement l'indice du token (mot, sous-mot, lettre) dans la séquence afin de l'encoder et l'ajouter à la représentation. Pour cela, les premières méthodes consistaient en un enchaînement de sinus et cosinus sur les différentes dimensions de l'embedding, prenant en entrée l'indice de la position du texte dans la séquence tel que :

$$\begin{aligned} PE_{pos,2i} &= sin(pos/1000^{\frac{2i}{d_h}}) \\ PE_{pos,2i+1} &= cos(pos/1000^{\frac{2i}{d_h}}) \end{aligned} \tag{4.1}$$

avec pos la position du token dans la séquence, d_h la dimension de l'embedding et i l'indice de la dimension courante de l'embedding. Cette méthode permet de toujours avoir un encodage positionnel différent entre chaque token. Les cosinus et sinus étant des fonctions périodiques, encoder les positions sur des fréquences différentes pour chaque indice d'embedding permet d'avoir une position unique par token dans la représentation. Bien que cet encodage de position ait fait ses preuves pour la compréhension de textes, les images sont des données à deux dimensions, ainsi chaque patch n'a pas une seule coordonnée, mais deux, à savoir l'indice sur la hauteur et sur la largeur (h, w). Cette méthode est donc peu efficace pour représenter leur position.
L'encodage spatial dans les ViTs a donc été un aspect fondamental dans le développement de leur architecture. Initialement entraînés sur des images naturelles, ces modèles ont conduit à l'étude de diverses façons d'encoder la position spatiale des patchs. Une première méthode consiste à adapter l'encodage de position des Transformers (voir équation 4.1) pour utiliser une position à deux coordonnées. Ainsi, la position h est encodée sur la première moitié de l'embedding et la position w sur l'autre partie, tel que

$$\begin{aligned} PE_{h,2i} &= sin(h/1000^{\frac{2i}{d_h/2}}); \\ PE_{h,2i+1} &= cos(h/1000^{\frac{2i}{d_h/2}}); \\ PE_{w,2i} &= sin(w/1000^{\frac{2i}{d_h/2}}); \\ PE_{w,2i+1} &= cos(w/1000^{\frac{2i}{d_h/2}}); \\ PE &= [PE_h, PE_w] \end{aligned} \tag{4.2}$$

avec h la position spatiale du patch sur la hauteur, w la position spatiale du patch sur la longueur et $PE_h, PE_w \in \mathbb{R}^{1\times d_h/2}$ les embeddings de position respectifs à la hauteur et à la longueur. La cohérence spatiale de ces embeddings de positions peut être évaluée en faisant le produit scalaire d'une position donnée avec les autres positions des patchs sur l'image. Si les positions spatialement proches ont des scores de similarité élevés, cela veut dire que les représentations des positions PE sont spatialement correctes. La figure 4.1 montre deux exemples de scores de similarité utilisant cette méthode. Comme montré par cette figure, les positions similairement proches dans l'espace de représentation sont les positions orthogonales et non les positions spatialement proches. Cela peut s'expliquer par le fait que les coordonnées h et w sont encodées



a) Similarités avec la position centrale b) Similarités avec la position (0,0)

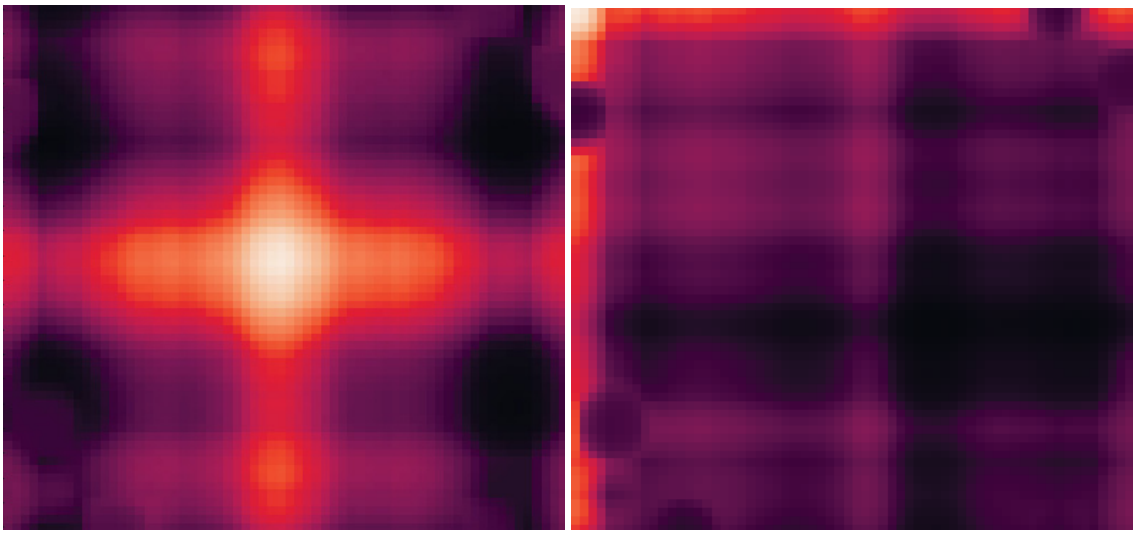
Figure 4.1 Similarité entre un vecteur de position choisi et les autres, utilisant l'encodage de position absolue 2d

séparément et non de manière uniforme. Ainsi cette méthode ne permet pas de representer correctement la position des patchs dans l'espace de représentation.

Une autre approche consiste à superviser un vecteur de position par indice unidimensionnel dans la séquence de patchs N (voir section 2.2.1). Ces vecteurs étant directement supervisés lors de l'entraînement, ils n'ont pas besoin de prendre les deux coordonnées spatiales (h, w). Les similarités de positions ont été projetées sur la figure 4.2.

Ainsi, bien que légèrement plus performante que la méthode précédente (Dosovitskiy *et al.*, 2020), cette approche présente le même problème et tend à rapprocher les positions orthogonales sans uniformité dans les autres directions. Ces encodages de positions sont dit absolu (Absolute Positional Encoding, APE), car ils encodent la position brute des patchs dans la séquence.

Une approche différente consiste à encoder la position relative (Relative Positional Encoding, RPE) entre les patchs. Un exemple de cette méthode est présent dans l'architecture Swin Transformer, introduisant un biais spatial dans les fenêtres d'attention (voir section 2.3.2). Ce



a) Similarités avec la position centrale b) Similarités avec la position (0,0)

Figure 4.2 Similarités entre un vecteur de position choisi et les autres, issues du modèle PaliGEMMA

biais est différent pour chaque patch au sein d'une même fenêtre mais se répète entre chaque fenêtre, cet encodage positionnel est donc local et non global. Ainsi, bien que ces méthodes se soient montrées efficace en achevant de bons résultats, ces dernières semblent mal adaptées pour représenter correctement les positions spatiales des patchs, particulièrement dans le contexte d'image de documents.

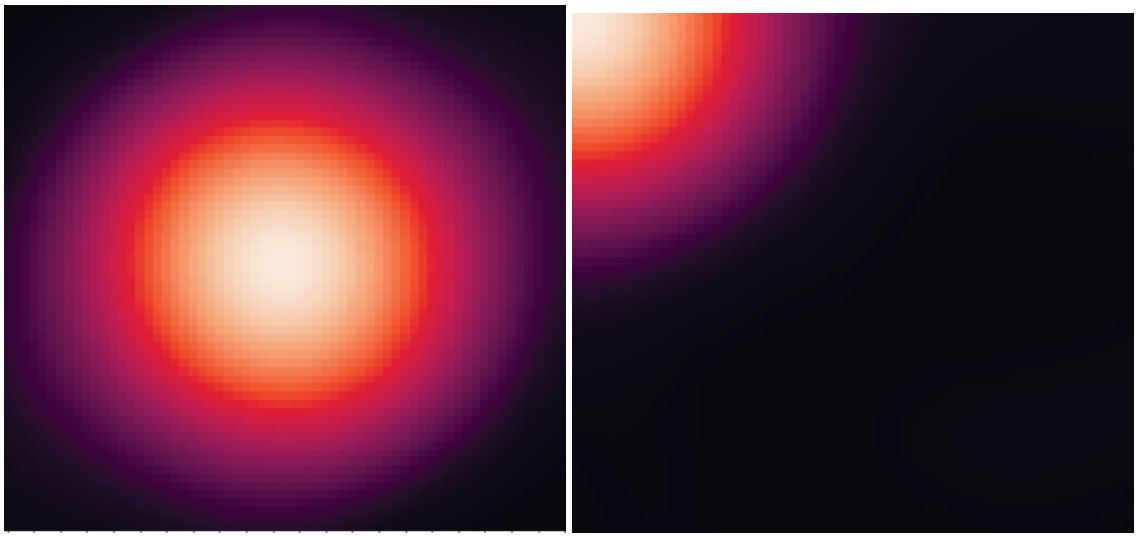
Une position spatiale adaptée serait telle qu'illustrée sur la figure 4.3. Sur cette dernière, les similarités de positions se propagent dans toutes les directions de manière homogène et sont spatialement cohérentes. Li, Si, Li, Hsieh & Bengio (2021) ont proposé une autre méthode se basant sur les caractéristiques de Fourier (Fourier Features). En apprenant une projection linéaire $W_\tau \in \mathbb{R}^{M \times \frac{D}{2}}$, avec $M = 2$ dans le cas d'une image, ils créent un espace de représentation uniforme pour les positions de patchs (h, w). Les caractéristiques de Fourier sont ensuite extraites de cette représentation suivant l'équation :

$$\begin{aligned} pe &= [h, w].W_\tau \\ ffpe &= \frac{1}{\sqrt{D}}[cos(pe), sin(pe)] \end{aligned} \tag{4.3}$$

avec $pe \in \mathbb{R}^{\frac{D}{2}}$ la représentation de la position et $ffpe \in \mathbb{R}^{D}$, les caractéristiques de Fourier extraites de la représentation pe. Cette technique permet d'avoir une propagation de similarité dans toutes les directions et non uniquement dans les directions orthogonales. Les auteurs ont démontré que le noyau gaussien tel que montré sur la figure 4.3 pouvait être obtenu en initialisant les poids de la projection linéaire W_τ par une distribution normale. Cependant, ils laissent cette projection être supervisée lors de l'entraînement afin que le modèle puisse apprendre une représentation pertinente pour la tâche. Afin d'ajouter plus de capacité d'apprentissage à la représentation, les auteurs ont ajouté un petit MLP de deux couches, permettant d'ajouter de la non linéarité afin d'enrichir et de rendre plus flexible la représentation :

$$ffpe' = GeLU(W^1_{ffpe} ffpe + B^1_{ffpe})W^2_{ffpe} + B^2_{ffpe} \tag{4.4}$$

où $W^1_{ffpe} \in \mathbb{R}^{D \times H}, B^1_{ffpe} \in \mathbb{R}^{H}, W^2_{ffpe} \in \mathbb{R}^{H \times D}$ et $B^2_{ffpe} \in \mathbb{R}^{D}$, avec H la dimension interne au MLP, habituellement $D < H$ afin d'apprendre des relations complexes. Les auteurs ont démontré



a) Similarités avec la position centrale b) Similarités avec la position (0,0)

Figure 4.3 Similarités entre un vecteur de position choisi et les autres, obtenues par l'extraction des caractéristiques de Fourier

que cette méthode permettait de meilleurs résultats sur les tâches de vision. Cependant, cette

méthode reste peu exploitée dans l'état de l'art et, au meilleur de notre connaissance, n'a pas été étudiée pour des tâches de représentation de documents. Les propriétés de cette approche permettant un encodage spatial des positions plus précis que les méthodes précédentes de l'état de l'art de DocVQA, on peut supposer qu'enrichir la représentation des modèles de vision actuels avec cette approche pourrait améliorer leur représentation de la structure des documents.

4.2 Méthodologie et Architecture

Les encodeurs visuels de la tâche de DocVQA se basent sur les papiers de fondation et intègrent l'encodage de position soit au début du modèle lors de la division de l'image en patch dans le cas d'un encodage de type APE (voir section 2.2.1) ou directement dans le mécanisme d'attention dans le cas RPE (voir section 2.3.2). Ajouter la position uniquement au début entraîne inexorablement une dilution de cette information dans les couches plus profondes. Cependant, la position est importante même en sortie de l'encodeur, afin que le modèle de langue puisse avoir un ordre de lecture correct des patchs afin de retrouver de manière efficace la réponse à la question posée. Ainsi, dans le cas de DocVQA, la partie la plus propice pour ajouter la position semblerait être la sortie de l'encodeur visuel. Cependant, afin que l'encodeur visuel puisse lui aussi apprendre à représenter la structure sémantique et spatiale du document, intégrer la position au début de son architecture ne doit pas non plus être négligée.
Ainsi, pour assurer une évaluation optimale de l'intégration de cette position dans l'encodeur visuel, la position d'insertion de cette dernière sera testé au début, à la sortie de chaque bloques, à la sortie du modèle et enfin à chacune des positions simultanément (voir figure 4.4).

Le module de position FFpos (Fourier Features position) suit donc les équations 4.3 et 4.4, prenant en entrée les paires de coordonnées (h, w) de chaque patch et retournant leur représentation $ffpe' \in \mathbb{R}^{N \times D}$. Ce dernier est ajouté à l'ensemble des vecteurs de patchs $z_l \in \mathbb{R}^{N \times D}$ tel que

$$z_l^{ffpe} = z_l + ffpe' \tag{4.5}$$

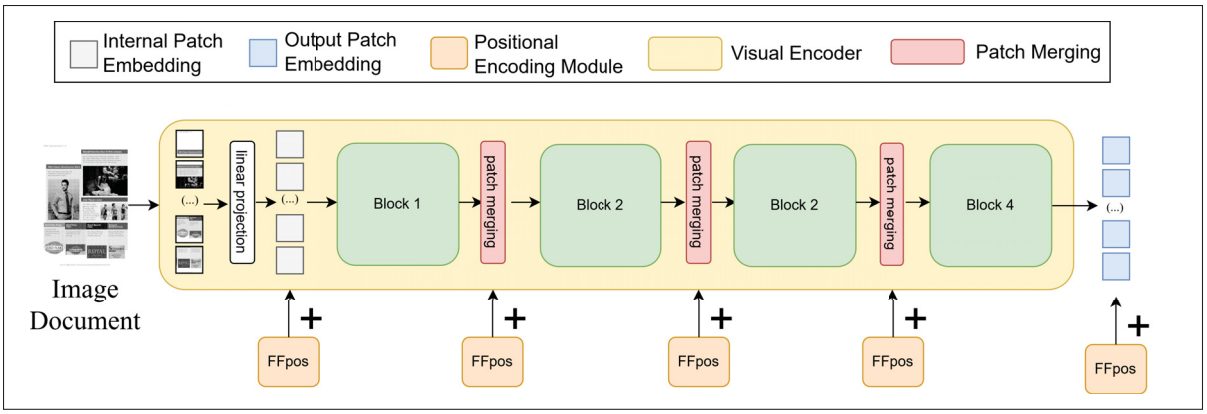


Figure 4.4 Emplacement des insertions des encodages de positions dans l'architecture

avec $z_l^{ffpe} \in \mathbb{R}^{N \times D}$ l'ensemble des vecteurs de patchs, enrichi par la position des caractéristiques de Fourier. Dans le cas où l'encodeur visuel est un Swin Transformer, l'ajout de la position se fait à la sortie de la couche "patch merging" (voir section 2.3).

Comme ce module a pour but d'enrichir les représentations des modèles de l'état de l'art, il est important de l'ajouter en affinant le modèle sans qu'il "oublie" sa connaissance actuelle de la tâche (caractéristiques à extraire et représenter). Pour cela, une stratégie d'entraînement divisée en 3 étapes est choisie (voir figure 4.5).

La première étape consiste à entraîner les modules FFpos, en gelant tous les autres poids du modèle afin d'intégrer la position sans endommager la connaissance actuelle du modèle. La deuxième étape consiste ensuite à entraîner entièrement l'encodeur visuel ainsi que la projection multimodale. Cela permet de laisser le module de vision apprendre de nouvelles caractéristiques et d'enrichir sa représentation du document grâce à l'ajout des modules de positions. Enfin, la dernière étape consiste à entraîner tous les poids du modèle de bout-en-bout, afin d'aligner le modèle de langue avec la nouvelle représentation du document, enrichie par la position. Afin de limiter les ressources de calcul nécessaires à cette étape, l'entraînement de bout-en-bout est réalisé avec la méthode QLoRA (voir section 3.2). Afin de permettre au modèle plus de flexibilité dans l'apprentissage des positions, chaque FFpos module possède ses propres poids

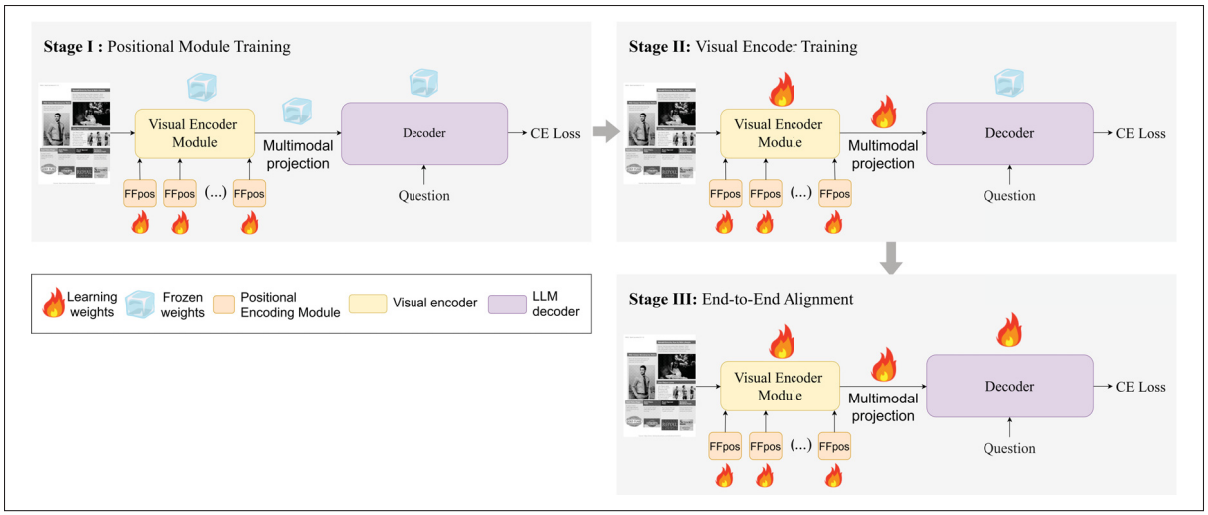


Figure 4.5 Entraînement des modèles enrichis par l'encodage positionnel proposé

et est indépendant des autres modules. Pour chaque étape, l'erreur est calculée en utilisant la fonction d'entropie croisée (voir équation 3.8).

Chaque position d'insertion étudiée est aussi évaluée sur les tâches de classification de document et d'analyse de structure sémantique de document suivant la méthodologie de la section 3.2.2.

CHAPITRE 5

UNE OUVERTURE SUR LE MULTI-PAGE : ADAPTER UN MODÈLE END-TO-END AUX DOCUMENTS COMPOSÉS DE PLUSIEURS PAGES

Ce mémoire a pour objectif de développer un encodeur visuel capable de représenter les images de documents scannés dans un espace de représentation multimodal afin de résoudre la tâche de DocVQA. Pour cela, deux sous objectifs sont réalisés, premièrement réduire la taille d'un encodeur visuel de fondation afin de réduire la complexité de calcul sans diminuer la qualité de la représentation afin de garder des performances compétitives avec les LVLMs (chapitre 3) ; deuxièmement intégrer un module d'encodage spatial précis afin d'améliorer la représentation structurel du document et de faciliter les ordres de lecture du modèle de langue (chapitre 4). Ainsi, les expériences ont été réalisées sur des documents d'une seule page afin de limiter le coût de calcul des entraînements. Cependant, beaucoup de scénarios impliquent des documents de plusieurs pages. Par ailleurs, traiter plusieurs pages de documents augmente considérablement le nombre de patchs à analyser, ce qui entraîne un coût de calcul bien supérieur (VRAM). De plus, cela dilue l'information à retrouver, ce qui peut complexifier la recherche de cette dernière par le modèle de langue. La tâche de réponse à des questions visuelles sur des documents multi-page (MP-DocVQA) entraîne donc de nouveaux défis. Ainsi, basée sur les résultats des deux sous-objectifs de ce mémoire, une exploration du multi-page est proposée afin d'ouvrir de nouvelles directions et de potentiels travaux futurs.

5.1 MP-DocVQA : une tâche récente et peu étudiée

La tâche de réponse à des questions visuelles sur des documents multi-page (MP-DocVQA) est apparue avec la sortie du dataset du même nom par Tito, Karatzas & Valveny (2023). Composé des mêmes documents que le dataset DocVQA, ce dernier propose des questions sur des documents allant jusqu'à 20 pages. Les méthodes évaluées sur ces datasets sont soumises et affichées publiquement sur le site internet Robust Reading Competition par Mathew, Tito, Karatzas, Manmatha & Jawahar (2020b).

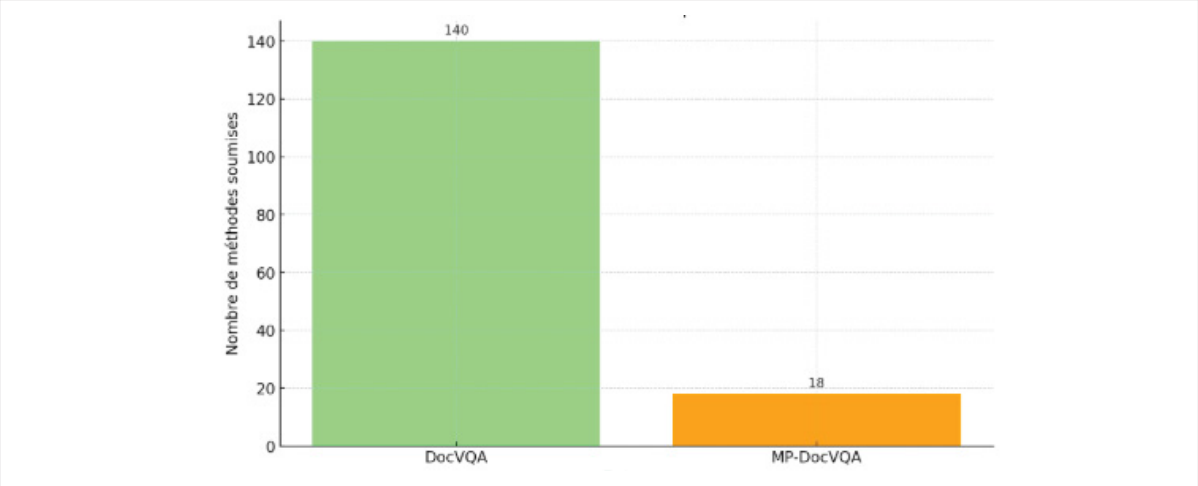


Figure 5.1 Nombre d'approches soumises par ensemble de données pour DocVQA et MP-DocVQA

La figure 5.1 montre le nombre d'approches publiquement évaluées sur ces derniers. Comme affiché sur cette dernière, là où le nombre d'approches soumises pour la base DocVQA est de 140, celles pour MP-DocVQA sont seulement au nombre de 19, ce qui montre l'aspect encore peu étudié de cette tâche.

Tableau 5.1 État de l'art sur MP-DocVQA

Méthode	# Param(B)	OCR	Tiling	Fusion Tot	Acc (%) ↑	ANLS (%)↑
Toutes les pages dans le décodeur						
Gram, Blau *et al.* (2024)	0.859	X		X	19.98	80.32
DocOwl2, Hu *et al.* (2024)	8		X		50.78	69.42
HiVT5, Tito *et al.* (2023)	0.316	X		X	79.63	62.01
Longformer, Tito *et al.* (2023)	0148	X		X	71.17	52.87
BigBird, Tito *et al.* (2023)	0.131	X		X	67.54	49.29
LayoutLMv3, Tito *et al.* (2023)	0.125	X		X	51.94	45.38
Sélecteur de réponse						
ScreenAI, Baechler *et al.* (2024)	5	X			77.88	77.1
ScreenAI, Baechler *et al.* (2024)	5				?	72.9
Sélecteur de page						
Pix2Struct, Kang *et al.* (2024)	0.273			X	81.55	61.99
Sélecteur de page + top-k pages dans le décodeur						
M3DocRAG, Cho, Mahata, Irsoy, He & Bansal (2024)	10B				81.05	84.44
FRAG-LLaVA-OV, Huang, Radhakrishnan, Yu & Kautz (2025)	7B				?	79.1
FRAG-InternVL2, Huang *et al.* (2025)	8B				?	77.8

Le tableau 5.1 regroupe différentes solutions de l'état de l'art de MP-DocVQA. Les approches traitant le multi-page peuvent être divisées en quatre familles. Premièrement, les méthodes donnant toutes les pages encodées ainsi que la question au décodeur (Tito *et al.*, 2023; Blau *et al.*, 2024; Hu *et al.*, 2024). Cette approche permet de rechercher la réponse parmi plusieurs pages sans changer l'architecture, mais augmente le coût de calcul dans le modèle de langue. De plus, les modèles de petite taille ont des résultats faibles (ANLS, voir équation 6.1) du fait de la dilution de l'information dans le contexte, comme LayoutLMv3 (45.38%), BigBird (49.29%) ou encore Longformer (52.87%). Certaines méthodes de cette catégorie obtiennent tout de même de meilleurs résultats comme DocOwl2 par Hu *et al.* (2024) qui utilise un compresseur afin de réduire le nombre de patchs des documents avant de les envoyer au décodeur, ce qui permet de réduire le coût de calcul tout en atteignant un score de 69.42% d'ANLS. Cependant, ce modèle nécessite toujours une capacité de calcul conséquente du fait de sa taille (8B de paramètres). D'autre part, Gram par Blau *et al.* (2024) utilise un encodeur multimodal avec de l'OCR et une fusion en amont, ce qui permet d'avoir un modèle de taille raisonnable (859M de paramètres) atteignant de meilleurs résultats (80.32%). Cependant, la fusion en amont empêche la réutilisation des embeddings des documents, ce qui nécessite d'encoder à nouveau chaque page du document pour chaque nouvelle question.

Afin d'éviter un coût de calcul supplémentaire trop important, une autre approche consiste à transformer le problème multi-page en mono-page (Baechler *et al.*, 2024). En donnant au modèle chaque page avec la question de manière indépendante afin de générer une réponse par page, sélectionnant ensuite celle ayant le plus haut score de probabilité en sortie du modèle, cette approche permet de réutiliser des modèles initialement entraînés sur des bases de données ayant une page par document. Bien que cela permette de réduire le coût de calcul, cette approche nécessite d'utiliser le décodeur pour chaque page, soit de manière séquentielle (une page après l'autre), ce qui augmente le temps pour obtenir la réponse correcte de manière linéaire avec le nombre de pages, soit de manière parallèle, ce qui augmente l'empreinte sur la VRAM.

Pour remédier à cela, une autre approche consiste à utiliser un filtre en sortie de l'encodeur, prenant en entrée chaque page du document avec la question générant ainsi un score de probabilité par page, désignant si la page contient la réponse ou non. La page ayant le plus haut score est

ensuite envoyée au décodeur, ce qui permet d'éviter d'utiliser le modèle de langue pour chaque page. Kang *et al.* (2024) utilise cette approche avec un encodeur multimodal. Cependant, les auteurs se basent sur la fusion en amont, ce qui ne permet pas de réutiliser les embeddings de documents pour chaque nouvelle question.

Enfin une dernière approche consiste à utiliser un sélecteur de page, puis à envoyer les $top - k$ pages avec le plus haut score dans le décodeur. Le modèle M3DocRAG par Cho *et al.* (2024) combine à la fois l'approche de sélection de page après l'encodeur visuel, et l'envoi de plusieurs pages au décodeur prenant les $top - k$ pages ayant obtenu le plus haut score par le sélecteur. Contrairement à la méthode proposée par Kang *et al.* (2024), ces derniers utilisent un LVLM comme encodeur, ce qui leur permet d'utiliser une simple similarité entre les embeddings et la question pour calculer les scores de probabilité. Cependant, cette approche nécessite d'avoir un encodeur de grande taille puis un autre décodeur également de grande taille. Ici les auteurs ont utilisé le modèle ColPali (3B) par Faysse *et al.* (2024) pour encoder la question et les pages du document, puis le modèle Qwen2-VL pour générer la réponse (7B) par Wang *et al.* (2024). Les modèles ColPali et Qwen2-VL étant différents, les embeddings générés par ColPali ne sont pas réutilisables, ainsi Qwen2-VL prend en entrée les images originales des $top - k$ pages, et les encode de nouveau avant de générer la réponse. Ainsi bien que cette méthode achève de bons résultats (84.44% ANLS), le coût de calcul lié à son nombre de paramètres (10B) ainsi que sa nécessité d'encoder deux fois le document le rendent plus compliqués pour des déploiements industriels. La méthode FRAG proposée par Huang *et al.* (2025) utilise deux fois le même modèle de langue pour calculer les scores par page en utilisant un prompt adapté et générer la réponse avec les $top - k$ pages sélectionnées ainsi que la question. Cependant, cette méthode nécessite d'utiliser deux fois le modèle de langue, ce qui augmente considérablement la latence comme souligné dans l'étude d'ablation (voir section 6.4.3).

Ainsi, la tâche MP-DocVQA est un domaine encore peu étudié où les approches proposent des compromis entre embeddings réutilisables, coût de calcul et performance.

5.2 Méthodologie et Architecture

Cette section présente la méthodologie utilisée afin d'adapter le modèle construit lors du premier et du deuxième sous-objectifs pour la tâche de MP-DocVQA. Afin de répondre aux contraintes utilisateurs, la méthode doit pouvoir réutiliser les embeddings de document pour potentiellement construire des bases de données interrogeables (voir figure 0.2c) et éviter d'encoder plusieurs fois la même page pour chaque question. De plus, afin de pouvoir être utilisé dans des infrastructures limitées (GPU), il est préférable que l'adaptation du modèle n'entraîne pas une augmentation trop importante du nombre de paramètres. Enfin, pour éviter l'augmentation du coût de calcul, le modèle doit limiter le nombre de jetons dans le décodeur afin de ne pas augmenter le coût de son attention qui est quadratique.

Ainsi, l'approche choisie est celle de la sélection de page présentée dans la section précédente. Rajoutant seulement un module entre l'encodeur visuel et le modèle de langue, cette approche permet de sélectionner les $top - k$ pages à envoyer au modèle de langue.

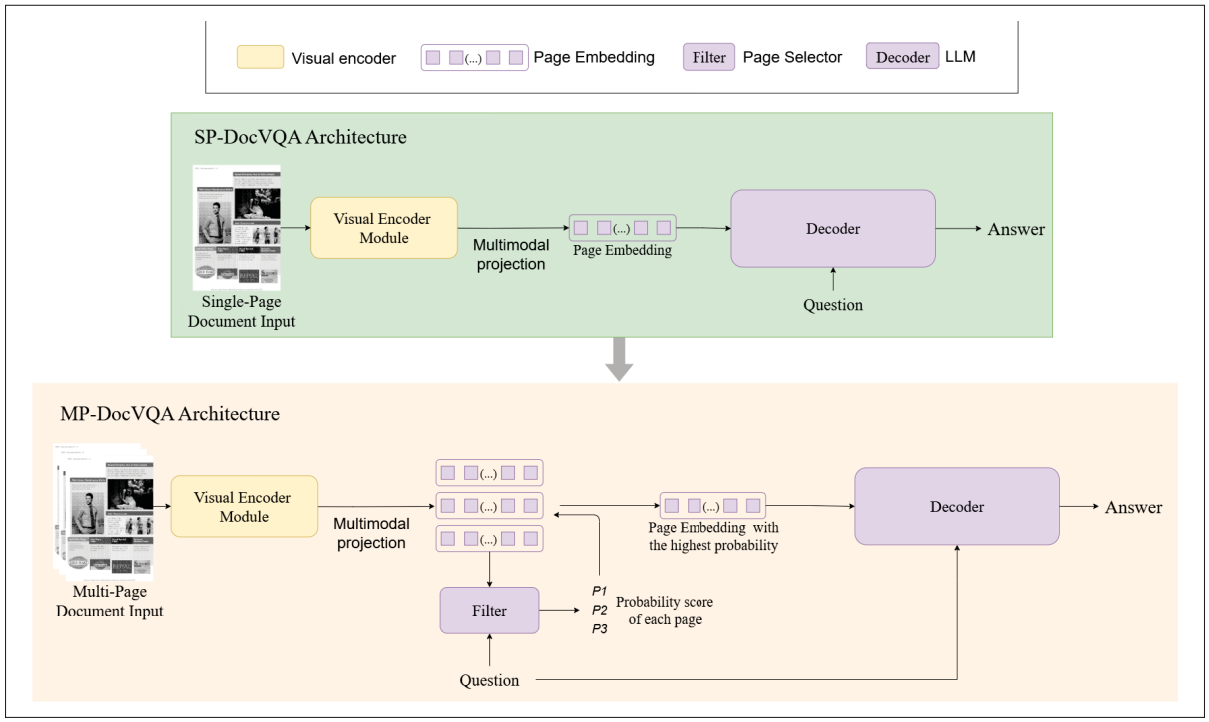


Figure 5.2 Schéma du modèle DIVE-Doc adapté pour MP-DocVQA

La figure 5.2 montre l'architecture proposée. Comme illustré sur cette dernière, un module filtre est inséré entre l'encodeur visuel et le modèle de langue. Le filtre prend en entrée la question encodée, c'est-à-dire la question divisée en N_t jetons (tokens), projetés par une couche linéaire dans le même espace de représentation que le modèle de langue. On note t la question encodée telle que $t \in \mathbb{R}^{N_t \times D_M}$ avec D_M la dimension de l'espace de représentation multimodal.

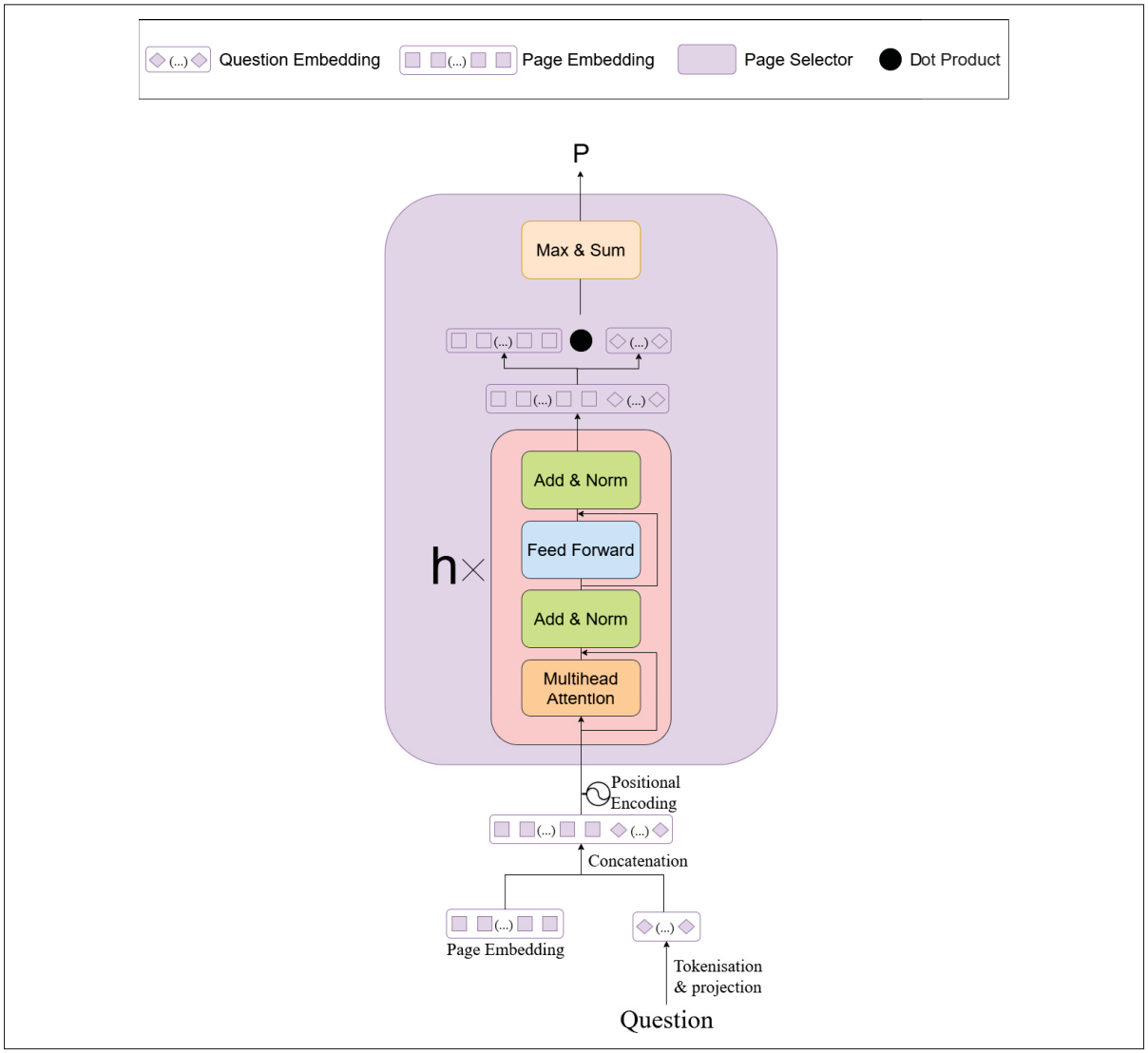


Figure 5.3 Schéma du sélecteur de page (filtre)

La figure 5.3 illustre l'architecture du filtre. Issu du modèle Gemma par Mesnard *et al.* (2024), il est composé de h couches de Transformer qui encodent la question t avec les patchs d'une page $v \in \mathbb{R}^{N_v \times D_M}$ où N_v est le nombre de patchs et D_M la dimension de l'espace de représentation du modèle de langue Gemma. Ces couches permettent ainsi de mettre en avant dans la représentation les informations relatives à la question contenue dans les patchs. Les couches sont composées d'un bloc d'attention afin d'encoder chaque élément en fonction des autres et de leurs pertinences (voir section 2.2.2.1), d'un projecteur multicouche afin d'ajouter de la non-linéarité et d'avantage de capacité de représentation (voir section 2.2.3). En sortie de ces blocs, une connexion résiduelle ainsi qu'une normalisation sont ajoutées afin d'enrichir les représentations précédentes par les nouvelles et d'éviter que les valeurs ne deviennent trop élevées. En sortie de la dernière couche, basé sur le papier de Cho *et al.* (2024), un score de probabilité est attribué à la page, calculé tel que :

$$s(t, v) = \sum_{i=1}^{N_t} max_{j \in [N_v]} t_i . v_j \tag{5.1}$$

avec t_i un jeton de la question et v_j un patch de la page. Ainsi, le score est calculé en faisant le produit scalaire entre chaque jeton et chaque patch. Pour chaque jeton, le plus haut score avec les patchs est conservé, puis chacun de ces scores est additionné, donnant le score final de la page. Ce processus est ensuite répété pour chaque page du document.

Le modèle de langue étant entraîné à prendre uniquement une page en entrée, seulement les embeddings v de la page ayant obtenu le plus haut score sont ensuite envoyés avec la question pour générer la réponse, ainsi le modèle prend en entrée la $top - 1$ page. Cela permet de limiter l'empreinte sur la VRAM mais empêche la résolution de questions ayant une réponse dispatchée sur plusieurs pages (cas appelé multi-hop).

Afin de ne pas augmenter l'empreinte du modèle sur la mémoire, les poids du filtre sont partagés avec ceux du modèle de langue. Pour cela, le filtre réutilise les h premières couches du décodeur, ne nécessitant donc pas d'entraînement ni d'allocation mémoire supplémentaire. Le modèle de langue étant composé de 26 couches, le filtre réutilise ses huit premières couches afin de limiter la latence (voir l'étude d'ablation section 6.4.3).

Le modèle est évalué en regardant à la fois la qualité des réponses et également si le score maximal retourné par le filtre correspond bien à la page contenant la réponse. Ainsi, cette approche propose un système bout-en-bout avec des embeddings réutilisables sans augmenter l'empreinte mémoire du modèle.

CHAPITRE 6

EXPÉRIMENTATIONS, RÉSULTATS ET DISCUSSIONS

Ce chapitre décrit les résultats obtenus pour les deux objectifs de ce mémoire, suivant les chapitres 3 et 4 ainsi que ceux de l'extension sur le multipage (chapitre 5). La section suivante contient les détails expérimentaux (base de données, mesure de performances, etc.) afin d'assurer la reproductibilité des résultats. Les trois dernières sections présentent respectivement le détail des résultats des expérimentations, ainsi que leurs analyses.

6.1 Base de données et métriques d'évaluation

6.1.1 Base de données

Pour les expérimentations principales sur la tâche de DocVQA (voir les sections 3.2 et 4.2), la base de données utilisée est celle du même nom (DocVQA) par Mathew *et al.* (2021). Elle contient près de 12.767 images de documents issues de l'industrie, de différentes structures (lettre, article, formulaire, etc.), et ayant des entités de modalités variées (photo, texte, graphique, etc.). Elle contient près de 99.000 questions, permettant d'aligner le modèle de langue avec l'encodeur visuel. Concernant les évaluations approfondies sur l'encodeur visuel (voir section 3.2.2), la base de données utilisée pour la tâche de classification de documents (DocCLS) se nomme RVL-CDIP (Harley *et al.*, 2015), contenant près de 400.000 images de documents, réparties sur 16 classes (lettre, article, etc.). Pour la tâche d'analyse de structure de documents (DLA), l'ensemble de données DocLayNet par Pfitzmann *et al.* (2022) a été utilisé. Ce dernier contient 80.863 images de documents, de différentes structures et distribution d'entités sémantiques (titre, tableau, photo, etc.). Enfin, pour l'ouverture sur le multi-page, la base de données choisie est MP-DocVQA par Tito *et al.* (2023). Cette dernière possède près de 46.000 questions posées sur 6.000 documents industriels, chacun pouvant contenir jusqu'à 20 pages, faisant un total de 48.000 images de pages de documents.

6.1.2 Métriques d'évaluation

Chaque tâche entraînant une sortie différente des autres, elles ont ainsi des métriques spécifiques. La tâche de DocVQA consiste à générer une réponse sous forme de texte numérique. Une métrique standard pour évaluer cette tâche est la Similarité de Levenshtein Normalisée Moyenne (Average Normalized Levenshtein Similarity, ANLS). Introduite par Biten *et al.* (2019), cette métrique mesure la similarité entre la réponse générée par le modèle et la réponse de référence (ground truth), en s'appuyant sur la distance de Levenshtein, normalisée pour tenir compte de la longueur des chaînes comparées. Son équation peut s'écrire :

$$\text{ANLS} = \frac{1}{N} \sum_{i=1}^{N} \text{sim}(p_i, g_i) \tag{6.1}$$

$$\text{sim}(p, g) = \begin{cases} 1 - \dfrac{\text{lev}(p, g)}{\max(|p|, |g|)} & \text{si } \dfrac{\text{lev}(p, g)}{\max(|p|, |g|)} < \tau \\ 0 & \text{sinon} \end{cases} \tag{6.2}$$

avec p la réponse prédite, g la réponse dans la base de données et lev la distance de Lcvenshtcin (1966), consistant à attribuer un nombre d'opérations minimales à effectuer sur p pour que $p = g$.

τ est un seuil qui permet de mettre à zéro la similarité si p et g diffèrent trop, il a pour valeur 0.5. Pour la tâche de classification de documents, la métrique choisie est la précision (accuracy) qui a pour formule

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{6.3}$$

avec $TP + TN$ les prédictions correctes (vrais positifs et vrais négatifs), et $TP + TN + FP + FN$ le nombre total de prédictions, où FP et FN sont respectivement les faux positifs et les faux négatifs. Enfin, pour la tâche d'analyse de structure de documents, les modèles ont été évalués en utilisant la moyenne des intersections sur l'union (mean Intersection over Union, mIoU). Cette métrique fait la moyenne de prédictions positives correctes pour chaque classe tel que :

$$\text{mIoU} = \frac{1}{C} \sum_{i=1}^{C} \frac{TP_i}{TP_i + FP_i + FN_i} \tag{6.4}$$

avec C le nombre de classes, et TP_i, FP_i, FN_i le nombre de vrais positifs, faux positifs et faux négatifs pour la classe i.

6.2 Compression d'un encodeur visuel de fondation par distillation

Cette section présente les résultats des expérimentations réalisées pour le premier objectif de ce mémoire (voir chapitre 3).

6.2.1 Détails des configurations

Le tableau 6.1 décrit les différents hyperparamètres utilisés lors des expérimentations du premier objectif (voir sections 3.2 et 3.2.2). Les expérimentations ont été réalisés sur 3 GPU v100, ayant chacun 32GB de VRAM.

Tableau 6.1 Détails des hyperparamètres des expérimentations du premier objectif

Étape d'entraînement/tâche	α	**Optimiseur**	**Époques**	**Taille des lots**
DocVQA (Distillation)	3e−4	Adam	20	16
DocVQA (Affinement)	3e−5	Adam	3	16
Classification	3e−4	Adam	5	16
Layout Analysis	9e−4	Adam	3	16

Comme expliqué dans la section 3.2, deux méthodes de distillation sont testées. Premièrement, FRD, qui impose une résolution d'entrée à l'étudiant afin que le nombre de patchs en sortie de ce dernier soit le même que celui du professeur. Cette méthode évite d'ajouter de la complexité au modèle, mais contraint la résolution, ne permettant pas de flexibilité. Une autre approche a donc été proposée (ARD), utilisant un module supplémentaire en sortie de l'étudiant qui aligne le nombre de patchs de ce dernier avec le professeur. Cette dernière permet de configurer une résolution d'entrée différente, pouvant être plus petite que la résolution imposée par la méthode FRD afin de limiter l'empreinte sur la VRAM ou plus grande, afin de permettre l'extraction de caractéristiques à plus petite échelle. Dans le cas de la distillation FRD, comme l'encodeur visuel du professeur a une résolution d'entrée $TRes \in \mathbb{R}^{896\times896\times3}$ et une séquence de

sortie $v_{TRes} \in \mathbb{R}^{4096\times1152}$, la résolution imposée pour l'étudiant sera de $MRes \in \mathbb{R}^{2048\times2048\times3}$. L'étudiant (DIVE-Doc) étant un encodeur de type Swin Transformer composé de 4 blocs, la sortie sera notée $v_{MRes} \in \mathbb{R}^{4096\times1024}$ (voir sections 2.3 et 3.2). Pour la méthode ARD, deux résolutions ont été testées. Une résolution petite $LRes \in \mathbb{R}^{1536\times1536\times3}$, entraînant une sortie $v_{LRes} \in \mathbb{R}^{2304\times1024}$, et une résolution grande $HRes \in \mathbb{R}^{2560\times1920\times3}$ entraînant une sortie de taille $v_{HRes} \in \mathbb{R}^{4800\times1024}$. Ainsi, $HRes$ aboutit à 704 patchs supplémentaires en comparaison à $MRes$, entraînant une granularité plus fine. $LRes$ résulte en 1792 patchs en moins que $MRes$, libérant ainsi de l'espace sur la VRAM. Ces sorties sont alignées avec le professeur par une interpolation telle qu'expliquée dans la section 3.2. Différentes méthodes d'interpolation ont été évaluées sur chaque résolution de la méthode ARD, ces résultats sont présentés dans la section 6.2.3.

6.2.2 Résultats

6.2.2.1 Évaluation sur la tâche DocVQA

Le tableau 6.2 présente les performances (ANLS) sur la tâche de DocVQA, en comparaison avec l'état de l'art. Les modèles étudiants DIVE-Doc obtiennent 82.67% pour la méthode FRD, 82.63% pour $ARD/HRes$ et 79.26% pour $ARD/LRes$. Là où les méthodes avec OCR (UDOP et LayoutLMv3) ont des performances de respectivement 84.70% et 78.76%, les modèles DIVE-Doc ont des performances compétitives sans reposer sur des outils externes tels que l'OCR. De plus, le LVLM Paligemma a un score de 84.77%, représentant un gap d'environ 2 points d'ANLS avec les modèles DIVE-Doc FRD et $ARD/HRes$. L'encodeur de DIVE-Doc ayant seulement 75 millions de paramètres contre 400 millions pour celui de Paligemma, ce gap est réalisé avec $\frac{1}{5}$ du nombre de paramètres de Paligemma. D'autre part, les modèles à petite échelle tels que Donut et Dessurt obtiennent des performances respectives de 66.26% et 63.22%, soulignant que les modèles DIVE-Doc obtiennent de meilleures performances que ces derniers, avec un gap minimum de 13 points d'ANLS entre DIVE-Doc $ARD/LRes$ et Donut. Les résultats de DIVE-Doc cités ci-dessus montrent que la meilleure performance obtenue est celle de la méthode FRD, puis $ARD/HRes$ et enfin $ARD/LRes$.

Tableau 6.2 Comparaison des résultats de l'objectif 1 avec l'état de l'art pour la tâche de DocVQA

Méthode	**Configuration du Modèle**			**ANLS Générale (%) ↑**
	#Params (VE)	**#Params Total**	**OCR**	
Paligemma, Beyer *et al.* (2024)	0.4(B)	3(B)		84.77
UDOP, Tang *et al.* (2023)	-	0.8(B)	✓	84.70
LayoutLMv3, Huang *et al.* (2022)	-	0.133(B)	✓	78.76
Donut, Kim *et al.* (2022)	0.075(B)	0.2(B)		66.26
Dessurt, Davis *et al.* (2022)		0.127(B)		63.22
DIVE-Doc FRD	0.075(B)	2.58(B)		**82.67**
DIVE-Doc $ARD/HRes$	0.075(B)	2.58(B)		82.63
DIVE-Doc $ARD/LRes$	0.075(B)	2.58(B)		79.26

Le tableau 6.3 présente les résultats détaillés par type de questions, des méthodes présentées dans le tableau 6.2. La résolution $ARD/HRes$ achève une performance de 61.48%, 58.68% et 85.34% pour les questions portant sur des aspects visuels tels que des figures, des photos ou encore la structure. Ainsi, elle surpasse les résultats de FRD ayant respectivement 59.33%, 49.96% et 85.00% sur ces catégories. Cependant, la méthode FRD achève une performance de 78.83% pour des questions portant sur du texte, là où $ARD/HRes$ a 77.64%.

Tableau 6.3 Résultats pour la tâche de DocVQA pour différentes catégories de questions

Méthode	**ANLS (%) par catégorie de questions ↑**			
	Figure	Texte	Photo	Structure
Paligemma, Beyer *et al.* (2024)	65.43	80.99	73.82	87.33
UDOP, Tang *et al.* (2023)	-	-	-	-
LayoutLMv3, Huang *et al.* (2022)	-	-	-	-
Donut, Kim *et al.* (2022)	39.60	46.43	29.69	69.87
Dessurt, Davis *et al.* (2022)	31.64	48.52	28.62	64.86
DIVE-Doc FRD	59.33	**78.83**	49.96	85.00
DIVE-Doc $ARD/HRes$	**61.48**	77.64	**58.68**	**85.34**
DIVE-Doc $ARD/LRes$	54.94	74.54	58.28	83.15

6.2.2.2 Évaluation de l'encodeur visuel

Cette section fournit des résultats sur les tâches de classification de document (DocCLS) et d'analyse de structure de document (DLA), utilisées afin d'évaluer l'encodeur visuel et de fournir des détails supplémentaires sur ce que cette composante du modèle a réellement appris pour la tâche de DocVQA (voir section 3.2.2). Afin d'avoir une évaluation plus complète, ces expérimentations, en plus d'être effectuées sur les modèles DIVE-Doc, ont également été conduites sur l'encodeur visuel de Donut et de Paligemma. Les résultats sont présentés sur le tableau 6.4.

Tableau 6.4 Résultats des encodeurs visuels sur les tâches de DocCLS et DLA

	Classification (Acc ↑)	**Analyse de la structure (IoU ↑)**				
Méthode	Générale	Moyenne	Texte	Titre	Liste	Note en pied de page
Paligemma, Beyer *et al.* (2024)	0.92	0.36	0.54	0.13	0.07	0.05
Donut, Kim *et al.* (2022)	0.89	0.37	0.54	0.08	0.07	0.05
DIVE-Doc FRD	0.90	0.41	0.58	0.1	0.06	0.06
DIVE-Doc $ARD/HRes$	0.90	0.30	0.50	0.07	0.05	0.04
DIVE-Doc $ARD/LRes$	0.90	0.39	0.54	0.14	0.07	0.06

Pour la tâche de classification, les encodeurs visuels ont tous une performance correcte avec peu de différences dans les résultats, allant de 0.89 pour Donut jusqu'à 0.92 pour Paligemma. Cependant, pour la tâche d'analyse de structure, les résultats chutent drastiquement. Les performances générales vont de 0.30 ($ARD/HRes$) à 0.41 (FRD). Dans les catégories d'entités, le texte semble être mieux segmenté/reconnu que des catégories de la même modalité mais plus précises sémantiquement comme le titre allant de 0.07 ($ARD/HRes$) à 0.13 (Paligemma), les liste de 0.05 ($ARD/HRes$) à 0.07 ($LRes/HRes$, Donut, Paligemma) et les notes en pieds de pages (footnote) allant de 0.04 ($ARD/HRes$) à 0.06 ($ARD/LRes$ et FRD). La figure 6.1 montre des visualisations qualitatives de cette tâche. Les modèles semblent segmenter et discriminer correctement les patchs/pixels d'entités de modalités différentes texte (violet) et photo (vert) par rapport à la vraie segmentation (ground truth). Cependant, pour les éléments dans la modalité de texte mais de classes sémantiques plus précises, comme le titre (orange), les listes (rouges) et les tableaux (gris), les modèles ont du mal à discriminer ces derniers correctement. L'image de

document sur la deuxième ligne voit une bonne partie de son tableau central classifié tel que du texte (violet) et certaines zones sont même classifiées telles que liste (rouge). Il en est de même pour la troisième image de document, qui voit ses listes (rouges sur la ground truth) classifiées telles que du texte par les modèles.

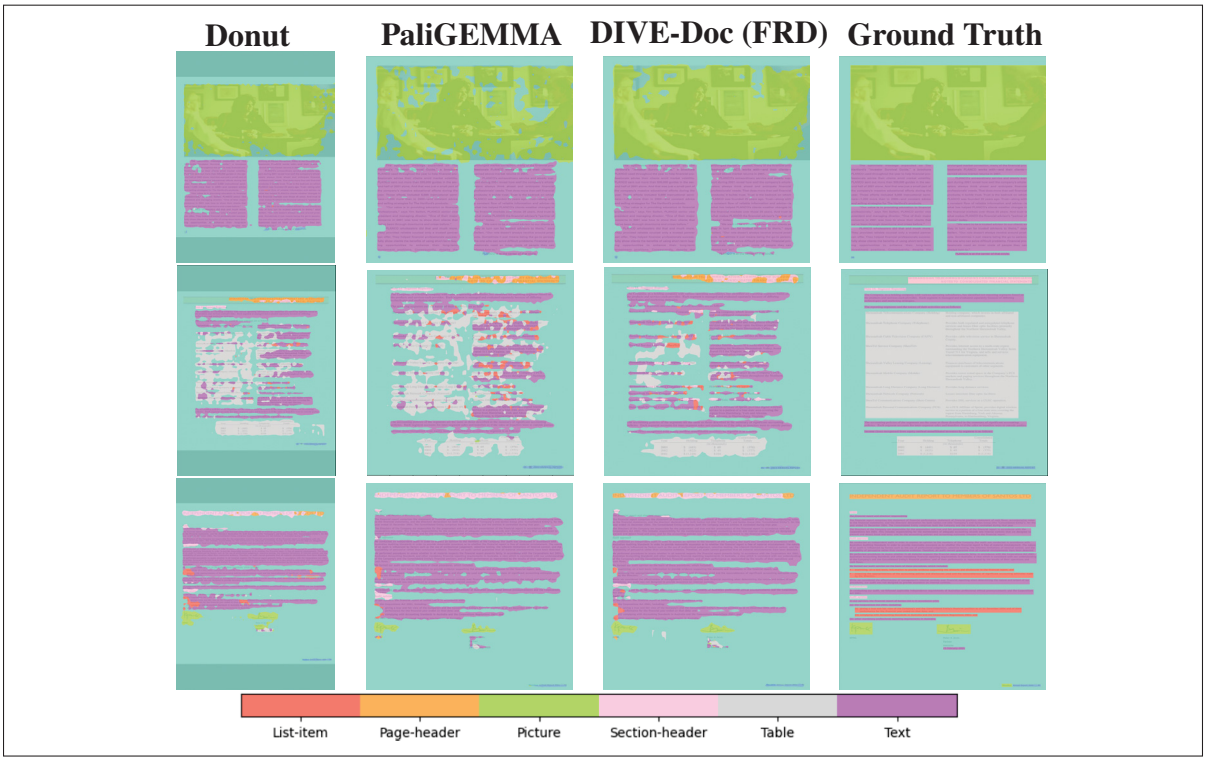


Figure 6.1 Analyse qualitative des encodeurs visuels de DocVQA pour l'analyse de structure de documents

6.2.3 Étude d'ablation

Cette section contient des études supplémentaires conduites pour analyser plus en détail les choix d'architecture.

Premièrement, une évaluation sur le coût de calcul a été réalisée entre les encodeurs visuels de Paligemma (le professeur) et les encodeurs visuels étudiants (DIVE-Doc FRD, $ARD/HRes$ et $ARD/LRes$). Pour cela, l'empreinte mémoire sur les GPU (VRAM) a été mesurée ainsi que la latence afin de comparer les gains de la distillation en temps de procédure.

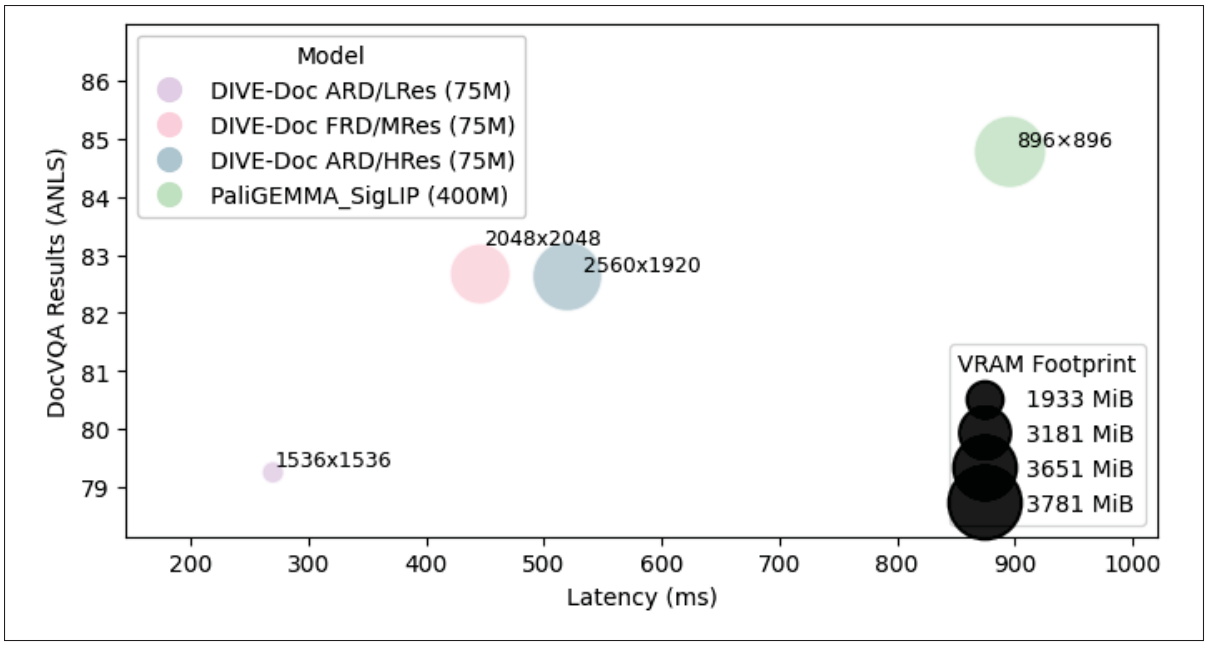


Figure 6.2 Comparaison entre l'efficacité (Latence/VRAM) et la performance (ANLS) des encodeurs visuels distillés

Les résultats sont affichés sur la figure 6.2. L'encodeur visuel de Paligemma a une latence par image d'environ 896ms. Là où les encodeurs visuels de DIVE-Doc FRD et $ARD/HRes$ ont une latence respective de 446 et 520ms par image. Ainsi, la réduction de l'encodeur visuel de Paligemma a permis de diminuer le coût de calcul de ce dernier en divisant la latence par un facteur de deux pour les méthodes FRD et $ARD/HRes$ tout en assurant des résultats compétitifs sur la tâche de DocVQA. Cependant, là où l'empreinte sur la VRAM est de 3781 MiB, celle des modèles $ARD/HRes$ et FRD est environ de la même grandeur avec respectivement 3651 et 3183 MiB. D'autre part, la méthode $ARD/LRes$ a un gap d'ANLS plus grand avec le professeur, comparée aux méthodes FRD et $ARD/HRes$. Cependant, sa résolution plus petite permet de diviser l'empreinte sur la VRAM par un facteur de deux (1933MiB) et la latence par trois (270ms).

De plus, différents types d'interpolations ont été évalués afin de définir le plus efficace pour les méthodes ARD. Avec cela, un encodeur étudiant de type SigLIP a été implémenté afin d'attester de l'efficacité d'un modèle hiérarchique ayant une architecture différente du professeur. Ainsi, le

modèle Paligemma$_T$ a été testé, composé d'un encodeur SigLIP de 80 millions de paramètres (même architecture que celui de Paligemma), prenant en entrée la même résolution $TRes$. Les évaluations ont été établies sur les modèles entraînés lors de la distillation (étape 1, voir section 3.2). Les résultats sont générés en réutilisant le décodeur de Paligemma, sans affinement ou entraînement supplémentaire, en utilisant la méthode SIMKD par Chen *et al.* (2022a). Les résultats sont affichés sur le tableau 6.5. Les résultats de Paligemma$_T$ (67.13%) sont inférieurs à

Tableau 6.5 Comparaison de l'ANLS (%) après distillation des modèles étudiant hiérarchique et non-hiérarchique

Méthode	# Params (M)	Bilinéaire	Bicubique	Sans Alignement
Prof : Paligemma, Beyer *et al.* (2024)	400	/	/	84.77
Étudiant : Paligemma$_T$	80	/	/	67.13
Étudiant : DIVE-Doc FRD	75	/	/	81.71
Étudiant : DIVE-Doc $ARD/HRes$	75	81.07	81.15	/
Étudiant : DIVE-Doc $ARD/LRes$	75	73.1	74.0	/

ceux des modèles DIVE-Doc, montrant l'utilité de l'architecture hiérarchique par rapport à une architecture réduite mais similaire au modèle de fondation. D'autre part, l'interpolation bicubique permet les meilleurs résultats pour les deux résolutions ($HRes \rightarrow 81.15\%$) et ($LRes \rightarrow 74.0\%$) contre ($HRes \rightarrow 81.07\%$) et ($LRes \rightarrow 73.1\%$) pour l'interpolation bilinéaire. Cependant, le gap pour $HRes$ étant très faible (0.08), la méthode bilinéaire a été choisie pour cette résolution, car elle ne fait intervenir que 4 pixels voisins avec de simples pondérations linéaires la rendant légèrement moins coûteuse en opérations que l'interpolation bicubique qui exploite un voisinage élargi de 16 pixels, nécessitant des calculs polynomiales de plus haut degré (3), entraînant plus de calculs.

6.2.4 Discussion

Les résultats à travers la tâche de DocVQA ont permis de démontrer la validité de ce premier objectif. En changeant l'architecture d'un encodeur visuel de fondation pour un modèle hiérarchique, la résolution a pu être augmentée, tout en réduisant la taille de l'encodeur par un facteur de cinq. Cette approche a ainsi permis de réduire le coût de calcul de l'encodeur visuel en

divisant la latence par deux et en minimisant la perte de performances. De plus, pour la méthode *ARD* permettant des résolutions plus variées, il s'est avéré qu'une résolution plus petite que celle imposée par la méthode *FRD* augmente le gap de performance avec le professeur mais améliore l'impacte sur le coût de calcul en divisant la latence par trois et l'empreinte sur la VRAM par deux, ce qui peut être bénéfique pour des environnements limités en puissance de calcul. D'autre part, la méthode *ARD* avec une résolution supérieure à la méthode *FRD* ne semble pas améliorer la performance d'un point de vue général mais uniquement sur des entités visuelles telles que les photos et les graphiques (voir tableau 6.3). La méthode d'alignement utilisée étant sans paramètres supervisés (sans apprentissage), il peut être supposé que certains détails de la représentation sont perdus lors de l'alignement de la séquence de sortie de l'étudiant avec celle du professeur, limitant ainsi les résultats de cette approche. Enfin, les expériences réalisées sur les tâches de classification et d'analyse de structure de documents apportent des détails supplémentaires sur ce qu'ont appris les encodeurs visuels. Pour chaque encodeur évalué, les performances sur la classification sont élevées, indiquant que ces modèles représentent correctement la structure des documents, leur permettant de les différencier et de les classifier. Cependant, pour la tâche d'analyse des structures des documents, les encodeurs visuels ont des performances assez faibles. L'analyse de leurs résultats montre que ces modèles arrivent à discriminer les éléments de modalités différentes (image et texte). Néanmoins, les éléments appartenant à une même modalité mais ayant une structure sémantique différente (liste, titre, pied de page, etc.) sont mélangés et ont du mal à être correctement segmentés. Cependant, pour la tâche de DocVQA, les modèles obtiennent de bons résultats pour les questions portant sur les structures sémantiques des documents (voir tableau 6.3). Ainsi, il peut être supposé que pour cette tâche, la représentation de la structure sémantique des éléments des documents est achevée par le LLM, là où l'encodeur visuel permet d'extraire les caractéristiques visuelles des images de documents.

6.3 Enrichissement des représentations avec la géométrie spatiale des documents

Cette section présente les résultats des expérimentations de l'objectif 2, voir chapitre 4.

6.3.1 Détails des configurations

Les hyperparamètres des expériences de l'objectif 2 (voir section 4.2 et 3.2.2) sont affichés sur le tableau 6.6. Les expérimentations ont été réalisées sur un GPU H100 de 80GB de VRAM. Afin

Tableau 6.6 Détails des hyperparamètres des expérimentations du premier objectif

Étape d'entraînement/tâche	α	**Optimiseur**	**Époques**	**Taille des lots**
DocVQA (Étape 1)	3e−4	Adam	2	16
DocVQA (Étape 2)	9e−5	Adam	3	16
DocVQA (Étape 3)	3e−5	Adam	3	16
Classification	3e−4	Adam	5	16
Layout Analysis	9e−4	Adam	3	16

d'évaluer la stratégie étudiée, les expérimentations ont été conduites sur le modèle DIVE-Doc (FRD), présenté et entraîné dans le cadre de l'objectif 1.

6.3.2 Résultats

Les sections suivantes présentent les résultats des expérimentations réalisées sur DIVE-Doc.

6.3.2.1 Évaluation sur la tâche DocVQA

Le tableau 6.7 présente les résultats de DIVE-Doc enrichie par le module de position FFpos en sortie du modèle sur l'ensemble de données DocVQA. Les résultats des autres positions sont affichés dans la section ablation 6.3.3. Le modèle initial a un encodeur visuel faisant 0.075 (B) de paramètres, ce qui représente une petite taille en comparaison avec les modèles de fondation qui ont des encodeurs d'environ 0.4 (B) (Paligemma). Comme affiché sur le tableau, le module de position FFpos (décris dans la section 4.2) ajoute 0.01(B) de paramètres à l'encodeur visuel de DIVE-Doc. Le modèle enrichi (DIVE-Doc + FFpos) atteint un score de 83.46% d'ANLS. Là

où le modèle atteint un score de 82.67% sans enrichissement, le module permet une amélioration de 0.81 point, passant d'un gap de 2.10 à 1.31 points d'ANLS avec le professeur Paligemma. Le tableau 6.8 affiche les résultats par catégorie de questions sur DocVQA. Le module FFpos entraîne un décroissement des performances pour les questions portant sur les figures passant de 59.33% à 57.77% d'ANLS. Cependant, il entraîne une amélioration sur les questions portant sur le texte (78.83% →79.29%) ou encore sur les questions liées à la structure du document (85.00% →85.44%). Enfin, le module a également permis une amélioration sur les questions de la catégorie photo, passant de 49.96% à 53.04%, représentant le gap le plus important avec le modèle initial (3.08 points d'ANLS).

Tableau 6.7 Comparaison des résultats de l'objectif 1 avec l'état de l'art pour la tâche de DocVQA

Méthode	**Configuration du Modèle**			**ANLS Générale (%) ↑**
	#Params (VE)	**#Params Total**	**OCR**	
Paligemma, Beyer *et al.* (2024)	0.4(B)	3(B)		84.77
UDOP, Tang *et al.* (2023)	-	0.8(B)	✓	84.70
LayoutLMv3, Huang *et al.* (2022)	-	0.133(B)	✓	78.76
Donut, Kim *et al.* (2022)	0.075(B)	0.2(B)		66.26
Dessurt, Davis *et al.* (2022)		0.127(B)		63.22
DIVE-Doc	0.075(B)	2.58(B)		82.67
DIVE-Doc + FFpos	0.085(B)	2.6(B)		**83.46**

Tableau 6.8 Résultats pour la tâche de DocVQA pour différentes catégories de questions

Méthode	**ANLS (%) par catégorie de questions ↑**			
	Figure	Texte	Photo	Structure
Paligemma, Beyer *et al.* (2024)	65.43	80.99	73.82	87.33
UDOP, Tang *et al.* (2023)	-	-	-	-
LayoutLMv3, Huang *et al.* (2022)	-	-	-	-
Donut, Kim *et al.* (2022)	39.60	46.43	29.69	69.87
Dessurt, Davis *et al.* (2022)	31.64	48.52	28.62	64.86
DIVE-Doc	**59.33**	78.83	49.96	85.00
DIVE-Doc + FFpos	57.77	**79.29**	**53.04**	**85.44**

6.3.2.2 Évaluation de l'encodeur visuel

Le tableau 6.9 affiche les résultats pour les tâches d'évaluation de l'encodeur visuel. Bien que le modèle initial de DIVE-Doc ait déjà une bonne représentation de la structure générale des documents en atteignant un score de 0.90, le module FFpos a permis d'améliorer ce dernier de 1%, passant à une performance de 0.91. Cela réduit le gap avec l'encodeur visuel de Paligemma qui a une performance de 0.92 (2% → 1%), et augmente celui avec Donut qui atteint 0.89 (1% → 2%). Cependant, l'ajout du module FFpos semble impacter négativement les résultats sur la tâche d'analyse de la structure des documents. Là où le modèle DIVE-Doc initial affiche une performance de 0.41 (mIoU), le module FFpos a entraîné une diminution de cette dernière (0.37).

Tableau 6.9 Résultats des encodeurs visuels sur les tâches de DocCLS et DLA

	Classification (Acc ↑)	**Analyse de la structure (IoU ↑)**				
Méthode	Générale	Moyenne	Texte	Titre	Liste	Note en pied de page
Paligemma, Beyer *et al.* (2024)	0.92	0.36	0.54	0.13	0.07	0.05
Donut, Kim *et al.* (2022)	0.89	0.37	0.54	0.08	0.07	0.05
DIVE-Doc	0.90	0.41	0.58	0.1	0.06	0.06
DIVE-Doc + FFpos	**0.91**	0.37	0.54	0.1	0.06	0.04

6.3.3 Étude d'ablation

Cette section présente les différentes études réalisées pour valider l'approche étudiée. Les résultats présentés en premier lieu sont issus des expérimentations réalisées en étudiant plusieurs zones d'insertion du module FFpos dans l'encodeur visuel. Comme décrit dans la section 4.2, les positions évaluées sont à l'entrée du modèle (d), à la fin de chaque niveau (bl), à la sortie du modèle (s) et enfin, sur toutes ces positions à la fois (dbls). Les résultats des performances générales pour la tâche de DocVQA sont présentés sur le tableau 6.10.

La performance la plus haute a été obtenue en intégrant la position à la sortie du modèle (83.46% ANLS), suivie de la position d'ajout à chaque niveau (83.36%) puis de l'insertion à chaque

Tableau 6.10 Résultats pour les différentes insertions de positions sur la tâche de DocVQA

Modèle	Insertion au début	Insertion à chaque block	Insertion à la sortie	ANLS général (%) ↑
DIVE-Doc				82.67
DIVE-Doc (d)	✓			82.84
DIVE-Doc (bl)		✓		83.36
DIVE-Doc (s)			✓	**83.46**
DIVE-Doc (dbls)	✓	✓	✓	83.19

position proposée (83.19%). L'ajout au début du modèle semble avoir eu le moins d'influence (82.84%), apportant une amélioration de 0.17 points d'ANLS contre 0.81 pour l'insertion en sortie.

Le tableau 6.11 montre les détails des résultats sur différents types de questions. Le module FFpos semble avoir réduit les performances pour les questions portant sur des figures passant de 59.33% d'ANLS pour le modèle initial à 56.74% pour l'ajout à chaque point d'insertion (dbls). Cependant, l'ajout à chaque point d'insertion a eu un fort impact sur les questions de la catégorie photo, avec un score de 58.82% d'ANLS, représentant ainsi une amélioration de 8.86 points d'ANLS. Contrairement aux performances générales, l'insertion du module FFpos uniquement à la sortie entraîne la plus petite amélioration sur la catégorie photo (3.08 points d'ANLS), là où l'ajout au début conduit à une progression de 5.37 points et l'ajout à chaque niveau améliore la performance de 5.17 points. Enfin, pour les questions liées à la structure des documents, le module FFpos entraîne une amélioration allant de 0.09 (d) à 1 point d'ANLS (dbls).

Tableau 6.11 Résultats pour les différentes insertions de positions sur la tâche de DocVQA

	ANLS (%) par catégorie de questions ↑			
Modèle	Figure	Texte	Photo	Structure
DIVE-Doc	**59.33**	78.83	49.96	85.00
DIVE-Doc (d)	56.80	77.43	55.33	85.09
DIVE-Doc (bl)	58.88	78.88	55.13	85.55
DIVE-Doc (s)	57.77	**79.29**	53.04	85.44
DIVE-Doc (dbls)	56.74	78.92	**58.82**	**86.00**

Le tableau 6.12 présente les résultats sur les tâches d'évaluation de l'encodeur visuel pour les différentes positions testées. Chaque position entraîne une amélioration de 1% sur la tâche de

classification passant de 0.90 à 0.91, excepté l'insertion à chaque position simultanément (dbls) qui conserve une performance de 0.90. Pour la tâche d'analyse de la structure, les différentes positions d'insertion entraînent également des résultats similaires, soit 0.37 points de mIoU pour la performance générale, ce qui montre une baisse des résultats par rapport au modèle initial qui affiche une performance de 0.41 points.

Tableau 6.12 Résultats des encodeurs visuels sur les tâches de DocCLS et DLA

	Classification (Acc ↑)	**Analyse de la structure (mIoU ↑)**				
Méthode	Générale	Moyenne	Texte	Titre	Liste	Note en pied de page
DIVE-Doc	0.90	0.41	0.58	0.1	0.06	0.06
DIVE-Doc (d)	0.91	0.37	0.54	0.09	0.06	0.04
DIVE-Doc (bl)	0.91	0.37	0.53	0.1	0.06	0.04
DIVE-Doc (s)	0.91	0.37	0.54	0.1	0.06	0.04
DIVE-Doc (dbls)	0.90	0.37	0.54	0.1	0.06	0.04

6.3.4 Discussion

L'intégration du module de position FFpos à la sortie de l'encodeur a démontré une amélioration des résultats de la tâche de DocVQA. L'ajout du module entraîne 10 millions de paramètres supplémentaires, ce qui représente une augmentation de 0.4% du nombre total de paramètres, tout en réduisant l'écart de performance avec le LVLM Paligemma par rapport au modèle DIVE-Doc initial. Ainsi, ce module permet à la fois d'améliorer les résultats avec une faible augmentation du nombre de paramètres. L'étude de ce module sur différentes catégories montre qu'il réduit cependant les résultats sur les questions portant sur des figures, ce qui suggère une perturbation de la représentation de ces dernières. DIVE-Doc a été pré-entraîné en distillant l'encodeur visuel de Paligemma qui est un modèle SigLIP. Ce dernier a lui-même été pré-entraîné sur des images naturelles (photo). La base de données de DocVQA contient très peu de questions sur des figures (environ 1000 sur les 50.000), sachant qu'une figure peut avoir plusieurs questions qui lui sont associées. Il peut donc être supposé que l'ajout d'une composante dans la représentation telle que la position n'a pas pu être suffisamment adaptée aux entités telles que les figures à cause de ce déséquilibre dans la distribution des données. Là où les performances sur les questions

portant sur des photos ont été améliorées de 3.08 points d'ANLS, ce qui a pu être aidé par le pré-entraînement de SigLIP. Pour les questions portant sur la structure des documents, les performances ont été légèrement améliorées (+0.44 points d'ANLS). Cela suit la discussion du premier objectif qui suggère que la compréhension sémantique de la structure des documents est achevée par le LLM, tandis que l'encodeur visuel extrait et représente les caractéristiques visuelles. D'autre part, l'ajout de la position à la sortie du modèle a également permis d'améliorer de 1% les résultats sur la tâche de classification de documents. Cependant, l'ajout de ce module a entraîné une diminution des résultats sur la tâche d'analyse de la structure des documents, et cela pour toutes les positions d'insertion (0.41 mIoU $\rightarrow$ 0.37 mIoU). Il peut être supposé que le module FFpos étant entraîné de bout-en-bout avec le LLM, il n'apprend pas de contexte (titre, tableau, etc.) sémantiquement lié à la position, ce dernier étant traité par le modèle de langue (voir section 6.2.4).

D'autre part, l'étude de l'insertion du module montre l'importance de l'endroit où est intégré ce dernier pour la tâche de DocVQA. La performance générale maximale est atteinte lorsque le module est ajouté uniquement à la sortie de l'encodeur visuel (83.46% d'ANLS). Cette position d'insertion entraîne également les meilleurs résultats pour les questions portant sur du texte (79.29%), soulignant l'importance de la position dans la représentation du modèle de langue pour l'ordre de lecture. Cependant, l'ajout en sortie entraîne la plus petite amélioration pour les questions de la catégorie photo, ce qui suit naturellement la discussion du premier objectif. L'encodeur visuel sert à extraire et à représenter les caractéristiques visuelles du document, les photos étant des entités visuelles, ajouter la position de manière plus précoce dans la représentation permet à l'encodeur visuel de mieux représenter ces dernières. Cependant, là où ajouter la position au début, à chaque niveau et à la sortie de l'encodeur visuel simultanément améliore la performance sur les questions portant sur des photos (+8.86 points), cette stratégie diminue l'amélioration sur les questions portant sur le texte et sur la performance générale par rapport à l'ajout de la position en sortie uniquement. Il peut être supposé qu'ajouter une information positionnelle de manière récurrente dans la représentation dilue d'autres informations utiles à la compréhension sémantique du LLM.

6.4 Ajout d'un module de filtrage pour étendre le modèle au multi-page

Cette section présente les résultats de l'adaptation du modèle DIVE-Doc au multi-page suivant la méthodologie présentée dans le chapitre 5.

6.4.1 Détails des configurations

Le modèle a été testé avec une résolution de page $MRes \in \mathbb{R}^{2048\times2048\times3}$, basé sur les résultats du premier sous-objectif (voir section 6.2.2.1). Cela conduit à un ensemble de patchs $v \in \mathbb{R}^{4096\times2048}$ pour chaque page avec 4096 le nombre de patchs et 2048 la dimension de l'espace de représentation multimodal. Le nombre de couches h pour le module filtre est de 8. L'évaluation de la performance a été effectuée en utilisant l'ANLS (voir équation 6.1) pour mesurer la qualité des réponses et l'accuracy (équation 6.3) pour l'évaluation du filtre sur la sélection de page.

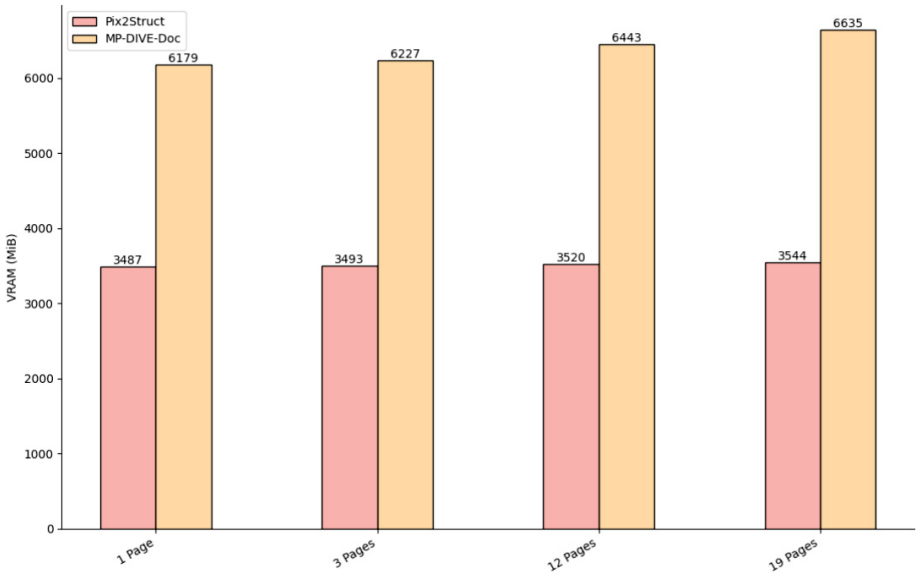
6.4.2 Résultats

Tableau 6.13 Résultats sur MP-DocVQA

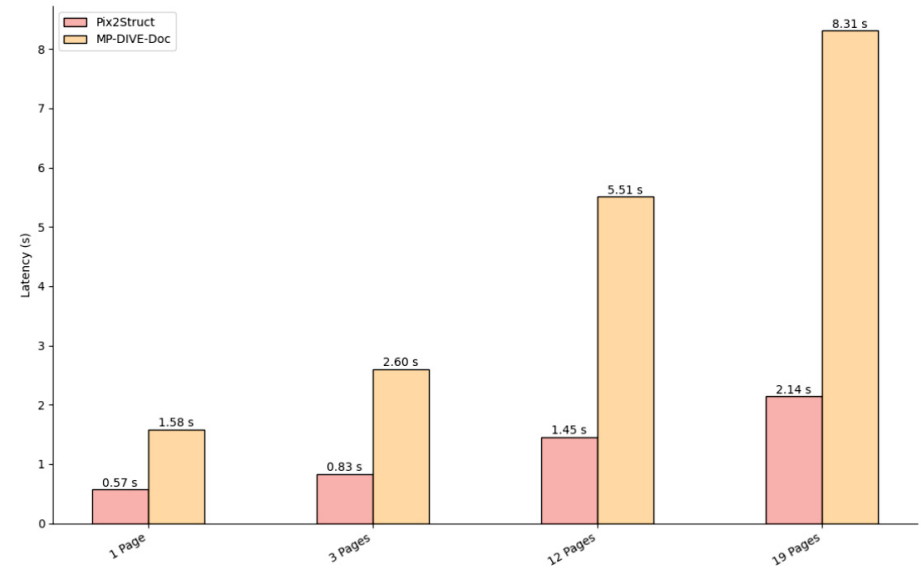
Méthode	# Param(B)	OCR	Tiling	Fusion Tot	Acc (%)↑	ANLS (%)↑
Toutes les pages dans le décodeur						
Gram, Blau *et al.* (2024)	0.859	X		X	19.98	80.32
DocOwl2, Hu *et al.* (2024)	8		X		50.78	69.42
HiVT5, Tito *et al.* (2023)	0.316	X		X	79.63	62.01
Longformer, Tito *et al.* (2023)	0148	X		X	71.17	52.87
BigBird, Tito *et al.* (2023)	0.131	X		X	67.54	49.29
LayoutLMv3, Tito *et al.* (2023)	0.125	X		X	51.94	45.38
Sélecteur de réponse						
ScreenAI, Baechler *et al.* (2024)	5	X			77.88	77.1
ScreenAI, Baechler *et al.* (2024)	5				?	72.9
Sélecteur de page + top-k pages dans le décodeur						
M3DocRAG, Cho *et al.* (2024)	10				81.05	84.44
FRAG-LLaVA-OV, Huang *et al.* (2025)	7				?	79.1
FRAG-InternVL2, Huang *et al.* (2025)	8				?	77.8
Sélecteur de page						
Pix2Struct, Kang *et al.* (2024)	0.273			X	81.55	61.99
MP-DIVE-Doc	2.58				76.25	70.72
MP-DIVE-Doc + FFpos	2.6				76.27	71.73

Le tableau 6.13 présente les résultats sur la tâche MP-DocVQA. Les modèles de ce mémoire adaptés ont été nommés MP-DIVE-Doc (chapitre 3) et MP-DIVE-Doc + FFpos (chapire 4). Ces derniers ont atteint une performance de 70.72% et 71.73% d'ANLS avec 2.6 milliards de paramètres sans utiliser d'OCR. En comparaison avec d'autres approches bout-en-bout, il concurrence des modèles tels que ScreenAI qui a une ANLS de 72.9% et DocOwl2 obtenant 69.42% d'ANLS. Ces derniers ayant respectivement 5 et 8 milliards de paramètres, les deux modèles MP-DIVE-Doc ont donc des résultats compétitifs avec deux à trois fois moins de paramètres. De plus, ils surpassent les modèles qui ont peu de paramètres et qui se basent sur la fusion en amont tels que LayoutLMv3 (45.38% ANLS), BigBird (52.87% ANLS) ou encore HiVT5 (62.01% ANLS). Le modèle Gram (80.32% d'ANLS) surpasse les méthodes MP-DIVE-Doc mais nécessite de prendre l'ensemble des pages dans le décodeur tout en se basant sur l'OCR pour représenter le document, ce qui augmente l'emplacement mémoire requis. De plus, il se base sur la fusion en amont, ce qui nécessite ainsi d'encoder l'image pour chaque question, ne permettant pas de stocker et de réutiliser les embeddings qu'il produit. En comparaison aux modèles sélectionnant les $top - k$ pages pour les envoyer dans leur décodeur, M3DocRAG (84.44% d'ANLS) et FRAG (79.1% et 77.8%) surpassent également MP-DIVE-Doc mais ont plus de paramètres (entre 7 et 10B). Ces méthodes faisant soit appel à différents modèles (M3DocRAG), ou itérant plusieurs fois sur le même modèle de bout-en-bout (FRAG) entraînent ainsi des systèmes plus complexes. Enfin, en comparaison avec Pix2Struct (61,99%) qui se base sur une méthode similaire de sélection de page mais utilise la fusion en amont, MP-Dive-DOC achève un gap supérieur de près de 8.73 points d'ANLS. La fusion en amont de Pix2Struct combinée à un module de sélection de page entraîné atteint une accuracy de 81.55% sur la page prédite, ce qui est supérieure au reste de l'état de l'art. La figure 6.3 compare l'efficience de ce modèle avec MP-DIVE-Doc. Du fait de son nombre de paramètres inférieur à celui de ce dernier, Pix2Struct a une latence bien inférieure à MP-DIVE-Doc qui est respectivement de 0.83 et 2.60 secondes pour un document de trois pages. Cette dernière évolue très peu avec l'augmentation du nombre de pages en comparaison à MP-DIVE-Doc, respectivement de 2.14 et 8.31 secondes pour 19 pages. De même, la faible taille de Pix2Struct lui permet d'avoir une empreinte réduite sur la VRAM (3544MiB) contre 6635MiB pour MP-DIVE-Doc. Cependant, la faible taille du

modèle l'empêche d'avoir une qualité de réponse compétitive (61.99% ANLS) par rapport à MP-DIVE-Doc (70.72% ANLS).



a) Comparaison de l'empreinte VRAM



b) Comparaison de la latence

Figure 6.3 Efficience des modèles MP-DIVE-Doc et Pix2Struct (Kang *et al.*, 2024)

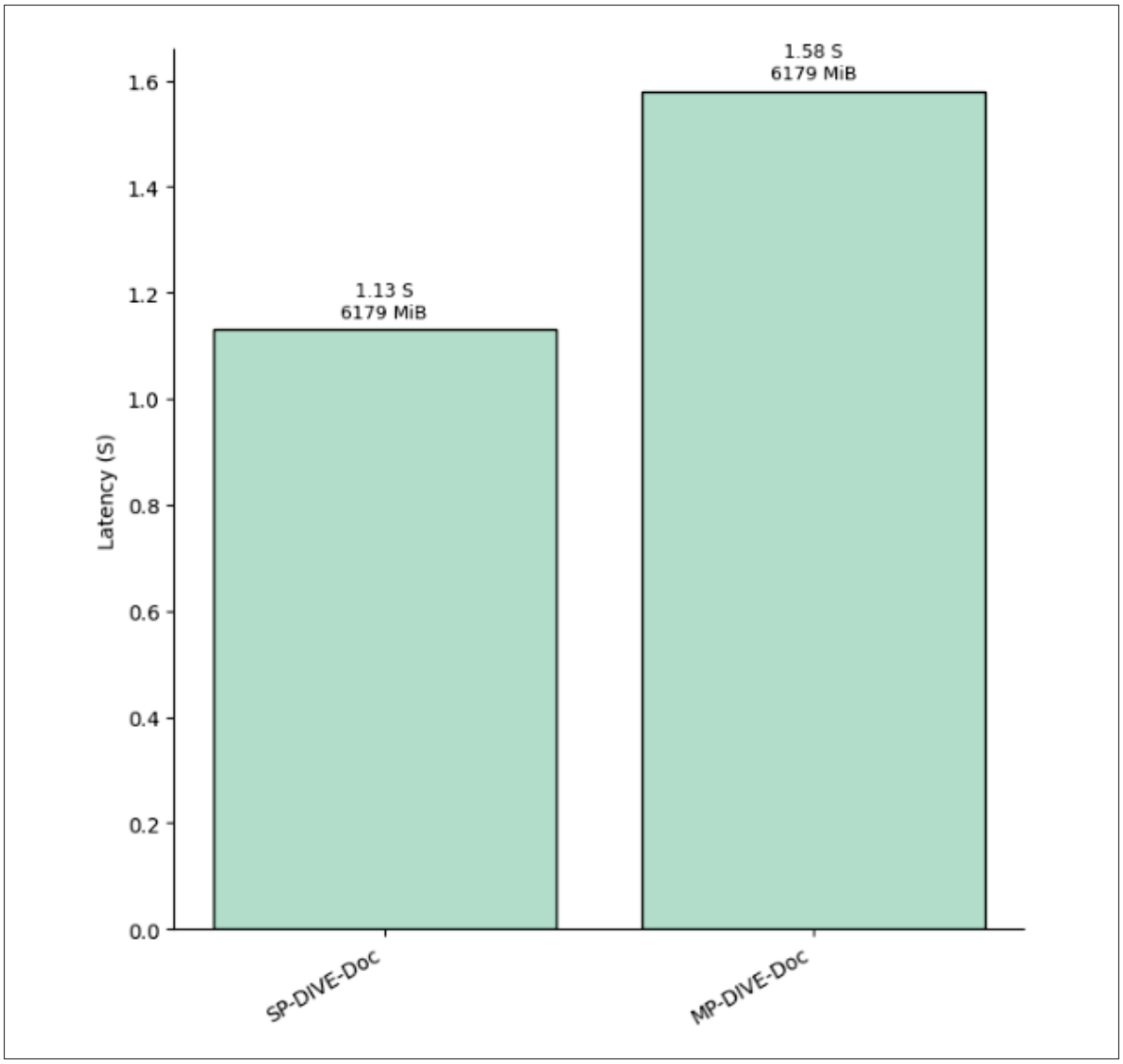
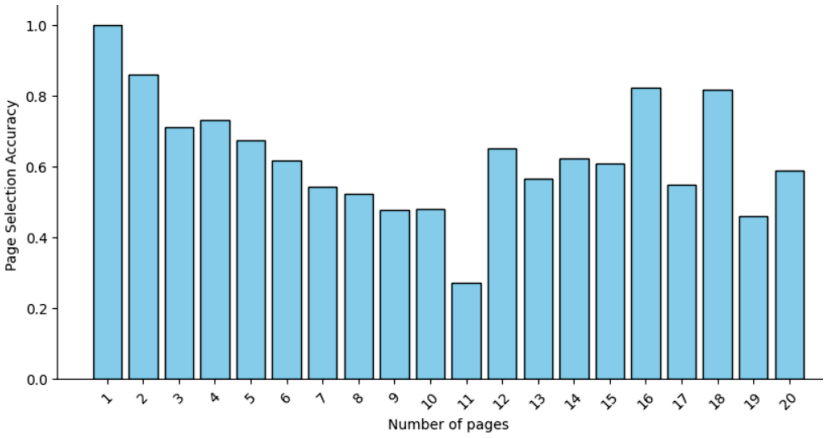
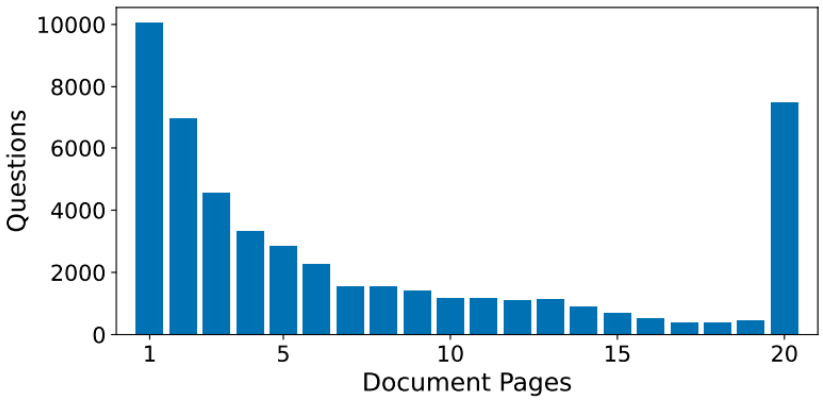


Figure 6.4 Comparaison du coût de calcul sur une page entre le modèle initial DIVE-Doc et MP-DIVE-Doc

La figure 6.4 compare le coût de calcul entre le modèle initial DIVE-Doc et le modèle adapté MP-DIVE-Doc. Pour cela, la latence et l'empreinte mémoire de chaque modèle ont été mesurées pour un document d'une page. Comme il est affiché, les deux modèles ont une faible différence de latence (1.13 et 1.58 secondes) pour retourner la réponse sur une seule page. De plus, leur empreinte sur la VRAM est la même, ce qui s'explique par le fait que le module filtre n'ajoute pas de nouveaux poids au modèle mais réutilise directement ceux du décodeur.



a) Comparaison de la performance de sélection de page en fonction du nombre de page des documents



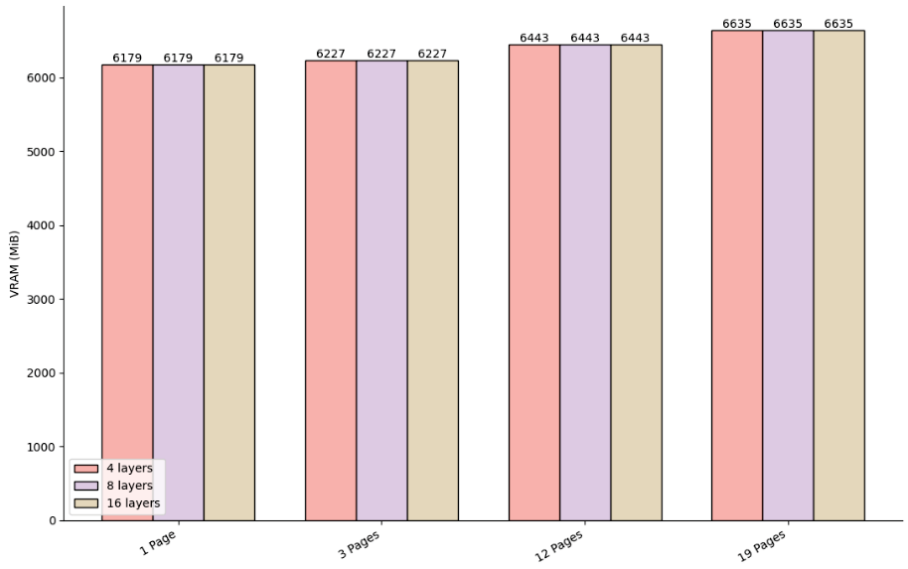
b) Nombre de question par taille de documents, tirée de Tito *et al.* (2023)

Figure 6.5 Nombre de question par taille de documents et performance du filtre

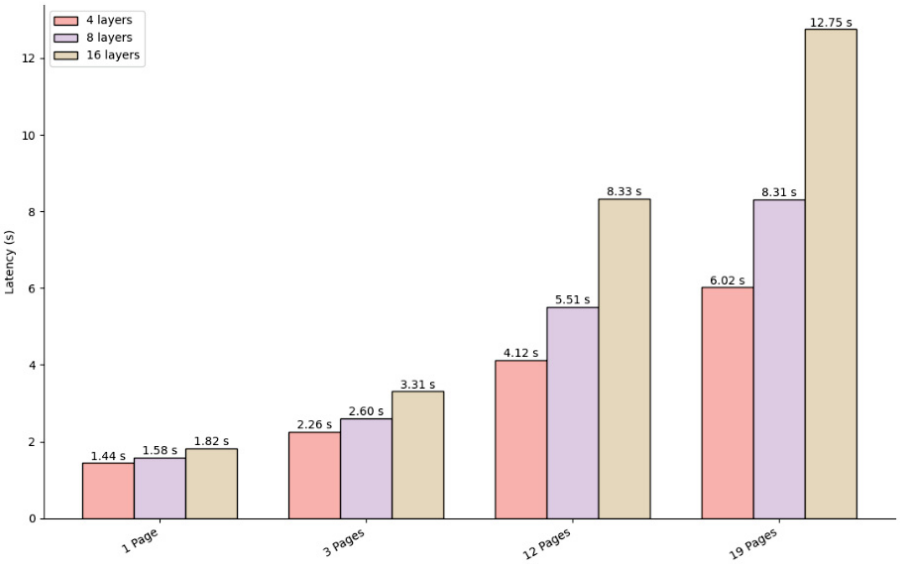
La figure 6.5a montre la performance du filtre de MP-DIVE-Doc sur l'ensemble de validation en fonction du nombre de pages des documents. Comme il est affiché sur ce dernier, les documents à page multiple où le filtre performe le mieux sont ceux ayant 2, 16 et 18 pages avec respectivement 0.86, 0.82 et 0.81 d'accuracy. Étrangement, l'évolution de la performance n'est pas linéaire avec l'augmentation du nombre de pages. La figure 6.5b montre le nombre de questions en fonction du nombre de pages des documents. Comme il est affiché, le nombre de questions pour des documents de 16 et 18 pages est bien inférieur par rapport à des documents entre 2 et 10 pages,

ce qui pourrait biaiser ces résultats. Cependant, on remarque que le nombre de questions pour des documents ayant 20 pages est supérieur au nombre de questions posées sur les autres tailles de documents. De plus, la figure 6.5a affiche une performance de 0.59 pour les documents de 20 pages, ce qui est similaire voir supérieur aux résultats pour des documents ayant par exemple 13 pages (0.57), 9 pages (0.48) ou encore 7 pages (0.54). Ainsi, il peut être supposé que la performance du filtre ne dépend pas directement du nombre de pages mais plutôt de la question posée, à savoir si des informations similaires mais non correctes peuvent être présentes sur les autres pages du document, ce qui pourrait induire le filtre en erreur.

6.4.3 Étude d'ablation



a) Comparaison de l'empreinte VRAM



b) Comparaison de la latence

Figure 6.6 Évaluation de l'impact du nombre de couche du module filtre sur le coût de calcul en fonction du nombre pages

La figure 6.6 présente une comparaison du coût de calcul en fonction du nombre de pages lorsque le nombre de couches du filtre est de 4, 8 et 16. Comme affiché, l'empreinte mémoire ne varie pas en fonction du nombre de couches utilisées mais plutôt en fonction du nombre de pages. Cela s'explique par le fait que le filtre réutilise les poids du LLM, ainsi peu importe son nombre de couches h, ces dernières sont dans tous les cas stockées sur la mémoire via l'architecture initiale. De plus, l'allocation mémoire supplémentaire pour chaque page reste minimale (1 page $\rightarrow$ 6179MiB, 12 pages $\rightarrow$ 6443MiB, 19 pages $\rightarrow$ 6635MiB), ce qui limite l'augmentation de l'emplacement requis sur la VRAM de 0.456MiB pour 18 pages supplémentaires. Cependant, la latence augmente significativement avec le nombre de couches et le nombre de pages. Pour un document d'une seule page, la latence est de 1.44 pour le filtre à 4 couches, de 1.58 secondes pour 8 couches et de 1.82 secondes pour 16 couches. Ces différences augmentent fortement avec le nombre de pages, atteignant respectivement jusqu'à 6.02, 8.31 et 12.75 secondes de latence pour les modules à 4, 8 et 16 couches. Le gap augmente donc avec le nombre de pages, passant de 0.38 secondes pour un document d'une page, à 6.73 secondes pour 19 pages entre les filtres à 4 et 16 couches.

La figure 6.7 montre l'accuracy pour la prédiction de la page contenant la réponse sur l'ensemble de validation entre les filtres à 4, 8 et 16 couches. Comme il est affiché, le module utilisant 8 couches obtient 78.23% d'accuracy alors que ceux à 4 et 16 couches atteignent respectivement une valeur de 73.54% et 72.89%. Cela peut s'expliquer par le fait que 4 couches seulement ne permettent pas d'encoder suffisamment la requête avec les embeddings pour en faire ressortir les informations relatives. D'autre part, à partir d'un certain nombre de couches, l'information de la réponse est dispatchée sur plusieurs jetons de l'image suite au mécanisme de l'attention, réduisant l'efficacité du calcul du score se basant sur la similarité d'un patch par jeton (voir équation 5.1). Ainsi, le module à 8 couches est plus performant, que ce soit en termes d'efficience (latence) ou de qualité de prédiction (accuracy).

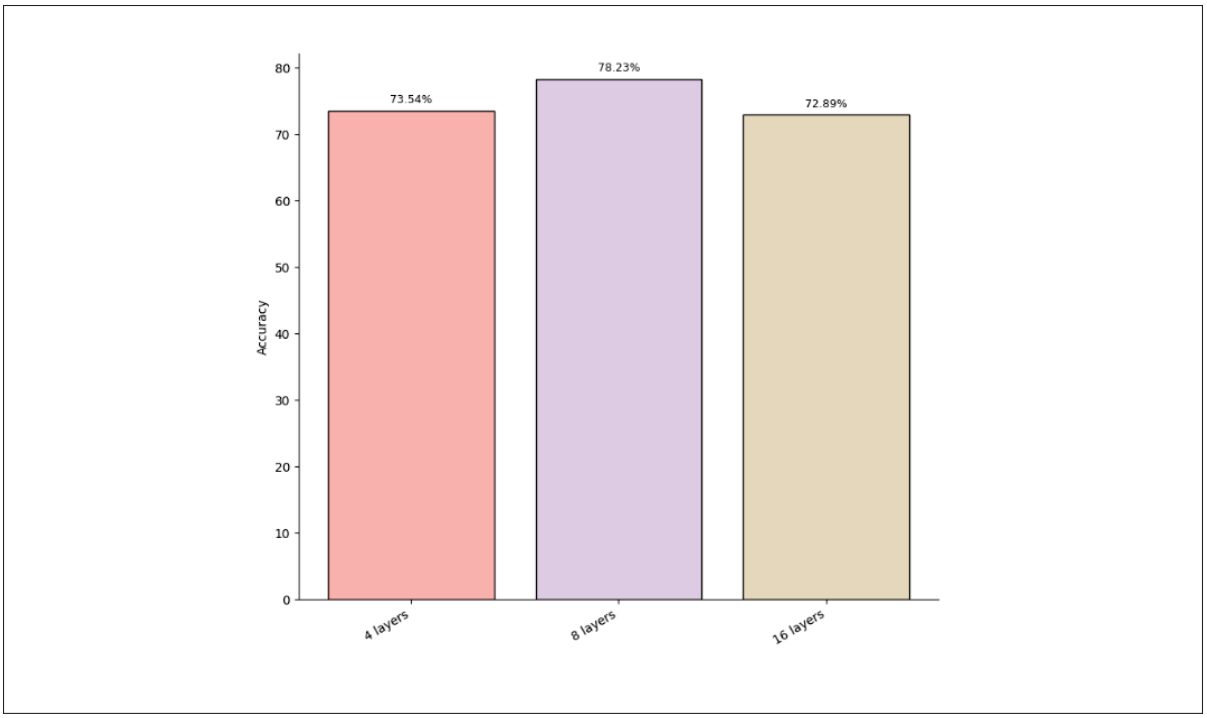


Figure 6.7 Comparaison de la qualité de prédiction de la page contenant la réponse entre les filtre à 8 et 16 couches

6.4.4 Discussion

Les modèles MP-DIVE-Doc ont ainsi obtenu des résultats compétitifs avec l'état de l'art. Achevant un score de 70.72% d'ANLS et 71.73% (+FFpos), ils surpassent les approches initiales ayant peu de paramètres qui se basent sur l'OCR et rivalisent aussi avec les méthodes bout-en-bout ayant beaucoup de paramètres telles que ScreenAI (5B) et DocOwl2 (8B). Il existe tout de même un gap avec certaines approches comme M3DocRAG (10B) qui utilise plusieurs modules pour encoder le document, atteignant ainsi une performance de 84,44% d'ANLS. Cependant, cette approche nécessite d'encoder le document une première fois pour effectuer la similarité, ainsi qu'une seconde fois pour être utilisable par le LLM génératif. Ainsi, cette méthode entraîne un coût de calcul plus important. D'autre part, dans le cas de stockage des embeddings pour des cas d'utilisation de recherche d'information dans des bases de données vectorielles (voir figure 0.2c), les approches de fusion en amont ne permettent pas

de sauvegarder les représentations des documents, ces derniers étant encodés spécifiquement pour chaque question. Les approches MP-DIVE-Doc se basant sur la fusion intermédiaire, les embeddings de documents qu'elle produit peuvent être sauvegardés et réutilisés pour de nouvelles questions/recherches d'information. De plus, les modèles MP-DIVE-Doc n'ont pas besoin de paramètres additionnels pour traiter le multi-page, ce qui limite l'empreinte du modèle sur la VRAM. Ainsi, la méthode proposée adapte le modèle bout-en-bout initial DIVE-Doc sans ajout de paramètres, tout en permettant la réutilisation des représentations des documents pour de nouvelles questions avec une approche de fusion intermédiaire. Il a été constaté que la performance du filtre n'était pas nécessairement décroissante en fonction du nombre de pages, ce qui suggère que la performance dépend plutôt de la question et des informations relatives contenues dans les pages, ce qui ouvre la porte à l'étude d'un critère plus poussé pour calculer le score par page. De plus, la latence augmente avec le nombre de pages, ce qui peut devenir une contrainte sur des documents contenant beaucoup de pages, ou même sur une extension de l'approche à des collections de documents. D'autre part, le LLM utilisé (gemma) étant entraîné à traiter un nombre limité de jetons, le mécanisme de sélection proposé retourne seulement une page, ce qui entraîne de nouveaux défis pour des cas d'utilisation où la réponse est dispatchée sur plusieurs pages (multi-hop).

CONCLUSION ET RECOMMANDATIONS

Ce mémoire a abordé la représentation d'image de document pour la tâche de réponse à des questions visuelles (DocVQA). Cette tâche a un rôle important dans le contexte de l'augmentation du nombre de documents numérisés pour des cas d'utilisation allant de l'extraction automatique de données à la recherche d'information spécifique dans des documents multi-page (MP-DocVQA). Elle nécessite à la fois une précision sur la qualité des informations extraites et un coût d'infrastructure restreint afin d'être déployable dans les secteurs industriels. Pour cela, représenter les documents dans un espace multimodal est essentiel afin que les modèles de langues puissent s'en servir afin de retrouver l'information à extraire. Ainsi, un modèle de type vision-language (VLM) a été développé afin de répondre à cette tâche sur des documents industriels comportant différents types d'information (texte manuscrit, illustration, graphique, etc.). Ce modèle nommé DIVE-Doc est composé d'un encodeur visuel qui prend en entrée une image de document numérisée et retourne sa représentation dans un espace multimodal. Cette dernière est ensuite envoyée avec une question au modèle de langue qui retourne la réponse extraite à partir de la représentation du document. Ce modèle bout-en-bout ne repose donc pas sur des outils extérieurs tels que l'OCR, ce qui réduit la complexité du système.

Afin de réduire le coût de calcul du modèle sans dégrader la qualité des réponses, ce dernier a été construit à partir d'un grand modèle de vision-language (LVLM). Ces architectures ayant beaucoup de paramètres, ce qui augmente leur latence ainsi que leur emplacement sur la mémoire de l'infrastructure, l'encodeur visuel du modèle initial a été réduit par distillation. Cette méthode a permis de changer l'architecture de l'encodeur visuel pour une structure hiérarchique ayant moins de paramètres, réduisant la latence par deux pour cette composante du modèle. De plus, l'architecture hiérarchique a permis de prendre en entrée une résolution d'image plus importante, conservant ainsi la qualité des réponses du modèle. Par ailleurs, une approche de distillation non conventionnelle a été proposée, permettant d'adapter la résolution des images de documents en fonction des besoins sans ajout de paramètres.

De plus, un module de représentation de la géométrie spatiale (structure) des documents a été intégré afin d'améliorer les ordres de lecture du modèle de langue, ce qui a permis d'affiner la qualité des réponses. Différentes positions d'insertion de ce dernier ont été testées, suggérant qu'un ajout en sortie de l'encodeur visuel est bénéfique afin de permettre à ce dernier de représenter le contenu avant de l'enrichir par sa position sur le document initial. Par ailleurs, une étude de l'encodeur visuel sur les tâches de classification et d'analyse de la structure sémantique des documents a été réalisée afin d'interpréter ce que le module de vision a appris à extraire et représenter. La tâche de classification a révélé que l'encodeur représente correctement la structure générale du document dans l'espace multimodal, lui permettant de discriminer les documents de différentes structures (lettre, formulaire, article, etc.). Cependant, la tâche d'analyse de la structure sémantique révèle que l'encodeur visuel ne discrimine pas les entités d'un même type d'information mais ayant une sémantique structurelle différente (titre, tableau, note en pied de page, etc.). Le modèle global (encodeur visuel et modèle de langue) ayant une bonne qualité de réponse sur les questions portant sur l'analyse de la structure sémantique, cela souligne que cet aspect est donc traité par le modèle de langue.

Enfin, une ouverture sur les documents multi-page a été proposée, en adaptant le modèle construit lors des sous-objectifs de ce mémoire. Afin de retrouver la page contenant la réponse, un module filtre a été ajouté entre l'encodeur visuel et le modèle de langue. Le filtre réutilise les huit premières couches du décodeur afin d'encoder la question avec les embeddings de chaque page et retourne un score de probabilité pour chacune de ces dernières. Les embeddings de la page ayant le plus haut score sont ensuite envoyés au LLM avec la question afin de générer la réponse. Ainsi, en réutilisant les paramètres du modèle de langue dans le filtre, les modèles MP-DIVE-Doc atteignent 70.72% et 71.73% ANLS, sans paramètres additionnels, surpassant des modèles plus lourds en efficacité tout en ayant des performances compétitives. Les modèles se basant sur la fusion intermédiaire, les représentations des documents qu'ils produisent sont

réutilisables pour chaque question. Ainsi, cela ancre le système dans des cas d'application réels tels que les rapports industriels multi-pages.

Le travail de ce mémoire a été réalisé afin de répondre à la tâche de DocVQA pour des cas d'utilisation industriels. Ainsi, la base de données utilisée pour entraîner et évaluer le modèle est essentiellement constituée d'informations sous forme de texte. Cependant, il pourrait être intéressant d'étudier l'architecture proposée sur d'autres bases de données comportant différentes distributions des types d'information tels que pour des documents infographiques, des rapports scientifiques et autres afin de répondre à des contextes d'utilisation différents.

D'autre part, l'approche de distillation proposée pour réduire la taille de l'encodeur visuel initial étant sans paramètres, cette dernière n'a pas permis de bénéficier totalement des documents de grande résolution. Ainsi, adapter un module d'alignement à un faible nombre de paramètres permettrait d'améliorer la qualité des réponses pour les infrastructures disposant d'une plus grande capacité de calcul. De plus, bien que la distillation de l'encodeur visuel ait permis de diminuer le coût de calcul du modèle, le LLM reste la composante la plus coûteuse en termes d'infrastructure. L'étude approfondie de l'encodeur visuel a démontré que l'analyse de la structure sémantique du document pour la tâche de DocVQA est effectuée par le modèle de langue. Ainsi, intégrer cette notion dans l'encodeur visuel permettrait de réduire la taille et la complexité du décodeur afin de faciliter son utilisation sur des infrastructures limitées.

Enfin, l'adaptation des modèles aux documents multi-pages par l'ajout d'un filtre permet de retrouver la réponse à une question parmi plusieurs pages d'un document. Cependant, il a été suggéré que la performance du filtre ne dépend pas directement du nombre de pages, mais plutôt du contexte (informations présentes dans les pages) ainsi que de la question. Ainsi, étudier d'autres critères de sélection pourrait permettre une meilleure discrimination des pages ayant des informations similaires mais non relatives à la question posée. D'autre part, la latence augmentant drastiquement avec le nombre de pages, des défis demeurent pour l'extension sur des collections de documents, pouvant contenir plusieurs centaines de pages. Un axe de développement pourrait être l'architecture de base de données vectorielle hiérarchique permettant un premier filtre

des pages candidates. Une autre difficulté consiste à retrouver la réponse lorsque celle-ci est dispatchée sur plusieurs pages. En effet, envoyer plusieurs pages au modèle de langue pour retrouver la réponse augmente considérablement le nombre de jetons à traiter et, par extension, le coût de calcul. Ainsi, une adaptation du filtre afin de prendre en entrée plusieurs pages pour une sélection modulable en fonction de la question et du nombre de pages pourrait être étudiée. D'autre part, les modèles hiérarchiques ayant fait leurs preuves pour allier coût de calcul et résultats dans le contexte de représentation d'image de documents, s'inspirer de leur architecture pour les modèles de langue pourrait être une piste intéressante à explorer afin de réduire l'évolution du coût de calcul en fonction du nombre de pages à traiter.

Contributions

Publication

Bencharef, R., Rahiche, A. & Cheriet, M. (2025). DIVE-Doc : Downscaling foundational Image Visual Encoder into hierarchical architecture for DocVQA. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 7547- 7556.

Distinction : Prix du meilleur article du workshop.

Open source

Code chapitre 3 (GitHub) : github.com/JayRay5/DIVE-Doc

Modèles chapitre 3 (HuggingFace) : huggingface.co/JayRay5/DIVE-Doc-FRD

Code chapitre 4 (GitHub) : github.com/JayRay5/DIVE_Doc_FFPos

Modèles chapitre 4 (HuggingFace) : huggingface.co/JayRay5/DIVEdoc_ffpos_end

Code chapitre 5 (GitHub) : github.com/JayRay5/MP-DIVE-Doc

BIBLIOGRAPHIE

Angela Tudico, N. A. N. (2022). National Archives Tops 200 Million Digitized Pages in Online Catalog. Repéré à https://www.archives.gov/news/articles/catalog-200-million-digitized-pages.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L. & Parikh, D. (2015, December). VQA : Visual Question Answering. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

ASTP. (2021). National Trends in Hospital and Physician Adoption of Electronic Health Records [Format]. Repéré à https://www.healthit.gov/data/quickstats/national-trends-hospital-and-physician-adoption-electronic-health-records.

Ba, J. & Caruana, R. (2014). Do deep nets really need to be deep ? *Advances in neural information processing systems*, 27.

Baechler, G., Sunkara, S., Wang, M., Zubach, F., Mansoor, H., Etter, V., Cărbune, V., Lin, J., Chen, J. & Sharma, A. (2024). Screenai : A vision-language model for ui and infographics understanding. *arXiv preprint arXiv :2402.04615*.

Bencharef, R., Rahiche, A. & Cheriet, M. (2025, October). DIVE-Doc : Downscaling foundational Image Visual Encoder into hierarchical architecture for DocVQA. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 7547-7556.

Beyer, L., Zhai, X., Royer, A., Markeeva, L., Anil, R. & Kolesnikov, A. (2022). Knowledge distillation : A good teacher is patient and consistent. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10925–10934.

Beyer, L., Steiner, A., Pinto, A. S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E. et al. (2024). Paligemma : A versatile 3b vlm for transfer. *arXiv preprint arXiv :2407.07726*.

Biten, A. F., Tito, R., Mafla, A., Gomez, L., Rusinol, M., Valveny, E., Jawahar, C. & Karatzas, D. (2019). Scene text visual question answering. *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4291–4301.

Blau, T., Fogel, S., Ronen, R., Golts, A., Ganz, R., Ben Avraham, E., Aberdam, A., Tsiper, S. & Litman, R. (2024). Gram : Global reasoning for multi-page vqa. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15598–15607.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. & Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.

Cao, J., Zhang, Y., Huang, T., Lu, M., Zhang, Q., An, R., Ma, N. & Zhang, S. (2025). MoVE-KD : Knowledge Distillation for VLMs with Mixture of Visual Encoders. *arXiv preprint arXiv :2501.01709*.

Chen, D., Mei, J.-P., Zhang, H., Wang, C., Feng, Y. & Chen, C. (2022a). Knowledge distillation with the reused teacher classifier. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11933–11942.

Chen, G., Choi, W., Yu, X., Han, T. & Chandraker, M. (2017). Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30.

Chen, P., Liu, S., Zhao, H. & Jia, J. (2021). Distilling knowledge via knowledge review. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5008–5017.

Chen, X., Cao, Q., Zhong, Y., Zhang, J., Gao, S. & Tao, D. (2022b). Dearkd : data-efficient early knowledge distillation for vision transformers. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12052–12062.

Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L. et al. (2024). Internvl : Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198.

Cho, J., Mahata, D., Irsoy, O., He, Y. & Bansal, M. (2024). M3docrag : Multi-modal retrieval is what you need for multi-page multi-document understanding. *arXiv preprint arXiv :2411.04952*.

Davis, B., Morse, B., Price, B., Tensmeyer, C., Wigington, C. & Morariu, V. (2022). End-to-end document recognition and understanding with dessurt. *European Conference on Computer Vision*, pp. 280–296.

Dettmers, T., Pagnoni, A., Holtzman, A. & Zettlemoyer, L. (2023). Qlora : Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36, 10088–10115.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2020). An image is worth 16x16 words : Transformers for image recognition at scale. *arXiv preprint arXiv :2010.11929*.

du Canada, G. (2025). Lignes directrices sur la numérisation. Repéré à https: //www.canada.ca/fr/bibliotheque-archives/services/gouvernement/information-disposition/documents/autorisations-disposition-pluri-institutionnelles/lignes-directrices-sur-numerisation.html.

Faghri, F., Pouransari, H., Mehta, S., Farajtabar, M., Farhadi, A., Rastegari, M. & Tuzel, O. (2023). Reinforce data, multiply impact : Improved model accuracy and robustness with dataset reinforcement. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17032–17043.

Fang, Z., Wang, J., Hu, X., Wang, L., Yang, Y. & Liu, Z. (2021). Compressing visual-linguistic model via knowledge distillation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1428–1438.

Faysse, M., Sibille, H., Wu, T., Omrani, B., Viaud, G., Hudelot, C. & Colombo, P. (2024). Colpali : Efficient document retrieval with vision language models. *arXiv preprint arXiv :2407.01449*.

Gao, Z., Chen, Z., Cui, E., Ren, Y., Wang, W., Zhu, J., Tian, H., Ye, S., He, J., Zhu, X. et al. (2024). Mini-InternVL : a flexible-transfer pocket multi-modal model with 5% parameters and 90% performance. *Visual Intelligence*, 2(1), 1–17.

Gou, J., Yu, B., Maybank, S. J. & Tao, D. (2021). Knowledge distillation : A survey. *International Journal of Computer Vision*, 129(6), 1789–1819.

Guardian, T. (2025). 'Spreadsheets of empire' : red tape goes back 4,000 years, say scientists after Iraq finds [Format]. Repéré à https://www.theguardian.com/science/2025/mar/15/stone-tablets-mesopotamia-iraq-red-tape-bureaucracy.

Han, S., Pool, J., Tran, J. & Dally, W. (2015). Learning both weights and connections for efficient neural network. 28.

Harley, A. W., Ufkes, A. & Derpanis, K. G. (2015). Evaluation of deep convolutional nets for document image classification and retrieval. *2015 13th international conference on document analysis and recognition (ICDAR)*, pp. 991–995.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hinton, G., Vinyals, O. & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv :1503.02531*.

Hu, A., Xu, H., Zhang, L., Ye, J., Yan, M., Zhang, J., Jin, Q., Huang, F. & Zhou, J. (2024). mplug-docowl2 : High-resolution compressing for ocr-free multi-page document understanding. *arXiv preprint arXiv :2409.03420*.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W. et al. (2022). Lora : Low-rank adaptation of large language models. *ICLR*, 1(2), 3.

Huang, D.-A., Radhakrishnan, S., Yu, Z. & Kautz, J. (2025). FRAG : Frame Selection Augmented Generation for Long Video and Long Document Understanding. *arXiv preprint arXiv :2504.17447*.

Huang, Y., Lv, T., Cui, L., Lu, Y. & Wei, F. (2022). Layoutlmv3 : Pre-training for document ai with unified text and image masking. *Proceedings of the 30th ACM international conference on multimedia*, pp. 4083–4091.

ImageNet. (2012). ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) [Format]. Repéré à https://www.image-net.org/challenges/LSVRC/2012/.

Johanne Roy, l. j. d. m. (2015). Des millions de dossiers à numériser au CHU de Québec. Repéré à https://www.journaldemontreal.com/2015/08/30/des-millions-de-dossiers-a-numeriser-au-chude-quebec.

Kang, L., Tito, R., Valveny, E. & Karatzas, D. (2024). Multi-page document visual question answering using self-attention scoring mechanism. *International Conference on Document Analysis and Recognition*, pp. 219–232.

Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D. & Park, S. (2022). Ocr-free document understanding transformer. *European Conference on Computer Vision*, pp. 498–517.

Kim, J., Park, S. & Kwak, N. (2018). Paraphrasing complex network : Network compression via factor transfer. *Advances in neural information processing systems*, 31.

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

Lcvenshtcin, V. (1966). Binary coors capable or ‘correcting deletions, insertions, and reversals. *Soviet physics-doklady*, 10(8).

LeCun, Y., Denker, J. & Solla, S. (1989). Optimal brain damage. *Advances in neural information processing systems*, 2.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

Li, Y., Si, S., Li, G., Hsieh, C.-J. & Bengio, S. (2021). Learnable fourier features for multi-dimensional spatial positional encoding. *Advances in Neural Information Processing Systems*, 34, 15816–15829.

Lin, S., Xie, H., Wang, B., Yu, K., Chang, X., Liang, X. & Wang, G. (2022). Knowledge distillation via the target-aware transformer. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10915–10924.

Liu, Y., Cao, J., Li, B., Hu, W., Ding, J. & Li, L. (2022). Cross-architecture knowledge distillation. *Proceedings of the Asian conference on computer vision*, pp. 3396–3411.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. & Guo, B. (2021). Swin transformer : Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022.

Mathew, M., Tito, R., Karatzas, D., Manmatha, R. & Jawahar, C. (2020a). Document visual question answering challenge 2020. *arXiv preprint arXiv :2008.08899*.

Mathew, M., Tito, R., Karatzas, D., Manmatha, R. & Jawahar, C. (2020b). Web page title. Repéré à https://rrc.cvc.uab.es/?ch=17&com=introduction.

Mathew, M., Karatzas, D. & Jawahar, C. (2021). Docvqa : A dataset for vqa on document images. *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2200–2209.

Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J. et al. (2024). Gemma : Open models based on gemini research and technology. *arXiv preprint arXiv :2403.08295*.

NG, A. [Nom d'usager]. (2017, Fevrier, 2). Andrew Ng : Artificial Intelligence is the New Electricity [Vidéo Youtube]. Repéré à https://www.youtube.com/watch?v=21EiKfQYZXc.

of Encyclopaedia Britannica, T. E. (2025). printing press [Format]. Repéré à https://www.britannica.com/technology/printing-press.

Parchami-Araghi, A., Böhle, M., Rao, S. & Schiele, B. (2024). Good teachers explain : Explanation-enhanced knowledge distillation. *European Conference on Computer Vision*, pp. 293–310.

Pfitzmann, B., Auer, C., Dolfi, M., Nassar, A. S. & Staar, P. (2022). Doclaynet : A large human-annotated dataset for document-layout segmentation. *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 3743–3751.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. & Sutskever, I. (2019). *Language Models are Unsupervised Multitask Learners*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. et al. (2021). Learning transferable visual models from natural language supervision. *International conference on machine learning*, pp. 8748–8763.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C. & Bengio, Y. (2014). Fitnets : Hints for thin deep nets. *arXiv preprint arXiv :1412.6550*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115, 211–252.

Sainte-Anne, U. (2025). Bases de données. Repéré à https://www.usainteanne.ca/bibliotheque/bases-de-donnees.

Shen, Z. & Xing, E. (2022). A fast knowledge distillation framework for visual recognition. *European conference on computer vision*, pp. 673–690.

Tang, Z., Yang, Z., Wang, G., Fang, Y., Liu, Y., Zhu, C., Zeng, M., Zhang, C. & Bansal, M. (2023). Unifying vision, text, and layout for universal document processing. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19254–19264.

Tao, C., Su, S., Zhu, X., Zhang, C., Chen, Z., Liu, J., Wang, W., Lu, L., Huang, G., Qiao, Y. et al. (2025). HoVLE : Unleashing the Power of Monolithic Vision-Language Models with Holistic Vision-Language Embedding. *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14559–14569.

Tito, R., Karatzas, D. & Valveny, E. (2023). Hierarchical multimodal transformers for multipage docvqa. *Pattern Recognition*, 144, 109834.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A. & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. *International conference on machine learning*, pp. 10347–10357.

Van Landeghem, J., Maity, S., Banerjee, A., Blaschko, M., Moens, M.-F., Lladós, J. & Biswas, S. (2024). DistilDoc : Knowledge distillation for visually-rich document applications. *International Conference on Document Analysis and Recognition*, pp. 195–217.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W. et al. (2024). Qwen2-vl : Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv :2409.12191*.

Wang, T., Zhou, W., Zeng, Y. & Zhang, X. (2022). Efficientvlm : Fast and accurate vision-language models via knowledge distillation and modal-adaptive pruning. *arXiv preprint arXiv :2210.07795*.

Wu, K., Zhang, J., Peng, H., Liu, M., Xiao, B., Fu, J. & Yuan, L. (2022). Tinyvit : Fast pretraining distillation for small vision transformers. *European conference on computer vision*, pp. 68–85.

Wu, Z., Chen, X., Pan, Z., Liu, X., Liu, W., Dai, D., Gao, H., Ma, Y., Wu, C., Wang, B. et al. (2024). Deepseek-vl2 : Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv :2412.10302*.

Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W. et al. (2020). Layoutlmv2 : Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv :2012.14740*.

Yang, C., An, Z., Huang, L., Bi, J., Yu, X., Yang, H., Diao, B. & Xu, Y. (2024a). Clip-kd : An empirical study of clip model distillation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15952–15962.

Yang, H., Yin, H., Shen, M., Molchanov, P., Li, H. & Kautz, J. (2023). Global vision transformer pruning with hessian-aware saliency. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18547–18557.

Yang, Z., Li, Z., Zeng, A., Li, Z., Yuan, C. & Li, Y. (2024b). ViTKD : Feature-based knowledge distillation for vision transformers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1379–1388.

Zhai, X., Mustafa, B., Kolesnikov, A. & Beyer, L. (2023). Sigmoid loss for language image pre-training. *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986.

Zhang, J., Peng, H., Wu, K., Liu, M., Xiao, B., Fu, J. & Yuan, L. (2022). Minivit : Compressing vision transformers with weight multiplexing. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12145–12154.

Zhang, L., Bao, C. & Ma, K. (2021). Self-distillation : Towards efficient and compact neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8), 4388–4403.

Zhang, Y., Xiang, T., Hospedales, T. M. & Lu, H. (2018). Deep mutual learning. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4320–4328.