

MEMOIRE

présenté à

L'Institut Supérieur d'Informatique de Mahdia

En vue de l'obtention

**Du diplôme de Mastère Professionnel en :
Technologies de Sciences des Données**

Par

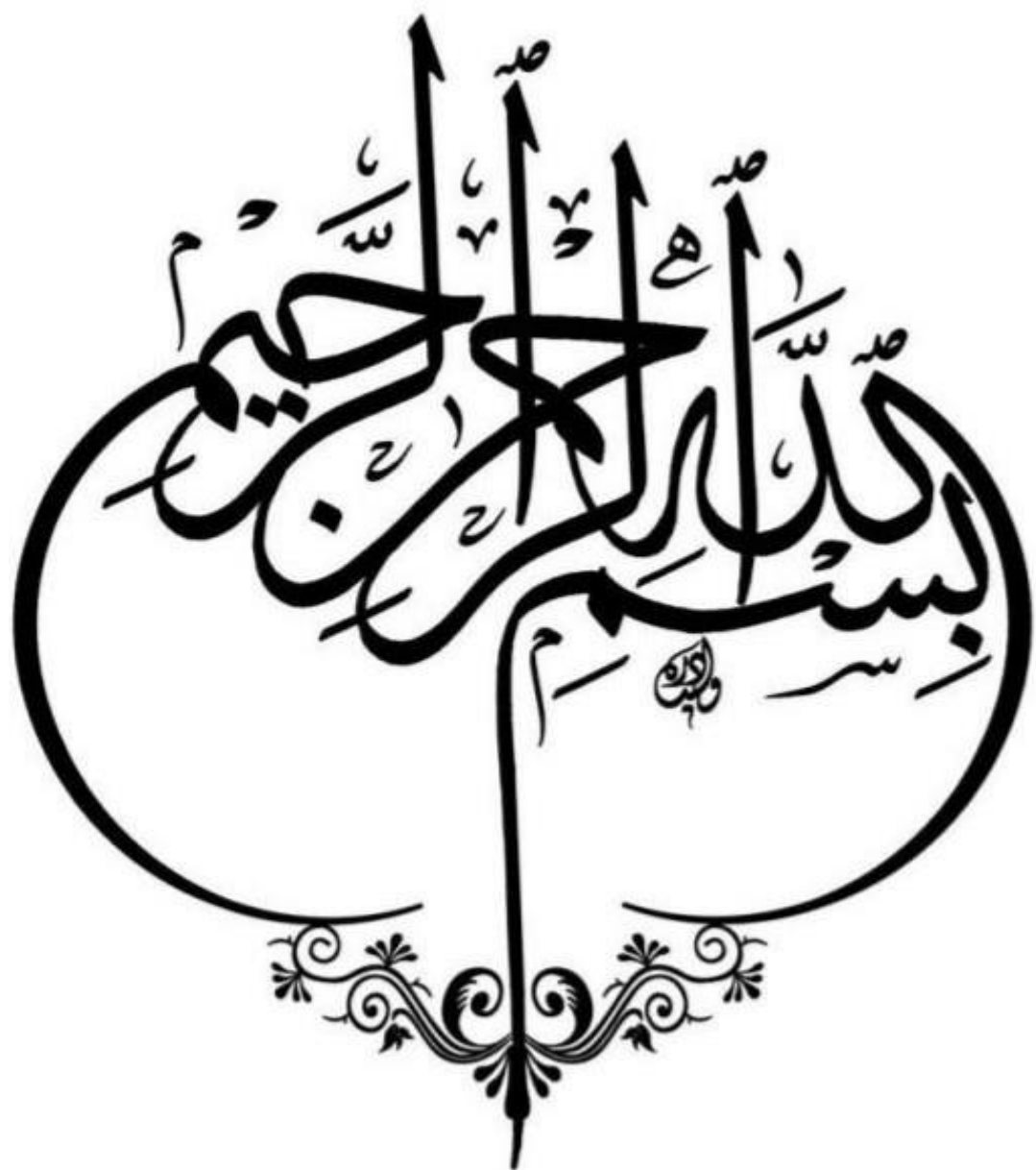
Sihem Jouini

Multimodal medical Visual Question Answering(VQA)

Soutenu le 11/10/ 2022 devant la commission d'examen :

Mr.	Président
Mr.	Jaafer Chaaouri	Encadrant
Mr.	Examineur

Année universitaire 2021/2022





DÉDICACE

A ma chère maman et mon cher papa
Pour tous les sacrifices que vous avez consentis, pour toutes les prières que vous
M'aviez
Faites, pour tout l'amour, l'affection, le soutien et l'encouragement que vous m'aviez
Toujours apportés tout au long de mes études pour faire de moi ce que je suis
Aujourd'hui.
Je vous dédie ce travail en signe de mon éternel attachement et de mon amour.
À ma sœur,
À mon cher petit frère
Vous étiez toujours la source de ma motivation. Tous mes remerciements Ne
suffisent.
A toute la famille **Jouini**
A tous mes amis de quartier A tous ceux qui ont su m'apporter aide et soutien aux
Moments propices.
A toute la promotion ISIMA 2022.
A tous mes amis pour les agréables moments passés ensemble, pour le soutien moral et
Pour la noblesse de vos actes.

Sihem



REMERCIEMENT

Avant tout, je tenais à remercier **Dieu** qui éclaire toujours mes horizons et me donne le courage et la passion pour atteindre mes objectifs.

Avec le mot d'appréciation et de respect, je remercie infiniment mon encadrant à l'ISIMA M. Jaafer CHAAOURI qui a accepté de me superviser et de me guider avec ses conseils précieux tout au long de la réalisation de ce travail.

Mes grands remerciements s'adressent aussi à tout le cadre éducatif de l'Institut Supérieure d'Informatique de Mahdia. je tiens, au terme de ce travail, à présenter mes vifs remerciements à toutes les personnes qui ont contribué, de près ou loin, à son bon déroulement.

Enfin je remercie tous les membres du jury d'avoir accepté d'évaluer ce travail.

Sihem

TABLE DES MATIERES

INTRODUCTION GENERALE	IX
1 CADRE GÉNÉRAL DU PROJET	1
1.1 Introduction	2
1.2 Contexte du projet	2
1.3 Problématique traitée	3
1.4 Conclusion.....	5
2 ANALYSE DE L'IMAGE MEDICALE	6
2.1 Introduction	7
2.2 Principe de l'imagerie médicale	7
2.3 Modalités d'imagerie médicale	7
2.3.1 Imagerie par Résonance Magnétique (IRM)	7
2.3.2 L'échographie ultrasonore.....	8
2.3.3 Image Rayon X.....	9
2.3.4 Endoscopie	9
2.3.5 La scintigraphie	10
2.4 Intérêt de l'imagerie médicale dans le processus de diagnostic médicale	10
2.5 Datasets pour le VQA médicale	11
2.5.1 VQA-MED-2018.....	11
2.5.2 VQA-RAD.....	12
2.5.3 VQA-MED-2019	12
2.5.4 pathVQA	13
2.5.5 VQA-Med-2020	14
2.6 Synthèse	15
2.7 Conclusion.....	15
3 ETAT DE L'ART SUR L'APPRENTISSAGE PROFOND(TRANFORMER).....	16
3.1 Introduction	17
3.2 Modèle Transformer.....	17
3.2.1 Fonctionnement et architecture	17
3.2.2 Notion d'attention :	19

3.2.3	Attention multi-têtes :	23
3.2.4	Encodage positionnel :	24
3.3	Le modèle « Transformer » pour le traitement du langage naturel	25
3.3.1	Qu'est-ce que le traitement du langage naturel	25
3.3.2	Variants du modèle« Transformer » pour les taches NLP	26
3.3.3	Bert	26
3.4	Le modèle « Transformer » pour la vision par ordinateur	28
3.4.1	Vision par ordinateur	28
3.4.2	Vision Transformer(ViT)	28
3.5	Conclusion.....	29
4	IMPLEMENTATION ET EVALUATION	30
4.1	Introduction	31
4.2	Environnement de travail	31
4.2.1	Environnement du développement	31
4.2.2	Installation des bibliothèques	32
4.3	Données d'expérimentations	34
4.3.1	Préparation et exploration des données	34
4.3.2	Critères d'évaluation :	37
4.4	Implémentation	39
4.4.1	implémentation du modèle BERT+ ViT	39
4.4.1.1	Architecture du modèle	39
4.4.1.2	Déploiement BERT+ViT	40
4.4.1.3.1	Accuracy et loss du modèle	44
4.4.1.3.2	Evaluation :	45
4.4.1.3.3	Résultats de VQA(inférence) :	46
4.4.2	Implémentation de BioBERT+ViT	49
4.4.2.1	Résultats expérimentaux	49
4.4.2.1.1	Accuracy et loss du modèle	49
4.4.2.1.1.1	Evaluation	50
4.1.2	Implémentation de BioBERT+Swin Transformer	51
4.1.2.1.1	Accuracy et loss du modèle	53
4.1.2.1.2	Evaluation	53
4.2	Comparaison des résultats	53
4.3	Conclusion.....	55
	CONCLUSION GENERALE	56
	Bibliographie & Webographie	58



LISTE DES FIGURES

Figure 1: Exemple de réponse au question visuel(VQA).....	2
Figure 2: Illustration du modèle de réponse au question visuel dans le domaine médicale(VQA)	4
Figure 3: Imagerie par Résonance Magnétique (IRM)	8
Figure 4: Echographie[5]	8
Figure 5: Image médicale rayons X[6]	9
Figure 6: Endoscopie	9
Figure 7: Scintigraphie	10
Figure 8: Echantillons d'images de l'ensemble de données VQA-MED-2018.....	12
Figure 9: Echantillons d'images de l'ensemble de données VQA-RAD.....	12
Figure 10: Echantillons d'images de l'ensemble de données VQA-MED-2019.....	13
Figure 11: Echantillons d'images de l'ensemble de données pathVQA	14
Figure 12: Echantillons d'images de l'ensemble de données VQA-MED-2020.....	14
Figure 13: Architecture du transformer	18
Figure 14: Décodeur du transformer[15].....	19
Figure 15: Vecteur de plongement de mots	19
Figure 16: Le vecteur de Plongements de mots traverse les couches d'encodeurs.....	20
Figure 17: vecteurs de poids en entrée	20
Figure 18: Calcule du score pour l'auto attention.....	21
Figure 19: Application de la fonctionnement softmax.....	22
Figure 20: Sortie Z de l'auto attention	22
Figure 21: Attention à multiple têtes.....	23
Figure 22: Calcule de l'auto attention dans huit têtes d'attention différentes.....	24
Figure 23:Concaténation de toutes les têtes d'attention et multiplication par la matrice de poids W_o	24
Figure 24: Encodage positionnel.....	25
Figure 25: Illustration du modelé pré-entraîné BERT	27
Figure 26: Fonctionnement du ViT [19]	29
Figure 27: Illustration du google colab	31
Figure 28: Logo du Pytorch	32
Figure 29: Logo de Sklearn.....	33

Figure 30: Logo du PIL	33
Figure 31: Logo du Numpy	33
Figure 32: Logo de la bibliothèque matplotlib	34
Figure 33: Importation des bibliothèques.....	35
Figure 34: Répartition des images et des paires Question-Réponse dans l'ensemble de données VQA-Med-2019	35
Figure 35: Distribution de nombre de mots par question	35
Figure 36: Fonction pour l'affichage d'un exemple aléatoire	36
Figure 37: Un échantillon d'une combinaison image/question/réponse	36
Figure 38: Création de l'espace de réponse	37
Figure 39: Structure du fichier "Answer Space"	37
Figure 40: Une représentation de l'architecture du modèle BERT-ViT.....	40
Figure 41: Portion de code d'une étape de tokenisation avec le modèle BERT	41
Figure 42: Portion de code d'une étape de préprocessing avec le modèle ViT.....	42
Figure 43: Portion de code de l'étape de fusion.....	43
Figure 44: Portion de code de construction de classifieur	43
Figure 45: Portion de code du modèle multimodale BERT-ViT	44
Figure 46: Tracé des indices de performance et de perte relatifs au modèle Bert +Vit	45
Figure 47: Un exemple de VQA en appliquant le modèle BERT-ViT	49
Figure 48: Tracé des indices de performance et de perte relatifs au modèle BioBERT+ViT	50
Figure 49: Différence entre ViT et Swin Transformer	52
Figure 50: Architecture de Swin transformer	52
Figure 51: Tracé des indices de performance et de perte relatifs au modèle BioBERT+Swin Transformer	53
Figure 52: Les résultats officiels de ImageCLEF@VQA-Med-2019	54



LISTE DES TABLEAUX

Tableau 1: Performance de Bert+Vit en termes des indices Wups, Accuracy, Précision, Recall, Loss	
.....	45
Tableau 2: Performance de BioBERT+Vit en termes des indices Wups, Accuracy, Precision, Recall, Loss	
.....	50
Tableau 3: Performance de BioBERT+Swin Transformer en termes des indices Wups, Accuracy, Precision, Recall, Loss	
.....	53
Tableau 4: Tableau comparatif des performances des trois modèles implémentés	
.....	54
Tableau 5: VQA-Med-2019 challenge accuracy	
.....	55



LISTE DES ACRONYMES

ADAM : ADaptive Moment estimation

BERT: Bidirectional Encoder Representations from Transformers

BioBERT : Bidirectional Encoder Representations from Transformers for Biomedical Text Mining

CV: Computer Vision

IA : Intelligence Artificielle

GPU: Graphics Processor Unit

IRM : L'imagerie par résonance magnétique

ML : Machine learning

MLP : Multi-Layer Perceptron

NLP : Naturel Language Processing

Swin : Shifted window Transformer

ViT : Vision Transformer

VQA : Visual Question Answering

WUPS : Wu et Palmer



INTRODUCTION GENERALE

De nos jours, les domaines d'application des algorithmes d'apprentissage automatique n'ont pas de limites en raison de la focalisation de la recherche scientifique et de son grand impact sur le confort dans notre vie quotidienne. Fondamentalement, des modèles statistiques récentes ont été déployées avec succès afin d'automatiser le traitement de la quantité croissante d'images, de vidéos et de textes. En particulier, les réseaux de neurones profonds, ont été adopté par les communautés de la vision par ordinateur (CV) et du traitement du langage naturel (NLP) grâce à leurs capacités à interpréter le contenu des images et des textes une fois entraînés sur de grands ensembles de données.

L'intégration du langage dans la reconnaissance visuelle a contribué à l'émergence des nouvelles applications multimodales telles que la réponse à des questions visuelles (VQA)[\[1\]](#), qui consiste à répondre à une question spécifique sur une image donnée. Il est donc nécessaire de combiner des techniques de CV qui permettent de comprendre le contenu de l'image avec des techniques de NLP qui permettent de comprendre la question et de produire la réponse. Récemment, le VQA a été appliquée à différents domaines spécifiques tels que le domaine médical. Il permet ainsi de concevoir des systèmes plus préventifs et personnalisés, apportant une amélioration du suivi médical des patients en répondant à leurs questions sans la nécessité d'une visite spéciale chez le médecin.

Dans ce cadre, nous avons proposé de comparer et d'évaluer les modèles d'apprentissage multimodal profond basé sur le modèle « Transformer » capable de répondre à des questions en langage naturel à partir du contenu visuel des images de radiologie associées, sans nécessiter d'inférence ou de contexte supplémentaire spécifique au domaine.

Le travail a été réalisé en deux parties ; une première partie théorique et une deuxième partie qui vise à l'implémentation et la réalisation concrète du projet. Ces deux parties seront préparées sur quatre chapitres: Le premier chapitre intitulé « cadre général du projet » est consacré à la présentation générale du projet, son contexte ainsi que la problématique traitée. Dans le deuxième chapitre, nous nous intéressons à l'imagerie médicale, son principe et son intérêt dans le processus de diagnostic médical, nous mettons en exergue les modalités d'imagerie ainsi

qu'un aperçu sur les différents datasets disponibles traitants le problème de réponse aux questions visuelles à partir des images médicales. Dans le troisième chapitre, nous détaillons le modèle « transformer » comme étant un sujet d'actualité et un état de l'art en apprentissage profond, nous décrivons son architecture, son fonctionnement et ses variants dans le domaine du traitement de langage naturel ainsi que de la vision par ordinateur. Le quatrième chapitre est consacré à l'implémentation de notre modèle VQA, nous présentons les bibliothèques utilisées dans notre projet, les données d'expérimentations ainsi qu'une étude comparative expérimentale des différents modèles utilisés en indiquant les résultats obtenus. Nous clôturons le rapport par une conclusion générale dans laquelle nous présentons une synthèse du travail réalisé ainsi que les éventuelles perspectives du projet.

CADRE GÉNÉRAL DU PROJET

Sommaire

1.1	Introduction	2
1.2	Contexte du projet.....	2
1.3	Problématique traitée	3
1.4	Conclusion.....	5

1.1 Introduction

Le premier chapitre est consacré à la présentation du cadre générale du projet. Dans un premier lieu, nous allons définir le contexte du sujet passant par la suite à la problématique traitée et la solution proposée.

1.2 Contexte du projet

Notre perception est par nature multimodale. Pour résoudre certaines tâches, il est donc pertinent d'utiliser et de prendre en compte les différentes modalités disponibles dans les données. Un sujet relativement nouvel d'apprentissage profond et où l'usage de différentes modalités peut se révéler très pertinent est la réponse aux questions visuels VQA(Visual Question Answering en anglais). Une tâche typique de VQA possède deux entrées : une image et une question en langage naturel sur cette image. Le système tente alors de répondre à la question en se basant sur le contenu de l'image.

La figure ci-dessous présente une illustration d'un système VQA :

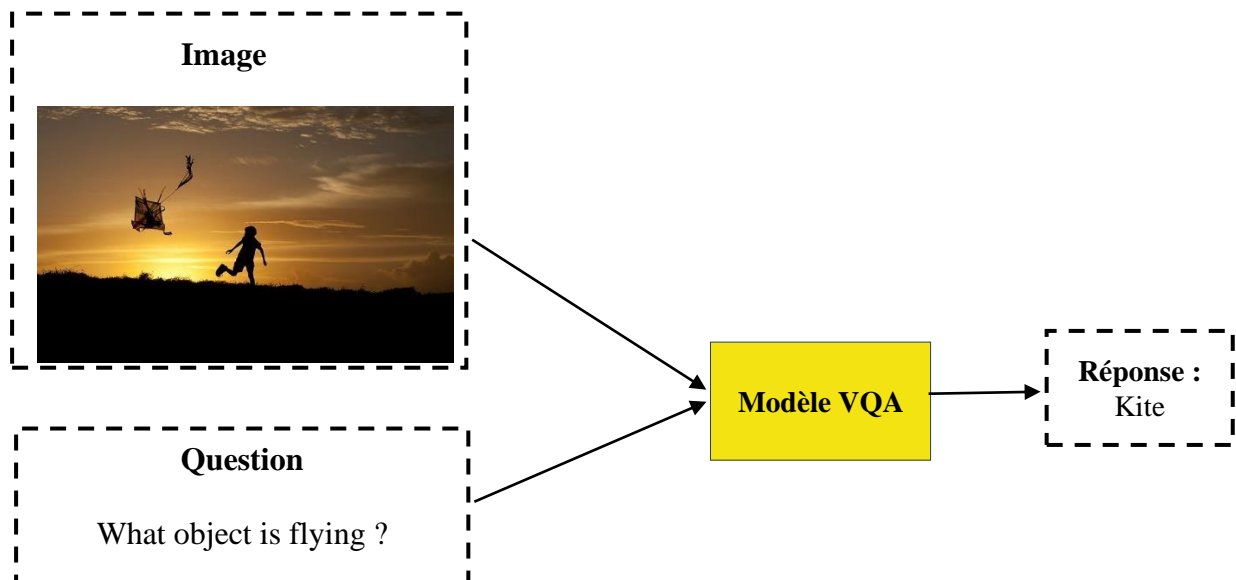


Figure 1: Exemple de réponse au question visuel(VQA)

Le domaine de la VQA se situe donc à la frontière entre la vision par ordinateur (CV) et le traitement du langage naturel (NLP). En effet, le véritable défi du VQA ne consiste pas simplement à utiliser les principes des deux domaines, mais à les combiner. Ce qui est difficile puisque les disciplines du CV et du NLP utilisent historiquement des méthodes et des

modèles différents pour résoudre leurs tâches respectives. Les systèmes VQA sont généralement posés comme un problème d'apprentissage automatique supervisé, ce qui signifie qu'ils nécessitent un ensemble de données étiquetées où chaque échantillon de données doit contenir une image, une question sur cette image et une réponse correcte à cette question.

Domaines d'applications du VQA :

La réponse aux questions visuelles a un large éventail d'applications pratiques possibles ; elle pourrait être utilisée pour :

➤ Une aide pour fournir des informations visuelles aux malvoyants

L'assistance aux personnes aveugles fait partie des objectifs de plusieurs applications VQA proposées ces dernières années. Ceci est principalement dû à la capacité du VQA à répondre aux questions quotidiennes qui peuvent aider les personnes malvoyantes à vivre sans barrières visuelles.

➤ éducation et patrimoine culturel

L'un des principaux aspects du VQA est sa forte corrélation avec la perception humaine. Bien que l'attention d'un système VQA puisse se concentrer sur des parties d'une image différentes de celles des humains, un robot éducatif peut être développé et testé en utilisant une architecture VQA pour formuler des questions et entamer un dialogue éducatif.

➤ Dans les scénarios de vidéosurveillance

L'adoption d'une approche du VQA dans les scénarios de vidéosurveillance peut aider les opérateurs à améliorer la compréhension d'une scène, ce qui les aide à prendre des décisions justes et rapides.

➤ un système d'aide au diagnostic médical

Le VQA peut également être utilisé dans un système d'aide au diagnostic médical. Ce sujet ouvre de nouveaux scénarios pour aider le personnel médical à prendre des décisions cliniques, ainsi que pour améliorer le diagnostic par le biais d'un "deuxième avis" informatisé.

1.3 Problématique traitée

Avec l'intérêt croissant de l'intelligence artificielle (IA) pour soutenir la prise de décision médical, la confiance des médecins dans l'interprétation d'images médicales complexes (Radiographie, IRM, Angiographie, Échographie, radiologie diagnostique) peut être considérablement renforcée par une « deuxième opinion » fournie par un système automatisé.

De plus, les patients peuvent être intéressés par la morphologie/physiologie et l'état pathologique des structures anatomiques autour d'une lésion qui a été bien caractérisée par leurs prestataires de soins de santé. Bien que les patients se tournent souvent vers les moteurs de recherche pour lever l'ambiguïté de termes complexes ou obtenir des réponses à des aspects relatifs à une image médicale, les résultats des moteurs de recherche peuvent être non spécifiques et erronés. En revanche, une VQA médicale peut être intégrée dans un système de consultation en ligne afin de fournir des réponses fiables à tout moment.

Le système de VQA médical devrait aider à la prise de décision clinique et améliorer l'engagement du patient. Contrairement à d'autres applications d'IA médicale souvent limitées à des maladies ou des types d'organes prédéfinis, le VQA médicale peut comprendre des questions de forme libre en langage naturel et fournir des réponses fiables et conviviale.

La figure ci-dessous présente une illustration de système VQA dans le domaine médical :

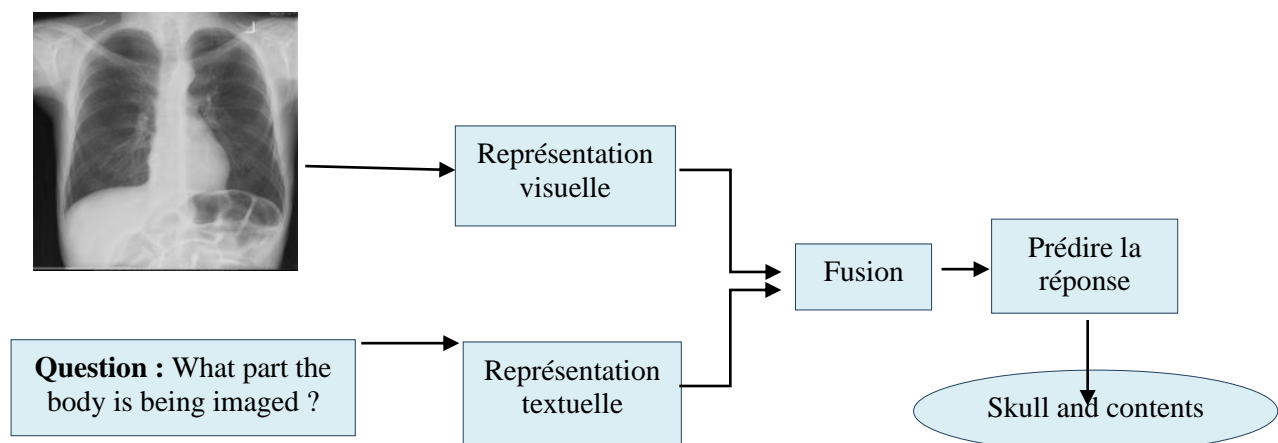


Figure 2: Illustration du modèle de réponse au question visuel dans le domaine médicale(VQA)

Fusion entre modalités :

Un facteur clé de la résolution d'un problème multimodal réside dans la fusion des différentes modalités. Cette étape sert à représenter conjointement les caractéristiques des différentes modalités prenant en entrée les caractéristiques du texte et d'image et fournit en sortie les caractéristiques multimodales fusionnées.

Type de fusion : c'est le niveau auquel la fusion a lieu entre les différentes modalités. Ceci correspond donc à l'architecture de fusion proposée. Il existe deux grands types d'architecture de fusion de données à distinguer : la fusion précoce, ou "fusion niveau caractéristiques" et la fusion tardive, ou "fusion niveau décision «On décrit ici très brièvement ces deux genres de fusion.

- **Fusion Tardive :** C'est la méthode de fusion la plus simple dans lequel les caractéristiques des différentes modalités sont extraites indépendamment de chaque source à l'étape initiale, et qu'elles sont fusionnées à une étape ultérieure. Leurs concaténations seront introduites dans des couches entièrement connectées pour prédire les réponses.
- **Fusion précoce:** Cette stratégie fusionne les données en entrée. Ici les caractéristiques pré-extraites sont combinées à partir de différentes sources et empilées pour construire un nouveau vecteur de caractéristiques unique . Il existe une grande variété de méthodes de fusion précoce, souvent adaptée pour un problème spécifique. Il est alors très difficile de savoir quelle méthode adopter face à un nouveau problème.

1.4 Conclusion

Dans ce chapitre, nous avons positionné le projet dans son contexte général. Dans un premier lieu, nous avons défini le problème de multimodalité comme étant un sujet d'actualité qui combine la vision par ordinateur et le traitement du langage naturel à la fois. Nous introduisons aux systèmes de réponse aux questions visuels , ses modes de fonctionnement et ses domaines d'application, En particulier, notre intérêt s'est porté sur le VQA dans le domaine médical que nous allons traiter dans ce projet. D'avantage, nous avons présenté l'étape de la fusion entre modalités, en mentionnant ses types.

Le chapitre suivant va comporter une étude de l'imagerie médicale et une généralité sur les datasets à exploré pour la tâche VQA médicale.

ANALYSE DE L'IMAGE MEDICALE

Sommaire

2.1	Introduction	7
2.2	Principe de l'imagerie médicale	7
2.3	Modalités d'imagerie médicale	7
2.3.1	Imagerie par Résonance Magnétique (IRM)	7
2.3.2	L'échographie ultrasonore	8
2.3.3	Image Rayon X	9
2.3.4	Endoscopie	9
2.3.5	La scintigraphie	10
2.4	Intérêt de l'imagerie médicale dans le processus de diagnostic médicale.....	10
2.5	Datasets pour le VQA médicale	11
2.5.1	VQA-MED-2018	11
2.5.2	VQA-RAD	12
2.5.3	VQA-MED-2019	12
2.5.4	pathVQA	13
2.5.5	VQA-Med-2020	14
2.6	Synthèse.....	15
2.7	Conclusion.....	15

2.1 Introduction

Ce chapitre est dédié en premier lieu à la présentation l'imagerie médicale ,son principe, ses modalités et son intérêt dans le processus de diagnostique médicale. La deuxième partie est consacrée à la présentation et à la comparaison des différents datasets disponibles qui traitent le problème de réponse aux questions visuelles à partir des images médicales.

2.2 Principe de l'imagerie médicale

L'imagerie médicale est un outil de diagnostic très puissant et largement utilisée dans la recherche. Son principal objectif est d'obtenir une image révélant un ensemble d'informations précises sur le fonctionnement d'un organe comme le cœur et ses artères ou tout autre organe nécessitant un diagnostic ou une approche chirurgicale.

L'image obtenue d'un organe ou d'un tissu est analysé directement par visionnement sur un écran. Dans d'autres cas, l'imagerie médicale utilise un film montrant les mouvements simultanés d'un organe ou une image en quantité d'où on pourra mesurer certaines valeurs biologiques comme la quantité de sang présent dans un organe malade. L'imagerie médicale compile plusieurs données et informations grâce à ses différentes techniques et en fait une gestion informatique exhaustive dans sa pratique[2].

2.3 Modalités d'imagerie médicale

2.3.1 Imagerie par Résonance Magnétique (IRM)

L'imagerie par résonance magnétique (IRM) est issue d'une série de découvertes scientifiques faites tout au long du 20ème siècle. Elle offre une gamme variée d'examens permettant de caractériser la fibrose et les tumeurs hépatiques par différentes modalités. l'IRM exploite les propriétés magnétiques des tissus, basée sur les propriétés magnétiques des noyaux atomiques, généralement des protons, qui sont abondants dans les tissus biologiques riches en eau et en graisses. Sur les tissus mous, elle facilite les contrastes qui peuvent être modulés grâce à l'utilisation de séquences spécifiques[3].

La figure ci-dessous présente un exemple d'une imagerie par résonance magnétique :

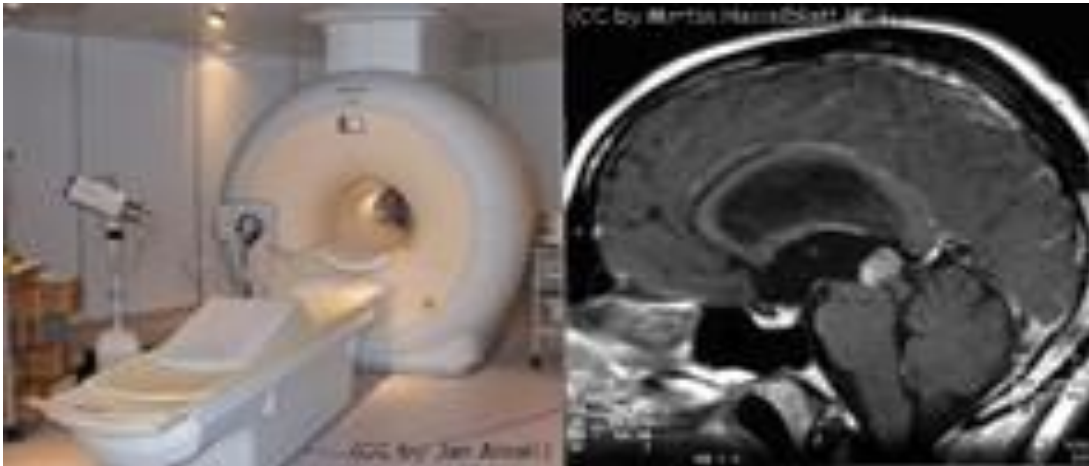


Figure 3: Imagerie par Résonance Magnétique (IRM)

2.3.2 L'échographie ultrasonore

L'échographie ultrasonore est une modalité d'imagerie médicale qui repose sur l'exposition des tissus aux ondes ultrasonores et sur la réception de leur écho. L'échographe se compose d'un écran et d'une sonde émettrice et réceptrice des ondes (appelée transducteur), soumises à un courant électrique, les micro-céramiques à la surface de la sonde vibrent et émettent des ondes ultrasonores. Ces ondes traversent les tissus et y font écho différemment selon leur densité : plus un tissu est dense, plus l'écho est important. Les ondes reviennent au niveau de la sonde, font vibrer les céramiques qui induisent un courant électrique traité par informatique[4].

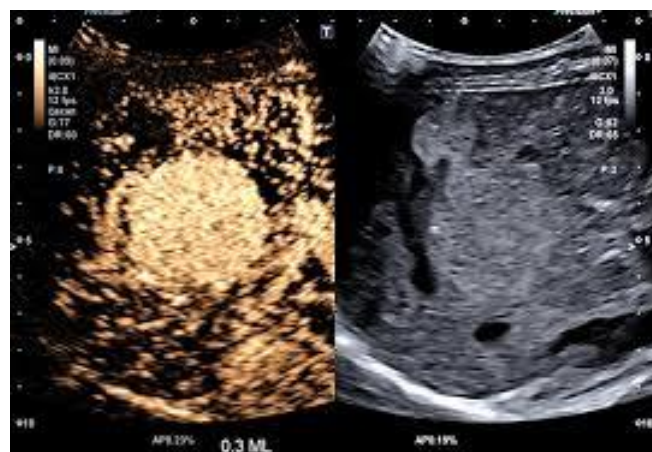


Figure 4: Echographie[5]

2.3.3 Image Rayon X

Les rayons X sont une forme de rayonnement électromagnétique, au même titre que la lumière visible, l'ultraviolet, l'infrarouge, les micro-ondes, les ondes radio ou les rayons gamma[4].

Les premières versions de tubes à rayons X ont été créées au début du XXème siècle et depuis un siècle, le principe physique régissant leur fonctionnement reste le même. Cependant, les matériaux et la technologie ont largement évolué et permettent des régimes de fonctionnement plus soutenus, ainsi que de meilleurs rendements de conversion énergétique.



Figure 5: Image médicale rayons X[6]

2.3.4 Endoscopie

L'endoscopie consiste à introduire une caméra (ou "endoscope") dans un conduit ou une cavité de l'organisme. Cette technique est le plus souvent utilisée pour rechercher visuellement la cause d'un trouble.



Figure 6: Endoscopie

2.3.5 La scintigraphie

La scintigraphie ou tomographie est une technique d'imagerie qui fait intervenir la médecine nucléaire. Un médicament radiopharmaceutique (isotopes radioactifs d'une molécule) est administré et les rayonnements sont analysés, une fois que le produit a été capté dans l'organe cible.

Cette technique permet d'obtenir une image fonctionnelle des organes ainsi que la présence d'éléments anormaux.

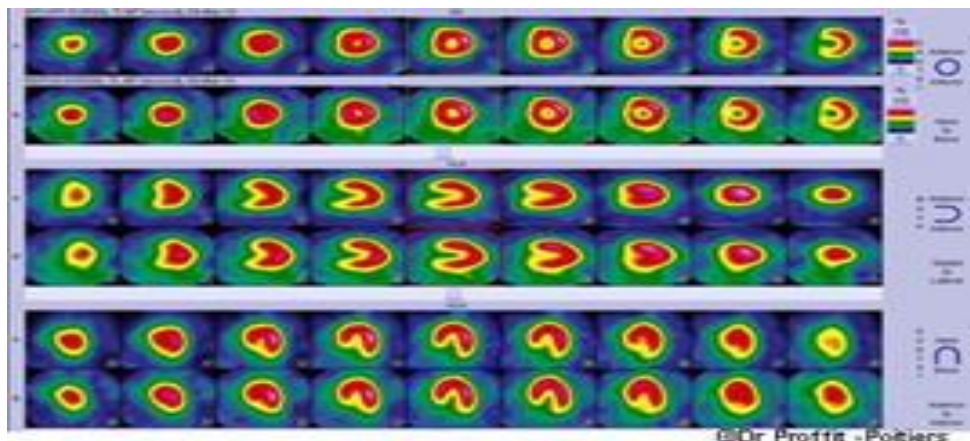


Figure 7: Scintigraphie

2.4 Intérêt de l'imagerie médicale dans le processus de diagnostic médicale

Pour le diagnostic de nombreuses maladies ainsi que pour les examens cliniques, nous utilisons principalement l'imagerie médicale. L'objectif de l'utilisation de l'imagerie peut être défini comme suit :

- **L'aide à l'intervention :** Chez les patients et sous échographie des ponctions sont effectuées pour bien visualiser la zone à prélever, essentiellement lorsqu'elle n'est pas concrète.
- **L'aide à la prise en charge et au suivi thérapeutique :** Il est possible de suivre le développement des maladies ou des fractures osseuses à travers la comparaison des phototypes prises au cours de temps et dans différents stades. Grâce à la grande utilisation de la scintigraphie en cancérologie, la vérification de l'efficacité du traitement en visionnant l'activité des cellules cancéreuses ou bien deviner des métastases. D'autre part, le support médical peut être un outil pour la modification d'attitude thérapeutique, au bénéfice des patients.

- **L'aide en médecine légale :** La radiologie est devenue une technique essentielle dans un certain nombre de situations spécifiques. Par exemple, dans l'évaluation des dommages corporels, mais aussi dans l'explication de certains décès traumatiques, qu'ils soient accidentels ou par arme à feu.
- **L'amélioration des connaissances :** La connaissance de l'activité cérébrale chez l'homme a été largement développée grâce à l'imagerie[7].

2.5 Datasets pour le VQA médicale

Le VQA médicale est techniquement plus difficile que le VQA du domaine général dans la mesure où la création d'un ensemble de données VQA médicale à grande échelle pose un défi, car l'annotation par des experts est coûteuse en raison des exigences élevées en matière de connaissances professionnelles, et que les paires de question-réponse ne peuvent pas être générées directement à partir d'images.

Dans cette partie nous allons discuter les ensembles de données médicaux pour la tâche VQA accessibles au public, en ce qui concerne la source des données, la quantité de données et les caractéristiques des tâches :

il existe 5 ensembles de données VQA médicaux disponibles à ce jour : VQA-MED-2018 ,VQA-RAD, VQA-MED-2019,PathVQA , VQA-MED-2020[8].

2.5.1 VQA-MED-2018

VQA-Med-2018 est un jeu de données proposé par le groupe ImageCLEF, et c'était le premier jeu de données publié dans le domaine médical. Les paires de QR ont été générées à partir des légendes en utilisant une approche semi-automatique. Tout d'abord, un système de génération de questions (QG) basé sur des règles a généré automatiquement des paires de question-réponse possibles en simplifiant les phrases, en identifiant les phrases de réponse, en générant des questions et en classant les questions candidates. Ensuite, deux annotateurs humains experts (dont un expert en médecine clinique) ont vérifié manuellement toutes les paires de question-réponse générées en deux passages. Respectivement, un passage assure l'exactitude sémantique, et un autre passage assure la pertinence clinique des images médicales associées[9].

Cet ensemble de données possède environ 6413 paires question-réponse avec 2866 images associés.

La figure ci-dessous montre quelques exemples d'images de l'ensemble de données VQA-MED-2018.

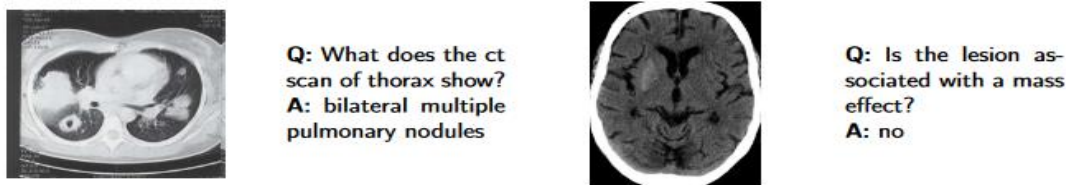


Figure 8: Echantillons d'images de l'ensemble de données VQA-MED-2018

2.5.2 VQA-RAD

VQA-RAD est un jeu de données spécifique à la radiologie proposé en 2018. L'ensemble d'images est équilibré et contient des échantillons de la tête, de la poitrine et du ventre provenant du dataset MedPix5.

Pour étudier la question dans une scène réaliste, l'auteur a présenté les images à des cliniciens pour recueillir des questions non guidées. Les cliniciens doivent produire des questions dans des structures libres et des structures de modèles. Ensuite, les paires d'AQ sont validées et classées manuellement pour analyser l'orientation clinique. Les types de réponses sont soit des questions fermées, soit des questions ouvertes. Bien qu'il ne s'agisse pas d'une grande quantité, l'ensemble de données VQA-RAD a permis d'acquérir des informations essentielles sur ce qu'un système VQA devrait être capable de répondre en tant que radiologue IA[10].

Cet ensemble de données possède environ 3515 paires question-réponse avec 315 images associés.

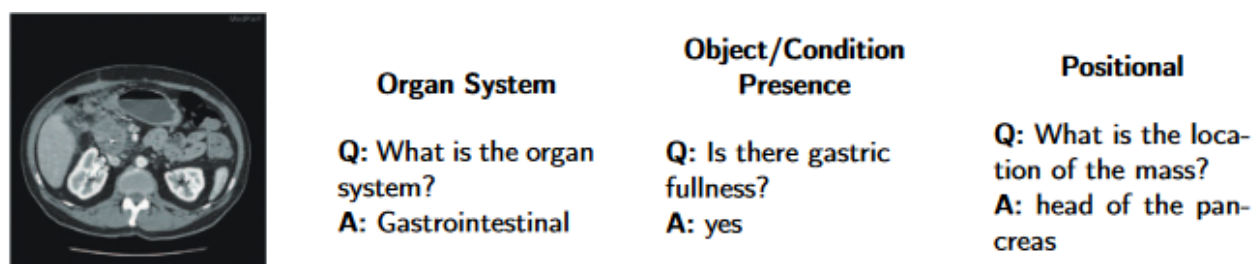


Figure 9: Echantillons d'images de l'ensemble de données VQA-RAD

2.5.3 VQA-MED-2019

VQA-Med-2019, est la deuxième édition du VQA-Med et a été publié pendant le défi ImageCLEF 2019. Inspiré par le VQA-RAD, VQA-Med-2019 a abordé les quatre catégories de questions les plus fréquentes : modalité, plan, système d'organes et anomalie. Pour chaque

catégorie, les questions suivent les modèles de centaines de questions naturellement posées et validées dans le cadre de VQA-RAD . Les trois premières catégories (modalité, plan et système organique) peuvent être abordées comme des tâches de classification, tandis que la quatrième catégorie (anomalie) présente un problème de génération de réponses[11].

Cet ensemble de données possède environ 15292 paires question-réponse avec 4200 images associés.

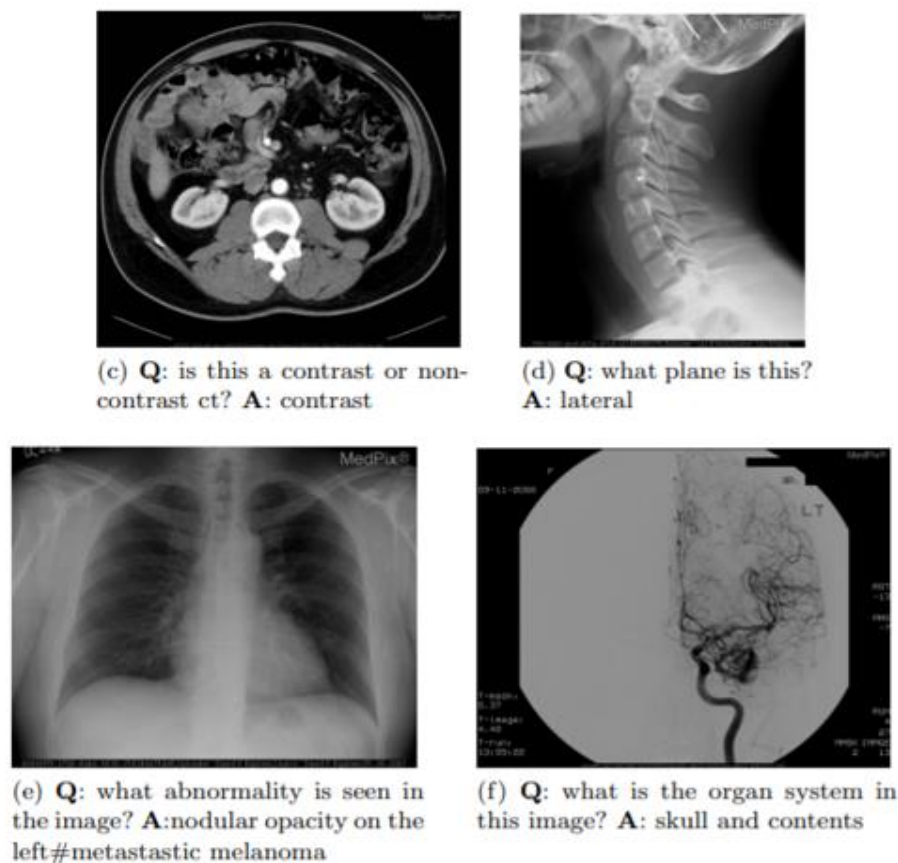


Figure 10: Echantillons d'images de l'ensemble de données VQA-MED-2019

2.5.4 pathVQA

PathVQA est un jeu de données explorant VQA pour la pathologie. Les images avec légendes sont extraites de ressources numériques (manuels électroniques et bibliothèques en ligne). L'auteur a développé un pipeline semi-automatique pour transférer les légendes en paires question-réponse, et les paires question-réponse générées sont vérifiées et révisées manuellement. La question peut être divisée en sept catégories : quoi, où, quand, dont, comment, combien/quantité, et oui/non. Les questions ouvertes représentent 50,2% de l'ensemble des questions. Pour les questions fermées "oui/non", les réponses sont équilibrées avec 8 145 "oui" et 8 189 "non". Les questions sont conçues en fonction de l'examen de certification des pathologistes de l'American Board of Pathology (ABP). Il s'agit donc d'un

examen visant à vérifier le "pathologiste AI" dans l'aide à la décision. L'ensemble de données PathVQA démontre que l'AQV médicale peut être appliquée à diverses scènes[12].

Cet ensemble de données possède environ 32799 paires question-réponse avec 4998 images associés.

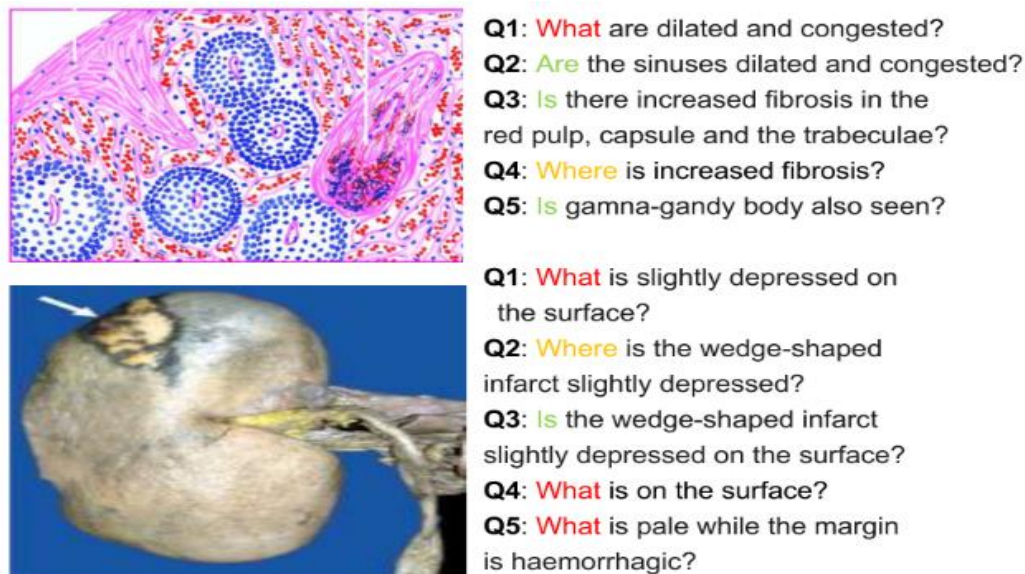


Figure 11: Echantillons d'images de l'ensemble de données pathVQA

2.5.5 VQA-Med-2020

VQA-Med-2020 est la troisième édition du VQA-Med et a été publié dans le cadre du défi ImageCLEF 2020. Les images sont sélectionnées avec la limitation que le diagnostic a été fait selon le contenu de l'image. Les questions portent spécifiquement sur les anomalies. Une liste de 330 problèmes d'anomalies est sélectionnée, et chaque problème doit apparaître au moins dix fois dans l'ensemble de données. Les paires QA sont générées Dans VQA-Med-2020, la tâche de génération de questions visuelles (VQG) est introduite pour la première fois dans le domaine médical. La tâche VQG consiste à générer des questions en langage naturel relatives au contenu de l'image. L'ensemble de données médicales VQG comprend 1001 images de radiologie et 2400 questions associées. Les questions de base sont générées par une approche à base de règles selon les légendes des images et révisées manuellement[13].

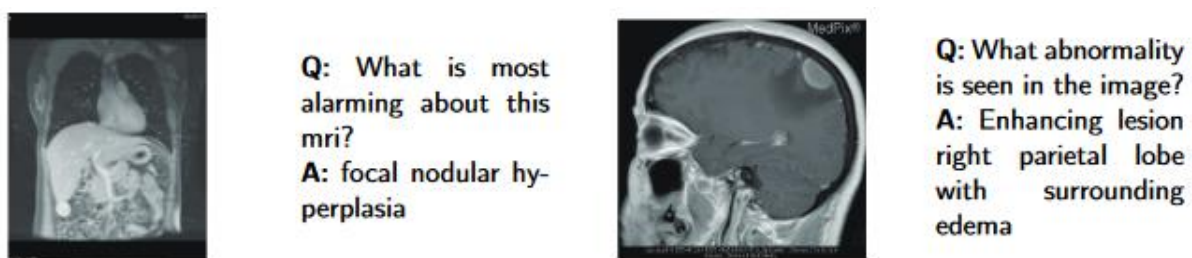


Figure 12: Echantillons d'images de l'ensemble de données VQA-MED-2020

2.6 Synthèse

Dans les sections ci-dessus, nous énumérons 5 ensembles de données médicales VQA en fonction de leur quantité, de leur source de données, de la création des QR et des catégories de questions. La quantité d'images varie de 315 à 91 060, tandis que la quantité de paires QR varie de 1 paire de QR par image à 10 paires de QR par image. La modalité d'imagerie comprend la radiographie x-ray, l'IRM, la tomographie et la pathologie. Les catégories de questions peuvent aller de 1 à 11 catégories.

En comparant les ensembles de données, nous constatons que la nature de la source de données est le facteur clé de la variation. Un type de source de données est constitué par les images accompagnées d'une description, comme une légende ou un rapport médical. VQA-Med-2018 est la première exploration de l'ensemble de données médicales VQA, elle utilise les images dans les articles afin que les images aient une description textuelle correspondante. Grâce à la transformation automatique et à la vérification manuelle, la description textuelle est convertie de phrases déclaratives en paires question-réponse. Le PathVQA utilise des images et du texte provenant de manuels scolaires et de la bibliothèque numérique. Pour ces deux ensembles de données, le problème clé est de savoir comment effectuer une transformation précise.

Un autre type de source de données est l'image avec une attribution catégorielle. VQA-RAD, VQA-Med-2019, VQA-Med-2020 proviennent tous de la base de données Med-Pix qui fournit des images avec des attributions. Par conséquent, le problème clé est de savoir comment obtenir de meilleures questions à partir d'images et de réponses.

Comme la recherche sur le VQA médicale en est encore à ses débuts, les ensembles de données actuels ne concernent que la radiologie et la pathologie. Il y a d'autres domaines à découvrir, comme l'ophtalmologie et la dermatologie, qui sont également populaires dans la recherche sur l'IA médicale et disposent déjà de bases de données existantes pour créer des tâches potentielles de VQA.

2.7 Conclusion

Dans ce chapitre, nous avons exploré les images médicales en général, ses caractéristiques ainsi que ses divers types. Aussi, nous avons positionné l'intérêt de l'analyse des images médicales par le biais des approches de l'intelligence artificielle. D'avantage, nous avons comparé les différents ensembles des données de VQA médicaux disponibles.

Le chapitre suivant va comporter une étude de l'état de l'art en apprentissage profond dans les deux domaines vision par ordinateur et traitement de langage naturel.

ETAT DE L'ART SUR L'APPRENTISSAGE PROFOND(TRANSFORMER)

Sommaire

3.1	Introduction	17
3.2	Modèle Transformer	17
3.2.1	Fonctionnement et architecture	17
3.2.2	Notion d'attention :	19
3.2.3	Attention multi-têtes :	23
3.2.4	Encodage positionnel :	24
3.3	Le modèle « Transformer » pour le traitement du langage naturel.....	25
3.3.1	Qu'est-ce que le traitement du langage naturel	25
3.3.2	Variants du modèle« Transformer » pour les taches NLP	26
3.3.3	Bert	26
3.4	Le modèle « Transformer » pour la vision par ordinateur	28
3.4.1	Vision par ordinateur	28
3.4.2	Vision Transformer(ViT)	28
3.5	Conclusion.....	29

3.1 Introduction

Au cours des dernières années, l'apprentissage profond a subi une grande révolution dans le domaine du traitement de langage naturel ainsi que de la vision par ordinateur, notamment avec l'apprentissage par transfert, les chercheurs ont développé de nouveaux modèles puissants qui remplacent éventuellement les réseaux de neurones convolutionnels (CNN) et les réseaux de neurones récurrents (RNN).

Dans ce chapitre, nous nous concentrons sur le nouveau modèle « Transformer » Ainsi, nous allons détailler leur architecture, son fonctionnement et ses variants .

3.2 Modèle Transformer

Le « Transformer » a été introduit en 2017 par une équipe de Google Brain, et qui est un modèle de Deep Learning (c'est-à-dire un réseau de neurones) de type seq2seq : il prend une séquence (une suite d'éléments du même type) en entrée et renvoie une séquence en sortie. Ce réseau se caractérise par le fait qu'il n'a pas besoin de traiter les données dans l'ordre, autrement dit, dans le contexte du NLP, il n'a pas besoin de traiter le début d'une phrase avant sa fin. De ce fait, Transformer a réussi à résoudre le problème de la parallélisation (par rapport aux RNN et CNN), il a également la particularité d'utiliser uniquement le mécanisme d'attention et aucun réseau récurrent ou convolutif, étant donné que l'attention accélère la vitesse de conversion du modèle d'une séquence à une autre[14].

3.2.1 Fonctionnement et architecture

Le modèle « Transformer » est composé essentiellement de deux composants: un composant d'encodage et un composant de décodage avec des connexions entre eux, illustrées respectivement dans la figure ci-dessous:

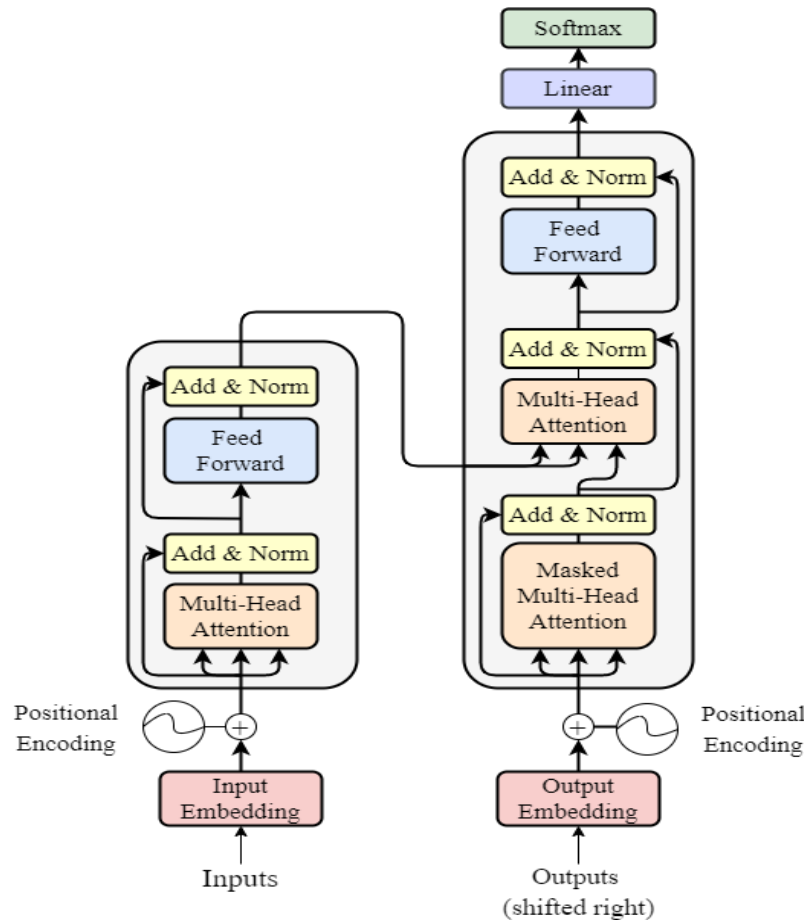


Figure 13: Architecture du transformer

❖ Encodeur

Le composant d'encodage est une pile de $N=6$ couches identiques. Chaque couche comporte deux sous-couches. La première est un mécanisme d'auto-attention à têtes multiples, et la seconde est un simple réseau de type feed-forward entièrement connecté. Le Transformer utilise une connexion résiduelle autour de chacune des deux sous-couches, suivie d'une normalisation des couches. Autrement dit, la sortie de chaque sous-couche est $\text{LayerNorm}(x + \text{Sublayer}(x))$, où $\text{Sublayer}(x)$ est la fonction mise en œuvre par la sous-couche elle-même. Pour faciliter ces connexions résiduelles, toutes les sous-couches du modèle, ainsi que les couches d'intégration, produisent des sorties de dimension $d_{model} = 512$.

❖ Décodeur

Le composant de décodage est une pile de décodeurs du même nombre. En plus des deux sous-couches de chaque couche de l'encodeur, le décodeur insère une troisième sous-couche, qui effectue une attention multi-tête sur la sortie de la pile de l'encodeur.

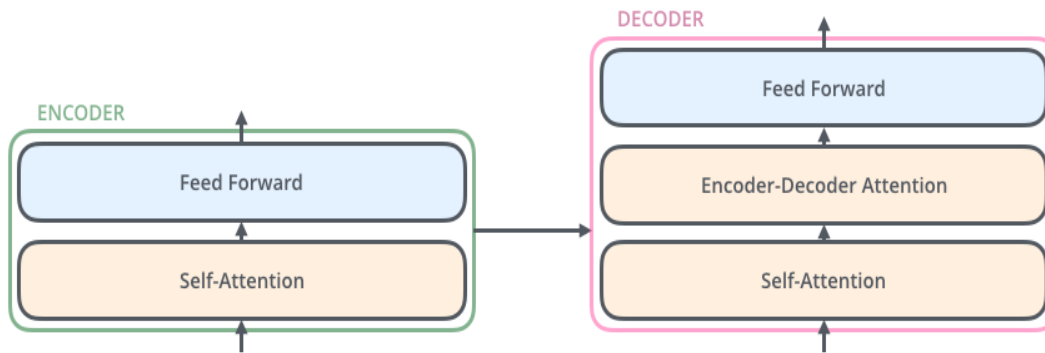


Figure 14: Décodeur du transformer[15]

Comme pour l'encodeur, le décodeur utilise des connexions résiduelles autour de chacune des sous-couches, suivies d'une normalisation des couches.

3.2.2 Notion d'attention :

L'intérêt du concept d'attention est de mesurer à quel point deux éléments de deux séquences sont liés. Dans un contexte de séquence à séquence en NLP, le mécanisme d'attention aura pour but de « dire, au reste du modèle, à quels mots de la séquence B il faut porter le plus d'attention quand on traite un mot de la séquence A ».

- **Word embeddding (Vecteur de plongements de mots en français) :**

Appelé aussi « vecteur d'intégration » ou « vecteur de plongement lexical », Comme on fait habituellement pour les applications en NLP, on commence par transformer et représenter chaque mot à l'entrée par un vecteur de nombres réels via un algorithme de plongement. Chaque mot est mis dans un vecteur de taille 512. Nous représentons ces vecteurs avec ces simples boîtes :



Figure 15: Vecteur de plongement de mots

Le processus de plongements de mots n'a lieu que dans l'encodeur inférieur. Le point commun à tous les encodeurs est qu'ils reçoivent une liste de vecteurs de la taille 512. Dans l'encodeur le plus bas il s'agit de plongements des mots mais dans les autres encodeurs, c'est la sortie de l'encodeur qui est juste en dessous.

La taille de la liste est un hyperparamètre que nous pouvons définir. Il s'agit essentiellement de la longueur de la phrase la plus longue dans notre ensemble de données d'entraînement.

Après que l'on fait les plongements de nos mots à l'entrée, chacun d'entre eux traverse chacune des deux couches de l'encodeur.

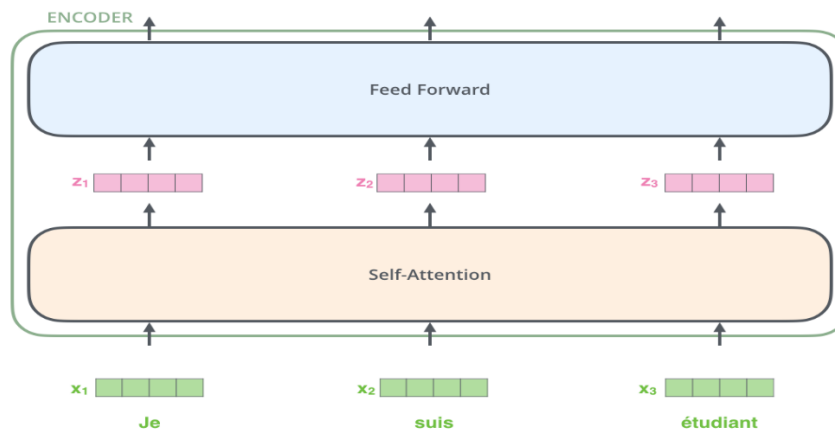


Figure 16: Le vecteur de Plongements de mots traverse les couches d'encodeurs

- **Auto attention:**

Pour calculer le vecteur d'auto attention ou « Scaled Dot-Product Attention », il faut considérer 3 vecteurs pour chaque entrée x_i de l'encodeur :

- Vecteur requête (Query Vector) qu'on appelle q .
- Vecteur clé (Key Vector) qu'on appelle k .
- Vecteur valeur (Value Vector) qu'on appelle v .

Ces vecteurs sont créés en multipliant le vecteur d'embedding du mot x_i par trois matrices que nous avons entraînées pendant le processus d'entraînement.

On a donc : $q_i = W_q x_i$, $k_i = W_k x_i$, $v_i = W_v x_i$. Notez que ces nouveaux vecteurs sont de plus petite dimension que le vecteur de plongements. Leur dimensionalité est 64, tandis que les vecteurs d'entrée/sortie ainsi que les vecteurs d'embedding ont une dimensionalité de 512. La figure ci-dessous présente les vecteurs de poids en entrée.

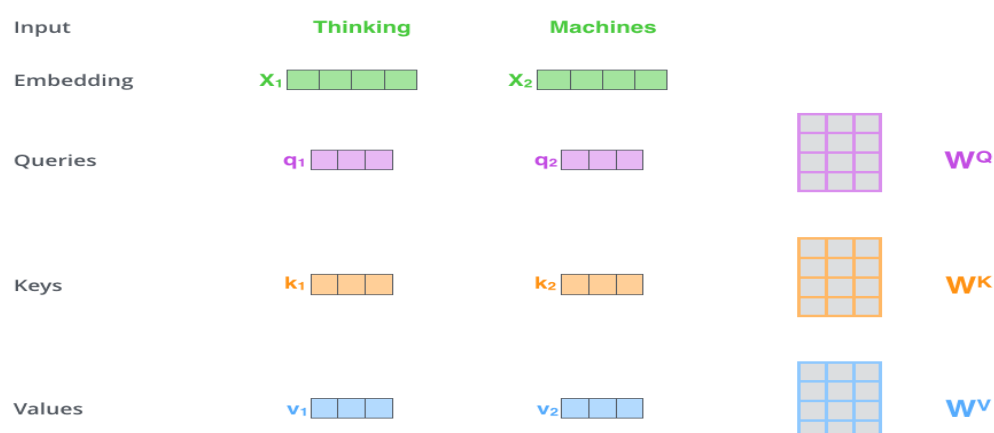


Figure 17: vecteurs de poids en entrée

L'étape suivante du calcul de l'auto-attention consiste à calculer un score. Où ce score détermine le degré de concentration à placer sur les autres parties de la phrase d'entrée au fur et à mesure que nous codons un mot à une certaine position. Le score est calculé en prenant le produit scalaire du vecteur de requête q avec le vecteur clé k du mot que nous évaluons. Donc, si nous traitons l'auto-attention pour le mot en position 1, le premier score serait le produit scalaire de: q_1 et k_1 . Le deuxième score serait le produit scalaire de q_1 et k_2 .

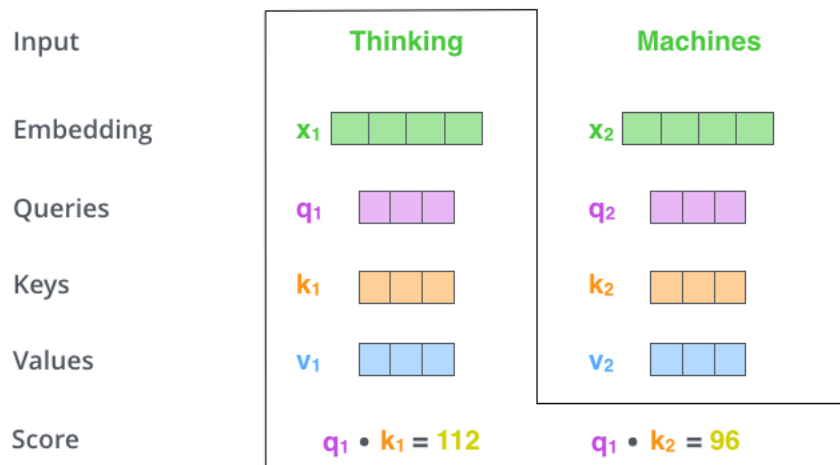


Figure 18: Calcule du score pour l'auto attention

La troisième et quatrième étapes consistent à diviser les scores par la racine carrée de la dimension des vecteurs clés utilisés $\sqrt{d_k}$ (ici on divise par $\sqrt{64}$). Cette stratégie mène aux gradients plus stables. En effet, la fonction *softmax* que nous appliquerons par la suite peut être sensible à de très grandes valeurs d'entrée. Cela tue le gradient et ralentit l'apprentissage, ou l'arrête complètement. Puisque la valeur moyenne du produit scalaire augmente avec la dimension de plongements de mots, il est utile de redimensionner un peu le produit scalaire pour empêcher les entrées de la fonction *softmax* de devenir trop grandes.

Le tout passe finalement à travers un softmax pour normaliser les scores pour qu'ils soient tous positifs et somment à 1.

La figure ci-dessous illustre l'application de la fonction *softmax* :

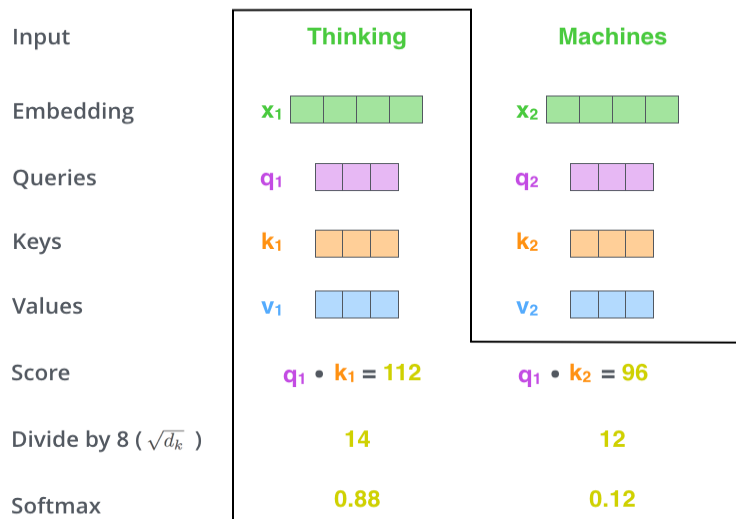


Figure 19: Application de la fonction softmax

Ce score *softmax* détermine à quel point chaque mot est exprimé à sa position. Il est donc logique que le mot qui a le score de *softmax* le plus élevé à sa position et le score des autres mots permet de déterminer leur pertinence par rapport au mot traité.

La cinquième étape est de multiplier chaque vecteur Value par le score *softmax* (en préparation de les sommer).

La sixième étape est de sommer les vecteurs Value pondérés. Ceci produit la sortie de la couche self-attention à cette position (pour le premier mot).

Les vecteurs z_i résultants peuvent être envoyés au réseau feed-forward.

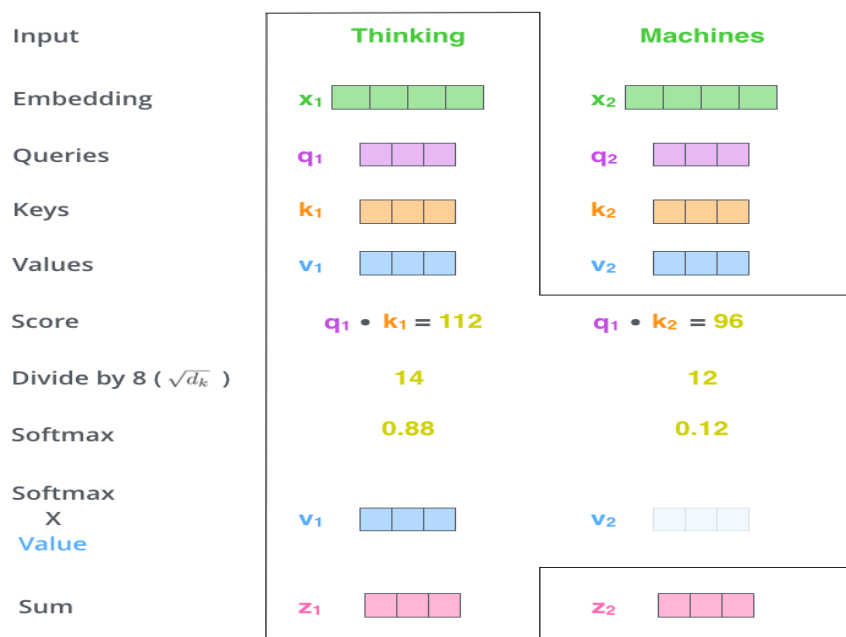


Figure 20: Sortie Z de l'auto attention

Toutes les étapes précédentes de calcul de l'auto attention sont résumés par la formule suivante :

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

3.2.3 Attention multi-têtes :

Dans le modèle « Transformer », le calcul est fait en parallèle par plusieurs blocs d'attention différent. Cela est appelé « Multi-head Attention ». Un ensemble de (W_q , W_k , W_v) est appelé une tête d'attention et chaque couche d'auto-attention dans un modèle de « Transformer », a plusieurs têtes d'attention. Bien qu'une tête d'attention cherche les jetons qui sont pertinents pour chaque jeton, avec plusieurs têtes d'attention, le modèle peut apprendre cette attention pour différentes définitions de pertinence. En effet, les recherches montrent que plusieurs têtes d'attention dans un « Transformer » permettent d'encoder une relation de pertinence qui est similaire à celle des humains. Par exemple, certaines têtes vont, pour chaque jeton, trouver une relation de pertinence avec le jeton suivant alors que d'autres têtes vont encoder une relation entre un verbe et son complément d'objet direct. Étant donné que les modèles « Transformer » ont plusieurs têtes d'attention, ils ont ainsi la possibilité de capturer différents niveaux de relation de pertinence. Les sorties de chaque tête sont alors concaténées ensemble et passées à la couche de réseau de neurones feed forward.

Dans le diagramme ci-dessus (K, Q, V) sont passés à travers des couches linéaires séparées (Dense) pour chaque tête d'attention.

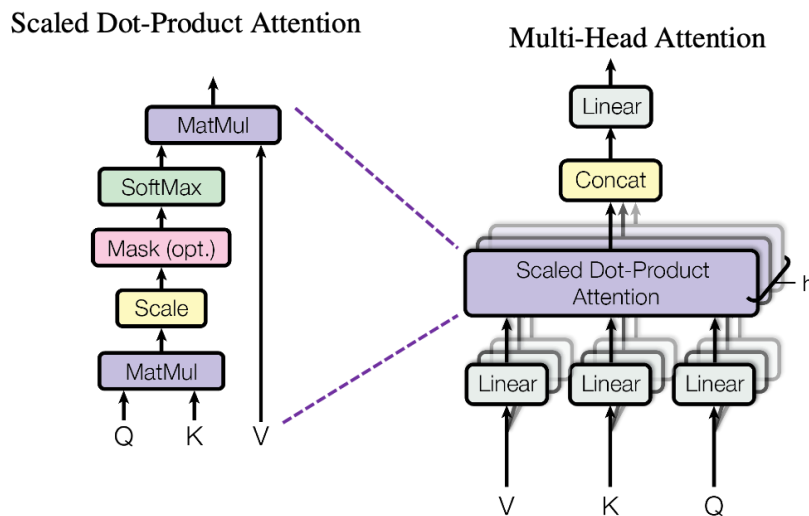


Figure 21: Attention à multiple têtes

Si nous faisons le même calcul d'auto-attention que nous avons décrit ci-dessus, huit fois avec des matrices de poids différentes, nous obtenons huit matrices Z différentes.

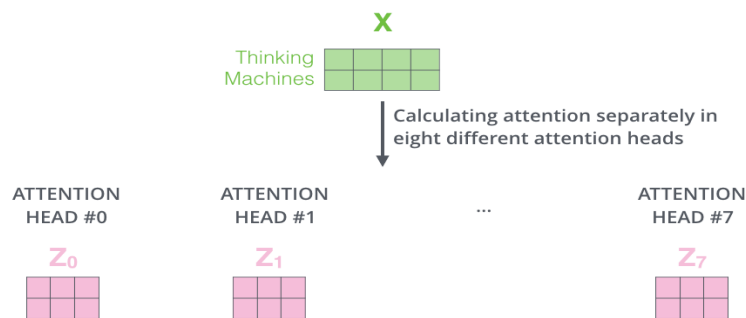


Figure 22: Calcul de l'auto attention dans huit têtes d'attention différentes

Et puisque la couche feed-forward n'attend qu'une seule matrice (un vecteur pour chaque mot), Nous avons donc besoin d'un moyen de condenser ces huit éléments en une seule matrice en les concaténant puis les multipliant par une matrice de poids supplémentaire W_o .

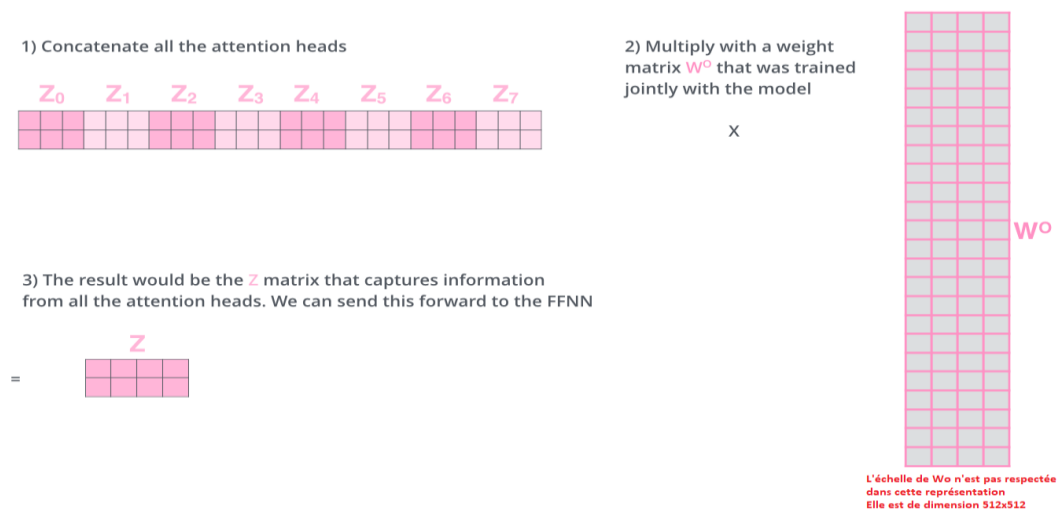


Figure 23: Concaténation de toutes les têtes d'attention et multiplication par la matrice de poids W_o

3.2.4 Encodage positionnel :

Les couches d'attention voient leur entrée comme un ensemble de vecteurs, sans ordre séquentiel. Ce modèle ne contient pas de couches récurrentes ou conventionnelles. Pour cette raison, un "codage positionnel" est ajouté pour donner au modèle des informations sur la position relative des jetons dans la phrase. Le vecteur de codage positionnel est ajouté au vecteur d'embedding (plongement de mots) et il doit avoir la même dimension d_{model} que le vecteur embedding, de sorte que les deux peuvent être additionnés.

Après avoir ajouté ce codage, les jetons seront plus proches les uns des autres en fonction de la similitude de leur sens et de leur position dans la phrase .

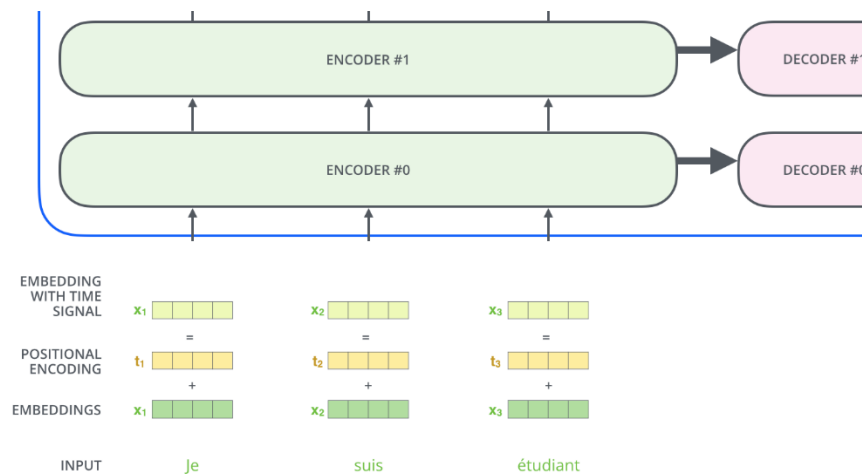


Figure 24: Encodage positionnel

3.3 Le modèle « Transformer » pour le traitement du langage naturel

3.3.1 Qu'est-ce que le traitement du langage naturel

Le traitement du langage naturel NLP (Natural Language Processing en anglais) est une technologie d'intelligence artificielle visant à permettre aux ordinateurs de comprendre le langage humain.

L'objectif de cette technologie est de permettre aux machines de lire, de déchiffrer, de comprendre et de donner sens au langage humain. D'importants progrès ont été réalisés dans ce domaine au cours des dernières années, et le traitement du langage naturel est aujourd'hui exploité pour une large variété de cas d'usage[16].

Globalement, nous pouvons distinguer deux aspects essentiels à tout problème de *NLP* :

- ❖ La partie « linguistique », qui consiste à prétraiter et transformer les informations en entrée en un jeu de données exploitable.
- ❖ La partie « apprentissage automatique » ou « Data Science », qui porte sur l'application de modèles de Machine Learning ou Deep Learning à ce jeu de données[17].

3.3.2 Variants du modèle« Transformer » pour les taches NLP

En se basant sur « Transformer », de nouveaux modèles de langues basés sur l'apprentissage par transfert ont été créés. L'idée derrière ces modèles de langue est de leur faire apprendre une tâche générale puis, par la suite de l'affiner sur une tâche plus spécifique afin de tirer parti des apprentissages de la tâche générale.

Nous allons maintenant présenter un modèle de langues basés sur l'architecture des « Transformers » que nous avons utilisés dans notre projet.

3.3.3 Bert

Une des révolutions les plus récentes en NLP qui a engendré la plupart des résultats de l'état de l'art actuel provient du BERT(Bidirectional Encoder Representations from Transformers) [18], un variant du modèle « Transformer » basé sur la langue

Dans BERT, seule la partie encodeur avec mécanisme d'auto-attention est utilisée, cependant l'information sur la position est apprise ce qui limite le modèle de langue final à une longueur de séquence maximale déterminée à l'entraînement (à savoir 512 jetons).

Profitant de l'architecture des encodeurs, et plus précisément de l'auto-attention, ce modèle de langue utilisé est bidirectionnel, c'est-à-dire que lorsque le modèle détermine la représentation d'un mot dans un contexte, il prend en compte autant l'historique du contexte, c'est-à-dire les mots à gauche, mais aussi la suite du contexte, soit les mots à droite. BERT profite donc de tous les avancements récents en matière d'attention afin de produire des plongements de mots qui contiennent l'information nécessaire à la tâche du modèle.

Pour BERT, le plongement est réalisé en plusieurs étapes. On procède d'abord à un prétraitement des textes en insérant des symboles spéciaux dans le texte brut, pour indiquer au transformateur certaines informations sur les mots et sur la composition du texte en entrée. Ensuite les mots sont décomposés en sous-mots (token) présents dans le vocabulaire et ces derniers seront représentés par leur identifiant dans le vocabulaire. Puis chaque token passera dans la couche du plongement lexical pour obtenir son vecteur de représentation (token embedding). BERT possède également deux autres types de plongement : le plongement de position (position embedding) et le plongement de segmentation (segmentation embedding). Le premier porte l'information de positionnement des phrases dans l'entrée tandis que le second porte l'information structurelle de la phrase. La représentation finale de chaque sous-mot en entrée est la somme des trois vecteurs de

représentation qui lui est associée, la figure 25 illustre comment ces trois plongements représentent une phrase donnée en entrée.

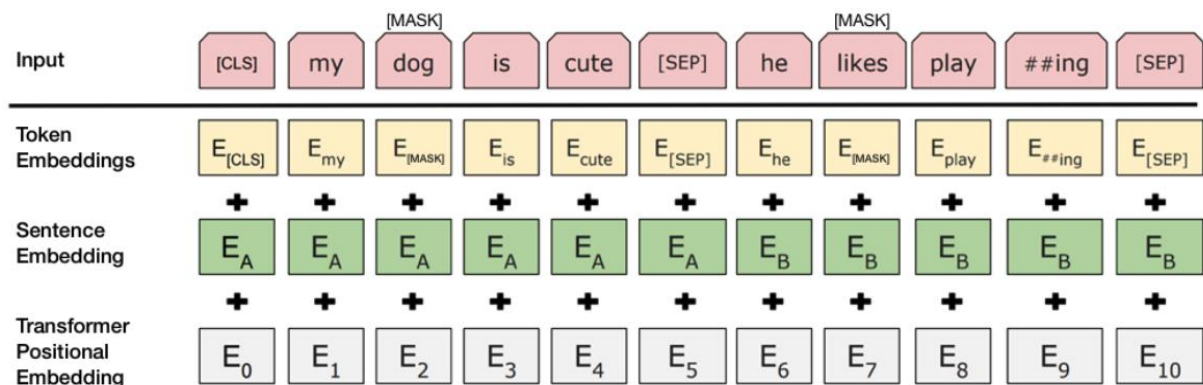


Figure 25: Illustration du modèle pré-entraîné BERT

Pour aider BERT à apprendre, des jetons spéciaux sont ajoutés dans la séquence d'entrée, nommé jeton [CLS] qui indique le début d'une séquence et le jeton [SEP] qui indique la fin d'une séquence. Comme BERT peut prendre jusqu'à deux séquences en entrée, il est possible d'avoir deux jetons [SEP]. Pour entraîner ce modèle de langue, Google propose l'approche suivante : BERT est un modèle de langue entraîné de façon non-supervisée sur un très grand corpus comme Wikipédia et se fait sur 2 tâches.

✚ **La première tâche :** permet au modèle d'apprendre les relations entre les mots (Masked Language Model MLM). Pour ce faire, le modèle apprend à prédire un mot masqué dans une séquence, comme le montre l'exemple suivant où le [MASK] remplace le mot président. En somme, 15% du corpus d'entraînement est masqué et c'est à BERT de retrouver le bon mot correspondant. Cela permet au modèle de produire des représentations de mots pertinentes pour tous les mots de la phrase.

Exemple: Barack Obama was the first African-American [MASK] of the United States.

✚ **La seconde tâche :** permet au modèle d'apprendre les relations entre les séquences. Pour ce faire, le modèle est entraîné avec deux phrases en entrée. Le modèle de langue doit alors apprendre à prédire si une séquence A est suivie d'une séquence B ou non, comme l'illustrent les deux exemples suivants. Dans le premier cas, la séquence B est la suite de la séquence A mais pas dans le second cas. Cela permet au modèle d'apprendre des bases de compréhension multi-phrases et aussi des bases d'implication textuelle (textual entailment).

Exemple 1 : The man went to the store. He bought a gallon of milk.

Exemple 2 : The man went to the store. Penguins are flightless birds.

Le modèle de langue pré-entraîné BERT se décline en deux versions majeures, une version de base qui compte 12 couches d'encodeurs, 12 têtes d'attentions et pas moins de 110 millions de paramètres. La version large compte 24 couches d'encodeurs, 16 têtes d'attentions et 340 millions de paramètres. BERT possède également des variantes pré-entraînées dans le domaine médical, telles que BioBERT et SciBERT..

3.4 Le modèle « Transformer » pour la vision par ordinateur

En raison du succès de Transformateurs dans le domaine du traitement du langage naturel (NLP), de nombreux chercheurs ont commencé à appliquer cette architecture à d'autres domaines tels que la vision par ordinateur.

3.4.1 Vision par ordinateur

La vision par ordinateur (computer vision en anglais) est le domaine de l'informatique qui vise à reproduire une partie de la complexité du système de vision humain et à permettre aux ordinateurs d'identifier et de traiter des objets dans des images et des vidéos de la même manière que les humains. Grâce aux progrès de l'intelligence artificielle et aux innovations en matière d'apprentissage profond et de réseaux neuronaux, le domaine a pu faire de grands bonds en avant ces dernières années. Ainsi, il a pu surpasser les humains dans certaines tâches liées à la détection et à l'étiquetage des objets.

3.4.2 Vision Transformer(ViT)

Maintenant que nous avons une idée approximative du fonctionnement de l'auto-attention à plusieurs têtes et des transformateurs, passons au ViT (Vision Transformer) qui est un modèle pré-entraîné du Transformer, a été publiée au début de 2020 par l'équipe de recherche de Google, et adopter pour les taches de vision par ordinateur [\[19\]](#).

Fonctionnement de ViT :

L'idée de ViT est d'utiliser un Transformer Encoder comme modèle de base pour extraire les caractéristiques de l'image, et de transmettre ces caractéristiques "traitées" dans un modèle de tête de Perceptron multicouche (MLP) pour la classification. La figure ci-dessous résume l'architecture de vision transformer :

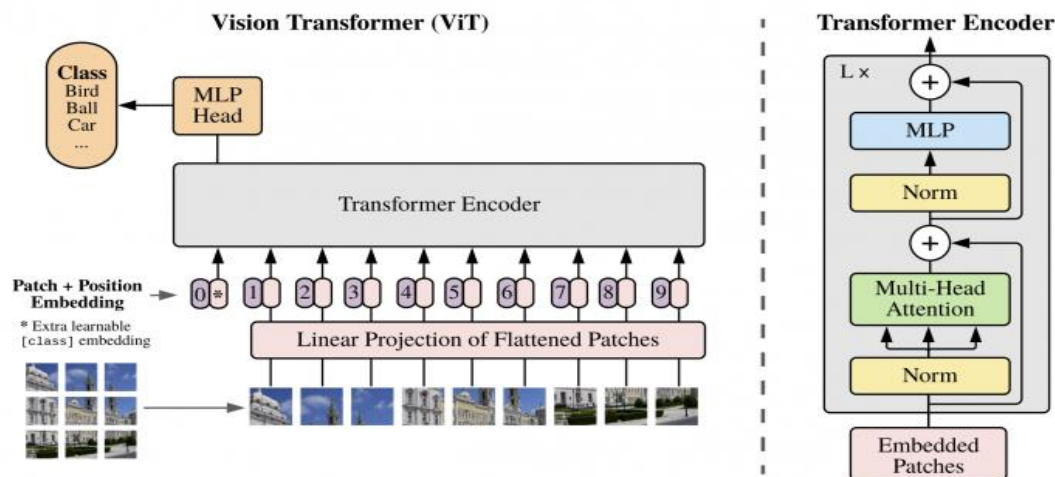


Figure 26: Fonctionnement du ViT [19]

Nous soulignons ci-dessous le principe de fonctionnement de ViT étape par étape:

1. Diviser une image en patches de tailles fixes.
2. Transformer les patches d'image en vecteurs de patches par une transformation linéaire(flattening) .
3. Combiner ces vecteurs de patches avec des intégrations positionnels (encodage positionnel).
4. Envoyez la séquence à l'encodeur « Transformer » qui la traite de la même manière que les vecteurs mots.

3.5 Conclusion

Au cours de ce chapitre, nous avons décrit les notions de base du modèle « Transformer ». Ensuite, nous avons présenté deux variants issus du « Transformer » pour le traitement du langage naturel et pour la vision par ordinateur. Dans le chapitre suivant, nous allons mener une étude analytique des modèles sélectionnées à partir des approches détaillées dans ce chapitre tout en les implémentant et les testant sur notre problématique de réponse aux questions visuelles à partir des images médicales.

IMPLEMENTATION ET EVALUATION

Sommaire

4.1	Introduction	31
4.2	Environnement de travail.....	31
4.2.1	Environnement du développement.....	31
4.2.2	Installation des bibliothèques	32
4.3	Données d'expérimentations	34
4.3.1	Préparation et exploration des données	34
4.3.2	Critères d'évaluation :	37
4.4	Implémentation	39
4.4.1	implémentation du modèle BERT+ ViT	39
4.4.1.1	Architecture du modèle	39
4.4.1.2	Déploiement BERT+ViT	40
4.4.1.3.1	Accuracy et loss du modèle	44
4.4.1.3.2	Evaluation :	45
4.4.1.3.3	Résultats de VQA(inférence) :	46
4.4.2	Implémentation de BioBERT+ViT	49
4.4.2.1	Résultats expérimentaux	49
4.4.2.1.1	Accuracy et loss du modèle	49
4.4.2.1.1.1	Evaluation	50
4.1.2	Implémentation de BioBERT+Swin Transformer	51
4.1.2.1.1	Accuracy et loss du modèle	53
4.1.2.1.2	Evaluation	53
4.2	Comparaison des résultats	53
4.3	Conclusion.....	55

4.1 Introduction

Nous présentons dans ce chapitre une étude comparative expérimentale de différentes méthodes de Réponse aux questions visuels. Cette étude expérimentale concerne les modèles basés sur le « Transformer ». Nous procédons donc à la présentation du processus d'implémentation tout en illustrant les résultats obtenus. La dernière partie de ce chapitre est dédiée à l'interprétation des résultats expérimentaux obtenus au cours de cette étude.

4.2 Environnement de travail

4.2.1 Environnement du développement

Nous avons choisi le langage python car il est largement utilisé dans le domaine d'apprentissage profond. Python offre plusieurs bibliothèques d'apprentissage automatique populaires qui peuvent être facilement chargées dans l'ordinateur portable et qui facilite le traitement d'images, de données, etc...

Comme environnement de développement, nous avons choisi l'environnement en ligne "google colab" (ou collaboratory), qui est un service cloud, offert par google, basé sur jupyter Notebook, cette plateforme permet d'entraîner des modèles machine learning directement dans le cloud sans avoir besoin d'installation. Une autre fonctionnalité intéressante que Colab propose aux développeurs est l'utilisation du GPU et qui est totalement gratuit.

Pendant de nombreuses années, Le groupe Meta a développé un package de développement complet d'Intelligence Artificielle appelé pytorch. Aujourd'hui, pytorch est un package open-source qui, depuis 2018, Meta l'a rendu accessible et gratuit pour un usage public.



Figure 27: Illustration du google colab

4.2.2 Installation des bibliothèques

Google Colab prend en charge la plupart des bibliothèques de l'apprentissage automatique disponibles sur le marché. Nous commençons par examiner comment installer ces bibliothèques dans l'éditeur de Google Colab. Pour installer une bibliothèque, nous devons rédiger l'instruction suivante :

```
!pip install  
ou  
!apt-get install
```

- Utilisation d'un GPU gratuit :

Google propose l'utilisation d'un GPU gratuit pour les notebooks Colab. Pour activer le GPU sur une machine de travail, nous sélectionnons la commande suivante :

```
Runtime/change runtime type
```

- Pytorch :

Est une bibliothèque logicielle Python, développée par Facebook et s'appuie sur Torch. PyTorch permet d'effectuer les calculs tensoriels nécessaires notamment pour l'apprentissage profond. Ces calculs sont optimisés et effectués soit par le processeur (CPU), soit lorsque c'est possible, par un processeur graphique (GPU) supportant CUDA.



Figure 28: Logo du Pytorch

- Scikit-learn :

Scikit-learn, encore appelé sklearn, est la bibliothèque la plus puissante et la plus robuste pour la machine learning en Python. Elle fournit une sélection d'outils efficaces pour l'apprentissage automatique et la modélisation statistique, notamment la classification, la régression et le clustering via une interface cohérente en Python. Cette bibliothèque, qui est en grande partie écrite en Python, s'appuie sur NumPy, SciPy et Matplotlib.



Figure 29: Logo de Sklearn

- PIL :

Python Imaging Library est une bibliothèque de traitement d'images pour le langage de programmation Python. Elle permet d'ouvrir, de manipuler, et de sauvegarder différents formats de fichiers graphiques. Elle est conçue de manière à offrir un accès rapide aux données contenues dans une image, et offre un support pour différents formats de fichiers tels que PPM, PNG, JPEG, GIF, TIFF et BMP.



Figure 30: Logo du PIL

- Numpy ;

Une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels. Ainsi, elle permet d'effectuer des calculs numériques. Le package Numpy introduit une gestion plus facile des tableaux de nombres. Pour utiliser NumPy, vous devez au préalable vous placer dans un environnement qui comprend cette bibliothèque, Pour importer le package numpy, on doit utiliser cette instruction :

```
import numpy as np
```



NumPy

Figure 31: Logo du Numpy

- Matplotlib :

Une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous forme de graphiques.

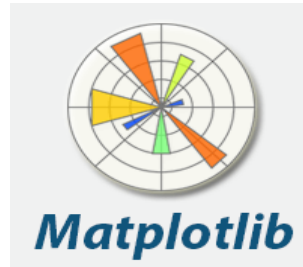


Figure 32: Logo de la bibliothèque matplotlib

4.3 Données d'expérimentations

Dans ce travail, nous allons exploiter la base de données publique 'VQA-Med 2019'. Ces données publiques sont issues d'une deuxième édition de imageCLEF challenge .

Cet ensemble de données se compose de 3 200 images médicales avec 12792 Paires question-réponse (QA) comme données d'apprentissage, 500 images médicales avec 2 000 paires QA comme données de validation et 500 images médicales avec 500 Paires question-réponse comme données de test.

Les données sont également réparties sur quatre catégories en fonction des types de questions qui sont : la catégorie de plan, la catégorie d'organe, la catégorie de modalité et la catégorie d'anomalie. Nous pouvons déterminer la catégorie de la question à partir des mots de la question, c'est-à-dire si le mot "plane" apparaît dans la question, alors c'est une question de plan. Alors que si les mots « organe » ou « partie » apparaissent dans la question, il s'agit alors d'une question d'organe. Si les mots 'normal', 'abnormal', 'alarm' ou 'wrong' apparaissent dans la question, donc c'est une question d'anomalie. Sinon, c'est une question de modalité.

4.3.1 Préparation et exploration des données

Notre base de données est composée de trois dossiers d'images et de trois fichiers csv : d'apprentissage , de validation et de test ,contiennent respectivement les paires question-réponse ainsi que les ID des images concernées .Une image peut avoir une ou plusieurs paires de question-réponse. Les dimensions des images introduites sont différentes.

En effet, nous commençons par importer les packages et les bibliothèques dont nous aurons besoin pour compiler le code.

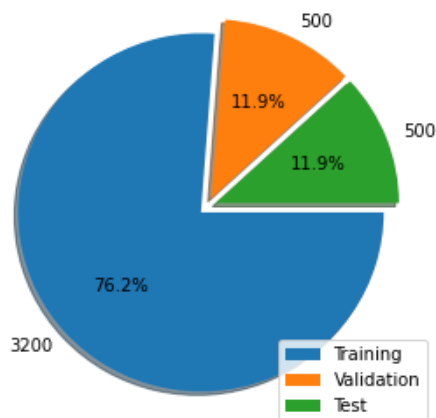
```
#Importation des librairies necessaires
import os
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from PIL import Image
import seaborn as sns

import torch
import torchvision.transforms.functional as fn
```

Figure 33: Importation des bibliothèques

La distribution des images et des paires Question-Réponse dans l'ensemble de données VQA-Med-2019 est illustré dans les figures ci-dessous :

Distribution des images :



Distribution des paires Question-Réponse :

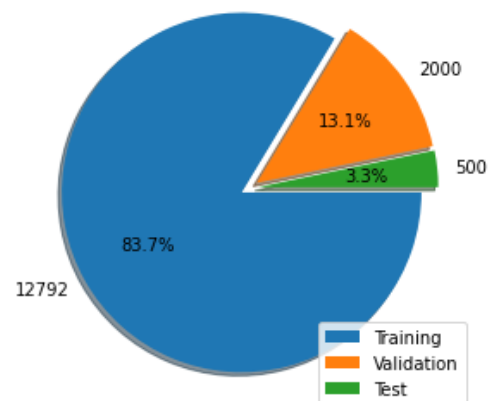


Figure 34: Répartition des images et des paires Question-Réponse dans l'ensemble de données VQA-Med-2019

Nous pouvons voir sur le graphique ci-dessus que la plupart des questions possèdent 7 mots , un minimum 4 mots et 11 maximum .

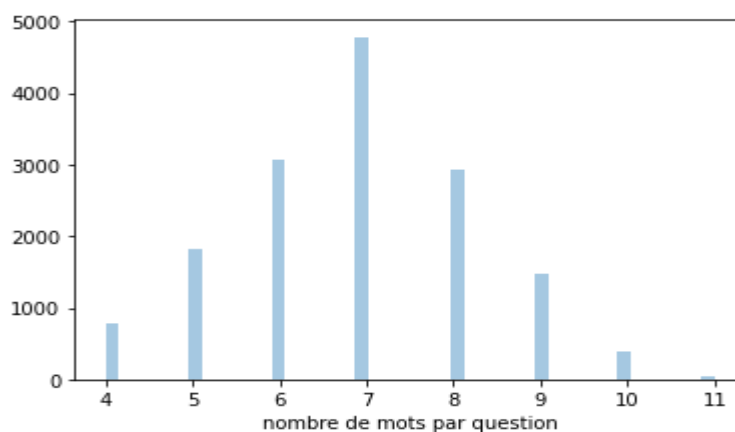


Figure 35: Distribution de nombre de mots par question

La portion de code ci-dessous permet de charger et d'inspecter un exemple aléatoire parmi les entrées présentes dans notre ensemble de données d'entraînement ou de validation tout en affichant les combinaisons Image/ Question /Réponse.

```
#exemple aléatoire

def showExample(mode, id=None):
    if mode=="training":
        data = dataset["training"]
    elif mode=="validation":
        data = dataset["validation"]
    else:
        data = dataset["test"]
    if id == None:
        id = np.random.randint(len(data))
    root_dir = '/content/drive/MyDrive/Sihem Jouini /VQA-Med2019/images'
    image = Image.open(os.path.join( root_dir, data[id]["img_id"] + ".jpg"))
    crop = fn.center_crop(image, output_size=[224])
    new_img = fn.resize(image, size=[224,224])
    display(new_img)
    print('Question: \t' + data[id]['question'])
    print('Réponse: \t' + data[id]['answer'] )

showExample(mode="training")
```

Figure 36: Fonction pour l'affichage d'un exemple aléatoire

Maintenant notre exemple est prêt à afficher en appelant la fonction `showExample ()`.



Figure 37: Un échantillon d'une combinaison image/question/réponse

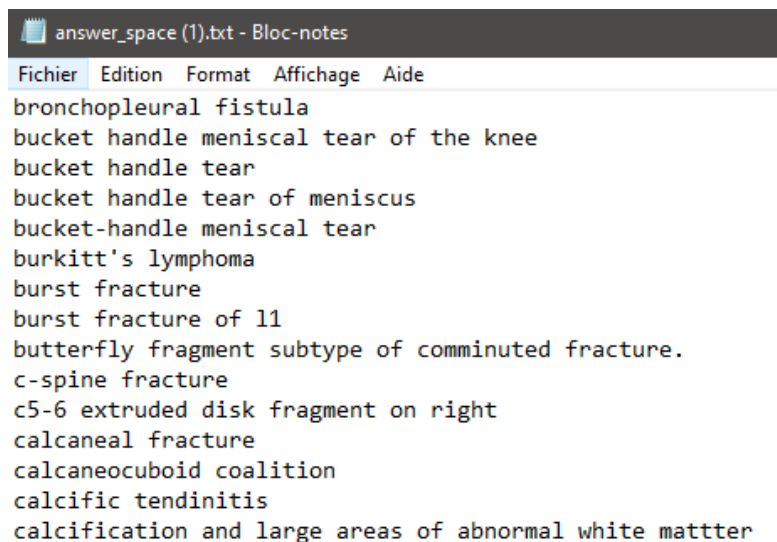
À ce stade, nous allons créer une liste de toutes les réponses possibles, de sorte que la partie de la tâche VQA qui consiste à générer des réponses peut être modélisée comme une classification multi-classe. Cette liste sera sauvegardée dans un fichier texte que nous appellerons « answer space ».

```
answer_space = []
for ans in train.answer.to_list():
    answer_space = answer_space + [ans] #if "," not in ans else answer_space + ans.split(",")

answer_space = list(set(answer_space))
answer_space.sort()
with open(os.path.join('/content/drive/MyDrive/Sihem Jouini /VQA-Med2019 (1)/answer_space.txt'), "w") as f:
    f.writelines("\n".join(answer_space))
```

Figure 38: Création de l'espace de réponse

La structure du fichier « Answer Space » est comme suit :



```
answer_space (1).txt - Bloc-notes
Fichier Edition Format Affichage Aide
bronchopleural fistula
bucket handle meniscal tear of the knee
bucket handle tear
bucket handle tear of meniscus
bucket-handle meniscal tear
burkitt's lymphoma
burst fracture
burst fracture of l1
butterfly fragment subtype of comminuted fracture.
c-spine fracture
c5-6 extruded disk fragment on right
calcaneal fracture
calcaneocuboid coalition
calcific tendinitis
calcification and large areas of abnormal white matter
```

Figure 39: Structure du fichier "Answer Space"

4.3.2 Critères d'évaluation :

— Accuracy :

L'accuracy est un choix valable d'évaluation pour les problèmes de classification qui sont bien équilibrés et non asymétriques ou sans déséquilibre de classe.

$$ACCURACY = \frac{TP + TN}{TP + TN + FP + FN}$$

- True positives (TP) : les vrais positifs sont les cas où la classe réelle de données est (Vrai) et la valeur prédite est également (Vrai)

- True negatives (TN) : les vrais négatifs sont les cas où la classe réelle des données est (Faux) et la valeur prédite est également (Faux)
- False positives (FP) : les faux positifs sont les cas où la classe réelle des données est (Faux) tandis que la valeur prédite est (Vrai).
- False negatives (FN) : les faux négatifs sont les cas où la classe réelle des données est (Vrai) et la valeur prédite est (Faux).

— **Indice de rappel (Recall) :**

L'indice de rappel calcule le nombre de positifs réels que notre modèle capture en les étiquetant comme positif (vrai positif) [TP]. En appliquant la même compréhension, nous savons que le rappel sera la métrique du modèle que nous utiliserons pour sélectionner notre meilleur modèle lorsqu'il y a un coût élevé associé aux faux négatifs [FN]. L'indice de recall est formulé comme suit :

$$RECALL = \frac{TP}{TP + FN}$$

— **Précision :**

La précision est une bonne mesure pour déterminer si les coûts des faux positifs [FP] sont élevés. la précision est donnée par la formule ci-dessous :

$$PRECISION = \frac{TP}{TP + FP}$$

— **Score de similarité Wu et Palmer (WUPS) :**

Pour évaluer les réponses ouvertes en langage naturel, il est nécessaire d'effectuer une correspondance exacte des chaînes. Cependant, il est trop difficile de déduire la relation sémantique entre la réponse prédite et la réponse vraie. Cela incite à utiliser d'autres métriques qui capturent efficacement la similarité sémantique des chaînes. L'une de ces mesures couramment utilisées est le score de similarité Wu et Palmer (WUPS).

WUPS calcule la similarité sémantique entre deux mots ou phrases en fonction de leur plus longue sous-séquence commune dans l'arbre taxonomique. Ce score fonctionne bien pour les réponses à un seul mot (par conséquent, nous l'utilisons pour notre tâche), mais peut ne pas fonctionner pour les expressions ou les phrases.

— **Score F1 :**

Le score F1 combine la précision et l'indice de rappel(RECALL) d'un classificateur en une seule métrique en prenant leur moyenne harmonique. Il est principalement utilisé pour

comparer les performances de deux classificateurs. Supposons que le classificateur A ait un rappel plus élevé et que le classificateur B ait une précision plus élevée. Dans ce cas, les scores F1 des deux classificateurs peuvent être utilisés pour déterminer celui qui produit les meilleurs résultats.

Le score F1 d'un modèle de classification est calculé comme suit :

$$F1_SCORE = \frac{2(P * R)}{P + R}$$

Avec P =Précision et R =Recall

4.4 Implémentation

Cette section concerne l'implémentation des différents modèles pré-entraînés profonds de « Transformer ». Ainsi, nous implémenterons les modèles « BERT », et « BioBERT », pour la tâche NLP et les modèles « ViT » et « Swin Transformer » pour la tâche de vision par ordinateur. Afin de comparer la performance de ces méthodes nous allons utiliser les critères d'évaluation mentionnés dans la section précédente.

4.4.1 implémentation du modèle BERT+ ViT

4.4.1.1 Architecture du modèle

Les modèles multimodaux peuvent prendre diverses formes pour capturer des informations à partir des modalités de texte et d'image. Ici, nous explorons deux variant de « Transformer » afin d'extraire les caractéristiques, d'image et de la question, puis on fusionne les deux caractéristiques pour effectuer la tâche VQA.

Notre modèle multimodal comprenant :

- **Un encodeur de texte :** qui peut être un modèle de « Transformer » basé sur du texte . ici on va utiliser le modèle BERT(Bidirectional Encoder Representations from Transformers).
- **Un encodeur d'image :** qui peut être un modèle de « Transformer » d'image ici on va utiliser le modèle pré-entraîné ViT(Vision Transformer)
- **Une couche de fusion simple :** qui concatène les caractéristiques textuelles et d'image et les fait passer à travers une couche linéaire pour générer une sortie intermédiaire .
- **Un classificateur :** qui est un réseau entièrement connecté avec une sortie ayant les dimensions égales à celle de l'espace de réponse.

La figure ci-dessous résume l'architecture de notre modèle :

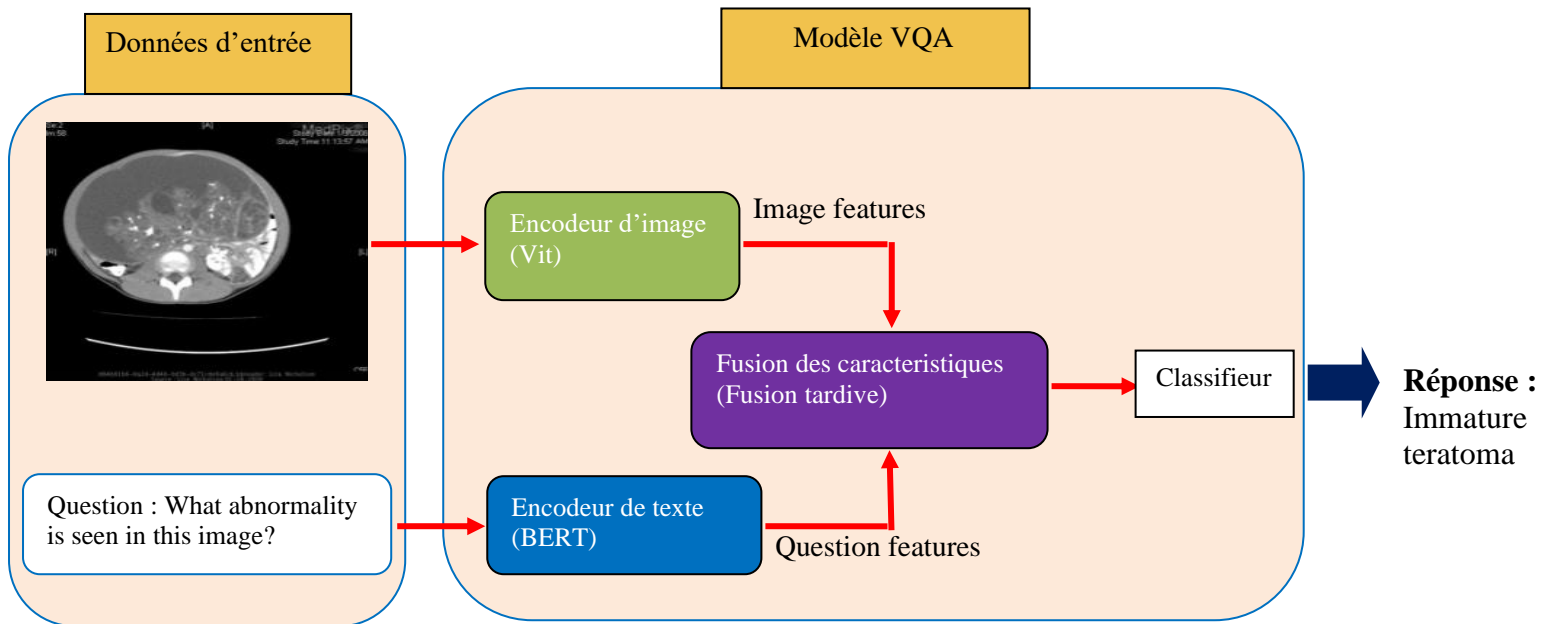


Figure 40: Une représentation de l'architecture du modèle BERT-ViT

4.4.1.2 Déploiement BERT+ViT

Dans ce qui suit, nous allons décrire les différentes étapes de déploiement de l'architecture multimodale avec BERT et ViT en se référant au package Pytorch.

- Nous commençons par l'installation du « Transformer » :

```
!pip install transformers
```

- Ensuite, nous devons importer les bibliothèques nécessaires au fonctionnement du modèle :

```
from transformers import (
    AutoTokenizer, AutoFeatureExtractor, AutoModel, TrainingArguments, Trainer )
```

- Passant à l'étape du prétraitement et création du Dataloader :

Nous utilisons le AutoTokenizer et l'AutoFeatureExtractor pour extraire les caractéristiques du question et d'image sur les modèle pré-entraîné. Nous sauvegardons ces caractéristiques dans les variables tokenizer et preprocessor.

```
tokenizer = transformers.AutoTokenizer.from_pretrained('bert-base-uncased')
preprocessor = transformers.AutoFeatureExtractor.from_pretrained('google/vit-base-patch16-224-in21k')
```

Un Dataloader est correspond à la méthode de chargement des données en entrée du modèle. Il traitera la question (texte) et l'image, et renverra le tokenisation de texte (avec des

masques d'attention) et les caractéristiques de l'image (essentiellement, les valeurs de pixel). Ceux-ci seront introduits dans notre modèle multimodal pour répondre aux questions.

Aucun modèle d'apprentissage profond ne peut travailler directement avec le texte, c'est pour cela, une tokenisation au niveau des caractères est nécessaire puisqu'elle permet de convertir le texte en chiffres que les modèles d'apprentissage profond peuvent utiliser pour le traitement.

Le modèle pré-entraîné de BERT a un tokenizer associé qui est construit sur des vocabulaires de grande taille. BERT extrait les caractéristiques implicites du texte en produisant en sortie : l' `input_ids`, `input_masks` et `segment_ids`. Tels que :

- `input_word_ids` : Mots codés à l'aide de BERT-tokenizer.
- `Mask_ids` : Séparation des jetons utiles et des jetons de remplissage (0 ou 1).
- `Segment_ids` : Utile pour l'apprentissage par paire des phrases.

La portion du code suivante est utilisée pour la tokenisation des questions avec le modèle BERT :

```
def tokenize_text(self, texts: List[str]):
    encoded_text = self.tokenizer(
        text=texts,
        padding='longest',
        max_length=24,
        truncation=True,
        return_tensors='pt',
        return_token_type_ids=True,
        return_attention_mask=True,
    )
    return {
        "input_ids": encoded_text['input_ids'].squeeze(),
        "token_type_ids": encoded_text['token_type_ids'].squeeze(),
        "attention_mask": encoded_text['attention_mask'].squeeze(),
    }
```

Figure 41: Portion de code d'une étape de tokenisation avec le modèle BERT

Les images aussi doivent subir une phase de prétraitement tel que le redimensionnement et la conversion en RGB, le code ci-dessous renverra les caractéristiques de l'image (essentiellement, les valeurs de pixel) Ceux-ci seront introduits dans notre modèle de « Transformer » multimodal .

```
def preprocess_images(self, images: List[str]):
    processed_images = self.preprocessor(
        images=[Image.open(os.path.join('/content/drive/MyDrive/Sihem Jouini /VQA-Med2019/images', img_id + ".jpg")).convert('RGB')
        for img_id in images],
        return_tensors="pt",
    )
    return {
        "pixel_values": processed_images["pixel_values"].squeeze(),
    }
```

Figure 42: Portion de code d'une étape de préprocessing avec le modèle ViT

- Maintenant, nos données sont prêtes ,Passons à l'étape du training du modèle.

Notre modèle multimodal comporte :

❖ Encodeur de texte :

```
self.text_encoder = AutoModel.from_pretrained(self.pretrained_text_name)
```

❖ Encodeur d'image :

```
self.image_encoder = AutoModel.from_pretrained(self.pretrained_image_name)
```

❖ Fusion entre les deux modalités :

Nous allons maintenant passer à l'étape de la fusion. De ce fait, on effectue la concaténation des sorties de l'encodeur de texte avec les sorties de l'encodeur de l'image. Ici, Nous utiliserons **nn.Sequential** pour créer un modèle de séquence au lieu de créer une sous classe de nn.module ; ici les données sont transmises à travers les modules dans l'ordre défini. dans ce cas nous allons créer un réseau de neurones avec une couche linéaire d'entrée dont le première paramètre soit la taille de l'entrée (**taille du texte encodé + taille d'image encodée**) et que le second paramètre soit la taille de sortie(**intermediate_dim=512**).

Nous ajoutons une fonction d'activation ReLU (Unité Linéaire Rectifié), qui permet d'effectuer un filtre sur nos données en définissant toutes les caractéristiques entrantes comme étant égales ou supérieures à 0. Lorsqu'on applique cette couche, tout nombre inférieur à 0 est transformé en zéro, tandis que les autres restent inchangés. Elle laisse passer les valeurs positives($x > 0$) dans les couches suivantes du réseau de neurones.

finalement, nous ajoutons une couche dropout lors du développement de notre modèle. La classe **torch.nn.Dropout()** sera utilisée pour ce faire. Certains des éléments du tenseur d'entrée sont désactivés au hasard par cette classe pendant l'entraînement. la couche dropout prend comme paramètre le taux de dropout p qui donne la probabilité qu'un neurone soit désactivé. Cette option a une valeur par défaut de 0,5, ce qui implique que la moitié des

neurones seront désactivés. Il est important de noter que le dropout peut réduire considérablement le risque de surajustement pendant le training.

La structure d'un bloc issu de cette étape de fusion est définie comme suit :

```
# Couche de fusion
self.fusion = nn.Sequential(
    nn.Linear(self.text_encoder.config.hidden_size + self.image_encoder.config.hidden_size, intermediate_dim),
    # ReLu (Unité Linéaire Rectifiée)==>fonction d'activation
    nn.ReLU(),
    # La méthode du dropout consiste à désactiver des sorties de neurones aléatoirement
    nn.Dropout(0.5),
)
```

Figure 43: Portion de code de l'étape de fusion

❖ Classifieur :

le classifieur est une simple couche linéaire, qui calcule les scores de chacune des classes. Dans notre ensemble de données VQA-Med-2019, on distingue 1749 classes d'étiquettes que nous avons sauvegardées dans le fichier « answer_space ». L'étiquette ayant le score le plus élevé sera celle que le modèle prédit. Dans cette couche linéaire, nous devons spécifier le nombre de caractéristiques d'entrée et le nombre de caractéristiques de sortie qui correspond au nombre de classes.

PyTorch dispose de nombreuses fonctions de perte standard dans le module torch.nn. et Puisque nous modélisons la tâche VQA comme une classification multi-classes, il est naturel d'utiliser Cross-Entropy Loss comme fonction de perte. Il est facile de définir la fonction de perte et de calculer les pertes :

La portion de code suivante illustre la construction du classifieur :

```
#classifieur
self.classifier = nn.Linear(intermediate_dim, self.num_labels)

self.criterion = nn.CrossEntropyLoss()
```

Figure 44: Portion de code de construction de classifieur

L'architecture finale de notre modèle est comme suit :

```
class MultimodalVQAModel(nn.Module):
    def __init__(self, pretrained_text_name, pretrained_image_name, num_labels= len(answer_space), intermediate_dim= 512, dropout=0.5):
        super(MultimodalVQAModel, self).__init__()
        self.num_labels = num_labels
        self.pretrained_text_name = pretrained_text_name
        self.pretrained_image_name = pretrained_image_name

        #encodeurs d'image et de texte
        self.text_encoder = AutoModel.from_pretrained(self.pretrained_text_name)
        self.image_encoder = AutoModel.from_pretrained(self.pretrained_image_name)

        # Couche de fusion
        self.fusion = nn.Sequential(
            nn.Linear(self.text_encoder.config.hidden_size + self.image_encoder.config.hidden_size, intermediate_dim),
            # ReLu (Unité Linéaire Rectifiée)=>fonction d'activation
            nn.ReLU(),
            # La méthode du dropout consiste à désactiver des sorties de neurones aléatoirement(0.5 pour les couches cachées)
            nn.Dropout(0.5),
        )

        #classifieur
        self.classifier = nn.Linear(intermediate_dim, self.num_labels)

        self.criterion = nn.CrossEntropyLoss()
```

Figure 45: Portion de code du modèle multimodale BERT-ViT

Hyperparamètre :

Le déploiement de ce premier modèle sera mené en choisissant les métriques suivantes : Puisque nous modélisons la tâche VQA comme une classification multi-classes, il est naturel d'utiliser *Cross-Entropy Loss* comme fonction de perte, ainsi que l'optimiseur "Adam". Le modèle devait s'entraîner sur 10 époques (epochs) avec un batch size=32 et une dimension=512. Aussi, L'hyper paramètre de décrochage (DropOut) est réglé à 0,5 .

4.4.1.3 Résultats expérimentaux

4.4.1.3.1 Accuracy et loss du modèle

Les deux figures suivantes illustrent les critères de performances qui ont été mesuré sur l'ensemble de données d'apprentissage pour la problématique de réponse aux questions visuels des images médicales.

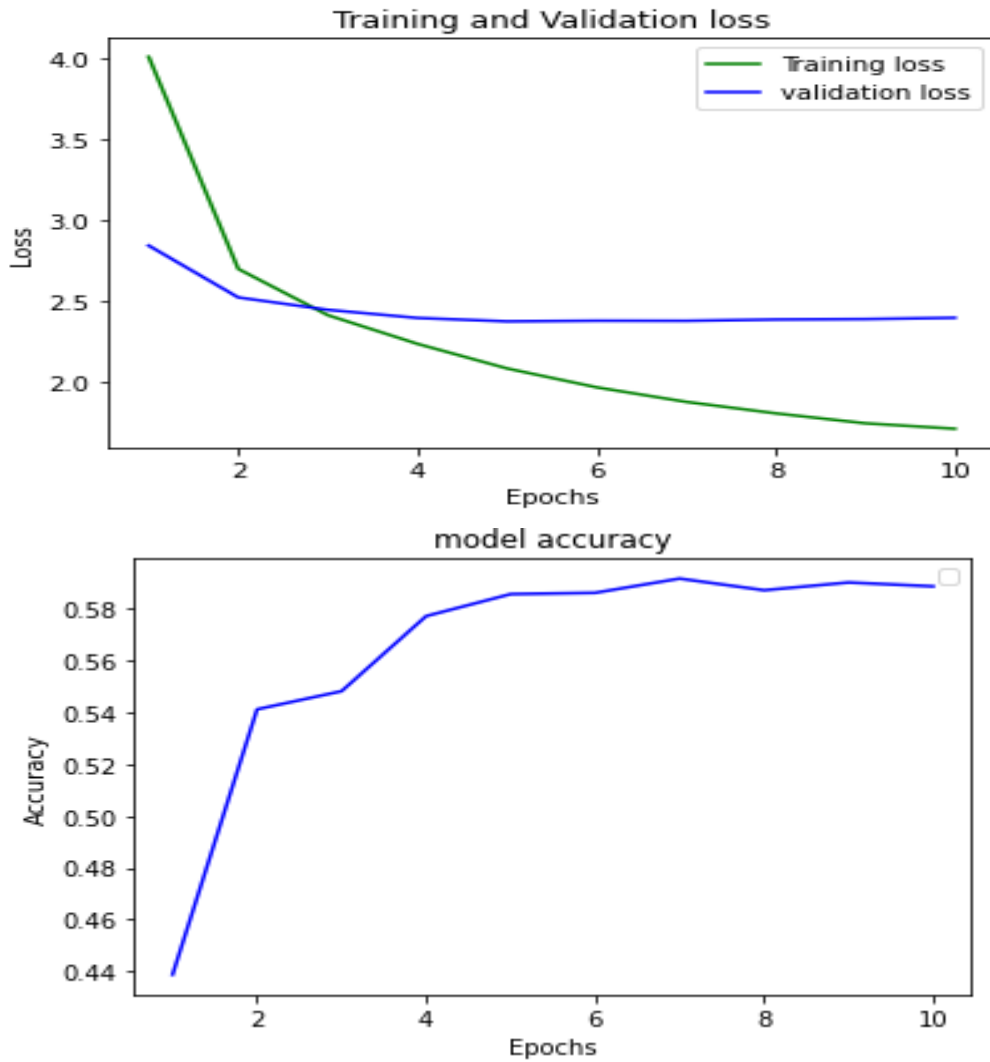


Figure 46: Tracé des indices de performance et de perte relatifs au modèle Bert +Vit

Sur la figure ci-dessus, nous illustrons le tracé de la perte et de la précision du modèle Bert+Vit. Nous distinguons l'étiquette bleue qui indique une précision de la phase de validation (Accuracy) de 59%. En fait, la précision de la validation d'apprentissage du modèle augmente progressivement à chaque époque. Nous notons une différence entre la perte d'apprentissage (1.87) et la perte de validation (2.37)

4.4.1.3.2 Evaluation :

Les performances d'algorithme, en termes de réponse aux questions visuels, sont détaillées dans le tableau suivant afin d'évaluer l'apprentissage et la validation de ce premier modèle.

WUPS	ACCURACY	PRECISION	RECALL	LOSS
59.35	59.17	56.33	67.61	40%

Tableau 1: Performance de Bert+Vit en termes des indices Wups, Accuracy, Précision, Recall, Loss

Nous obtenons un taux de 59,35% pour le score Wups qui est un taux plus au moins satisfaisant avec une perte (Loss) qui s'élève à 40 %. A partir de ces performances, nous pouvons interpréter ces résultats comme étant moyennes. De ce fait, nous déduisons que le modèle Bert+ViT n'est pas suffisamment capable de résoudre le problème de VQA dans le domaine médical.

4.4.1.3.3 Résultats de VQA(inférence) :

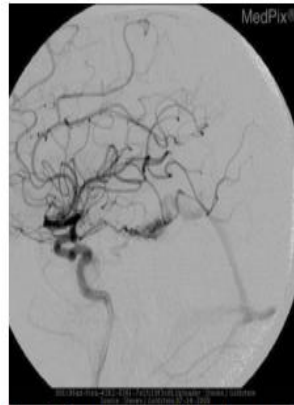
Nous avons sélectionné un échantillon d'images issu de la base de données initiale qui a été testé sur le modèle Bert+Vit. Sur les figures suivantes, nous illustrons l'image de test, la question et la réponse originale de cette image. Aussi, nous illustrons la prédiction déduite du modèle en question. Nous pouvons remarquer des cas de ressemblance et d'autres cas de non ressemblance entre les réponses réelles et la prédiction du modèle.



Question: what is abnormal in the x-ray?

Answer: ankylosing spondylitis

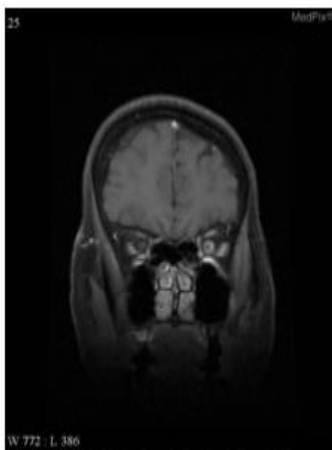
Predicted Answer: ankylosing spondylitis



Question: what is most alarming about this angiogram?

Answer: arteriovenous malformation (avm)

Predicted Answer: arteriovenous malformation (avm)



Question: what abnormality is seen in the image?

Answer: meningiomas

Predicted Answer: meningioma



Question: what organ system is visualized?

Answer: genitourinary

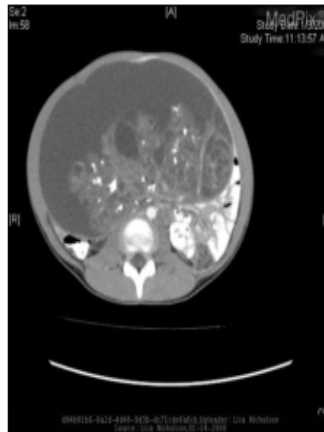
Predicted Answer: genitourinary



Question: what is most alarming about this x-ray?

Answer: spinal osteochondroma

Predicted Answer: rickets



Question: what abnormality is seen in the image?

Answer: immature teratoma

Predicted Answer: acute appendicitis



Question: what organ system is evaluated primarily?

Answer: lung, mediastinum, pleura

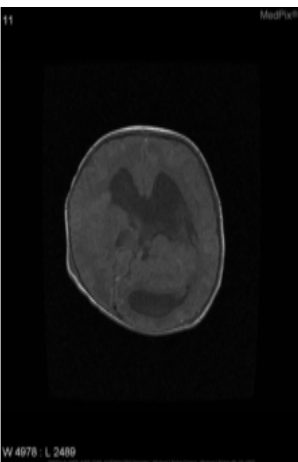
Predicted Answer: lung, mediastinum, pleura



Question: is this a ct scan?

Answer: no

Predicted Answer: yes



Question: what abnormality is seen in the image?

Answer: schizencephaly

Predicted Answer: bas - barium swallow



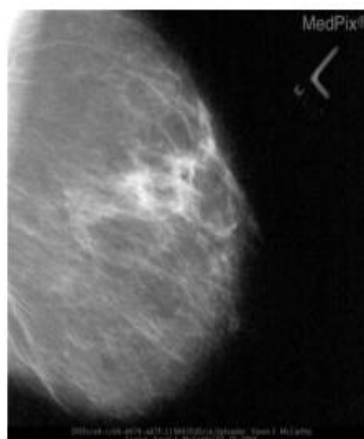
Question: what imaging modality is used?

Answer: xr - plain film

Predicted Answer: xr - plain film



Question: what is abnormal in the ct scan?
 Answer: ossification of stylohyoid ligament
 Predicted Answer: blowout fracture of orbit



Question: what organ is this image of?
 Answer: breast
 Predicted Answer: breast



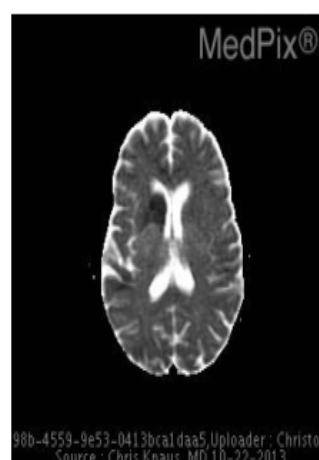
Question: is this a contrast or noncontrast ct?
 Answer: noncontrast
 Predicted Answer: contrast



Question: what imaging modality is used?
 Answer: xr - plain film
 Predicted Answer: xr - plain film



Question: what organ system is shown in the image?
 Answer: musculoskeletal
 Predicted Answer: musculoskeletal



Question: what part of the body is being imaged?
 Answer: skull and contents
 Predicted Answer: skull and contents

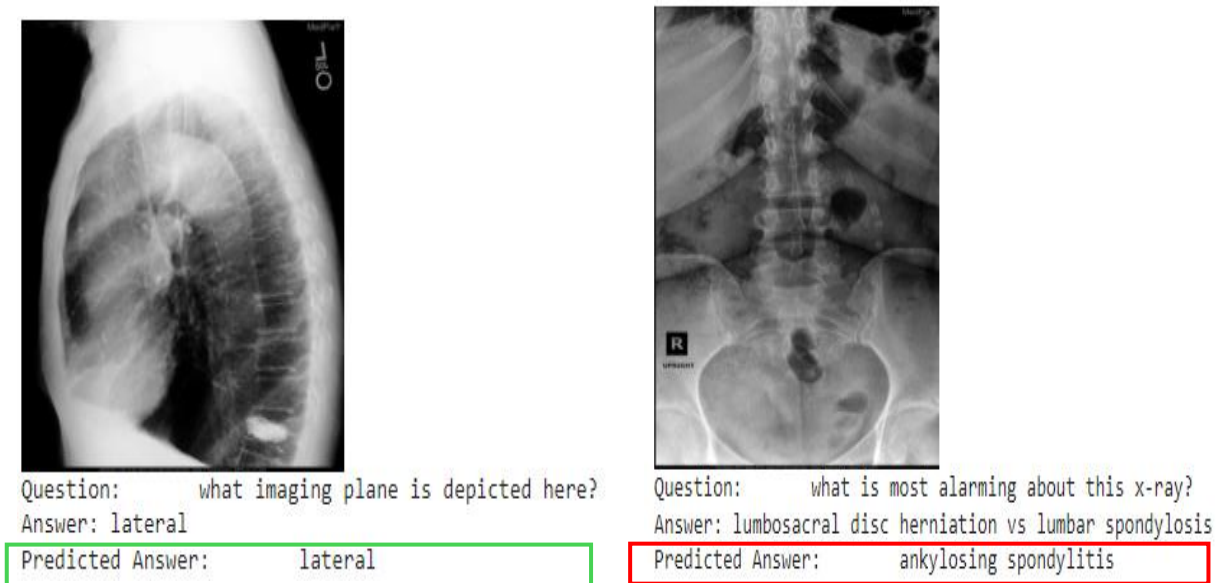


Figure 47: Un exemple de VQA en appliquant le modèle BERT-ViT

4.4.2 Implémentation de BioBERT+ViT

Ici on a choisi de garder la même architecture du modèle, juste on va utiliser le sous-modèle pré-entraîné BioBERT (Bidirectional Encoder Representations from Transformers for Biomedical Text Mining) pour extraire les caractéristiques sémantiques d'une question donnée., BioBERT est un modèle de représentation du langage spécifique au monde médical, pré-entraîné sur des corpus biomédicaux à grande échelle. Avec presque la même architecture, BioBERT surpasse largement BERT et les modèles de l'état de l'art précédents dans une variété de tâches d'exploration de texte biomédical. Parmi ces tâches, on peut citer : la reconnaissance d'entités nommées biomédicales, extraction de relations biomédicales et réponse à des questions biomédicales[20] .

Afin d'extraire les caractéristiques textuelles qui peuvent représenter la phrase de question, nous encodons la phrase de question pour obtenir les ids `_input`, l'attention `mask` et `token_type` ids, puis nous les entrons dans BioBERT.

4.4.2.1 Résultats expérimentaux

4.4.2.1.1 Accuracy et loss du modèle

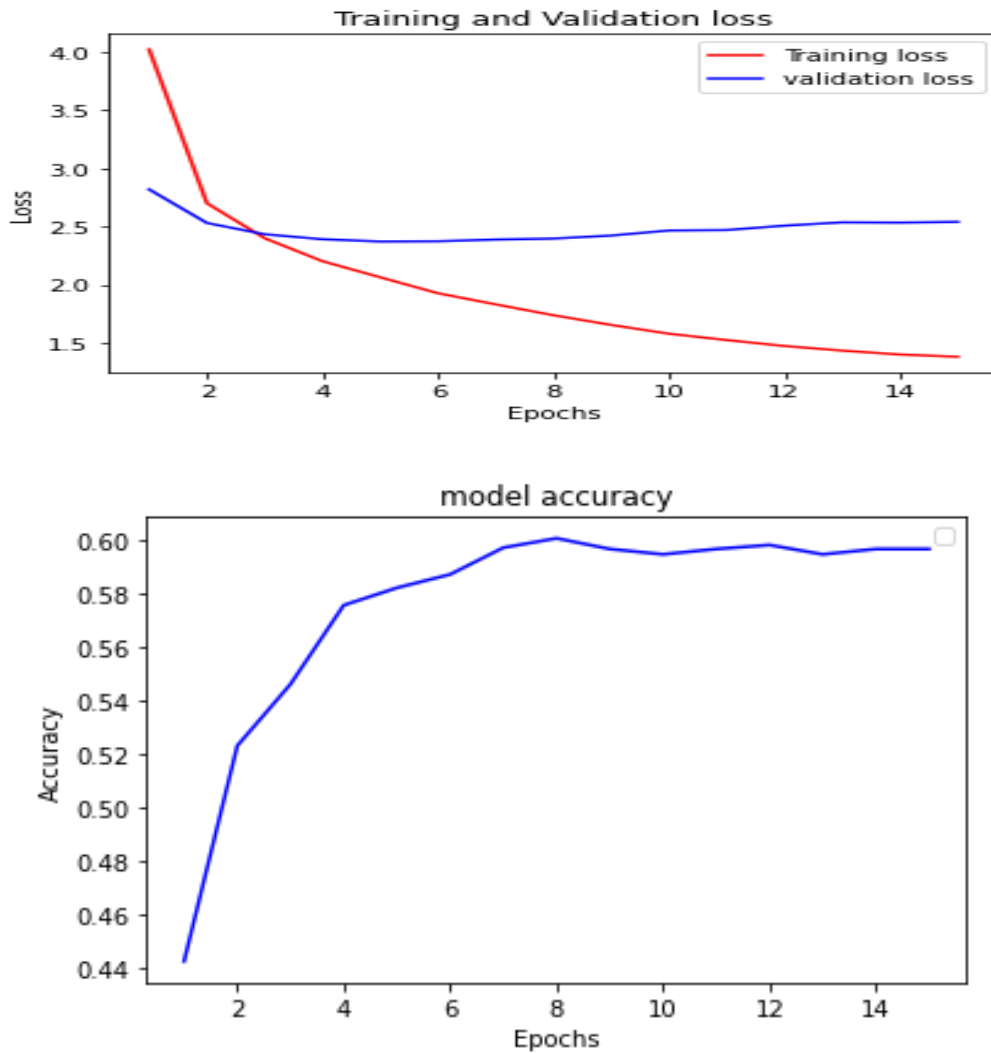


Figure 48: Tracé des indices de performance et de perte relatifs au modèle BioBERT+ViT

4.1.1.1.1 Evaluation

WUPS	ACCURACY	PRECISION	RECALL	LOSS
60.24	60.08	60.73	72.93	40%

Tableau 2: Performance de BioBERT+Vit en termes des indices Wups, Accuracy, Precision, Recall, Loss

Sur le tableau suivant, nous avons présenté les résultats collectés de différents indices de performance. En comparant ces résultats avec ceux du modèle BERT-ViT, nous pouvons déduire que les performances du modèle BioBERT sont plus optimales à celle du modèle BERT.

4.1.2 Implémentation de BioBERT+Swin Transformer

Les limites du ViT :

Le modèle ViT a connu un énorme succès et a été capable de surpasser la plupart des architectures telles que le CNN . Cependant, ce modèle ne pouvait bien résoudre que des tâches de classification simples et ne réussit pas bien à se généraliser dans des tâches de vision complexes .

Les chercheurs ont introduit le modèle Swin Transformer pour résoudre ce problème. qui a introduit des fenêtres glissantes (utilisées dans les CNN) dans les « Transformer », ce qui les fait ressembler aux ConvNets .

Parmi les problèmes de ViT , on peut citer :

- Complexité de calcul quadratique. Cette complexité est due au fait que le modèle ViT calcule l'auto-attention de manière globale. Au fur et à mesure que nous augmentons la taille des images, le temps de calcul augmente de façon quadratique.
- Les ViTs divisent d'abord les images en patches pour garder la longueur de la séquence dans les limites de calcul. Cela ne pose pas beaucoup de problèmes lors de la résolution de tâches de classification. Mais ce processus est devenu un véritable problème pour les tâches qui nécessitent un traitement détaillé pour chaque pixel, comme dans la détection d'objets et la segmentation sémantique. En raison de sa complexité de calcul, le modèle a rencontré des difficultés lorsqu'il s'agissait de résoudre des tâches de vision par ordinateur plus générales et plus denses.

Architecture de Swin Transformer :

Le modèle Swin Transformer est un modèle étudié par l'équipe de recherche de Microsoft en Asie[21]. Le mot Swin est un acronyme qui signifie "fenêtre décalée". Ce concept de fenêtre décalée n'est pas nouveau pour la communauté des chercheurs. Il est utilisé dans les CNN depuis de nombreuses années. C'est l'une des caractéristiques des CNN qui leur a permis d'exceller dans le domaine de la vision par ordinateur, car elle a apporté une grande efficacité. Cependant, elle n'avait jamais été utilisée dans les « Transformer » auparavant.

Ce modèle utilise toujours les patches comme dans le modèle ViT. Cependant, au lieu d'utiliser une taille unique comme dans ViT (16 par 16px), le Swin Transformer commence par de petits patches dans la première couche du « Transformer ». Le modèle fusionne ces couches en de plus grandes dans les couches plus profondes. Il prend une image et la divise en patches de 4px par 4px. Chaque patch est une image colorée à trois canaux. Ainsi, un patch

a une dimensionnalité totale de 48 caractéristiques. C'est-à-dire $4 \times 4 \times 3 = 48$. Il est ensuite transformé linéairement en une dimensionnalité appelée C, de votre choix.

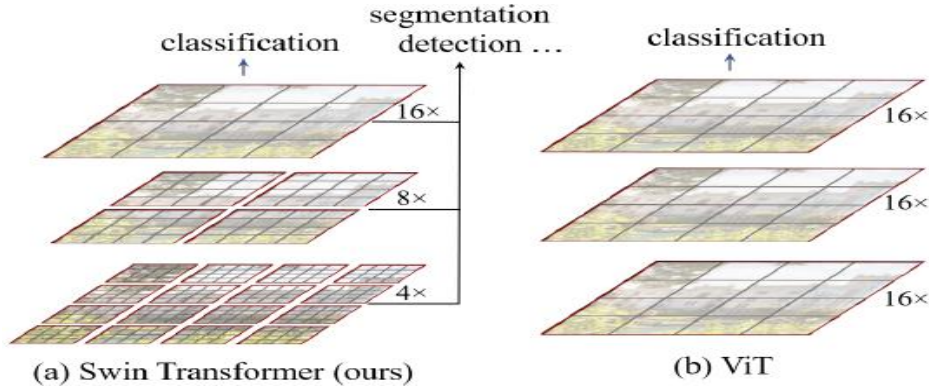


Figure 49: Différence entre ViT et Swin Transformer

Jusqu'à présent, par rapport aux ViTs, les patches d'image sont de plus petite taille. La valeur C, détermine la taille de votre modèle Transformer. nous avons différentes variantes de modèle Swin Transformer: Une variante Swin-Tiny avec C=96, et le Swin-Large avec C=192. Dans ce cas, le C est 96 et 192. La valeur, C détermine le nombre de paramètres cachés dans les couches entièrement connectées.

Swin Transformer calcule l'auto-attention uniquement dans la fenêtre locale et non globalement comme c'est le cas avec le modèle ViT. La sortie est ensuite fusionnée par une couche de fusion, elle concatène les vecteurs des groupes de patches 2x2 voisins dans l'image. À chaque fois, la fenêtre d'attention se déplace par rapport à la couche précédente. Par exemple, si dans la première couche l'attention était limitée au voisinage de ces régions, dans la couche suivante, les régions sont déplacées (comme dans la convolution stridente).

Les patches qui ont atterri dans des fenêtres séparées dans la première couche et ne pouvaient pas communiquer, peuvent maintenant le faire dans la deuxième couche. Ces patches résultants sont fusionnés par la couche de fusion. Ce processus est répété en fonction du nombre de couches choisi.

Voici un résumé de l'architecture du Swin Transformer :

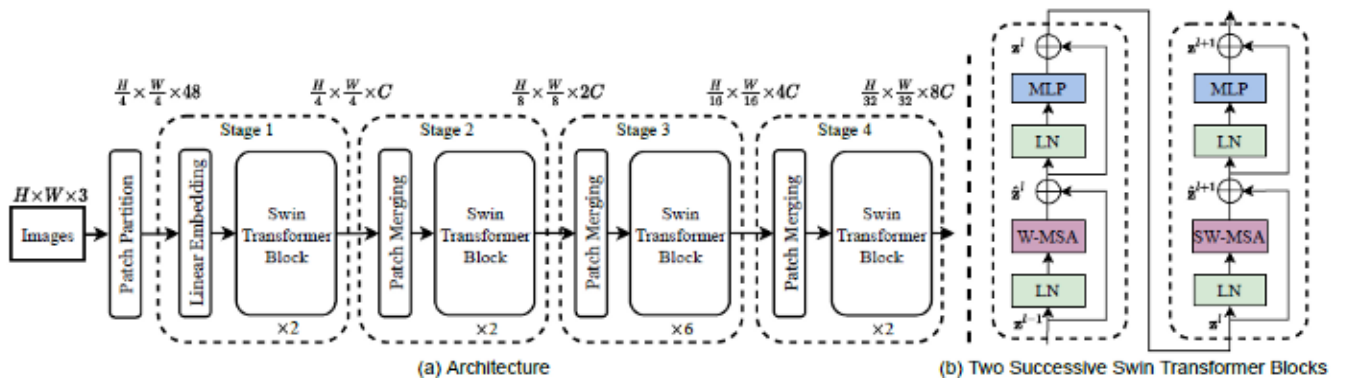


Figure 50: Architecture de Swin transformer

4.1.2.1.1 Accuracy et loss du modèle

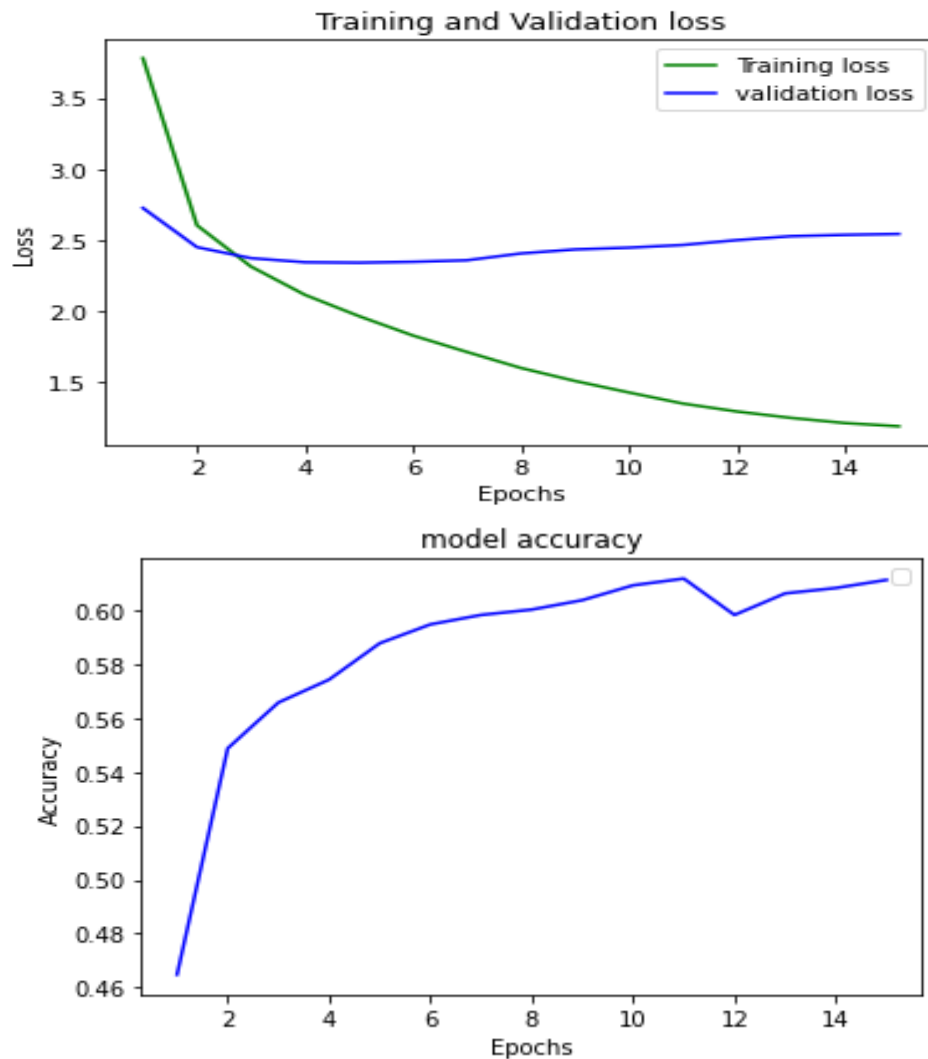


Figure 51: Tracé des indices de performance et de perte relatifs au modèle BioBERT+Swin Transformer

4.1.2.1.2 Evaluation

WUPS	ACCURACY	PRECISION	RECALL	LOSS
61.29	61.18	70.42	80.63	37%

Tableau 3: Performance de BioBERT+Swin Transformer en termes des indices Wups, Accuracy, Precision, Recall, Loss

4.2 Comparaison des résultats

Evaluation des résultats :

Les performances des trois algorithmes, en termes de réponse aux questions visuels, sont détaillées sur le tableau suivant. Ainsi, nous avons mené la comparaison en utilisant les critères d'évaluation suivants : RECALL,PRECISION,ACCURACY et WUPS.



	ACCURACY	WUPS	PRECISION	RECALL
BERT+ViT	59.17	59.35	56.33	67.61
BioBERT+ViT	60.08	60.24	60.73	72.93
BioBERT+Swin Transformer	61.18	61.29	70.42	80.63

Tableau 4: Tableau comparatif des performances des trois modèles implémentés

Interprétation des résultats :

Les valeurs maximales des indices d'évaluation utilisés dans ces runs indiquent que les valeurs enregistrées par le modèle Swin Transformer avec BioBERT proposé dans ce travail sont les plus élevées.

ImageCLEF 2019 Challenge :






Δ #	Participant	Accuracy (Strict)	BLEU
● 01.	 Hanlin	0.624	0.644
● 02.	 yan	0.62	0.64
● 03.	 minhvu	0.616	0.634
● 04.	 TUA1	0.606	0.633
▲ 05.	 UMMS	0.566	0.593

Figure 52: Les résultats officiels de ImageCLEF@VQA-Med-2019

le tableau ci-dessous présente les résultats des 14 équipes participantes à ImageCLEF 2019 challenge. Le meilleur résultat global a été obtenu par l'équipe Hanlin, avec une précision de 0,624 et un score BLEU de 0,644[22].

<i>Groupe</i>	<i>Run ID</i>	<i>Modèles</i>	<i>Accuracy</i>
Halin	26889	VGG16 + BERT	0.624
Yan	26853	non mentionné	0.620
Minhvu	26881	Resnet + BERT	0.616
TUA1	26822	Resnet + BERT	0.606
UMMS	27306	Resnet + LSTM	0.566
AIOZ	26873	non mentionné	0.564
IBM Research AI	27199	VGG16 + LSTM	0.558
LIST	26908	Densenet + LSTM	0.556
Turner.JCE	26913	VGG16 + LSTM	0.536
JUST19	27142	non mentionné	0.534
Team-Pwc-Med	26941	Resnet + LSTM	0.488
Techno	27079	VGG16 + LSTM	0.462
Dear stranger	26895	non mentionné	0.210
Abhishekthanki	27307	VGG16 + LSTM	0.160

Tableau 5: VQA-Med-2019 challenge accuracy

La précision de notre modèle se situe en 4eme place, ce qui montre que nos résultats sont satisfaisants.

4.3 Conclusion

Dans ce chapitre, nous avons implémenté trois modèles profonds de de réponse aux questions visuels basé sur Transformer et nous avons comparé et interprété les résultats obtenus.



CONCLUSION GENERALE

Ce projet de fin d'études avait pour but de comparer et d'évaluer les variantes d'apprentissage multimodal profond issu du modèle « Transformer » capable de répondre à des questions en langage naturel à partir du contenu visuel des images de radiologie associées. Principalement, l'objectif est de justifier l'intérêt du « Transformer » face aux réseaux de neurones convolutionnels et récurrentes lors de la combinaison de la vision et du NLP. Ce travail a été divisé en quatre grandes parties: tout d'abord, nous avons fait une étude générale du projet, en définissant le problème de la multimodalité, nous avons introduit les systèmes de réponse aux questions visuelles, ainsi que ses domaines d'application et plus précisément le domaine médical. En outre, nous avons présenté l'étape de la fusion entre les modalités.

Ensuite, nous avons fait une étude sur l'imagerie médicale en mentionnant ses différentes modalités ; nous avons également fait une étude comparative entre les différents ensembles des données de VQA médicaux disponibles. Dans le troisième chapitre, nous avons décrit les notions de base du modèle « Transformer ». Ensuite, nous avons présenté deux sous-modèle issus du « Transformer » pour le traitement du langage naturel et pour la vision par ordinateur. Et finalement, Dans le quatrième chapitre, nous avons décrit l'implémentation de trois modèles pré-entraînés profonds appliqués dans la réponse aux question visuels: le BERT, le BioBERT, le ViT et le Swin Transformer. En terminant, à partir de différents tests, nous avons procédé à l'évaluation des trois modèles implémentés tout en utilisant des critères de performance.

En résumé, A partir des résultats obtenus, nous avons réussi à résoudre la problématique de VQA via un modèle de « Transformer » multimodale avec une fusion tardive sous PyTorch. Le modèle que nous avons proposé avec BioBERT et Swin Transformer est celui qui a donné les résultats optimaux avec une précision de 0,611. Néanmoins, cette tâche reste difficile et ardue, Cette difficulté augmente encore plus avec la nature inhérente de l'imagerie médicale.

En guise de perspective, ce travail reste prêt pour toute amélioration envisageable: Pour les travaux futurs, nous continuerons à améliorer les performances de notre modèle actuel, en

Introduisant des méthodes plus avancées de sorte que la tâche VQA sera traité comme un problème de génération au lieu de problème classification multi-classe et les appliquant à d'autres ensembles de données. D'autre part, nous pouvons penser à intégrer notre modèle dans une application web pour donner plus de valeur à notre solution.

Bibliographie & Webographie

- [1] Aishwarya Agrawal*, Jiasen Lu*, Stanislaw Antol*, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, Devi Parikh. VQA: Visual Question Answering. arXiv:1505.00468v7 [cs.CL] 27 Oct 2016, **consulté le 17/04/2022.**
- [2]<https://www.indexsante.ca/chroniques/284/imagerie-medicale.php>, **consulté le 22/08/2022.**
- [3]<http://dspace.univmsila.dz:8080/xmlui/bitstream/handle/123456789/6602/620.pdf?sequence=1&isAllowed=y>, **consulté le 22/08/2022.**
- [4]<https://www.cea.fr/comprendre/Pages/sante-sciences-du-vivant/essentiel-sur-imagerie-medicale.aspx>, **consulté le 22/08/2022.**
- [5] <https://naxos.biomedicale.univ-paris5.fr/diue/>, **consulté le 10/08/2022.**
- [6]<https://actualitte.com/article/23712/technologie/securite-les-aeroports-americaains-des-rayons-x-a-la-crise-de-zele>, **consulté le 10/09/2022.**
- [7]<https://www.aredoc.com/index.php/publication/imagerie-medicale-de-levolution-a-la-revolution/>, **consulté le 25/08/2022.**
- [8] Zhihong Lina, Donghao Zhangb, Qingyi Taoc, Danli Shid, Gholamreza Haffarie, Qi Wuf, Mingguang Heg and Zongyuan Geb,h,i,* .Medical Visual Question Answering: A Survey. arXiv:2111.10056v2 [cs.CV] 19 Jan 2022, **consulté le 06/04/2022.**
- [9] Hasan, S.A., Ling, Y., Farri, O., Liu, J., Müller, H., Lungren, M.P., 2018. Overview of ImageCLEF 2018 medical domain visual question answering task., in: CLEF (Working Notes). **consulté le 03/09/2022.**
- [10] Lau, J.J., Gayen, S., Abacha, A.B., Demner-Fushman, D., 2018. A dataset of clinically generated visual questions and answers about radiology images. Scientific Data 5, 1–10. **consulté le 03/09/2022.**
- [11] Ben Abacha, A., Hasan, S.A., Datla, V.V., Liu, J., Demner-Fushman, D., Müller, H., 2019. VQA-Med: Overview of the medical visual question answering task at imageclef 2019, in: CLEF2019 Working Notes, CEUR-WS.org, Lugano, Switzerland. **consulté le 03/09/2022.**

- [12] He, X., Zhang, Y., Mou, L., Xing, E., Xie, P., 2020. PathVQA: 30000+ questions for medical visual question answering. arXiv preprint arXiv:2003.10286. **consulté le 05/09/2022.**
- [13] Ben Abacha, A., Datla, V.V., Hasan, S.A., Demner-Fushman, D., Müller, H., 2020. Overview of the VQA-Med task at ImageCLEF2020: Visual question answering and generation in the medical domain, in: CLEF 2020 Working Notes, CEUR-WS.org, Thessaloniki, Greec. **consulté le 10/05/2022.**
- [14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin , Attention Is All You Need. arXiv:1706.03762v5 [cs.CL] 6 Dec 2017, **consulté le 11/03/2022.**
- [15] <https://jalammar.github.io/illustrated-transformer/>, **consulté le 11/03/2022.**
- [16] <https://www.lebigdata.fr/traitement-naturel-du-langage-nlp-definition>, **consulté le 27/08/2022.**
- [17] <https://datascientest.com/introduction-au-nlp-natural-language-processing>, **consulté le 27/08/2022.**
- [18] J. Devlin, M.-W. Chang, K. Lee et K. Toutanova, “Bert : Pre-training of deep bidirectional transformers for language understanding,” arXiv preprint arXiv :1810.04805, 2018, **consulté le 03/09/2022.**
- [19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby . An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929v2 [cs.CV] 3 Jun 2021, **consulté le 08/04/2022.**
- [20] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. arXiv:1901.08746v4, **consulté le 06/05/2022.**
- [21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. arXiv:2103.14030v2 [cs.CV] 17 Aug 2021, **consulté le 27/08/2022.**
- [22] <https://www.aicrowd.com/challenges/imageclef-2019-vqa-med/leaderboards>, **consulté le 17/06/2022.**

Épigraphe



« Tu peux tous accomplir dans la vie si tu as le courage de rêver, l'intelligence d'en faire un projet réaliste, et la volonté de voir ce projet mené à bien. »

Vauvenargues