



Impact de la représentation de l'information sur les performances et la validation des modèles d'apprentissage profond

Rebecca Leygonie

► To cite this version:

Rebecca Leygonie. Impact de la représentation de l'information sur les performances et la validation des modèles d'apprentissage profond. Intelligence artificielle [cs.AI]. Université Paris Cité, 2024. Français. ⟨NNT : 2024UNIP7115⟩. ⟨tel-05045347⟩

HAL Id: tel-05045347

<https://theses.hal.science/tel-05045347v1>

Submitted on 24 Apr 2025

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

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



HAL Authorization

Université Paris Cité, Laboratoire d'Informatique Paris Descartes (LIPADE, Équipe SIP)
École Doctorale Informatique, Télécommunications et Électronique (EDITE, ED130)
Centre d'Expertise Données & Intelligence Artificielle (CEDIA) du Ministère des Armées

Impact de la représentation de l'information sur les performances et la validation des modèles d'apprentissage profond

Par **REBECCA LEYGONIE**

Thèse de doctorat en :
INFORMATIQUE

Spécialité :
TRAITEMENT DU SIGNAL ET DES IMAGES

Dirigée par **LAURENT WENDLING**

Présentée et soutenue publiquement le 10/12/2024

Devant un jury composé de :

ROLF INGOLD, PROF. ÉMÉRITE	Université de Fribourg,	Rapporteur
LARBI BOUBCHIR, PROF.	Université Paris 8	Rapporteur
SALIMA BENBERNOU, PROF.	Université Paris Cité	Présidente du Jury
CHARLOTTE PELLETIER, MCF	Université Bretagne Sud	Examinatrice
LAURENT WENDLING, PROF.	Université Paris Cité	Directeur de thèse
SYLVAIN LOBRY, MCF	Université Paris Cité	Membre invité
GUILLAUME VIMONT, PhD	Ministère des Armées	Membre invité

À Jacob.

Remerciements

Je tiens tout d’abord à remercier le Secrétariat Général pour l’Administration (SGA) du Ministère des Armées et des Anciens Combattants pour avoir rendu possible cette recherche par leur investissement. Je suis particulièrement reconnaissante envers Guillaume et Valérie d’avoir veillé à ce que je puisse réaliser ma thèse dans les meilleures conditions.

Je souhaite remercier les rapporteurs, Larbi Bouchir et Rolf Ingold, pour leurs remarques très pertinentes sur le présent manuscrit. Je remercie aussi les examinatrices, Charlotte Pelletier et Salima Benbernou qui suivent mes travaux depuis le début.

Bien entendu, je remercie mes encadrants, Laurent et Sylvain, de pour m’avoir donné l’opportunité de réaliser cette thèse. Merci à vous pour votre patience et votre bienveillance qui m’ont permis de m’épanouir totalement dans mes travaux. Votre accompagnement a été un atout précieux dans l’accomplissement de ce travail.

Plus précisément, merci Laurent d’être toujours optimiste et de trouver une issue à chaque problème. Quant à Sylvain, merci d’avoir compris très rapidement que le stress faisait partie intégrante de ma personne, d’avoir su le gérer, et surtout d’en rigoler avec moi.

Merci aux membres de l’équipe SIP, avec lesquels j’ai passé de super mardi/mercredi, bien qu’en chaises musicales. Plus précisément, merci à Olivier, Nathan, Robin, Momo, Lucrezia, Hichem et bien sûr Basile, sans qui aucune information administrative ne serait parvenue jusqu’à moi.

Toujours dans un contexte académique, je souhaite remercier Monsieur Mariage, mon professeur en master, pour m’avoir montré ce qu’était la passion de la recherche, car c’est cette vision qui m’a donné envie de faire une thèse.

Ma thèse m’a permis de faire de nombreuses et fructueuses rencontres. J’ai notamment eu le plaisir de rencontrer Milena Suarez Castillo, membre du Lab Santé de la Direction de la Recherche, des Études, de l’Évaluation et des Statistiques (Drees). Cette rencontre a donné lieu à une convention pour des travaux sur les données du Système National des Données de Santé, et je t’en suis très reconnaissante.

Je remercie tous les membres du Lab Santé avec qui j’ai eu le plaisir d’interagir. Je souhaite tout particulièrement exprimer ma gratitude envers Javier Nicolau et Gladys Baudet qui m’ont aidée à établir des ponts entre mes travaux et le domaine médical.

Enfin, je remercie sincèrement les experts médicaux Albert Vuagnat, Constance Prieur et Diane Naouri pour leur patience et leurs efforts de compréhension lorsque je tentais de transposer des concepts de deep learning dans le contexte médical.

Ma thèse a été initiée par mon entrée au sein du Centre d'Expertise Données & IA (anciennement Labo BI) du Ministère des Armées. Pour cela, je remercie Guillaume et Valérie, d'avoir cru en moi, il y a plus de 4 ans, lorsque que je cherchais un stage, sans connaître le nom de la Ministre. Merci à Valérie pour ta bienveillance, parfois même maternelle, toi qui as fait un peu du « Labo BI » ma deuxième famille. Merci à Guillaume de donner libre cours à mes envies de recherche et à mes idées alambiquées, merci de me laisser la liberté de m'épanouir dans ce qui me passionne et de me faire confiance.

Merci à tous les membres du « Labo », avec qui les souvenirs sont infinis, dont parmi eux, les déjeuners avec vue sur Paris, les journées cohésion, les Time's Up et bien sur les stand-up ...

Plus particulièrement, merci à Hugues, Princina, Alix, Benoit et Guillaume N°2.

Viviane, Théo, Jul, le bureau 4038, mes « collègues », mes amis. Sans votre présence ces 3 années n'auraient pas été aussi mémorables. Merci Viviane de me soutenir dans tous mes choix, de compenser mes lacunes dans les choses pratiques du quotidien, d'être à l'écoute et de répondre au téléphone à n'importe quel moment de la journée. Merci Jul de ne pas me soutenir dans tous mes choix, ce qui m'a permis de rationaliser beaucoup de situations. Merci Théo, mon binôme doctorant, pour ton honnêteté, souvent pas dosée. Je te discerne le titre de gagnant du carnet des problèmes. Merci à vous pour tous ces fous rires, pour toutes nos longues discussions sur la vie, l'avenir du labo, et évidemment les croustis.

Emma Kopp, mon ancienne colloc, mon amie d'enfance, merci de toujours me rappeler à la mesure quand je « ne dose pas », merci pour les popotes, les soirées devant HPI, et les domac en rentrant de soirée.

Merci Olivia de m'avoir appris à gérer des situations d'adulte, de m'avoir si bien conseillée tout en gardant tes avis tranchés. Merci à la famille Pariente-Cebula, à qui je dois la plupart de mes articles, rédigés dans le cadre merveilleux de Formentera.

"Le thé" merci pour ces facetimes "tous" les dimanches des 4 coins du monde, vous m'aurez bien fait rigoler. Benjamin, merci de me divertir avec tes expériences loufoques comme rester 10 jours assis sur un plot, ta détermination est source d'inspiration. Ann Perryn, merci pour tes expressions verbales et faciales plus surprenantes et hilarantes les unes que les autres, ton audace et ton "self-love" sont sources d'admiration.

Merci aux organisateurs du colloque ORASIS de m'avoir permis de tomber, sans mauvais jeu de mots, après un malaise vagal sur la presqu'île de Giens, sur Alexandre.

Alex, plus communément connu sous le nom de « ma vie », merci d'avoir rendu ma dernière année de thèse si légère. Le combo entre nos moments de lores en conférences et nos week-ends à Henridorff étaient tout ce dont j'avais besoin. Tu m'as appris à "aimer" les randonnées (petite pensée pour Nico et Martial), chiller dans les maisons rustiques en Cévennes, m'installer dans les restaurants familiaux des Vosges du Nord et bien sûr, savourer le Munster.

A mes amis d'Orasis, de Peyresq, et aux amis de mes amis d'Orasis et de Peyresq, merci de m'avoir montré que je n'étais pas la seule pour qui la thèse n'était pas un long fleuve tranquille. Ce lore autour de la recherche française m'a bien régalié et ce n'est que le début.

Merci à ma famille, famille de docteurs, d'être si inspirante. Nana, Dédé, vous m'avez appris la valeur de l'épanouissement dans la réussite.

Amatxi et Aïtatxi, c'est un honneur de réaliser ma thèse dans l'université dans laquelle vous avez fait vos années de médecine.

Papa, tu m'as appris la rigueur, l'ambition et la valeur de la persévérance. Si un jour je veux devenir Présidente de la République, sois sûr que je te demanderai des séances de coaching.

Maman, merci d'être si inspirante, dans ta vie professionnelle autant que personnelle. Merci de me donner du contenu à raconter à mes amis du bureau, qui n'ont que les mots « Claudia Senik » à la bouche. Merci de penser que je suis un génie dès que j'arrive à régler un problème tel que le navigo de Martin. Merci de me montrer un exemple que j'ai envie de suivre, d'être si fière de Jacob Martin et moi et de répondre à mes 10 coups de fils par jour. Merci d'être là.

Jean Olivier, toi qui m'as forcée à faire mes devoirs au collège, grâce à toi je connais la définition des phrases alambiquées et l'importance d'avoir des *Figolus* dans son sac quand on part en journée sport avec toi !

Merci à ma très grande fratrie, Jacob, Adrien, Martin, la deuxième portée de papa, Tamina, Thomien, Max, et la troisième portée, Danil et Polina.

Jacob, tu as été un pilier essentiel à ces trois années, et dans ma vie en général. Merci d'avoir écouté mes idées de recherche, de les avoir challengées pour que j'arrive au bout de mon raisonnement et de m'avoir appris à modéliser mathématiquement mes problèmes. Merci pour nos semaines « TT » à Turin, en Sardaigne et bien sûr à l'Île de Ré, qui ont été les plus productives de ma thèse. Merci pour tes brillants jeux de mots, nos crêpes nut' rue Mouffetard, les déménagements en UberVan et les petits frichtis HelloFresh pratiquement tous les soirs de la semaine quand j'habitais rue Lacépède. Merci pour ta présence. Merci d'être mon exemple depuis toujours.

Adrien, mon deuxième frère docteur, merci de me rappeler tous les jours que j'ai de la chance d'être si bien encadrée dans mes travaux de recherche et pour tous ces moments qui m'ont permis de penser à autre chose : nos après-midi crapette, nos planches à l'institut, nos bières aux *Sciences*, nos escapades à Lisbonne, Deauville et à l'Île de Ré.

Martin, mon petit frère Bordeliskhou, merci de me rendre si fière, d'avoir ton nouveau coté "clanique", de m'entraîner au Kumo et au Set, de ne pas dire que je suis nulle au tennis, et surtout, d'accepter de ne pas jouer les mêmes morceaux de piano que moi, pour que maman aime aussi quand je joue.

Tamina, ma petite sœur adorée, merci de penser que ce que je fais est toujours bien, même quand c'est la cata. J'espère arriver un jour à la hauteur de l'estime que tu me portes.

Je remercie tout mon entourage, qui a fortement contribué à mon bien être pendant cette thèse : Albert, Dominique, Evelyne, Dasha, Vincent, Armelle, Juliette, David, Eliot, Anouk, Eugène, Raph, Alban et mes amis, Blumsi noodles, Ann Pergola, Lulu, Nem, Em la crem, Mitsou, Dende, Rubz, Louis, Did, Anna, Pat, Elaïa, Adri2, Esther [IndexError : list index out of range...]

Pour finir, je tiens à rendre hommage à mes arrières grands parents maternels Berek, Sura, Moshè et Rywka, dont je porte fièrement le prénom. L'histoire et le courage de ces fondateurs de la famille « Senik-Rozencwajg » ont été une source d'inspiration et de persévérance dans les moments de doute, me donnant la force d'aller jusqu'au bout de cette aventure.

Grâce à toutes les personnes citées, et bien d'autres, ces trois années de thèse, qui auraient pu être les pires, ont été les meilleures de ma vie.

Résumé

Titre : Impact de la représentation de l'information sur les performances et la validation des modèles d'apprentissage profond.

Mots clefs : Représentation de l'information ; Données à haute dimension ; Fléau de la dimension ; Détection d'anomalies ; Explicabilité

Résumé : Les modèles d'apprentissage profond font face à plusieurs défis qui limitent leur efficacité et leur application dans divers domaines. Le fléau de la dimension est l'un de ces défis, il désigne la difficulté croissante d'analyser les données à mesure que le nombre de caractéristiques augmente. Un autre défi auquel font face les modèles est la complexité des données. Celle-ci peut se caractériser par la multitude de dimensions, la non-linéarité des données, et la présence de bruit, altérant ainsi l'identification de l'information utile. Enfin, la validation des modèles d'apprentissage profond pose un défi crucial. Ces modèles sont souvent considérés comme des boîtes noires, car leurs processus de prise de décision sont difficiles, voire impossibles à interpréter.

Pour surmonter ces défis, diverses approches ont été développées, visant soit à intégrer de nouveaux mécanismes adaptés à des types de données spécifiques, soit à améliorer la robustesse et l'interprétabilité globale des modèles. La tendance actuelle s'oriente vers des modèles plus robustes et plus volumineux, caractérisés par un grand nombre de paramètres et de couches, capables de traiter une grande variété de données et de tâches. Cependant, cette augmentation de la complexité structurelle des modèles engendre une évolution du coût computationnel, ce qui soulève de nouveaux défis en termes de ressources et d'efficacité.

Face à ces enjeux, cette thèse se concentre sur l'analyse de l'impact de la représentation

de l'information sur les problématiques liées à l'apprentissage profond. Elle cherche à déterminer si, pour un objectif donné, le choix de la représentation de l'information peut influencer significativement les résultats. Pour répondre à cette question, trois études ont été menées. La première consiste à transposer des données unidimensionnelles (1D) en représentations bidimensionnelles (2D), pour pallier le fléau de la dimension en exploitant les capacités des réseaux de neurones convolutifs sur des données initialement non structurées en 2D. La deuxième étude explore une nouvelle approche de détection d'anomalies basée sur l'analyse des caractéristiques d'importance révélées par des cartes de saillance, obtenues par un modèle d'explicabilité. Enfin, la troisième étude examine la possibilité de valider l'apprentissage d'un modèle de type Transformer en représentant ses poids d'auto-attention sous forme de graphe et en les évaluant par rapport à une référence établie, aussi représentée sous forme de graphe.

Les résultats de ces trois études montrent que la représentation de l'information peut être un levier d'amélioration des modèles d'apprentissage profond. Cela ouvre des perspectives intéressantes pour le développement de solutions plus efficaces, plus explicables et mieux adaptées à une variété de problèmes complexes dans des domaines nécessitant une grande transparence des processus décisionnels, comme la médecine.

Abstract

Title : Impact of information representation on the performance and validation of deep learning models.

Keywords : Information Representation ; High-dimensional Data ; Curse of Dimensionality ; Anomaly Detection ; Explainability

Abstract : Deep learning models face several challenges that limit their effectiveness and application in various fields. The curse of dimensionality is one of these challenges ; it refers to the increasing difficulty of analyzing data as the number of features increases. Another challenge faced by models is the complexity of data. This can be characterized by the multitude of dimensions, the non-linearity of data, and the presence of noise, thus altering the identification of useful information. Finally, the validation of deep learning models poses a crucial challenge. These models are often considered as black boxes, because their decision-making processes are difficult, if not impossible, to interpret.

To overcome these challenges, various approaches have been developed, aimed either at integrating new mechanisms adapted to specific data types, or at improving the robustness and overall interpretability of models. The current trend is towards more robust and larger models, able to handling a wide variety of data and tasks. However, this increase in the structural complexity of models leads to an evolution in computational cost, which raises new challenges in terms of resources and efficiency.

Given these challenges, this thesis focuses on analyzing the impact of information repre-

sentation on deep learning issues. It aims to determine whether, for a given objective, the choice of information representation can significantly influence results. To address this question, three studies were conducted. The first consists of transposing one-dimensional (1D) data into a two-dimensional (2D) representation, to overcome the curse of dimensionality by exploiting the capabilities of convolutional neural networks on data not initially structured in 2D. The second study explores a new approach to anomaly detection based on the analysis of important features revealed by heatmaps, obtained by an explainability model. Finally, the third study examines the possibility of validating the learning of a Transformer-type model by representing its self-attention weights as a graph and evaluating them against an established reference, also represented as a graph.

The results of these three studies show that information representation can be a powerful way of improving deep learning models. This opens up interesting perspectives for the development of more effective, more explainable and better-adapted solutions to a variety of complex problems in fields requiring high transparency of decision-making processes, such as medicine.

Table des matières

Remerciements	i
Résumé	v
Abstract	vii
Liste des figures	xvii
Liste des tableaux	xix
Liste des abréviations	xxi
Introduction	1
1 Apprentissage profond pour le traitement d'images et de données temporelles	7
1.1 Réseaux de neurones convolutifs pour le traitement des images	7
1.1.1 Évolution des réseaux de neurones convolutifs	9
1.2 Réseaux de neurones récurrents pour le traitement des séries temporelles .	15
1.2.1 Évolution des réseaux de neurones récurrents	16
1.2.2 Transformers	18
1.3 Traitement des séries temporelles d'images	25
2 Représentation des données tabulaires en images pour pallier le fléau de la dimension	29
2.1 Introduction aux défis liés au fléau de la dimension	29
2.2 Approches de réduction de dimension	31
2.2.1 Réduction de dimension des données statiques	31
2.2.2 Réduction de dimension des données séquentielles	32
2.2.3 Limites des approches de réduction de dimension	33
2.3 Alternative à la réduction de dimension	34
2.3.1 Génération d'images à partir de données tabulaires	35
2.3.2 Contexte de notre étude	37
2.4 Études des limites des CNN sur des images bruitées	38
2.4.1 Étude de l'information résiduelle nécessaire pour la classification d'objets figuratifs	39
2.4.2 Étude de l'impact du bruit sur la classification d'images abstraites .	43

2.4.3	Conclusion des études préliminaires	50
2.5	Transformation de données séquentielles en images et application d'un modèle hybride	51
2.5.1	Méthodologie	51
2.5.2	Modèle de classification	52
2.6	Expérimentations	53
2.6.1	Construction du jeu de données	53
2.6.2	Des données médicales aux séries temporelles multivariées	54
2.6.3	Modèles et paramétrisation	55
2.7	Résultats et discussions	57
2.8	Conclusion et perspectives	59
3	Représentation des images en cartes de saillance pour la détection d'anomalies	61
3.1	Introduction aux défis liés à la détection d'anomalies	61
3.2	État de l'art	64
3.2.1	Détection d'anomalies au niveau de l'image	64
3.2.2	Test <i>a contrario</i>	67
3.2.3	Explicabilité	69
3.3	Détection d'anomalies par une approche <i>a contrario</i>	74
3.3.1	Définition du problème	74
3.3.2	Calcul d'un score de normalité	75
3.3.3	Probabilité d'appartenir à la distribution de référence	76
3.4	Expérimentations	77
3.4.1	Jeux de données	77
3.4.2	Application du modèle proposé	80
3.5	Résultats et discussions	84
3.5.1	Tâche auxiliaire	84
3.5.2	Comparaison avec d'autres méthodes	85
3.5.3	Limitations de l'approche proposée	87
3.6	Conclusion et perspectives	88
4	Évaluation des poids d'auto-attention d'un Transformer par représentation sous forme de graphe	89
4.1	Introduction des défis liés à la validation de l'apprentissage des Transformers	89
4.2	État de l'art	92
4.2.1	Transformers appliqués aux dossiers de santé électroniques	92
4.2.2	Explicabilité des Transformers basée sur l'auto-attention	94
4.2.3	Positionnement de notre approche	98
4.3	Évaluation de l'apprentissage d'un Transformer	98
4.3.1	Création de la matrice d'attention globale	99
4.3.2	Génération du graphe d'auto-attention	100
4.3.3	Évaluation de l'auto-attention apprise par un Transformer	101
4.4	Expérimentations	102
4.4.1	Modèle fondé sur des mécanismes d'attention	102
4.4.2	Cas d'usage	103
4.4.3	Génération du graphe d'auto-attention G' à partir de Behrt en inférence	105

4.4.4	Création du graphe de référence $\tilde{G}$	107
4.4.5	Évaluation de G' par rapport à $\tilde{G}$	108
4.5	Résultats et discussions	109
4.6	Conclusion et Perspectives	111
Conclusion et Perspectives		112
Bibliographie		119
Annexes		139
A	Données du Système National des Données de Santé	139
B	Guide de décodage d'un QR code	147
C	Exemples d'images obtenues après transformations des données du SNDS	151
D	Codes CIM-10	152

Liste des figures

1.1	Exemple du résultat des opérations successives de convolution et de <i>max pooling</i> .	8
1.2	Évolution des performances sur ImageNet et du nombre de paramètres des architectures CNN.	12
1.3	Schéma simplifié d'un bloc résiduel où x représente l'entrée du bloc résiduel, et $f(x)$ représente la transformation appliquée à x par les couches du bloc. L'opération $f(x) + x$ ajoute directement l'entrée initiale x à la sortie de la fonction $f(x)$. . .	14
1.4	Schéma d'un réseau de neurones récurrent. (a) représente le flux d'informations entre t et $t + 1$. (b) représente la cellule du RNN au temps t	16
1.5	Schéma des cellules LSTM (a) GRU (b).	17
1.6	Schéma simplifié de l'architecture d'un Transformer, représentant un seul bloc d'encodeur et de décodeur. Étapes numérotées de 1 à 5, avec 1 : conversion de la séquence d'entrée, 2 : traitement par l'encodeur, 3 : préparation de la séquence cible, 4 : traitement par le décodeur et 5 : génération de la séquence de sortie.	19
1.7	Schéma du pré-entraînement de BERT : le modèle est pré-entraîné simultanément sur deux tâches prenant la même séquence en entrée. La séquence est composée de deux phrases A et B (séparées par le marqueur ["SEP"]), dont certains mots sont masqués. La première tâche consiste à prédire les mots masqués de la séquence. La deuxième tâche, une classification binaire, évalue si la phrase B suit logiquement la phrase A.	23
1.8	Schéma d'un Vision Transformer (ViT)	25
1.9	Schéma d'un modèle CNN-LSTM	26
1.10	Schéma d'une cellule ConvLSTM	27
2.1	Trois images qui correspondent à la classe "a été enceinte". L'image A est figurative et représente une femme enceinte par le dessin de sa silhouette. Les images B et C sont abstraites et représentent une grossesse par des pixels correspondant à des événements liés au suivi de la grossesse.	38
2.2	Représentation d'une image à classifier dont une surface, un pourcentage de pixels variant de 10% à 100%, est bruitée par un changement de couleur tirée aléatoirement.	40
2.3	Exemples de QR codes avec le logo de <i>Facebook</i> ou de <i>Twitter</i> superposé.	41
2.4	Deux images contenant le logo <i>Facebook</i> segmenté. Les segments sont séparés par une distance de <i>Manhattan</i> , 50 pixels pour l'image de gauche et 190 pixels pour celle de droite.	41

2.5	Schéma des expérimentations. 1) Génération des QR codes contenant les informations obtenues lors du pass sanitaire européen. 2) Ajout de bruit. 3) Entraînement d'un classifieur sur les images bruitées. 4) Comparaison des performances du classifieur et celles d'un décodeur classique de QR codes pour la tâche de détection de la constante de validité du pass.	44
2.6	QR codes dont un pourcentage de la surface totale est bruitée en inversant la valeur des pixels (un pixel noir devient blanc et inversement).	45
2.7	Exemples de QR codes bruités utilisant différentes techniques de distorsion. (a) la distorsion aléatoire, qui simule l'étalement de l'encre en fonction du paramètre σ . (b) la distorsion par enroulement, qui simule le gondolage du papier en fonction du paramètre mag . (c) la sélection de l'avant/arrière-plan, qui simule une baisse de luminosité.	46
2.8	Arbre représentatif de la distribution du jeu de données d'entraînement. Chaque classe est représentée par 7 920 observations dont 1 980 par correction d'erreur. Pour chaque correction d'erreur, 165 observations sont générées par type de bruit : inversement de pixels de 0% à 40% inclus, distorsion aléatoire pour $\sigma \in \{2; 5; 20\}$, distorsion par enroulement pour $mag \in \{10; 20; 50\}$ et sélection de l'avant/arrière-plan	47
2.9	Courbes d'exactitude du décodeur déterministe et du classifieur sur des QR codes bruités, par inversement de pixel, de 0% à 100%. Une ligne rouge en pointillés située à exactitude=50% est incluse pour faciliter la visualisation.	48
2.10	Courbes de l'exactitude du décodeur déterministe et du classifieur sur des QR codes générés à partir de séquence de type https://constante/username avec constante = Facebook ou Twitter.	50
2.11	Comparaison de l'inférence des modèles entraînés sur différentes classes en fonction du pourcentage de bruit par inversement de pixels (de 0% à 100%).	50
2.12	Schéma de l'approche proposée. Tout d'abord, la série temporelle multivariée X_i est transformée en une série temporelle d'images $\tilde{X}_i$. Chaque image de cette série temporelle est ensuite transmise à un CNN unique pour extraire des vecteurs de caractéristiques. La série de vecteurs de caractéristiques est ensuite utilisée comme entrée d'un modèle LSTM pour obtenir une prédiction.	51
2.13	Architecture du modèle CNN-LSTM utilisé. Le modèle se compose de trois couches convolutives 2D avec 3, 16 et 32 filtres ayant respectivement une taille de 3×3 , 5×5 et 3×3 . Une normalisation par lot [8] est placée après la première et la deuxième couche, et la fonction d'activation ReLU est utilisée pour chaque couche. Après ces couches convolutives, les caractéristiques extraites sont aplaties et passées à une couche entièrement connectée. Cette couche entièrement connectée a une sortie de taille 50, correspondant au vecteur final des caractéristiques extraites. L'architecture CNN est suivie par un LSTM avec un état caché (h_t) et un état de cellule (c_t) de taille 50. Le LSTM est suivi d'une couche de <i>dropout</i> avec une probabilité de 0,4 et d'une couche entièrement connectée avec une sortie de taille 6.	56

2.14	Architecture du modèle LSTM utilisé. Le modèle comporte deux couches LSTM successives. La première couche LSTM a un état de cellule c_{t_1} et un état caché h_{t_1} , tous deux de taille 100. La deuxième couche LSTM a un état de cellule c_{t_2} et un état caché h_{t_2} , tous deux de taille 50. Ces couches LSTM sont suivies d'une couche entièrement connectée de taille 6 en sortie, correspondant au nombre de classes à prédire.	56
2.15	Architecture du modèle 1DCNN-LSTM utilisé. Le modèle est composé de 3 couches convolutives 1D avec 16, 16 et 32 filtres ayant respectivement une taille de 5, 3 et 3. La fonction d'activation ReLU est utilisée pour chaque couche. La partie LSTM du modèle a un état caché (h_t) et un état de cellule (c_t) de taille 100 et une couche entièrement connectée avec une sortie de taille 6.	56
2.16	Matrices de confusion de notre modèle (CNN-LSTM) et des modèles 1DCNN-LSTM, LSTM et BERT, pour la tâche de classification de 6 pathologies.	59
3.1	Deux choux-fleurs âgés de 42 jours. Celui de gauche est anormal et celui de droite est normal	62
3.2	Schéma de l'approche proposée pour la détection d'anomalies. Pour chaque nouvelle observation, nous obtenons la carte de saillance associée en appliquant Grad-CAM à un modèle entraîné sur une tâche auxiliaire. Nous comparons ensuite cette carte de saillance à la distribution de référence des cartes de saillance issues d'images normales en utilisant une approche <i>a contrario</i> pour déterminer si la nouvelle observation est normale ou non.	63
3.3	Schéma de la méthodologie de notre approche. Étant donné un ensemble d'images normales de la classe c , nous récupérons les cartes de saillance associées pour établir la distribution de référence $\tilde{H}_c$. Pour une nouvelle image x de la classe c , nous récupérons la carte de saillance h associée et calculons, pour chaque pixel, la probabilité qu'il suive la distribution de référence. Nous moyennons ensuite ces probabilités pour obtenir le score de normalité $S_c(h)$, qui sera comparé à un seuil pour savoir dans quelle hypothèse du test <i>a contrario</i> l'image se trouve.	75
3.4	Exemples d'images des jeux de données <i>GrowliFlower</i> , <i>PlantDoc</i> et <i>PlantVillage</i> , utilisés dans notre étude. (a) images d'origine, (b) superposition des images d'origine et des cartes de saillance obtenues par Grad-CAM, (c) cartes de saillance obtenues par Grad-CAM.	78
3.5	Exemples de séries temporelles d'images de deux choux-fleurs du jeu de données <i>GrowliFlower</i> , superposées, ou non, avec les cartes de saillance. Pour chaque pas de temps, sont indiqués : l'âge du chou-fleur, l'âge prédit par le modèle et les annotations associées indiquant la présence/nature d'une anomalie. "Aucune" signifie qu'il n'y a pas d'anomalie associée à la plante.	83
3.6	Courbes de précision/rappel de notre modèle et de f-AnoGAN sur les jeux de données <i>GrowliFlower</i> , <i>PlantDoc</i> et <i>PlantVillage</i>	87

4.1	Schéma de l'approche proposée. Étant donné une séquence de diagnostics que l'on peut retrouver dans le SNDS (cf. annexe A) donnée en entrée d'un Transformer, nous extrayons les poids d'auto-attention appris par le modèle entre chaque paire de diagnostics et les représentons sous forme de graphe, où chaque nœud est un diagnostic et chaque arc qui relie le nœud A au nœud B représente le poids d'auto-attention que le nœud A accorde au nœud B. Nous évaluons ce graphe par rapport à un graphe représentant l'expertise médicale afin d'évaluer la pertinence des poids d'auto-attention appris par le modèle.	90
4.2	Étapes de la méthodologie proposée : à partir des séquences données en entrée d'un Transformer, nous identifions les tokens d'importance par analyse des gradients. Une matrice d'attention globale est construite dans laquelle chaque token attribue un poids (noté w) à tous les autres tokens d'un vocabulaire global, à condition que les deux tokens coexistent dans la même séquence. Ensuite, nous extrayons un sous graphe d'auto-attention G' en partant des tokens d'importance (nœuds bleus) et en ajoutant tous les tokens (nœuds noirs) auxquels ils sont liés dans la matrice d'attention globale. Un arc reliant deux nœuds dans G' est pondéré par le poids d'auto-attention que le nœud source accorde au nœud cible. Nous évaluons la pertinence des poids d'auto-attention appris par le modèle à l'aide de la fonction d'évaluation E , qui compare le graphe d'attention G' au graphe représentant la vérité terrain, noté $\tilde{G}$	99
4.3	Fréquence d'apparition des différents diagnostics de la classe <i>maladies hypertensives</i> , dans l'échantillon global contenant les historiques médicaux de 1,19 millions d'individus.	105
4.4	Graphes du cas d'usage de l'accouchement. (a) graphe issu de la matrice d'attention globale G , (b) sous-graphe d'auto-attention G' généré à partir des diagnostics d'importance (en bleu) et ceux à qui ils sont liés dans G . Les libellés des codes sont listés dans la Table 4 de l'annexe D.	106
4.5	Graphes du cas d'usage de l'hypertension. (a) graphe issu de la matrice d'attention globale G , (b) graphe d'auto-attention G' généré à partir des diagnostics d'importance (en bleu) et ceux à qui ils sont liés dans G . Les libellés des codes sont listés dans la Table 4 de l'annexe D.	107
4.6	Graphes issus des expertises médicales pour le cas d'usage de l'accouchement. Les deux graphes sont générés en demandant aux médecins de relier les diagnostics noirs à un ou plusieurs diagnostics bleus si il existe une corrélation contextuelle entre eux. Les graphes (a) et (b) sont respectivement les résultats du protocole effectué avec les médecins 1 et 2.	108
4.7	Graphes issus des expertises médicales pour le cas d'usage de l'hypertension. Les deux graphes sont générés en demandant aux médecins de relier les diagnostics noirs à un ou plusieurs diagnostics bleus si il existe une corrélation contextuelle entre eux. Les graphes (a) et (b) sont respectivement les résultats du protocole effectué avec les médecins 1 et 2.	109

4.8	Courbes illustrant l'évolution du score d'évaluation en fonction du seuil d'auto-attention utilisé pour sélectionner les arcs dans le graphe d'auto-attention G' . (a) correspond au cas d'usage de l'accouchement, et (b) au cas d'usage des maladies hypertensives. Dans les deux figures, la courbe bleue représente l'évaluation du graphe par rapport à l'expertise médicale du premier médecin, tandis que la courbe orange représente l'évaluation par rapport au deuxième médecin. Une ligne rouge en pointillés située à score=0 est incluse pour faciliter la visualisation des scores d'évaluation devenant négatifs.	110
9	Schéma simplifié de l'architecture des bases de données individuelles. Les tables du <i>Programme de Médicalisation des Systèmes d'Information</i> (PMSI), du <i>Datamart de Consommation Inter Régime</i> (DCIR) et du <i>Centre d'épidémiologie sur les causes médicales de décès</i> (CépiDC) peuvent être jointes grâce à un identifiant patient. . .	142
10	Schéma de jointure de trois tables du DCIR (une principale et deux tables à joindre), par la méthode <i>SCALPEL-Flattening</i>	144
11	Schéma de jointure des tables principales et patient du PMSI par la méthode <i>SCALPEL-Flattening</i> . Le résultat de cette jointure est la table <i>centrale</i>	144
12	Schéma de jointure de la table <i>centrale</i> (résultat de la jointure illustrée dans la Figure 11) du PMSI et des autres tables par la méthode <i>SCALPEL-Flattening</i>	145
13	Différents masques de QR code avec leurs règles d'inversion associées. Les variables i et j correspondent respectivement aux coordonnées des positions horizontale et verticale du module (bit).	147
14	Schéma des différentes parties d'un QR code	148
15	Sens de lecture des bits correspondant au format du QR code.	149
16	Sens de lecture des octets d'un QR code.	149
17	Déchiffrage du nombre de caractères à décoder.	149
18	Positions des octets à décoder.	150
19	Négatif des images moyenne de chaque classe. Où (c_1) : tumeurs; (c_2) : maladies endocriniennes, nutritionnelles et métaboliques; (c_3) : troubles mentaux et comportementaux, (c_4) : maladies du système circulatoire, (c_5) : maladies du système ostéo-articulaire, des muscles et du tissu conjonctif et (c_6) : lésions traumatiques, empoisonnements et certaines autres conséquences de causes externes.	151

Liste des tableaux

1.1	Liste des architectures principales de classification d'images et les contributions associées.	10
2.1	Résultats du pourcentage maximal de bruit tel que le modèle arrive à apprendre, pour des logos entiers ou fragmentés dont la position est fixe ou aléatoire.	42
2.2	Comparaison des performances du classifieur et du décodeur en fonction du bruit utilisé durant la génération des QR codes. Pour la distorsion aléatoire et la distorsion par enroulement, nous présentons les performances moyennes obtenues.	48
2.3	Performances et temps d'apprentissage des modèles de classification.	57
3.1	Observations in-situ des traits phénotypiques de développement et de stress des plantes du jeu de données <i>GrowliFlower</i>	79
3.2	Liste des classes du jeu de données <i>PlantDoc</i> . En bleu les classes normales, en noir les anormales. Pour chaque tuple, le premier élément correspond au fruit ou au légume de la feuille concernée, et le deuxième élément à l'état de la feuille.	80
3.3	Liste des classes du jeu de données <i>PlantVillage</i> . En bleu les classes normales, en noir les anormales. Pour chaque tuple, le premier élément correspond au fruit ou au légume de la feuille concernée, et le deuxième élément à l'état de la feuille.	81
3.4	Paramétrisation des modèles d'apprentissage pour la tâche auxiliaire en fonction du jeu de données.	82
3.5	Performances des modèles d'apprentissage pour la tâche auxiliaire en fonction du jeu de données.	84
3.6	Résultats de la détection d'anomalies. Nous rapportons l'aire sous la courbe (AUC), le score $F1$, la précision et le rappel de chaque modèle sur chaque jeu de données utilisé.	86
4.1	Spécificités des échantillons utilisés pour les deux tâches d'apprentissage du modèle Behrt. La tâche 1 correspond à la prédiction de mots masqués, la tâche 2 correspond à l'affinage du modèle pré-entraîné sur la tâche 1.	103
2	Contenu du SNDS. La colonne de gauche liste les informations qu'on peut trouver dans le SNDS et celle de droite les informations qu'on ne peut pas trouver dans le SNDS.	140
3	Décodage des octets avec l'ajout du masque pour obtenir les valeurs ASCII correspondantes	150
4	Liste (non exhaustive) des codes CIM10 et leur signification	152

Liste des abréviations

1D Une Dimension

2D Deux Dimensions

3D Trois Dimensions

ACI Analyse en Composantes Indépendantes

ACP Analyse en Composantes Principales

ADL Analyse Discriminante Linéaire

ADN Acide Désoxyribonucléique

ARN Acide Ribonucléique

ATIH Agence Technique de l'Information sur l'Hospitalisation

AUC *Area Under Curve*

BERT *Bidirectional Encoder Representations from Transformers*

Bi-LSTM *Bidirectional Long Short-Term Memory*

BMDS *Bayesian MultiDimensional Scaling*

BOSSEnsemble *Bag-of-SFA-Symbols Ensemble*

CAM *Class Activation Map*

CDeepEx *Contrastive Deep Explanation*

CEDIA Centre d'Expertise Données et Intelligence Artificielle

CEM *Contrastive Explanation Method*

CIM Classification Internationale des Maladies

CNAM Caisse Nationale de l'Assurance Maladie

CNN *Convolutional neural network*

CépiDC Centre d'épidémiologie sur les causes médicales de décès

DCIR Datamart de Consommation Inter Régime

DFT *Discrete Fourier Transform*

DM *Diffusion Maps*

Drees Direction de la Recherche, des Études, de l'Évaluation et des Statistiques

DWT *Discrete Wavelet Transform*

EDLT *Enabling Deep Learning for Generic Classification Tasks*

EGB *Échantillon Généraliste des Bénéficiaires*

EHR *Electronic Health Record*

EPFL *École Polytechnique Fédérale de Lausanne*

FFT *Fast Fourier Transform*

FN *Faux Négatifs*

FP *Faux Positifs*

GAN *Generative Adversarial Network*

GAP *Global Average Pooling*

GCT *Graph Convolution Transformer*

GNN *Graph Neural Network*

GPT *Generative Pre-Training*

GPU *Graphics Processing Unit*

Grad-CAM *Gradient-weighted Class Activation Mapping*

GRU *Gated Recurrent Unit*

HAD *Hospitalisation À Domicile*

HiTANet *Hierarchical Time-aware Attention Network*

IEEE *Institute of Electrical and Electronics Engineers*

IG *Integrated Gradients*

IGTD *Image Generator for Tabular Data*

ILSVRC *ImageNet Large Scale Visual Recognition Challenge*

INSERM *Institut National de la Santé et de la Recherche Médicale*

Isomap *Isometric Mapping*

KNN *K-Nearest Neighbors*

KPCA *Kernel PCA*

LE *Laplacian Eigenmap*

LIME *Local Interpretable Model-agnostic Explanations*

LLE *Locally Linear Embedding*

LLM *Large Language Model*

LPP *Locality Preserving Projections*

LSTM *Long Short-Term Memory*

MCO *Médecine, Chirurgie et Obstétrique*

MDS *MultiDimensional Scaling*

MLM *Masked Language Modeling*

MNIST *Modified National Institute of Standards and Technology database*

MSE *Mean Square Error*

MVU *Maximum Variance Unfolding*

NAS *Neural Architecture Search*

NFA *Nombre de Fausses Alarmes*

NIR *Numéro d’Inscription au Répertoire*

NSP *Next Sentence Prediction*

OCCNN *One-Class Convolutional Neural Network*

OCSVM *One-Class Support Vector Machines*

PMAN *Probabilistic Masked Attention Network*

PMSI *Programme de Médicalisation des Systèmes d’Information*

QR code *Quick Response code*

RAPT *RepresentAtion by Pre-training time-aware Trans-former*

REFINED *REpresentation of Features as Images with NEighborhood Dependencies*

ReLU *REctified Linear Unit*

ResNet *Residual Network*

RFDS *Relative Filtered Driving Score*

RFE *Recursive Feature Elimination*

RISE *Randomized Input Sampling for Explanation*

RNN *Recurrent Neural Network*

RVB *Rouge Vert Bleu*

SAX *Symbolic Aggregate approXimation*

SHAP *SHapley Additive exPlanations*

SNDS *Système National des Données de Santé*

Ssd *Self-supervised outlier Detection*

SSR *Soins de Suite et de Réadaptation*

SVD *Singular Value Decomposition*

SVDD *Support Vector Data Description*

t-SNE *t-distributed Stochastic Neighbor Embedding*

TAPER *Time-Aware Patient EHR Representation*

TSF *Time Series Forest*

TSN *Temporal Segment Networks*

TSS *Transcription Start Site*

UAV *Unmanned Aerial Vehicle*

ViT *Vision Transformer*

VN Vrais Négatifs

VP Vrais Positifs

XAI *Explainable Artificial Intelligence*

Introduction

Dans le domaine de l'apprentissage automatique, les modèles d'apprentissage profond sont devenus des outils incontournables pour exploiter de vastes volumes de données [1]. Les modèles les plus récents, principalement basés sur des réseaux de neurones artificiels, s'inspirent du fonctionnement du cortex cérébral humain, où des neurones interconnectés traitent et transmettent des informations.

Historiquement, un réseau de neurones artificiels est composé de couches successives de neurones [2]. Chaque neurone reçoit des données, les transforme via des poids et des fonctions d'activation, puis transmet le résultat à la couche suivante. Cette structure hiérarchique permet d'extraire progressivement des caractéristiques de plus en plus abstraites et de haut niveau à partir des données brutes.

Les réseaux de neurones profonds sont utilisés pour de nombreux axes thématiques. Quand ils traitent des images, ils entrent dans le domaine de la vision par ordinateur, accomplissant des tâches telles que la reconnaissance d'objets, la segmentation sémantique et la classification d'images. Pour les séries temporelles, ils sont par exemple appliqués à la prédiction de valeurs futures, la détection d'anomalies et la modélisation de séries temporelles. Dans le traitement du langage naturel, ils peuvent être utilisés pour des tâches comme la traduction automatique, la génération de texte et la classification de documents. En réalité, les applications et les tâches d'apprentissage possibles sont nombreuses et variées, dépendant essentiellement de la nature des données d'entrée et du contexte d'utilisation.

Cependant, malgré leur succès, les modèles d'apprentissage profond font face à plusieurs défis. L'un des principaux défis est **le fléau de la dimension** [3], qui désigne la difficulté croissante d'optimiser les modèles à mesure que la dimensionnalité des données augmente. **La complexité des données** représente également un obstacle, car il est souvent difficile d'extraire automatiquement les caractéristiques les plus pertinentes. De plus, dans le cadre d'une tâche d'entraînement supervisée, le **manque de données annotées** de qualité complique l'entraînement des modèles. En effet, lorsque les jeux de données sont peu représentatifs et que la tâche nécessite de capturer des caractéristiques complexes, le modèle risque de faire face au phénomène de surapprentissage [4], c'est-à-dire qu'il peut ne pas avoir assez d'exemples variés pour généraliser son apprentissage au-delà des données d'entraînement, entraînant ainsi une performance sous-optimale sur de nouvelles données.

Pour surmonter ces défis, plusieurs approches ont été développées. L'augmentation de la taille des modèles, en termes de nombre de paramètres et de couches, permet de capturer plus de nuances dans les données, mais au prix d'une complexité computationnelle accrue.

Des mécanismes de régularisation, comme le dropout [5] - une technique qui désactive aléatoirement certains neurones durant l'entraînement - sont utilisés pour prévenir le surapprentissage afin de garantir une meilleure généralisation du modèle. L'augmentation de données [6] aide à enrichir les jeux de données en générant des variations artificielles des données existantes. Des techniques comme le transfert d'apprentissage¹ [7] et l'affinage de modèle² (*fine-tuning*) permettent de réutiliser et d'adapter des modèles pré-entraînés sur de grandes bases de données, réduisant ainsi le besoin de vastes ensembles de données annotées. Enfin, les méthodes de normalisation des données, telles que la normalisation par lots [8], aident à stabiliser et accélérer l'entraînement des réseaux de neurones.

Un autre défi rencontré par l'application de modèles profonds est **la validation** de leur apprentissage, surtout dans des domaines sensibles tels que la médecine [9]. Contrairement à certains modèles d'apprentissage automatique comme les arbres de décision [10], dont les processus décisionnels peuvent être retracés et expliqués, les réseaux de neurones sont souvent considérés comme des boîtes noires [9], rendant l'explication de leurs décisions plus difficile. Ce problème a donné naissance à un domaine de recherche appelé l'Intelligence Artificielle Explicable (*Explainable Artificial Intelligence* (XAI)) [11]. Les recherches en XAI se concentrent principalement sur l'identification et l'interprétation des caractéristiques qui ont permis au modèle de faire sa prédiction. L'identification de ces caractéristiques, dites d'importance, permet aux experts humains de valider la logique des décisions prises par les réseaux de neurones.

Les recherches actuelles sur les défis liés à l'application des réseaux de neurones se concentrent sur l'intégration de nouveaux mécanismes dans les architectures pour surmonter divers problèmes, en fonction des données en entrée et des modèles utilisés. Dans nos travaux, nous nous intéressons spécifiquement à l'impact de la représentation de l'information sur les défis liés à l'apprentissage profond. Plus précisément, notre question de recherche est la suivante : **comment la représentation de l'information influence-t-elle l'atteinte d'un objectif donné ?**

Pour répondre à cette question, nos travaux se divisent en trois objectifs de recherche, dont les expérimentations associées ont été validées sur des données réelles :

Objectif 1 : pallier le fléau de la dimension lors de l'application des réseaux de neurones sur des données à haute dimension. Nous répondons à cet objectif en transformant les données tabulaires à haute dimension en images, permettant l'application de réseaux de neurones convolutifs (CNN). Ces modèles étant moins sensibles au nombre élevé de dimensions grâce à leurs opérations de convolution, leur application permet de pallier le fléau de la dimension [3].

Objectif 2 : identifier les images contenant des anomalies complexes à détecter.

-
1. Apprentissage par transfert : technique utilisant un modèle pré-entraîné sur une tâche source comme base pour une nouvelle tâche cible. Les couches pré-entraînées peuvent être gelées pour servir d'extracteurs de caractéristiques fixes, tandis que de nouvelles couches ajoutées sont entraînées spécifiquement pour la tâche cible.
 2. Affinage du modèle : processus d'adaptation d'un modèle pré-entraîné à une tâche spécifique en ajustant ses paramètres. Cette technique permet de transférer les connaissances acquises sur de vastes ensembles de données à des applications plus ciblées, en mettant à jour les poids de certaines ou toutes les couches du réseau. L'affinage optimise ainsi les performances du modèle sur la nouvelle tâche tout en préservant les représentations générales apprises lors du pré-entraînement.

Pour répondre à cet objectif, nous exploitons les cartes de saillance générées par un modèle d'explicabilité appliqué à un modèle entraîné sur une tâche prédictive auxiliaire. Ces cartes, révélant les régions significatives des images, permettent de déterminer, à l'aide d'un test *a contrario* [12], si ces régions sont caractéristiques des données normales ou anormales

Objectif 3 : évaluer les liens d'auto-attention appris par un modèle Transformer.

Nous répondons à cet objectif en représentant les poids d'auto-attention appris par un Transformer [13] sous forme de graphe pour évaluer de manière automatisée les relations entre les caractéristiques, en les comparant à une référence établie, elle aussi représentée sous forme de graphe.

Les travaux de cette thèse ont été réalisés dans le cadre d'un co-encadrement avec le *Centre d'Expertise Données et Intelligence Artificielle* (CEDIA) du Ministère des Armées. Ces recherches ont émergé d'un besoin identifié par le CEDIA, qui collaborait avec le *Service de Santé des Armées* sur des problématiques liées à la pandémie du Covid-19, en utilisant les données du *Système National des Données de Santé* (SNDS). Ces données, présentées dans l'annexe A, sont de très haute dimension, ce qui rend les approches d'apprentissage automatique classiques peu performantes en raison du fléau de la dimension. Pour répondre à ce problème, qui correspond au premier objectif de recherche, nous souhaitons évaluer l'impact de la transformation des données tabulaires à haute dimension en images, afin d'appliquer des modèles convolutifs spécialement conçus pour traiter les images. Pour cela, nous travaillons avec un échantillon du SNDS fourni par le Service de Santé des Armées.

Cette étude a été précédée par deux études préliminaires visant à déterminer les limites des modèles convolutifs sur des images non conventionnelles par rapport aux jeux de données traditionnels comme ImageNet [14] ou COCO [15]. L'objectif de la première étude préliminaire est de déterminer le niveau maximum de bruit qu'une image figurative (i.e. identifiable à l'œil nu) peut contenir tout en permettant à un modèle de classification d'apprendre correctement. La deuxième étude préliminaire consiste à étudier le comportement d'un CNN sur des images abstraites (non identifiables visuellement), telles que des QR codes, en présence de bruit.

Objectif 1 : pallier le fléau de la dimension lors de l'application des réseaux de neurones sur des données à haute dimension.

Contributions :

1. Nous proposons une méthode de transposition de données multidimensionnelles en images pour pallier le fléau de la dimension.
2. Nous proposons l'utilisation d'un modèle hybride léger, en termes de paramètres et de couches, permettant la reconnaissance de motifs dans les images et l'analyse des corrélations entre les différents instants d'une série temporelle.
3. Nous validons notre approche pour la tâche de classification multi-classes sur des données de santé bruitées.

Pour répondre au deuxième objectif de recherche, nous étudions la possibilité d'utiliser les résultats d'apprentissage d'un modèle performant sur une tâche donnée pour une nouvelle tâche complexe : la détection d'images anormales. Plus précisément, nous utilisons les zones

d'importance révélées par un modèle d'explicabilité, appliqué à un modèle performant sur une tâche auxiliaire, comme entrée pour le modèle de détection d'anomalies basé sur un test *a contrario*. Pour cette étude nous travaillons avec des images issues de jeux de données du domaine de la télédétection pour la détection de maladies des plantes.

Objectif 2 : identifier les images contenant des anomalies complexes à détecter.

Contributions :

1. Nous proposons une méthode qui entraîne un modèle sur une tâche prétexte triviale pour obtenir des cartes de saillance, obtenues par un modèle d'explicabilité, révélant les zones d'importance des images. Sur cette base, nous développons une approche *a contrario* pour la détection d'anomalies.
2. Nous introduisons une formulation du problème de détection des anomalies des plantes qui ne nécessite aucune connaissance préalable des anomalies.
3. Nous validons notre approche sur trois jeux de données du domaine de la télédétection.

Pour répondre au troisième objectif de recherche sur l'évaluation des poids d'auto-attention appris par un modèle Transformer [13], nous proposons de les représenter sous forme de graphe, que nous évaluons par rapport à une référence établie, aussi représentée sous forme de graphe. Nous appliquons notre approche sur un cas d'usage médical pour évaluer les liens entre les diagnostics appris par le modèle par rapport à l'expertise médicale. Cette étude est réalisée en collaboration avec le *Lab Santé* de la *Direction de la Recherche, des Études, de l'Évaluation et des Statistiques* (Drees³), du Ministère de la Santé. Plus précisément, nous utilisons un modèle Transformer, entraîné par le Lab Santé sur un échantillon du SNDS, pour prédire la prochaine visite à l'hôpital en se basant sur le parcours de soins hospitaliers d'un individu.

Objectif 3 : évaluer les liens d'auto-attention appris par un modèle Transformer.

Contributions :

1. Nous proposons une méthode qui représente sous forme de graphe les interactions et les poids d'auto-attention entre les tokens appris par un Transformer.
2. Nous proposons une approche pour modéliser l'expertise médicale sous forme de graphe, utilisé comme référence.
3. Nous introduisons une nouvelle métrique pour évaluer la pertinence des connexions apprises par le modèle en les comparant à une référence établie.
4. Nous validons notre approche sur deux cas d'usage médicaux, visant à évaluer les interactions entre les diagnostics sur lesquels le modèle s'appuie pour prédire le diagnostic de la prochaine visite à l'hôpital.

Les différentes études réalisées traitent soit des images, soit des séries temporelles, soit des séquences d'images. Ainsi, le Chapitre 1 présente les différentes familles d'architectures

3. <https://drees.solidarites-sante.gouv.fr>

pour traiter ces trois types de données.

La suite des chapitres est organisée en fonction des objectifs présentés. Ainsi, le Chapitre 2 présente l'étude sur la transposition de données à haute dimension en images pour pallier le fléau de la dimension. Les études préliminaires sont incluses dans ce chapitre. Le Chapitre 3 présente l'étude sur la représentation des images en cartes de saillance pour la détection d'anomalies. Enfin, le Chapitre 4 présente l'étude sur la validation des modèles Transformers en représentant leurs poids d'auto-attention sous forme de graphes. Chaque chapitre contient un état de l'art sur les approches utilisées dans le domaine de recherche abordé. Enfin, nous concluons cette thèse et discutons des perspectives dans la dernière partie.

Chapitre 1

Apprentissage profond pour le traitement d'images et de données temporelles

Ce manuscrit explore des stratégies d'apprentissage profond pour trois objectifs de recherche, chacun traitant de types de données spécifiques. Le premier objectif vise à pallier le fléau de la dimension lors de l'application des réseaux de neurones sur des données à haute dimension, en transformant des données séquentielles en séries temporelles d'images. Le deuxième objectif consiste à identifier les images contenant des anomalies complexes à détecter, en utilisant la capacité des réseaux de neurones convolutifs comme extracteurs de caractéristiques. Enfin, le troisième objectif se focalise sur l'évaluation d'un mécanisme interne de réseaux appliqués aux séries temporelles, plus précisément l'auto-attention apprise par les Transformers. Ainsi, les travaux de recherche présentés couvrent trois types de données distincts : les images, les séries temporelles et les séquences d'images. Dans ce cadre, ce chapitre retrace le développement des modèles d'apprentissage profond, conçus pour surmonter les contraintes spécifiques à ces types de données. Les concepts exposés permettent ainsi une meilleure compréhension des travaux que nous présentons par la suite.

La Section 1.1 décrit les réseaux de neurones convolutifs et leur évolution pour le traitement d'images. La Section 1.2 présente l'évolution des réseaux de neurones récurrents et les Transformers, développés pour traiter les séries temporelles. Enfin, la Section 1.3 explore les techniques développées spécifiquement pour traiter les séries temporelles d'images.

1.1 Réseaux de neurones convolutifs pour le traitement des images

Les réseaux de neurones convolutifs (CNN) sont une catégorie spécifique de réseaux de neurones utilisés principalement pour traiter les images. Ils sont utilisés pour les tâches de vision par ordinateur, comme la reconnaissance d'images, la détection d'objets ou la

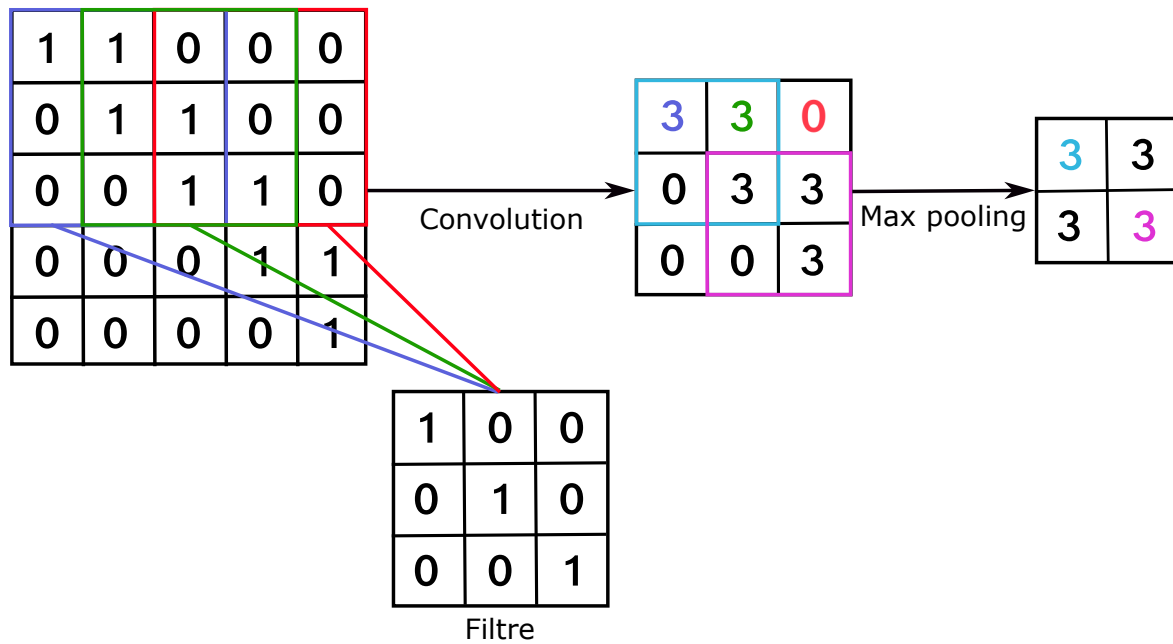


FIGURE 1.1 – Exemple du résultat des opérations successives de convolution et de *max pooling*.

segmentation d'images. Ces réseaux tirent parti de la structure spatiale des données pour réduire la complexité du modèle et améliorer l'efficacité de l'apprentissage. La structure multicouche du CNN lui permet d'extraire des caractéristiques de bas et de haut niveau. Ces caractéristiques peuvent être transférées à des tâches génériques grâce à l'apprentissage par transfert [7]. De plus, leur capacité à apprendre des caractéristiques visuelles hiérarchiques à partir de données brutes les rend également utiles dans d'autres domaines tels que l'analyse de séquences temporelles et le traitement du langage naturel, où les données peuvent être structurées de manière similaire à des grilles.

Les principaux mécanismes propres aux CNN sont les couches de convolution et les couches de *pooling*. Les couches de convolution appliquent un ensemble de filtres (aussi appelés noyaux ou *kernels*) à l'image d'entrée pour créer des cartes de caractéristiques (*feature maps*). Ces filtres permettent de détecter des caractéristiques spatiales telles que les bords, les textures, ou des formes spécifiques. Les couches de *pooling* quant à elles sont utilisées pour réduire les dimensions des cartes de caractéristiques, ce qui diminue la quantité de calcul et de mémoire nécessaire pour les opérations suivantes et aide à rendre le modèle plus généralisable. Les opérations de *pooling* les plus courantes sont le *max pooling*, qui prend le maximum dans une fenêtre de valeurs et l'*average pooling*, qui prend la moyenne d'une fenêtre de valeurs. La Figure 1.1 montre un exemple visuel de l'application d'une couche de convolution sur une image, suivie d'une couche de *max pooling*.

L'évolution récente des technologies matérielles, notamment les *Graphics Processing Unit* (GPU), a permis de développer et de déployer efficacement les CNN pour diverses applications, car ces réseaux reposent sur des séries de calculs matriciels que les GPU sont spécifiquement conçus pour traiter rapidement en parallèle. Dans la partie suivante, nous présentons l'évolution des principales architectures CNN pour la tâche de classification d'images, en mettant l'accent sur les innovations des modèles qui répondent aux éven-

tuelles limites des modèles qui le précèdent. La Table 1.1 recense les principaux modèles de classification d'images basés sur des architectures convolutives, avec les contributions associées.

1.1.1 Évolution des réseaux de neurones convolutifs

Le *Neocognitron* [16], proposé en 1980, est considéré comme l'architecture pionnière des réseaux de neurones convolutifs (CNN). Il a ouvert la voie au développement des CNN tels que nous les connaissons aujourd'hui. Peu de temps après, le premier CNN multicouche, *ConvNet* [17], est introduit. Il utilise l'algorithme de rétropropagation [28], contrairement au *Neocognitron* qui utilise de l'apprentissage par renforcement non supervisé. Le nom "réseau de neurone convolutif" provient du design de *LeNet-5* [18], développé pour la tâche de reconnaissance de chiffres manuscrits sur le jeu de données MNIST¹. Cependant, la notoriété des architectures CNN plus récentes revient au défi de classification *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) [29], lancé en 2010, qui permet de comparer les modèles sur un même jeu de données, *ImageNet* [14]. Depuis sa création, le jeu de données *ImageNet* a été enrichi et contient aujourd'hui environ 14 millions d'images et 21 000 classes.

AlexNet [19], qui a remporté le concours ILSVRC en 2012, est considéré comme le premier CNN avec une architecture profonde, composée de 8 couches contre 5 pour *LeNet-5*. Bien que la profondeur du réseau améliore la généralisation pour différentes résolutions d'images, le principal inconvénient associé à l'augmentation de la profondeur est le surapprentissage [4]. Pour y remédier, *AlexNet* introduit plusieurs stratégies d'optimisation de paramètres, comme le *dropout*, qui désactive aléatoirement des neurones pendant l'entraînement pour forcer le modèle à apprendre des caractéristiques plus robustes. De plus, l'utilisation d'une couche qui normalise les valeurs à travers les canaux pour accélérer la convergence, et l'utilisation de l'unité linéaire rectifiée (ReLU) comme fonction d'activation permettent d'améliorer l'efficacité et le temps de l'entraînement.

Malgré ses avancées significatives dans le domaine de la vision par ordinateur, *AlexNet* présente certaines limites, notamment une mauvaise compréhension des mécanismes internes du réseau. Ces limitations révèlent des défis dans l'optimisation de la topologie du réseau et la sélection des paramètres pour améliorer la performance sur des images complexes. Pour répondre à ces problématiques, *ZFNet* [20], qui a remporté le deuxième prix à l'ILSVRC 2013, a introduit une méthode pour visualiser et comprendre l'activité interne des couches du réseau. En utilisant un réseau déconvolutionnel multicouche, *ZFNet* permet de projeter les sorties des couches convolutionnelles en motifs d'images visuellement perceptibles, offrant une interprétation détaillée de la représentation des caractéristiques apprises. Cette visualisation a révélé que seulement quelques neurones étaient actifs dans les premières couches d'*AlexNet*, tandis que d'autres restaient inactifs, et que les caractéristiques extraites par la deuxième couche souffraient d'*aliasing*². Cela signifie que les motifs ou les textures reconnaissables dans ces visualisations ne sont pas aussi clairs ou précis qu'ils devraient l'être. Au lieu de capturer fidèlement les détails fins ou les variations

1. <http://yann.lecun.com/exdb/mnist/> 2. L'*aliasing* est un effet qui se produit lorsqu'un signal est échantillonné à une fréquence insuffisante pour capturer toutes ses composantes de haute fréquence. Cela peut entraîner une distorsion ou une mauvaise représentation du signal original.

Année	Modèle	Contributions
1980	<i>Neocognitron</i> [16]	• Précurseur des CNN
1989	<i>ConvNet</i> [17]	• Inspiré du cortex visuel • Utilisation de la rétropropagation • Partage des poids • Réduction de dimension (<i>pooling</i>)
1998	<i>LeNet-5</i> [18]	• Premier CNN
2012	<i>AlexNet</i> [19]	• Plus profond que <i>LeNet-5</i> • Utilisation de <i>ReLU</i> et <i>dropout</i>
2014	<i>ZFNet</i> [20]	• Visualisation de l'activité des couches
2014	<i>VGG</i> [21]	• Architecture profonde • Petits filtres de convolutions
2015	<i>GoogLeNet</i> [22]	• Blocs <i>Inception</i> • Utilisation de plusieurs têtes de classification
2015	<i>Highway</i> [23]	• Mécanisme de connexion entre couches
2016	<i>ResNet</i> [24]	• Blocs résiduels
2017	<i>DenseNet</i> [25]	• Flux d'informations entre les couches
2017	<i>MobileNet</i> [26]	• Convolutions séparables en profondeur
2019	<i>EfficientNet</i> [27]	• Approche dynamique • Utilisation des couches <i>Squeeze and excitation</i> • Utilisation de la fonction d'activation <i>Swish</i>

TABLE 1.1 – Liste des architectures principales de classification d'images et les contributions associées.

subtiles dans les données d'entrée, le processus d'extraction de caractéristiques produit des résultats qui peuvent être déformés ou simplifiés de manière incorrecte. En réponse à ces découvertes, Zeiler et al. ont procédé à des ajustements de la topologie et des paramètres d'*AlexNet*, notamment en réduisant la taille des filtres et le pas de déplacement dans les deux premières couches convolutionnelles, pour retenir un nombre maximal de caractéristiques utiles. Ces modifications ont conduit à une amélioration notable de la performance des CNN, démontrant l'importance de la visualisation des caractéristiques pour l'identification et la correction des lacunes dans la conception des réseaux de neurones profonds.

Les architectures présentées précédemment sont limitées par l'utilisation de grands filtres, ce qui entraîne une complexité computationnelle élevée. De plus, une architecture peu profonde limite l'apprentissage des caractéristiques visuelles complexes. Pour dépasser ces limites, le modèle VGG [21], qui a remporté le deuxième prix à l'ILSVRC 2014, propose une architecture plus profonde, allant jusqu'à 19 couches avec des filtres de petite tailles. Les auteurs montrent que l'utilisation successive de filtres de taille (3×3) permet d'atteindre le même champ réceptif qu'un filtre plus grand, tout en réduisant le nombre de paramètres nécessaires. Ainsi, ils augmentent la profondeur du réseau et introduisent davantage de non-linéarités, permettant une meilleure capture des caractéristiques complexes tout en optimisant la capacité d'apprentissage du modèle. Le modèle VGG est largement utilisé car son architecture est plus régulière et plus facile à comprendre que celle d'*AlexNet* et de *ZFNet*. Cependant, malgré cette conception uniforme, la principale limitation associée à VGG est son nombre élevé de paramètres (cf. Figure 1.2, qui compare le nombre de paramètres des modèles), ce qui rend le calcul coûteux et difficile à déployer sur des systèmes aux ressources limitées.

Contrairement à VGG qui privilégie la profondeur architecturale, Szegedy et al. ont introduit *GoogLeNet* [22], aussi appelé *Inception V1*, qui a gagné le premier prix à l'ILSVRC 2014 en proposant une approche plus efficiente en termes de calcul. L'objectif principal de *GoogLeNet* était d'atteindre une haute précision avec un coût computationnel réduit. Il a introduit le concept de module *Inception* dans les CNN, par lequel il incorpore des transformations convolutionnelles multi-échelles et des opérations de *pooling* au même niveau, permettant au modèle de capturer des informations à différentes échelles ou résolutions spatiales. *GoogLeNet* est composé de 3 têtes de classification pour favoriser l'apprentissage de caractéristiques utiles à différentes profondeurs du réseau, faciliter la convergence et agir comme une forme de régularisation. L'une des principales innovations de *GoogLeNet* est l'utilisation de noyaux de différentes tailles en parallèle, ce qui permet d'augmenter la largeur du modèle au lieu de sa profondeur. L'idée qui a conduit cette technique est qu'une telle architecture permet d'extraire simultanément des caractéristiques à différentes échelles. Malgré ses innovations, *GoogLeNet* présente des défis tels que la perte d'information due à l'utilisation de goulots d'étranglement. En effet, afin de réduire la complexité du modèle et mélanger l'information entre les différents canaux, *GoogLeNet* intègre des filtres de taille (1×1) comme couches de réduction dimensionnelle avant l'application de filtres de plus grande taille. De plus, la topologie hétérogène de l'architecture nécessite des ajustements des paramètres, comme la sélection du nombre de canaux ou la combinaison optimale des tailles de filtres dans les modules *Inception*, rendant la gestion et l'utilisation de *GoogLeNet* particulièrement complexes.

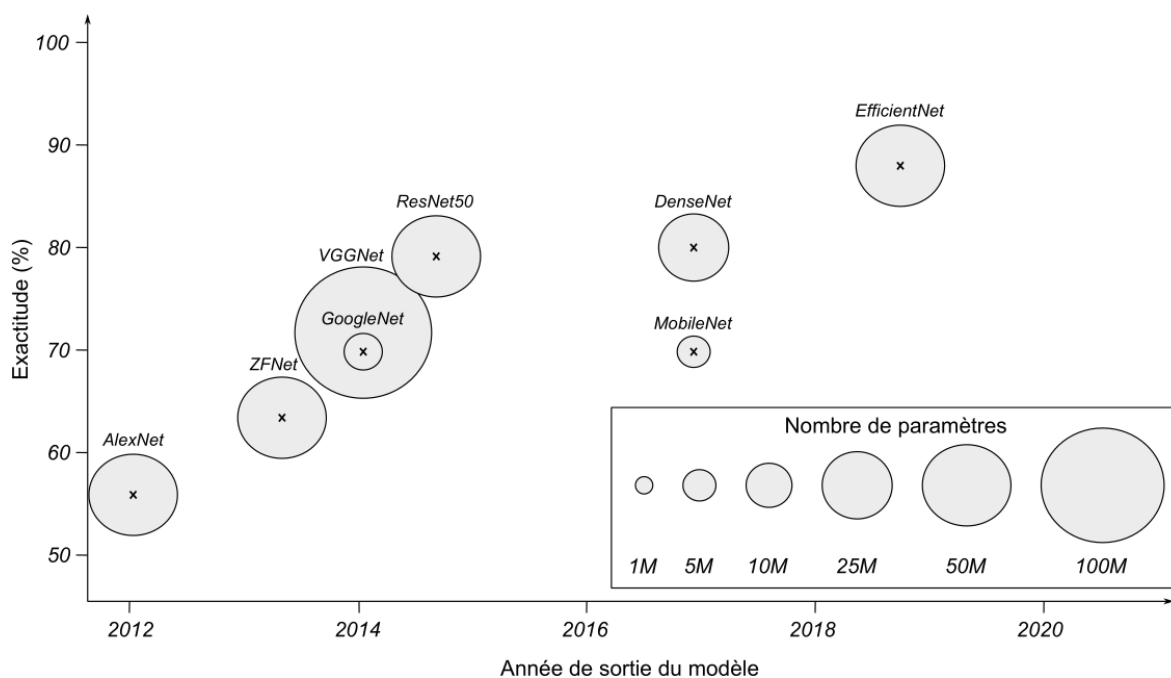


FIGURE 1.2 – Évolution des performances sur ImageNet et du nombre de paramètres des architectures CNN.

En se basant sur l'intuition que la capacité d'apprentissage peut être améliorée en augmentant la profondeur du réseau, Srivastava et al. ont introduit les réseaux *Highway* [23]. Sachant qu'un des défis majeurs des réseaux profonds est la convergence, les réseaux *Highway* utilisent leur profondeur pour acquérir une représentation plus riche des caractéristiques en intégrant un mécanisme de connectivité entre les couches. Ce mécanisme introduit des portes adaptatives qui régulent le flux d'information, permettant à chaque couche de déterminer quelle proportion du signal doit être transformée et quelle proportion doit être directement transmise aux couches suivantes. Cette régulation dynamique du flux d'information, inspirée des réseaux LSTM permet aux *Highway* d'avoir un temps de convergence bien meilleur que les modèles aux architectures profondes mais étroites. En effet, l'article d'introduction du modèle révèle que la performance des réseaux convolutifs diminue lorsque le nombre de couches dépasse 10, tandis que les réseaux *Highway* maintiennent une convergence rapide même à 900 couches, illustrant leur efficacité dans la gestion de la profondeur du réseau.

Toujours dans l'idée de créer des connexions directes entre les couches non successives pour faciliter l'apprentissage des réseaux profonds, *ResNet* (*Residual Network*) [24] a introduit l'apprentissage résiduel, marquant une évolution significative dans la conception des CNN. Avec son architecture de 152 couches, surpassant largement *AlexNet* et *VGG* en terme de profondeur mais avec moins de complexité computationnelle, *ResNet* a excellé au concours ILSVRC 2015. Comme les réseaux *Highway*, *ResNet* exploite la profondeur du réseau pour atteindre une représentation de caractéristiques de haute qualité. La clé de son succès réside dans les blocs résiduels qui, en facilitant le flux de gradients, contournent le problème de la

disparition du gradient³ (*vanishing gradient*) [30]. Ce problème, résultant de la mécanique de la rétropropagation, ralentit considérablement l'apprentissage et limite l'évolution des performances. L'utilisation des blocs résiduels permet ainsi l'entraînement efficace de réseaux très profonds avec des architectures comportant 50, 101 ou 152 couches. Ces réseaux ont démontré une performance supérieure en termes d'erreur de classification sur *ImageNet*, comparativement à un réseau de référence, conçu avec 34 couches, utilisé pour illustrer un exemple classique de réseau profond sans connexions résiduelles.

L'innovation des blocs résiduels est dans leur capacité à créer une "autoroute de gradient", où les gradients peuvent influencer directement les poids des premières couches, rendant les mises à jour plus efficaces et permettant de réduire l'erreur avec l'ajout de couches, contrairement aux réseaux traditionnels où l'ajout de couches supplémentaires peut augmenter l'erreur. La structure de base d'un bloc résiduel, schématisée dans la Figure 1.3, peut être simplifiée par la formule :

$$\text{Sortie} = f(x) + x \quad (1.1)$$

où, x représente l'entrée du bloc résiduel, et $f(x)$ représente la transformation appliquée à x par les couches du bloc (par exemple, des convolutions, des normalisations par lots, des activations, etc.). L'opération $f(x) + x$ est ce qu'on appelle une "connexion résiduelle", car elle ajoute directement l'entrée initiale x à la sortie de la fonction $f(x)$. L'idée est de permettre au réseau d'apprendre plus facilement la fonction identité : si une couche n'est pas nécessaire, le réseau peut simplement apprendre à mettre ses poids à zéro, laissant passer l'entrée inchangée via la connexion résiduelle. Ainsi, même si les couches supplémentaires ne parviennent pas à apporter une amélioration significative, le réseau conserve l'information de l'entrée initiale. Cela aide le gradient à se propager directement à travers les réseaux profonds sans être atténué, facilitant l'apprentissage.

L'utilisation de *ResNet* pour des apprentissages *from scratch*⁴, par transfert ou par affinage est très répandue, en raison de son excellent compromis entre la taille du réseau, en termes de paramètres et de couches, et la complexité computationnelle.

Tout comme les réseaux *Highway* et *ResNet*, *DenseNet* [25] a été conçu pour aborder le problème de la disparition du gradient. Contrairement à *ResNet*, qui utilise des transformations d'identité additives pour préserver l'information à travers les couches, *DenseNet* améliore la conservation et l'utilisation de l'information en connectant directement chaque couche à toutes celles qui la suivent. Cette approche assure que les cartes de caractéristiques de chaque couche sont utilisées comme entrées pour les couches suivantes, facilitant un flux d'information plus riche à travers le réseau. Bien que *DenseNet* offre un mécanisme efficace pour le flux d'informations et l'accès direct aux gradients, réduisant potentiellement le surapprentissage, cette architecture peut devenir coûteuse en termes de paramètres avec l'augmentation du nombre de connexions.

Dans un contexte où la tendance était à la création d'architectures convolutives toujours

3. La disparition du gradient est un phénomène qui se produit lors de l'entraînement des réseaux de neurones profonds, où le gradient devient extrêmement petit (proche de zéro) au fur et à mesure qu'il est rétropropagé à travers les couches. Cela rend l'apprentissage très lent ou impossible pour les couches les plus profondes du réseau. 4. Apprentissage *from scratch* (à partir de zéro) : méthode d'entraînement où tous les paramètres du réseau sont initialisés aléatoirement et ajustés uniquement à partir des données d'entraînement fournies, sans utiliser de connaissances préexistantes.

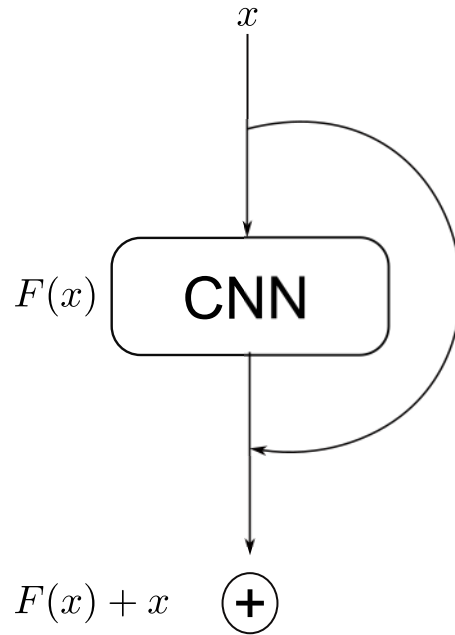


FIGURE 1.3 – Schéma simplifié d'un bloc résiduel où x représente l'entrée du bloc résiduel, et $f(x)$ représente la transformation appliquée à x par les couches du bloc. L'opération $f(x) + x$ ajoute directement l'entrée initiale x à la sortie de la fonction $f(x)$.

plus grandes pour améliorer les performances, *MobileNet* [26] est apparu comme une solution innovante pour les dispositifs ayant des contraintes strictes de mémoire et de calcul. *MobileNet* a introduit les convolutions séparables en profondeur, une technique qui décompose une convolution classique en deux étapes : une convolution en profondeur pour traiter les informations spatiales dans chaque canal séparément, et une convolution ponctuelle pour fusionner les caractéristiques entre les canaux. Cette approche réduit significativement le nombre de paramètres et la complexité computationnelle du modèle, tout en maintenant des performances compétitives.

MobileNetV2 [31] a poursuivi sur cette lancée en ajoutant des connexions résiduelles et en apportant des ajustements architecturaux pour réduire davantage le nombre de paramètres et la complexité computationnelle des modèles, facilitant ainsi leur déploiement sur des appareils avec des ressources limitées. Ces évolutions marquent un tournant vers des modèles CNN adaptés aux besoins des applications mobiles et embarquées, où l'efficacité computationnelle est cruciale.

Enfin, alors que tous les modèles présentés se caractérisent par des structures statiques et spécialisées, *EfficientNet* [27] propose une approche dynamique, ajustant de manière systématique certains paramètres de son architecture pour équilibrer efficacement performance et efficacité computationnelle. En proposant une architecture de base commune et en manipulant trois paramètres clés – la largeur (nombre de canaux par couche), la profondeur (nombre de couches) et la résolution (taille de l'image d'entrée), *EfficientNet* démontre qu'il est possible de concevoir des modèles offrant un bon compromis entre performance de classification et coût computationnel en ajustant ces dimensions de manière équilibrée. L'ajustement se fait en deux temps : d'abord une phase de calibration sur un modèle de

base pour déterminer les ratios optimaux entre ces trois dimensions, puis une phase de mise à l'échelle où ces ratios sont appliqués uniformément pour créer différentes versions du modèle, garantissant ainsi un équilibre optimal entre les paramètres.

EfficientNet intègre également des innovations telles que l'utilisation de couches *Squeeze-and-Excitation* [32], une technique d'attention qui recalibre adaptativement l'importance des canaux de caractéristiques, et la fonction d'activation *Swish*⁵ [33]. Cette dernière combine les propriétés des fonctions ReLU et sigmoïde, offrant une alternative qui améliore l'apprentissage dans les réseaux profonds. Ces deux techniques contribuent significativement à l'amélioration des performances de classification.

Aujourd'hui, l'utilisation des *Vision Transformers* (ViT) [34], qui ne sont pas des réseaux de neurones convolutifs, est largement répandue pour le traitement des images. Comme cette architecture a été initialement introduite pour les données séquentielles, nous la présentons à la fin de la Section 1.2.2.

1.2 Réseaux de neurones récurrents pour le traitement des séries temporelles

Les séries temporelles sont définies par l'observation des variations d'une donnée au cours du temps. Elles peuvent être univariées, suivant une seule variable, ou multivariées, suivant plusieurs variables simultanément. Leur complexité réside dans la dépendance temporelle et la saisonnalité, qui nécessitent des modèles capables de capturer des relations séquentielles et des tendances à long terme. Cela les rendent particulièrement difficiles à traiter par rapport aux données statiques.

Introduits en 1986, les réseaux de neurones récurrents (RNN) [28] sont une catégorie de réseaux de neurones conçus spécifiquement pour traiter des séquences de données, comme les séries temporelles. Contrairement aux réseaux de neurones traditionnels, qui traitent chaque entrée de manière indépendante, les RNN ont la capacité de maintenir un état caché. Cet état caché joue un rôle fondamental dans les RNN en capturant les informations des données précédemment traitées dans la séquence. Il fonctionne comme une forme de mémoire à court terme, permettant au réseau de conserver et d'utiliser le contexte des entrées précédentes lors du traitement de nouvelles données. Cette caractéristique des RNN leur permet de mieux comprendre et analyser les séquences où l'ordre et les relations temporelles entre les éléments sont cruciaux.

La Figure 1.4a détaille comment les informations passent entre chaque cellule RNN au fil du temps. Un RNN reçoit deux entrées à chaque pas de temps : l'état caché précédent h_{t-1} et l'entrée actuelle de la séquence x_t . Il calcule le nouvel état caché h_t et produit une sortie y_t . Ensuite, l'état caché actuel h_t est utilisé au pas de temps suivant, $t + 1$, et le processus tourne en boucle jusqu'à la fin de la séquence d'entrée. Cette architecture permet aux RNN

5. La fonction d'activation Swish, définie par $f(x) = x * \text{sigmoid}(x)$, combine les propriétés de ReLU et de la fonction *sigmoïde*. Elle préserve les avantages de ReLU pour les valeurs positives tout en permettant une activation douce et non nulle pour certaines valeurs négatives, grâce à l'influence de la *Sigmoïde*.

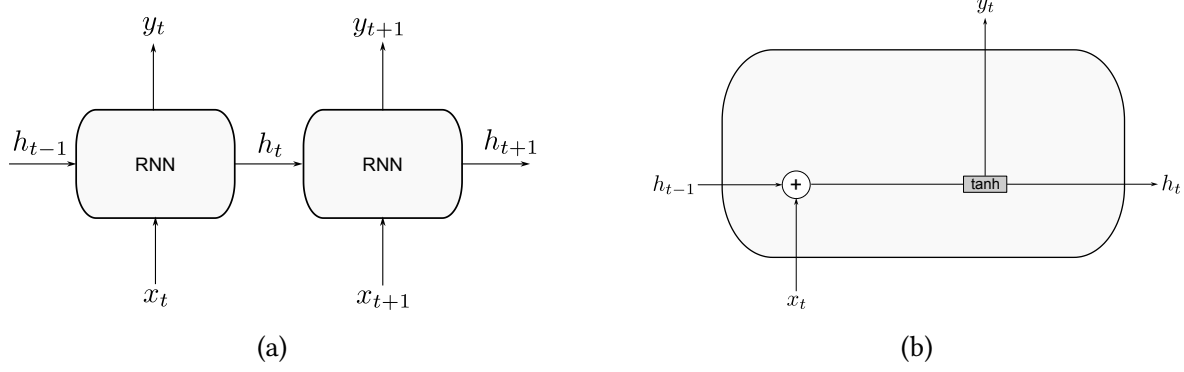


FIGURE 1.4 – Schéma d'un réseau de neurones récurrent. (a) représente le flux d'informations entre t et $t + 1$. (b) représente la cellule du RNN au temps t

de saisir les dépendances entre tous les éléments d'une séquence. La Figure 1.4b montre une illustration du calcul effectué dans une cellule RNN à chaque pas de temps. Plus précisément, le calcul est comme suit :

$$\begin{aligned} h_t &= \tanh(W_x x_t + W_h h_{t-1} + b_h) \rightarrow \text{état caché} \\ y_t &= W_y h_t + b_y \rightarrow \text{sortie} \end{aligned} \quad (1.2)$$

Où W_x , W_h et W_y sont des matrices de poids, et b_h , b_y sont des biais, tous appris par le modèle au cours de l'entraînement. Ces paramètres permettent au modèle d'ajuster les valeurs produites afin d'améliorer sa flexibilité et ses performances.

Bien que les RNN soient puissants, ils ont des limites lorsqu'il s'agit de saisir des dépendances à long terme, principalement à cause du phénomène de disparition du gradient [30], déjà évoqué pour motiver des mécanismes introduits dans les CNN.

1.2.1 Évolution des réseaux de neurones récurrents

L'architecture *Long Short-Term Memory* (LSTM) [35] a été initialement introduite en 1997 pour répondre aux limites des RNN standards sur la mémorisation des informations sur des séquences longues. La cellule type d'un LSTM, schématisée dans la Figure 1.5a, a été conçue pour permettre au réseau d'apprendre ce qu'il faut retenir et ce qu'il faut oublier. Elle se compose de quatre portes : la porte d'entrée i_t , la porte d'oubli f_t , la porte de l'état de la cellule g_t et la porte de sortie o_t . En plus de ces portes, les LSTM utilisent un état de cellule c_t considérée comme de la mémoire additionnelle.

Chaque porte remplit une fonction spécifique en contrôlant le flux d'informations de l'état caché précédent h_{t-1} et de l'entrée actuelle x_t . La porte de l'état de la cellule (g_t) détermine les nouvelles informations à ajouter à l'état de la cellule (c_t). Les portes d'oubli (f_t) et d'entrée (i_t) décident respectivement des informations à retenir de l'état de la cellule antérieur (c_{t-1}) et des nouvelles données à incorporer (x_t). Finalement, la porte de sortie (o_t) modifie l'état de la cellule mise à jour (c_t) pour produire l'état caché actuel (h_t), qui sert aussi de sortie (y_t). Les calculs des portes et des sorties sont formulés comme suit :

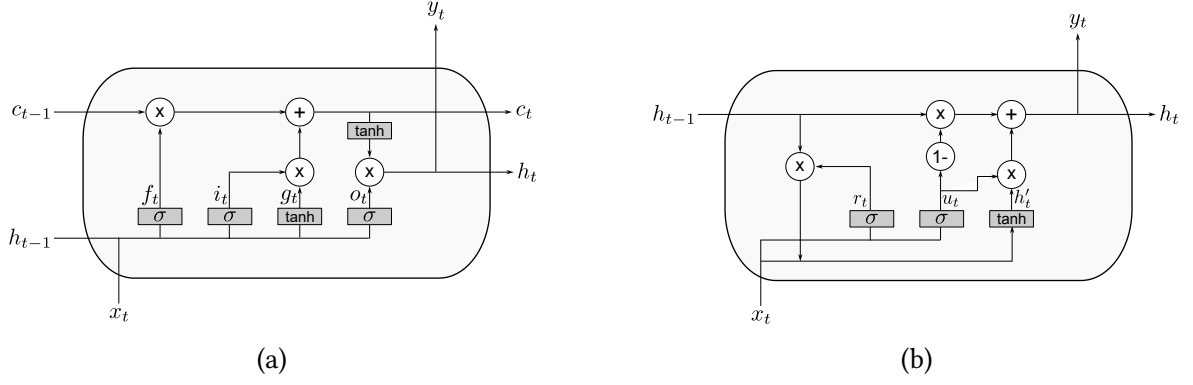


FIGURE 1.5 – Schéma des cellules LSTM (a) GRU (b).

$$\begin{aligned}
 f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \rightarrow \text{porte d'oubli} \\
 i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \rightarrow \text{porte d'entrée} \\
 o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \rightarrow \text{porte de sortie} \\
 g_t &= \tanh(W_g x_t + U_g h_{t-1} + b_g) \rightarrow \text{porte état de cellule} \\
 c_t &= f_t \odot c_{t-1} + i_t \odot g_t \rightarrow \text{état de cellule} \\
 h_t &= o_t \odot \tanh(c_t) \rightarrow \text{état caché} \\
 y_t &= h_t \rightarrow \text{sortie}
 \end{aligned} \tag{1.3}$$

Où σ correspond à la fonction *sigmoïde*, $W_f, W_i, W_o, W_g, U_f, U_i, U_o, U_g$ sont des matrices de poids et b_f, b_i, b_o, b_g des biais.

Bien que les LSTM représentent une avancée significative dans le traitement des dépendances à long terme dans les séquences de données, ils présentent des limitations dans la prise en compte des informations contextuelles futures. Pour répondre à ces limites, le modèle *Bidirectional Long Short-Term Memory* (Bi-LSTM) [36] a été introduit. Bi-LSTM est une extension de l'architecture LSTM qui considère à la fois le contexte passé et futur dans les tâches de modélisation de séquences. En effet, alors que les LSTM traditionnels traitent les données d'entrée uniquement dans la direction avant, Bi-LSTM se compose de deux couches LSTM parallèles : l'une traite la séquence d'entrée dans la direction avant, tandis que l'autre la traite dans la direction arrière. Ce traitement bidirectionnel permet au modèle de capturer des informations à partir des contextes passé et futur, engendrant une meilleure compréhension des dépendances temporelles au sein de la séquence.

Les modèles LSTM et Bi-LSTM représentent des progrès notables dans la capacité de capturer les dépendances à long terme au sein des séquences de données mais leur complexité structurelle entraîne un coût computationnel élevé, dû à un nombre accru de paramètres. Pour répondre à cette problématique, Cho et al. ont introduit l'architecture *Gated Recurrent Unit* (GRU) [37], une variante de LSTM avec une architecture simplifiée. En effet, cette architecture combine la porte d'entrée et la porte d'oubli du LSTM en une seule porte de mise à jour. Ainsi, une unité GRU, schématisée dans la Figure 1.5b, a deux composantes principales : une porte de mise à jour u_t , une porte de réinitialisation r_t . Contrairement aux LSTM, les GRU ne dépendent pas de l'état de la cellule. Les portes sont calculées sur l'état caché précédent h_{t-1} et l'entrée actuelle x_t . Dans les GRU, la porte de mise à jour u_t est calculée pour spécifier

ce qu'il faut garder de l'étape précédente par h_{t-1} et ce qu'il faut intégrer de la nouvelle entrée par h'_t . La porte de réinitialisation r_t influe également sur la quantité d'informations à oublier du passé. La cellule GRU peut être formulée mathématiquement de la manière suivante :

$$\begin{aligned}
 r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r) \rightarrow \text{porte de réinitialisation} \\
 u_t &= \sigma(W_u x_t + U_u h_{t-1} + b_u) \rightarrow \text{porte de mise à jour} \\
 h'_t &= \tanh(W_{h'} x_t + r_t \odot U_{h'} h_{t-1} + b_{h'}) \rightarrow \text{porte état actuel} \\
 h_t &= (1 - u_t) \odot h_{t-1} + u_t \odot h'_t \rightarrow \text{état actuel} \\
 y_t &= h_t \rightarrow \text{sortie}
 \end{aligned} \tag{1.4}$$

Où $W_r, W_u, W_{h'}, U_r, U_u, U_{h'}$ sont des matrices de poids et $b_r, b_u, b_{h'}$ sont des biais. Bien que les GRU nécessitent moins de ressources computationnelles que les LSTM, ils peuvent être moins performants sur des tâches nécessitant la modélisation de dépendances séquentielles complexes en raison de leur architecture simplifiée et de leur capacité légèrement inférieure à gérer les dépendances à long terme. Ainsi, le choix entre GRU et LSTM doit être guidé par un équilibre entre la complexité de la tâche et les contraintes de ressources disponibles.

1.2.2 Transformers

Les réseaux récurrents représentent une avancée majeure pour le traitement de données temporelles. Cependant, ils sont contraints par leur mémoire à court terme qui limite le traitement des longues séquences, par leur difficulté à capturer les dépendances à long terme dans les données, ainsi que par leur difficulté à être parallélisés. En effet, les RNN traitent les données de façon séquentielle, ce qui accroît le coût computationnel de l'entraînement des modèles.

Pour pallier ces problèmes, notamment pour la tâche de traduction, les Transformers [13] ont été introduits en 2017. L'architecture des Transformers, repose sur deux concepts fondamentaux pour traiter les mots d'une phrase, représentés sous la forme d'une série de tokens⁶ :

- l'encodage de la position des tokens d'une séquence, qui permet au modèle de prendre en compte l'importance de l'ordre des tokens ;
- le mécanisme d'attention, détaillé par la suite, qui, en pondérant l'importance jointe des tokens, permet au modèle de faire une prédiction en fonction de l'ensemble de la séquence plutôt que de considérer chaque token séparément.

Contrairement aux RNN, les Transformers gèrent les tâches de traitement séquence à séquence sans suivre un ordre séquentiel dans le traitement des données. Cette approche élimine la nécessité d'une récurrence, permettant ainsi un traitement parallèle des données et une accélération significative de l'entraînement. Pour pallier l'absence de traitement séquentiel et intégrer une notion d'ordre dans les séquences, les Transformers utilisent un

6. Un token est une unité de base dans le traitement du texte, généralement un mot, une partie de mot, ou un caractère spécial. Dans le contexte des Transformers, les tokens sont les éléments individuels dans lesquels un texte est divisé pour être traité par le modèle. Le processus de conversion d'un texte en tokens s'appelle la tokenisation.

mécanisme de codage de position basé sur des fonctions sinus et cosinus. Ces fonctions sont appliquées de manière à générer des vecteurs pour chaque position dans la séquence : le sinus pour les positions paires et le cosinus pour les impaires, qui sont ensuite additionnés aux représentations vectorielles des mots. Cette méthode assure que le modèle conserve une information sur la position relative des mots dans la séquence. De plus, le modèle Transformer est composé d'un encodeur et d'un décodeur. L'encodeur traite la séquence d'entrée et génère une représentation continue, tandis que le décodeur utilise cette représentation ainsi que la sortie précédente pour générer la séquence suivante, permettant ainsi une traduction ou une génération de texte étape par étape.

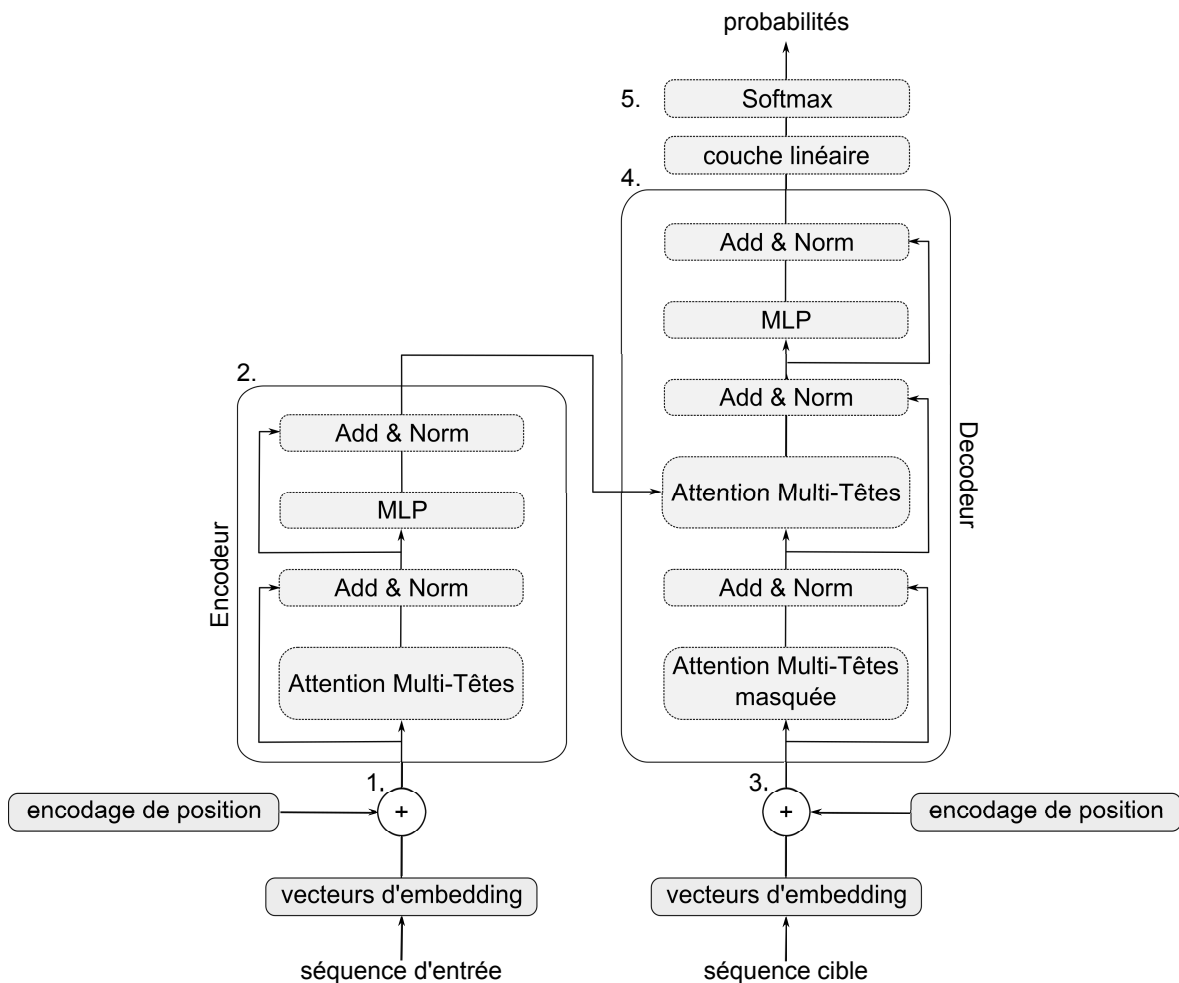


FIGURE 1.6 – Schéma simplifié de l'architecture d'un Transformer, représentant un seul bloc d'encodeur et de décodeur. Étapes numérotées de 1 à 5, avec 1 : conversion de la séquence d'entrée, 2 : traitement par l'encodeur, 3 : préparation de la séquence cible, 4 : traitement par le décodeur et 5 : génération de la séquence de sortie.

La Figure 1.6 schématise l'architecture d'un Transformer. Les étapes clés de l'apprentissage d'un Transformer, en prenant l'exemple de la tâche de traduction pour laquelle il a été initialement introduit, sont :

1. **Conversion de la séquence d'entrée :** Initialement, la séquence d'entrée est transformée en représentation vectorielle. Ce processus commence par la tokenisation de

la séquence en sous-mots (parties de mots) ou caractères, qui sont ensuite convertis en indices de vocabulaire. Ces indices sont transformés en représentations vectorielles auxquelles sont ajouté un codage de position, qui permet de préserver l'information séquentielle. Le résultat final est un ensemble de vecteurs représentant chaque token et sa position dans la séquence.

2. **Traitement par l'encodeur** : Ces vecteurs sont ensuite transmis à une série d'encodeurs. Chaque encodeur utilise un mécanisme d'attention multi-têtes pour capturer les relations contextuelles entre les différents éléments de la séquence. Ce processus enrichit la représentation des vecteurs avec des informations contextuelles et relationnelles, résultant en une représentation encodée de la séquence d'entrée qui condense son sens global et les interactions entre ses éléments.
3. **Préparation de la séquence cible** : En parallèle au traitement de l'entrée, le modèle prépare la séquence cible. Cette étape diffère selon qu'on soit en phase d'entraînement ou d'inférence. Lors de l'entraînement, la vraie séquence cible est utilisée : elle est tokenisée, convertie en indices, puis en représentations vectorielles, et un codage de position lui est ajouté, tout comme pour la séquence d'entrée. Un token spécial indiquant le début de la séquence est ajouté au début. Lors de l'inférence, seul ce token de début est utilisé pour initialiser le processus, car la séquence cible sera générée progressivement.
4. **Traitement par le décodeur** : Les vecteurs de la séquence cible sont traités par une série de décodeurs de manière autorégressive, ce qui signifie que la séquence de sortie génère un élément à la fois, chaque nouvel élément s'appuyant sur ceux précédemment générés. Dans chaque décodeur, le processus commence par une auto-attention masquée, où chaque position de la séquence cible ne peut accéder qu'aux positions précédentes, préservant ainsi l'ordre de génération. Ensuite, un module d'attention est appliqué, utilisant la représentation encodée de la séquence d'entrée pour guider la génération. Cette étape permet au décodeur d'accéder à l'information pertinente de la séquence source à chaque étape de la génération. Ce processus produit une représentation de la séquence cible qui est à la fois cohérente avec le contexte généré jusqu'à présent et alignée avec le sens de la séquence d'entrée.
5. **Génération de la séquence de sortie** : La représentation finale produite par le décodeur est passée à travers une couche de sortie, qui la convertit en une distribution de probabilités sur l'ensemble du vocabulaire cible. Cette distribution indique la probabilité de chaque token d'être le prochain dans la séquence de sortie. Le modèle construit alors la séquence finale en sélectionnant les tokens selon un critère défini, généralement en maximisant la probabilité ou en utilisant une méthode d'échantillonnage. Ce processus de génération et de sélection se répète pour chaque position de la séquence, jusqu'à ce qu'un token de fin soit produit ou qu'une longueur maximale soit atteinte.

Les informations contextuelles et relationnelles sont calculées avec le mécanisme d'auto-attention. Ce processus s'articule autour de la couche d'attention, qui opère en utilisant trois vecteurs : Requête (Query), Clé (Key) et Valeur (Value). Ces vecteurs sont obtenus par des projections linéaires de la représentation vectorielle de chaque token de la séquence, facilitant ainsi l'extraction et la comparaison d'informations à travers le mécanisme d'attention.

- **Requête (Query)** : Ces vecteurs servent à interroger le jeu de données, ciblant les

informations spécifiques que le modèle cherche à comprendre ou à associer.

- **Clé (Key)** : Ces vecteurs sont utilisés pour évaluer la pertinence des informations en fonction des requêtes, permettant de déterminer l'importance relative des différentes parties de la séquence.
- **Valeur (Value)** : Ces vecteurs contiennent l'information concrète à extraire, sélectionnée sur la base des analyses réalisées avec les vecteurs Requête et Clé.

Dans le contexte d'une séquence complète, les vecteurs Requête, Clé et Valeur sont regroupés en matrices Q , K , et V , où chaque ligne correspond aux projections d'un token. Ce regroupement permet au mécanisme d'attention de traiter efficacement les interactions entre tous les tokens de la séquence, capturant ainsi les relations complexes et les dépendances à longue distance au sein du texte.

La formulation mathématique de l'Attention, s'exprime comme suit :

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^\top}{\sqrt{d_k}} \right) V \quad (1.5)$$

où d_k est la dimension des vecteurs de clés et de requêtes. Le produit matriciel $QK^\top$ mesure la similarité entre les vecteurs Requête et Clé, influençant le degré d'attention que chaque élément porte aux autres. La division par $\sqrt{d_k}$ normalise ce produit, aidant à maintenir la stabilité du gradient pendant l'apprentissage en évitant des valeurs extrêmement grandes. L'application de la fonction softmax convertit ensuite ces scores de similarité normalisés en poids d'attention, déterminant ainsi l'importance relative de chaque vecteur Valeur (les lignes de la matrice V) dans la représentation finale.

Notons que le mécanisme d'attention est un concept général permettant au modèle de se concentrer sur des parties spécifiques de l'entrée, l'auto-attention est une variante où chaque élément interagit avec tous les autres de la même séquence ; la fonction Attention est la formulation mathématique de ce processus, tandis que le module d'attention est l'implémentation concrète de ce mécanisme dans l'architecture du modèle.

L'architecture Transformer inclut un module d'attention multi-têtes qui réalise plusieurs calculs d'attention en parallèle, chacun étant effectué par une tête d'attention distincte. Chaque tête apprend ses propres projections linéaires pour les vecteurs Requête, Clé et Valeur à partir des mêmes représentations vectorielles d'entrée. Ces têtes travaillent indépendamment, permettant au modèle de capturer différents types de relations et de nuances dans les données. Après que chaque tête ait réalisé ses calculs d'attention, leurs sorties sont concaténées puis projetées linéairement pour former la sortie finale de la couche. Cette approche permet au Transformer de comprendre et d'intégrer de multiples relations et subtilités pour chaque token, améliorant ainsi sa capacité à traiter les nuances linguistiques et contextuelles complexes.

Dans la structure du Transformer, le mécanisme d'attention est utilisé à trois endroits distincts, chacun ayant un rôle spécifique dans le traitement de la séquence :

1. **Auto-attention dans l'Encodeur** : la séquence d'entrée est analysée par rapport à elle-même. Cela permet à chaque mot de la séquence de tenir compte de l'ensemble de la séquence lors de sa transformation, favorisant ainsi une compréhension contextuelle enrichie de chaque élément par rapport à tous les autres.

2. **Auto-attention masquée dans le Décodeur** : la séquence cible examine sa propre structure interne, mais de manière masquée. Chaque élément de la séquence ne peut porter attention qu'aux éléments précédents, préservant ainsi l'aspect autorégressif de la génération. Ce processus permet de maintenir la cohérence et la pertinence contextuelle lors de la génération de la séquence de sortie.
3. **Attention Encodeur-Décodeur dans le Décodeur** : la séquence cible porte attention à la séquence d'entrée. Cela signifie que lors de la génération de chaque mot de la séquence cible, le modèle prend en compte l'intégralité de la séquence d'entrée, permettant ainsi une traduction fidèle et informée par le contexte initial.

En résumé, ce processus permet d'obtenir une analyse des relations contextuelles des tokens de la séquence d'entrée et de générer une séquence cible pertinente, grâce à l'intégration dynamique des informations à travers les mécanismes d'auto-attention⁷ et d'attention⁸ Encodeur-Décodeur.

La performance des Transformers est aussi due à d'autres fonctions clés qui renforcent leur capacité à traiter efficacement des séquences complexes. Parmi elles, les connexions résiduelles facilitent la préservation du flux d'informations à travers les couches du réseau, en ajoutant directement l'entrée de chaque sous-couche à sa sortie, après normalisation (cf. couches *Add & Norm* dans la Figure 1.6). Cette approche aide à lutter contre les problèmes de disparition ou d'explosion⁹ des gradients, garantissant ainsi un entraînement plus stable et efficace. La combinaison de tous ces mécanismes contribue à la flexibilité et à la puissance des Transformers, leur permettant de capturer avec précision une vaste gamme de relations contextuelles et syntaxiques au sein des données traitées.

Les modèles de type Transformer ont connu une évolution remarquable ces dernières années [38], marquant une avancée significative dans le domaine du traitement du langage naturel. Ces modèles se déclinent en trois grandes catégories architecturales : celles basées uniquement sur l'encodeur, celles utilisant exclusivement le décodeur, et les architectures séquence à séquence qui combinent les deux.

Malgré leur puissance, l'entraînement de ces modèles *from scratch* reste un défi de taille, nécessitant des volumes de données conséquents pour atteindre des performances optimales. Pour surmonter cette contrainte, l'utilisation de modèles pré-entraînés s'est largement répandue. Ces modèles, initialement entraînés sur de vastes corpus, peuvent être affinés pour des tâches particulières, permettant ainsi d'exploiter la puissance des Transformers sans subir les coûts computationnels élevés liés à leur entraînement initial.

Les applications de ces différentes architectures sont variées. Les modèles basés sur l'encodeur sont optimisés pour les tâches de compréhension, telles que la classification de textes ou l'analyse de sentiments. Les architectures utilisant uniquement le décodeur se distinguent dans la génération de texte, exploitant leur capacité à prédire la probabilité des séquences de tokens. Enfin, les modèles séquence à séquence, combinant encodeur

7. L'auto-attention se concentre sur les éléments de la même séquence, en considérant les relations entre chaque élément et tous les autres, y compris lui-même. 8. Le mécanisme d'attention se concentre sur les éléments d'une séquence en fonction de leur pertinence par rapport à un contexte ou une requête spécifique. Elle est souvent utilisée dans les tâches de séquence à séquence. 9. L'explosion du gradient est un phénomène qui se produit lors de l'entraînement des réseaux de neurones, où les valeurs du gradient deviennent excessivement grandes durant la rétropropagation, entraînant des mises à jour de poids instables et potentiellement la divergence du modèle.

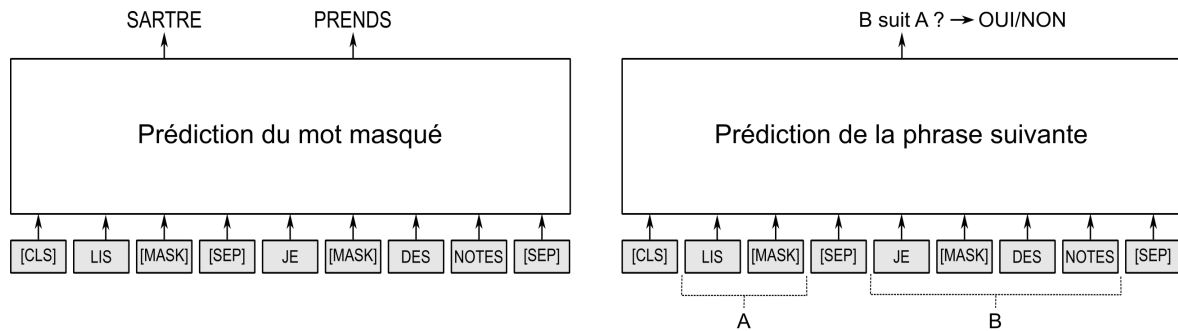


FIGURE 1.7 – Schéma du pré-entraînement de BERT : le modèle est pré-entraîné simultanément sur deux tâches prenant la même séquence en entrée. La séquence est composée de deux phrases A et B (séparées par le marqueur "[SEP]"), dont certains mots sont masqués. La première tâche consiste à prédire les mots masqués de la séquence. La deuxième tâche, une classification binaire, évalue si la phrase B suit logiquement la phrase A.

et décodeur, sont particulièrement efficaces pour les tâches de transformation comme la traduction automatique ou le résumé de textes.

Parmi les modèles n'utilisant que la partie encodeur du Transformer, le modèle *Bidirectional Encoder Representations from Transformers* (BERT) [39] est couramment utilisé pour diverses tâches de traitement du langage naturel. L'architecture de BERT est composée de plusieurs encodeurs empilés. Elle se distingue par sa capacité à comprendre le contexte des mots d'un texte de manière bidirectionnelle. En effet, son mécanisme d'attention est capable de prendre en compte les deux directions du token en cours, la gauche et la droite. BERT est pré-entraîné sur deux tâches simultanées pour apprendre les relations générales entre les tokens : la modélisation du langage masqué (MLM) et la prédiction de la phrase suivante (*Next Sentence Prediction* (NSP)). La Figure 1.7 schématise le pré-entraînement de BERT sur les deux tâches, qui sont effectuées avec la même séquence en entrée : une séquence de deux phrases, séparées par un marqueur "[SEP]", dont certains mots sont masqués. La première tâche consiste à prédire les mots masqués tandis que la deuxième consiste à faire une classification binaire qui évalue si la deuxième phrase est une suite logique de la première phrase. BERT est très souvent utilisé pour d'autres tâches après affinage (*fine-tuning*).

De nombreuses variantes de BERT ont été introduites, telles que XLNet [40], qui utilise une approche autoregressive basée sur une méthode de permutation pour prédire les tokens dans différents ordres, apprenant ainsi les dépendances contextuelles sans masquage explicite ; RoBERTa [41], qui optimise l'entraînement de BERT en supprimant la tâche de prédiction de la phrase suivante et en introduisant un masquage dynamique ; et DistilBERT [42], qui réduit le nombre de paramètres et la complexité computationnelle de BERT en utilisant des techniques de distillation des connaissances, c'est-à-dire en entraînant un modèle plus compact pour imiter le comportement d'un modèle plus volumineux.

Le modèle T5, ou *Text-to-Text Transfer Transformer* [43] se distingue par son approche unifiée séquence à séquence, où toutes les tâches sont formulées comme une transformation de texte en texte. Autrement-dit, T5 traite systématiquement l'entrée et la sortie comme des séquences de texte. Le processus de T5 inclut une phase de pré-entraînement et une phase d'affinage. Lors du pré-entraînement, il utilise une technique appelée *Span Corruption*,

où certaines parties du texte sont masquées et doivent être prédites par le modèle. À la différence de la tâche de prédiction du mot masqué utilisée par BERT, T5 masque des séquences continues (spans) de tokens, et ces séquences peuvent être de longueur variable. Le modèle est ensuite affiné pour des tâches spécifiques en transformant les entrées en texte et en générant les sorties correspondantes. L'architecture de T5 repose sur le modèle Transformer original avec des modules d'encodage et de décodage. L'encodeur transforme la séquence d'entrée en une représentation intermédiaire, tandis que le décodeur utilise cette représentation pour générer la séquence de sortie.

Ces dernières années, les modèles n'utilisant que la partie décodeur du Transformer ont connu un essor remarquable, notamment avec l'introduction du modèle *Generative Pre-Training* (GPT) [44]. GPT a introduit le concept d'un pré-entraînement non supervisé, où le modèle apprend à partir de grandes quantités de texte non étiqueté en prédisant le mot suivant dans une séquence. Cette phase est suivie d'un apprentissage supervisé pour l'affinage du modèle sur des tâches spécifiques en utilisant des données annotées. Cette approche, qui a considérablement amélioré les performances en génération de texte, est aujourd'hui adoptée par de nombreux modèles basés sur les Transformers.

Les modèles GPT ont connu plusieurs évolutions significatives, marquées par l'élargissement des capacités et l'augmentation de la taille des architectures en termes de paramètres. Le premier modèle, GPT-1 [44], dont l'architecture contient 117 millions de paramètres, est capable de générer du texte, de traduire des langues, de rédiger différents types de contenus créatifs et de répondre aux questions de manière informative. GPT-2 [45], dont l'architecture contient 1,5 milliard de paramètres, est capable de générer des textes encore plus proches de l'humain que GPT-1 et peut rédiger différents types de contenus créatifs. GPT-3 [46], contenant 175 milliards de paramètres, est capable d'effectuer un large éventail de tâches, notamment répondre aux questions de manière informative, même si elles sont ouvertes, générer différents formats de textes, tels que des poèmes, des scripts, des morceaux de musique, des mails, des lettres, etc. Enfin, GPT-4 [47], dont la taille est estimée à environ 1,8 trillion de paramètres, peut produire des textes à la sonorité plus naturelle et résoudre des problèmes avec plus de précision que son prédécesseur. Il peut également traiter des images en plus du texte.

Inspirés par le succès des Transformers dans le traitement du langage naturel, les *Vision Transformers* (ViT) [34] représentent une évolution significative dans le domaine de la vision par ordinateur. Contrairement aux approches traditionnelles de traitement d'images, qui s'appuient sur la manipulation directe des pixels ou des caractéristiques locales, les ViT traitent les images en les découpant par patch et en les considérant comme des séquences d'éléments d'entrée (cf. Figure 1.8). Chaque patch est ensuite projeté dans un espace de représentation vectoriel et traité à travers plusieurs couches de Transformer, qui utilisent l'attention pour capturer des relations globales entre les patches, indépendamment de leur position initiale dans l'image. Cette approche permet aux ViT de surmonter certaines limitations des approches classiques de vision par ordinateur, présentées précédemment, notamment la capacité à capturer des dépendances à longue distance entre les caractéristiques de l'image. En se focalisant sur les relations entre tous les patches d'une image, les ViT peuvent théoriquement apprendre des motifs visuels complexes sans être contraints par la structure fixe et localisée des opérations de convolution.

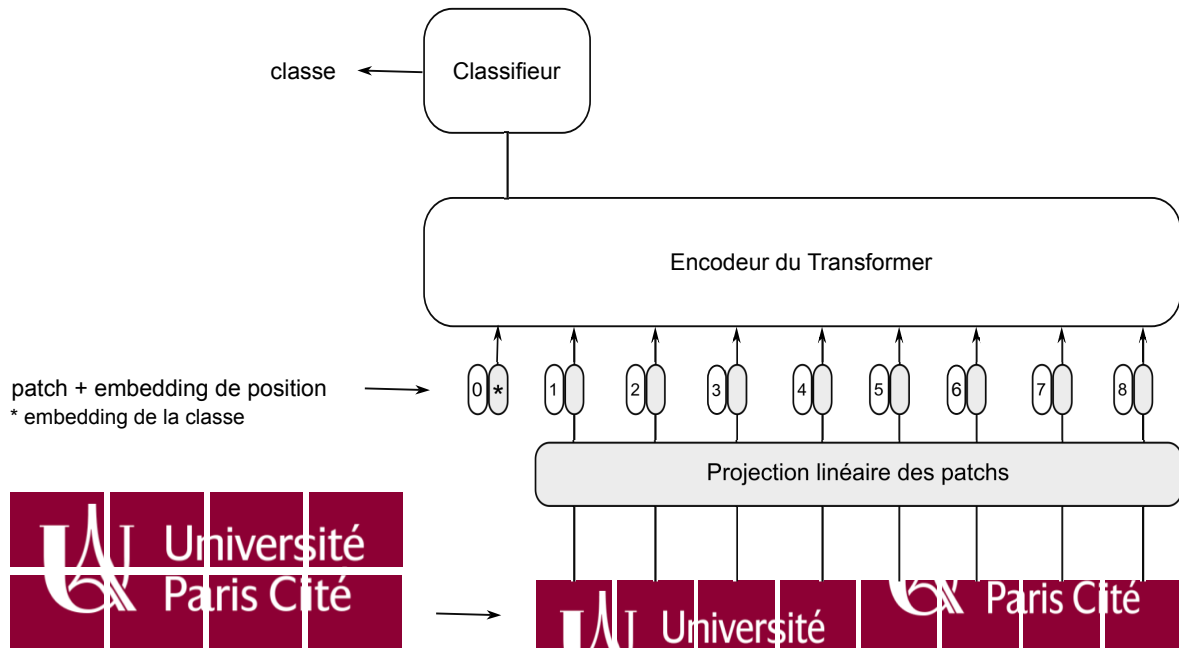


FIGURE 1.8 – Schéma d'un Vision Transformer (ViT)

1.3 Traitement des séries temporelles d'images

Le traitement des séries temporelles d'images est un domaine à l'intersection de l'analyse d'images et de l'étude des séries temporelles. Les séries temporelles d'images combinent les caractéristiques spatiales inhérentes aux images avec la dimension temporelle propre aux séries temporelles, créant ainsi un défi analytique multidimensionnel.

Face à cette dualité de caractéristiques, Ji et al. ont proposé le modèle 3D CNN [48], une extension des CNN, qui s'appliquent normalement sur des données à 2 dimensions, en adaptant leur architecture pour intégrer une troisième dimension : la temporalité. Pour cela, les 3D CNN, introduisent des convolutions tridimensionnelles au cœur de leur processus de convolution. Ces convolutions 3D manipulent une séquence d'images, utilisant un noyau 3D pour parcourir non seulement la largeur et la hauteur, mais aussi la profondeur temporelle de la séquence. Grâce à cette construction, les cartes de caractéristiques de la couche de convolution 3D sont connectées à plusieurs images contiguës de la couche précédente, ce qui permet de capturer des informations sur le mouvement.

Bien que les 3D CNN offrent une approche directe pour traiter conjointement les informations spatiales et temporelles, ils sont limités par leur difficulté à capturer les dépendances à long terme. Pour pallier cette problématique, l'utilisation d'un modèle combinant deux modèles spécialement conçus pour traiter les deux natures de données différentes, peut être utile. Le modèle hybride CNN-LSTM [49], par exemple, est composé d'un modèle convolutif, suivi d'un modèle récurrent, ce qui permet de capturer les informations spatiales et temporelles des données. En effet, le CNN est chargé de l'extraction des caractéristiques spatiales des images, qui sont ensuite données en entrée d'un LSTM, qui analyse leur évolution au fil du temps, permettant de modéliser la dynamique des phénomènes observés à travers la série d'images. L'architecture d'un CNN-LSTM, qui prend en entrée une série temporelles

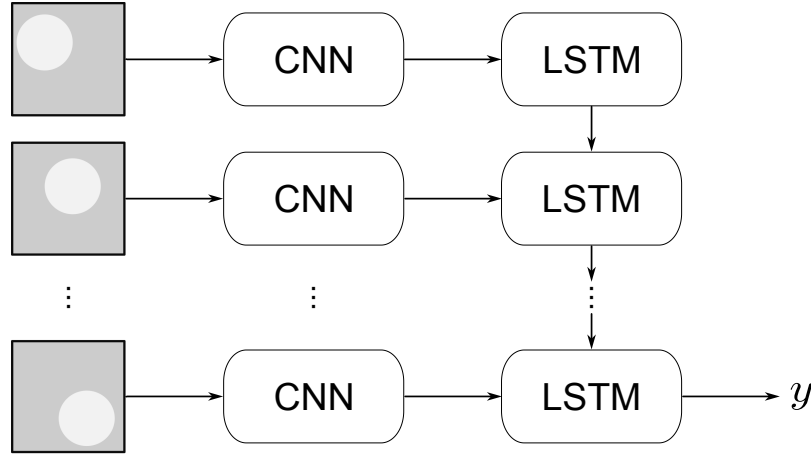


FIGURE 1.9 – Schéma d'un modèle CNN-LSTM

d'images et qui fait un prédiction y , est montrée dans la Figure 1.9.

Une variante du CNN-LSTM est le *Convolutional LSTM* (ConvLSTM) [50]. Inspiré du LSTM, le ConvLSTM est conçu pour traiter des séquences de données, mais il innove en remplaçant les opérations internes de multiplication matricielle par des convolutions. Cette modification permet au modèle de traiter efficacement les données ayant des dimensions spatiales. Comme le LSTM, le ConvLSTM se compose de quatre portes - la porte d'entrée I_t , la porte d'oubli F_t , la porte de l'état de la cellule G_t et la porte de sortie O_t - d'un état de cellule C_t et d'un état caché H_t (cf. Figure 1.10). Le ConvLSTM étend l'architecture du LSTM en appliquant des convolutions dans le calcul de ces portes, ce qui change radicalement la façon dont les données sont traitées. Au lieu de manipuler des vecteurs, le ConvLSTM opère sur des cartes de caractéristiques multidimensionnelles, permettant une analyse fine des informations spatiales contenues dans les séquences d'images. Le calcul des portes, des états et de la sortie d'une cellule ConvLSTM est proche de ceux du LSTM (cf. équation 1.3) et se fait de la manière suivante :

$$\begin{aligned}
 F_t &= \sigma(W_F * X_t + U_F * H_{t-1} + b_F) \\
 I_t &= \sigma(W_I * X_t + U_I * H_{t-1} + b_I) \\
 O_t &= \sigma(W_O * X_t + U_O * H_{t-1} + b_O) \\
 G_t &= \tanh(W_G * X_t + U_G * H_{t-1} + b_G) \\
 C_t &= F_t \odot C_{t-1} + I_t \odot G_t \\
 H_t &= O_t \odot \tanh(C_t) \\
 Y_t &= H_t
 \end{aligned} \tag{1.6}$$

Où σ correspond à la fonction *sigmoïde*, $W_F, W_I, W_O, W_G, U_F, U_I, U_O, U_G$ sont des matrices de poids; b_F, b_I, b_O, b_G des biais et $*$ est l'opération de convolution.

De la même façon que le GRU simplifie l'architecture du LSTM pour diminuer le coût computationnel, le ConvGRU [51] est l'architecture simplifiée du ConvLSTM qui combine l'état de la cellule et l'état caché en réduisant le nombre de portes. Comme le ConvLSTM, il remplace les multiplications matricielles par des convolutions.

Poursuivant l'exploration des modèles capables de traiter efficacement les dimensions

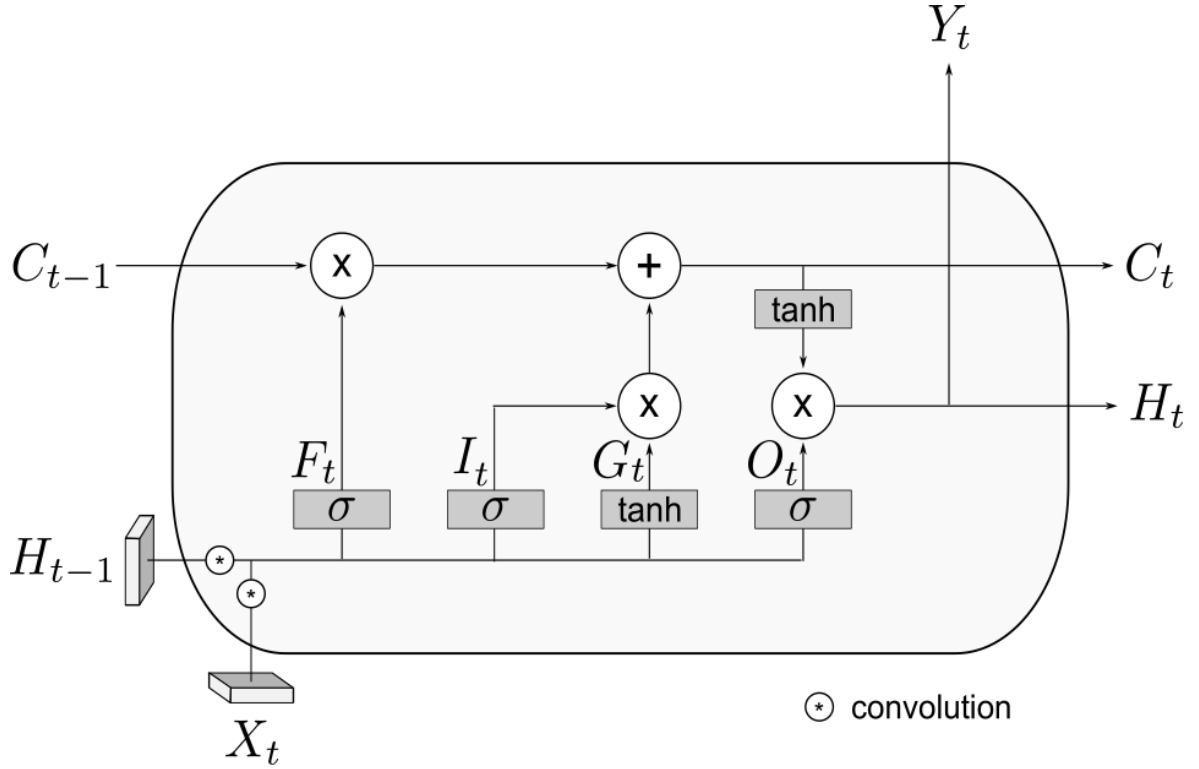


FIGURE 1.10 – Schéma d'une cellule ConvLSTM

spatiale et temporelle des données, notamment dans des contextes tels que la reconnaissance d'actions dans les vidéos, les architectures à deux flux, ou *Two-Stream Convolutional Networks* [52], ont marqué une étape importante dans le domaine de la vision par ordinateur. Ces réseaux exploitent une approche parallèle qui traite séparément mais simultanément les caractéristiques spatiales et temporelles des vidéos, afin d'extraire un vecteur de caractéristiques spatio-temporelles. Pour ce faire, le modèle à deux flux se compose de deux réseaux convolutifs distincts : un réseau spatial et un réseau temporel. D'une part, le réseau spatial est conçu pour analyser chaque image individuelle de la séquence vidéo, capturant ainsi les caractéristiques spatiales telles que la forme, la couleur et la texture des objets dans l'image. Cette analyse permet au modèle de comprendre ce qui est présent dans chaque image, offrant une perception des éléments statiques de la scène. D'autre part, le réseau temporel se focalise sur les variations entre les images consécutives de la séquence vidéo, traitant les différences d'apparence qui résultent du mouvement. Pour cela, il utilise des cartes de flux optique, qui quantifient le mouvement de chaque pixel entre deux images consécutives. Le réseau temporel est donc capable de saisir les mouvements et les actions en analysant comment les objets se déplacent ou changent de position au fil du temps. Une fois que les deux réseaux ont traité les données, leurs vecteurs de caractéristiques respectifs sont combinés, pour produire un vecteur de caractéristiques spatio-temporelles unifié. Ce vecteur combiné offre une représentation complète des informations contenues dans la vidéo, englobant à la fois l'aspect statique des scènes et la dynamique des mouvements.

Le problème des architectures à deux flux est leur incapacité à modéliser la structure temporelle à long terme à cause de leur accès limité au contexte temporel. En effet, ils sont conçus pour fonctionner uniquement sur une seule image (réseau spatial) ou sur

une seule pile d'images dans un court laps de temps (réseau temporel). Pour résoudre ce problème, Wang et al. ont introduit les réseaux de segments temporels ou *Temporal Segment Networks* (TSN) [53], une approche modulaire qui divise les vidéos en segments et traite chaque segment individuellement. Plus précisément, les TSN sont, comme les réseaux à deux flux, composés d'un réseau à flux spatial et un réseau à flux temporel mais contrairement aux réseaux à deux flux, qui travaillent sur des images uniques ou des piles d'images, les TSN opèrent sur un ensemble de courtes séquences échantillonnées de manière éparse dans l'ensemble de la vidéo. Chaque extrait de cet ensemble produit sa propre prédiction préliminaire et la prédiction finale est calculée comme un consensus de toutes les prédictions préliminaires.

Enfin, parmi les avancées significatives dans le traitement des séries temporelles d'images, les variantes du ViT illustrent l'adaptation de l'architecture Transformer pour exploiter à la fois la dimension spatiale et temporelle des données. Le modèle *TimeSformer* [54] étend l'architecture ViT en appliquant séparément l'attention temporelle et spatiale, permettant une analyse efficace des séquences vidéo en capturant des dynamiques temporelles fines au-delà des capacités des CNN. Quant au modèle ViViT [55], il utilise une approche hybride, combinant l'attention factorisée en espace et en temps pour traiter efficacement de gros volumes de données. Au-delà de ces architectures génériques, des variantes spécialisées émergent pour répondre aux spécificités de certains types de données, comme les données satellitaires [56].

Chapitre 2

Représentation des données tabulaires en images pour pallier le fléau de la dimension

2.1 Introduction aux défis liés au fléau de la dimension

Face aux données à haute dimension, telles que les séries temporelles, les modèles d'apprentissage automatique, y compris profond, peuvent se heurter au fléau de la dimension (*curse of dimensionality*) [3]. Ce phénomène se traduit par plusieurs défis significatifs. Tout d'abord, la dilution de la densité des données : à mesure que le nombre de dimensions augmente, le volume de l'espace de données accroît exponentiellement, nécessitant une quantité proportionnellement plus grande de données pour maintenir une densité suffisante. Cela rend l'apprentissage plus difficile et augmente le risque de surapprentissage [4], où les modèles s'adaptent trop spécifiquement aux données d'entraînement, incluant le bruit, au détriment de la généralisation. De plus, la complexité computationnelle s'élève, intensifiant les ressources requises pour le stockage et le traitement, rallongeant ainsi les temps d'entraînement. Enfin, les difficultés d'optimisation deviennent plus prévalentes, avec des problèmes tels que la disparition du gradient [30]. Ce phénomène se produit lorsque les gradients deviennent extrêmement petits lors de la rétropropagation à travers de nombreuses couches, rendant l'apprentissage inefficace pour les premières couches du réseau. Cela est particulièrement problématique dans les réseaux profonds et les architectures récurrentes traitant de longues séquences, où les dépendances à long terme sont difficiles à capturer.

Pour pallier les défis liés au fléau de la dimension, l'approche la plus couramment utilisée est de réduire les dimensions du jeu de données [57], ce qui consiste à transformer un espace à haute dimension en un espace à dimension réduite, tout en préservant les caractéristiques les plus pertinentes des données. Cependant, l'utilisation de ces approches engendre souvent une perte d'informations, ce que nous souhaitons éviter. Par exemple, dans le domaine médical, où les études sont souvent exploratoires, la réduction de dimensions peut avoir des conséquences significatives. En effet, l'utilisation de ces approches peut

engendrer une négligence des informations qui, à première vue, ne semblent pas essentielles, mais qui peuvent en réalité s'avérer cruciales pour l'étude d'un phénomène. Des données apparemment anodines pourraient révéler des cas particuliers ou des anomalies qu'il est important de prendre en compte.

Pour surmonter cette problématique, nous proposons une approche permettant d'exploiter l'exhaustivité des caractéristiques des données séquentielles à haute dimension sans subir le fléau de la dimension. Notre méthode consiste à transformer des données tabulaires représentées par des vecteurs unidimensionnels (1D) en vecteurs bidimensionnels (2D), sous forme d'images. Cela nous permet de tirer parti des performances d'un modèle basé sur une architecture convolutive pour l'extraction d'informations, couplé à un modèle récurrent pour capturer la temporalité des données. Cette approche améliore non seulement les performances de prédiction par rapport aux méthodes traditionnellement appliquées sur les données séquentielles, mais elle démontre également l'avantage de représenter l'information sous une autre forme pour obtenir de meilleurs résultats en moins de temps.

Contributions

1. Nous proposons une méthode de **transposition de données multidimensionnelles en images** pour pallier le *fléau de la dimension* [3].
2. Nous proposons l'utilisation d'un **modèle hybride léger**, en termes de paramètres et de couches, permettant la reconnaissance de motifs dans les images ainsi que l'analyse de la corrélation entre les différents pas de temps d'une série temporelle.
3. Nous validons notre approche pour la tâche de classification multi-classes sur des **données de santé bruitées** en la comparant à des modèles couramment utilisés pour le traitement de longues séquences temporelles.

Nos travaux sont précédés de deux études préliminaires permettant de définir les limites des CNN lorsqu'ils sont appliqués à des images où l'information à identifier n'est pas visuellement reconnaissable, que ce soit parce qu'elle est encodée ou en raison de la présence de bruit. Ces études sont essentielles pour notre travail car elles nous permettent d'anticiper les défis que nous rencontrerons lors de l'application des architectures convolutives sur les images résultant de la transposition de nos données tabulaires, qui seront de nature non figurative et potentiellement bruitées.

Plan du chapitre

Approches de réduction de dimension (Section 2.2) : état de l’art des approches de réduction de dimension et étude de leurs limites.

Alternative à la réduction de dimension (Section 2.3) : présentation des travaux existants sur la transposition de données tabulaires en images ainsi que le positionnement de notre approche.

Études des limites des CNN sur des images bruitées (Section 2.4) : présentation des deux études préliminaires pour définir les limites des CNN sur des images figuratives (Section 2.4.1) ou abstraites (Section 2.4.2) bruitées.

Transformation de données séquentielles en images et application d’un modèle hybride (Section 2.5) : méthodologie de notre approche.

Expérimentations (Section 2.6) : cas d’usage de détection de pathologies à partir de données de santé et comparaison avec des modèles traditionnels de traitement de séquences.

Résultats et discussions (Section 2.7)

Conclusion et perspectives (Section 2.8)

2.2 Approches de réduction de dimension

Dans cette partie, nous détaillons l’état de l’art des approches de réduction de dimension appliquées aux données non séquentielles (ou statiques), puis celles conçues pour les données séquentielles. Enfin, nous discutons des limites de ces approches.

2.2.1 Réduction de dimension des données statiques

Les approches de réduction de dimension des données statiques peuvent être classées en deux grandes catégories : la sélection de caractéristiques et l’extraction de caractéristiques.

Les approches de **sélection de caractéristiques** identifient et conservent uniquement les variables les plus significatives du jeu de données, en éliminant celles qui sont redondantes ou peu informatives. Parmi les méthodes populaires, on trouve l’élimination récursive de caractéristiques (*Recursive Feature Elimination* (RFE)) [58], les tests statistiques comme l’ANOVA [59] et le test du Chi-carré [60], ainsi que l’analyse du coefficient de corrélation de *Pearson* [61] pour détecter et retirer les variables fortement corrélées.

Les approches d’**extraction de caractéristiques** procèdent à une transformation des données initiales en un ensemble réduit de variables, généralement moins nombreuses,

qui capturent les informations essentielles présentes dans les données originales. Cette catégorie inclut l'*Analyse en Composantes Principales* (ACP) [62], qui réduit les dimensions en transformant les données initiales en un nombre plus petit de composantes non corrélées ; l'*Analyse Discriminante Linéaire* (ADL) [63], qui maximise la séparabilité entre les classes ; l'*Analyse en Composantes Indépendantes* (ACI) [64], qui cherche à transformer les données en composantes statistiquement indépendantes ; l'*Analyse Factorielle* [65], qui modélise les variables observées comme des combinaisons linéaires de facteurs inobservés, réduisant ainsi la complexité des données ; et *Singular Value Decomposition* (SVD) [66], qui décompose une matrice en trois composantes, révélant des caractéristiques essentielles de la matrice originale, telles que son rang et ses directions principales de variation. Pour réduire la dimensionnalité, SVD utilise les premiers composants de cette décomposition, qui correspondent aux éléments les plus significatifs.

Une autre manière de réduire les dimensions est d'utiliser des techniques d'apprentissage profond, comme les autoencodeurs [67], qui sont entraînés pour encoder les données en un espace latent plus restreint, représentant les informations les plus importantes. Les autoencodeurs fonctionnent en deux étapes principales : l'encodage et le décodage. L'encodeur compresse les données d'entrée en une représentation de dimension inférieure, tandis que le décodeur tente de reconstruire les données originales à partir de cette représentation compressée. En minimisant l'erreur de reconstruction, l'autoencodeur apprend à capturer les caractéristiques les plus importantes des données dans l'espace latent. Cette approche permet une réduction de dimension non linéaire, capable de capturer des relations complexes dans les données que les méthodes linéaires traditionnelles pourraient manquer.

2.2.2 Réduction de dimension des données séquentielles

Lors de l'analyse des séries temporelles, la dimensionnalité des données devient rapidement un défi majeur, non seulement en raison du nombre de caractéristiques mais aussi de la longueur des séquences temporelles. Ces dimensions accrues exacerbent les problèmes inhérents aux grandes dimensions, et les modèles conventionnels peinent souvent à capturer et à maintenir les dépendances temporelles cruciales qui caractérisent les données séquentielles. Cette incapacité à préserver les relations temporelles essentielles peut entraîner une perte significative d'information et de performance dans les tâches prédictives ou de classification. Pour contourner ce problème, comme pour les données qui ne contiennent pas d'aspect de temporalité, l'approche la plus évidente est de réduire les dimensions des données. L'enjeu des approches de réduction de dimension des données séquentielles est de simplifier l'espace de représentation des données tout en conservant les propriétés temporelles intrinsèques.

Bien qu'il soit possible d'utiliser des approches de réduction de dimension de données statiques, comme les autoencodeurs ou SVD sur des données séquentielles, il existe des méthodes couramment utilisées pour réduire les dimensions des données séquentielles tout en prenant en compte leur temporalité [68]. Les principales approches sont les suivantes :

- *Kernel PCA* (KPCA) [69] est une extension de PCA (*Principal Component Analysis*, que nous appelons ACP dans ce manuscrit) pour les données séquentielles. KPCA applique une fonction de noyau pour projeter les données dans un espace de dimension supérieure où les relations linéaires sont possibles, puis applique une ACP dans cet

espace transformé.

- *Laplacian Eigenmap* (LE) [70] construit un graphe reflétant la proximité entre les observations dans l'espace d'origine. Cette méthode génère ensuite une représentation en dimension inférieure qui préserve ces relations de voisinage, en minimisant une fonction de coût basée sur la matrice laplacienne du graphe.
- *Locality Preserving Projections* (LPP) [71] détermine une projection linéaire qui préserve les relations de voisinage entre les observations. Cette méthode s'apparente aux *Laplacian Eigenmaps*, mais opère dans un cadre plus global et linéaire.
- *Locally Linear Embedding* (LLE) [72] suppose que chaque point peut être reconstruit linéairement à partir de ses voisins les plus proches. LLE cherche à conserver ces reconstructions locales dans la représentation de dimension inférieure.
- *Isometric Mapping* (Isomap) [73] estime les distances géodésiques (distances sur la variété) entre les points en utilisant un graphe de voisinage et utilise ces distances pour effectuer une réduction de dimension qui préserve la structure globale.
- *Maximum Variance Unfolding* (MVU) [74] cherche à "déplier" les données en une représentation de dimension inférieure tout en maximisant les distances entre tous les points, ce qui aide à révéler la structure sous-jacente cachée dans les données.
- *Diffusion Maps* (DM) [75] utilise des probabilités de transition (calculées à partir des distances entre points) pour cartographier les données dans un espace où la distance entre les points reflète la probabilité de transition entre eux, capturant les structures de connectivité complexes.
- *Discrete Fourier Transform* (DFT) [76] transforme des séquences de données numériques en composantes fréquentielles pour identifier les fréquences dominantes et réduire les dimensions en ignorant les fréquences avec de faibles amplitudes
- *Discrete Wavelet Transform* (DWT) [77] décompose ces mêmes séquences en différentes échelles de temps et fréquences en utilisant des ondelettes, ce qui permet une analyse multi-résolution des données et une réduction efficace de la dimensionnalité en se concentrant sur les caractéristiques importantes.

2.2.3 Limites des approches de réduction de dimension

Les approches de réduction de dimension, qu'elles soient appliquées à des données statiques ou séquentielles, présentent plusieurs limites. Les méthodes de sélection de caractéristiques peuvent entraîner une perte d'informations importantes en éliminant des variables qui, bien que non significatives individuellement, pourraient contribuer de manière synergique à la compréhension des données. Les techniques d'extraction de caractéristiques génèrent de nouvelles variables en combinant les données d'entrée initiales de manière linéaire, c'est-à-dire en additionnant ces variables multipliées par des coefficients. Ce processus peut compliquer l'interprétation des résultats car ces nouvelles variables peuvent ne pas avoir de signification intuitive ou directe. De plus, ces méthodes, en se concentrant sur des combinaisons linéaires, peuvent parfois ne pas capturer adéquatement les relations complexes ou non linéaires présentes dans les données, ce qui limite leur efficacité pour certains types de problèmes.

Les autoencodeurs [67], en dépit de leur capacité à apprendre des représentations compactes, sont sujets au surapprentissage, surtout si le réseau est trop complexe par rapport à la taille du jeu de données. Ils nécessitent souvent un grand nombre de données pour l'entraînement et peuvent être sensibles aux paramètres de leur architecture, rendant leur mise en œuvre et leur optimisation potentiellement difficiles et coûteuses en termes de ressources computationnelles.

Les approches de réduction de dimension appliquées aux données séquentielles présentent des limites similaires. Premièrement, des méthodes comme *Singular Value Decomposition*, *Discrete Wavelet Transform*, *Diffusion Maps* et MVU ont une complexité computationnelle élevée, ce qui restreint leur utilisation pour le traitement de grandes bases de données dans des environnements avec des ressources de calcul limitées [68]. De plus, la sensibilité au bruit et aux données aberrantes est limitante pour les techniques telles que *Locally Linear Embedding* et *Locality Preserving Projections*. Ces méthodes peuvent produire des représentations déformées des données, réduisant ainsi la fiabilité des analyses. Par ailleurs, des problèmes tels que la création de graphes de voisinage déconnectés par *Laplacian Eigenmap* et l'instabilité topologique de Isomap soulignent des risques d'incohérence dans la préservation des structures de données sous-jacentes. Concernant la méthode KPCA, elle requiert un temps d'entraînement élevé, ce qui peut poser problème dans des contextes où la rapidité et l'efficacité sont nécessaires. Enfin, l'approche *Discrete Fourier Transform* peut entraîner la perte d'informations temporelles essentielles, limitant son utilité pour l'analyse des séries temporelles. Cette perte survient notamment lorsque le signal est enregistré trop lentement, coupé au milieu d'un motif, observé sur une durée insuffisante, soumis aux arrondis numériques, ou quand certaines fréquences sont ignorées lors de la reconstruction.

Face à ces limitations, il est devenu courant d'entraîner des modèles profonds, notamment car ils sont perçus comme moins sensibles au fléau de la dimension comparés aux modèles classiques ou peu profonds [78]. Toutefois, la tendance à augmenter la taille des architectures, que ce soit en termes de nombre de paramètres ou de couches, pour traiter des données à haute dimension sans réduction préalable, peut engendrer des coûts computationnels élevés et une propension accrue au surapprentissage [79].

Dans la partie suivante, nous introduisons une alternative à l'utilisation des approches de réduction de dimension, qui consiste à utiliser des architectures convolutives pour traiter les données tabulaires à haute dimension.

2.3 Alternative à la réduction de dimension

Dans le domaine de l'apprentissage automatique, le choix des modèles est souvent guidé par la nature des données à traiter. Par exemple, les réseaux de neurones récurrents (RNN) sont fréquemment utilisés pour les séries temporelles en raison de leur capacité à gérer des séquences de données et à capturer des dépendances temporelles (cf. Section 1.2). Plus récemment, l'utilisation des Transformers [13] a permis de pallier certains défis liés à la dimension des données, notamment grâce au mécanisme d'attention (cf. Section 1.2.2) qui peut faciliter la capture de dépendances à long terme, et à la parallélisation des calculs, qui permet de traiter plus efficacement de grandes quantités de données. Pour les images,

les réseaux de neurones convolutifs (CNN) sont privilégiés car ils exploitent la structure spatiale des images grâce à des opérations de convolution qui réduisent efficacement la dimensionnalité tout en préservant les caractéristiques essentielles (cf. Section 1.1).

Certaines architectures sont plus robustes au fléau de la dimension que d'autres. Abend affirme que les CNN sont particulièrement efficaces pour gérer la haute dimensionnalité [80] grâce à leur architecture qui réduit le nombre de paramètres indépendants par le partage de poids. De plus, les premières couches des CNN détectent des caractéristiques locales récurrentes, comme les bords ou les textures, avec efficacité, tandis que les couches plus profondes s'occupent de caractéristiques plus complexes et moins fréquentes, nécessitant moins de neurones. Cette structure permet aux CNN de nécessiter de moins de données pour l'apprentissage tout en conservant une précision élevée dans la reconnaissance de motifs, même avec des ensembles de données limités, dépassant ainsi les défis posés par la haute dimensionnalité. Madala et al. expliquent que les CNN sont robustes au fléau de la dimension principalement grâce à leur approche de traitement par patches [81]. Au lieu d'analyser l'image entière en une fois, les CNN traitent de petites portions (patches) de l'image successivement. Cette méthode réduit considérablement la dimensionnalité des données traitées à chaque étape du réseau. Cette approche permet aux CNN de gérer efficacement la haute dimensionnalité des images, tout en préservant les informations pertinentes. De plus, les opérations de convolution et de *pooling*, appliquées à ces patches, permettent d'extraire progressivement des caractéristiques de plus en plus abstraites, tout en réduisant la dimensionnalité spatiale des représentations.

Motivés par la robustesse des CNN face aux données à haute dimension, nous proposons de les utiliser pour surmonter les défis liés au fléau de la dimension. Notre méthode consiste à transformer des données séquentielles en formats structurés semblables à des images (cf. Section 2.5), afin de tirer parti des capacités inhérentes aux convolutions pour détecter des motifs complexes. Nous combinons ensuite ces capacités avec celles d'un réseau de neurones récurrent pour analyser les dépendances à long terme, utilisant des vecteurs de caractéristiques de dimension réduite. Ainsi, nous proposons une approche permettant d'éviter l'augmentation du coût computationnel lié à l'accroissement du nombre de paramètres dans un réseau, ainsi que la perte d'informations souvent associée aux méthodes de réduction de dimension utilisées en prétraitement.

Dans la Section 2.3.1, nous présentons les travaux existants sur la transformation de données tabulaires en images pour l'application de CNN et positionnons l'objectif de notre approche par rapport à ces travaux dans la Section 2.3.2.

2.3.1 Génération d'images à partir de données tabulaires

L'application des réseaux de neurones convolutifs sur des données tabulaires a suscité le développement de diverses méthodes de transformation [82]. L'idée est de convertir ces données en images pour permettre leur traitement par des CNN. Nous présentons ici les principales approches de transformation de données tabulaires en images.

Han et al. ont introduit EDLT (*Enabling Deep Learning for generic classification Tasks*) [83], une approche reposant sur une double exploitation de la corrélation de Pearson : d'une

part pour cartographier les interdépendances entre variables, d'autre part pour quantifier l'influence de chaque variable sur l'étiquette à prédire. La combinaison de ces analyses produit une matrice de réorganisation, qui est ensuite transformée en représentation synthétique adaptée à l'apprentissage par CNN. Cette structuration permet de capturer efficacement les motifs pertinents pour la classification.

Sharma et al. ont développé DeepInsight [84], une méthode qui utilise des techniques de réduction de dimension comme t-SNE [85], ACP ou KPCA pour projeter les vecteurs de caractéristiques dans un espace bidimensionnel. Une fois projetées, ces caractéristiques sont organisées de manière à former une image, où chaque caractéristique correspond à un pixel spécifique. L'intensité de chaque pixel reflète la valeur de la caractéristique correspondante. Cette approche permet de convertir des données tabulaires en représentations visuelles structurées, qui servent ensuite d'entrées aux CNN, tout en préservant les relations spatiales entre les caractéristiques pour une classification plus efficace.

Bazgir et al. ont développé REFINED (*REpresentation of Features as Images with NEighborhood Dependencies*) [86], une approche qui utilise le dimensionnement multidimensionnel bayésien (BMDS) pour projeter des vecteurs de caractéristiques sur un espace 2D, tout en minimisant la distorsion globale pour préserver la distribution originale des caractéristiques. Ce processus commence par calculer une matrice de distance euclidienne entre caractéristiques, utilisée par l'algorithme de MDS (*MultiDimensional Scaling*) [87] pour créer un espace 2D initial qui forme une carte de caractéristiques. Le BMDS est ensuite appliqué pour estimer les positions des caractéristiques dans cet espace, en veillant à ce que chaque pixel de l'image finale ne contienne qu'une seule caractéristique. L'optimisation de cette configuration est réalisée à l'aide d'un algorithme de *hill climbing*, qui ajuste les positions pour minimiser les différences de distances entre les nouvelles positions des caractéristiques et les distances estimées initialement. Le résultat est un espace de coordonnées distinct pour chaque caractéristique, qui sert de base pour créer une image. Ces images sont ensuite utilisées pour entraîner un CNN sur différentes tâches telles que la prédiction de la sensibilité aux médicaments anticancéreux, la classification des types de tumeurs, et la régression pour estimer l'efficacité des médicaments sur diverses lignées cellulaires cancéreuses.

Zhu et al. ont proposé IGTD (*Image Generator for Tabular Data*) [88], une méthode qui attribue un pixel en niveaux de gris pour chaque caractéristique dans une image, où l'intensité de la couleur du pixel reflète la valeur de la caractéristique. IGTD utilise une métrique de distance pour représenter les relations entre les caractéristiques des données tabulaires en relations entre pixels dans l'image. Il calcule les distances euclidiennes entre chaque caractéristique et tente d'assigner des pixels avec des distances similaires dans l'image. Ce processus est effectué en utilisant une matrice de classement des distances des caractéristiques et une matrice de classement des distances des pixels, en ordre croissant. IGTD optimise l'attribution des caractéristiques en minimisant les différences entre ces deux matrices. Comparé aux méthodes comme DeepInsight et REFINED, IGTD génère des représentations d'images plus compactes, préserve mieux la structure de voisinage des caractéristiques, et permet une transformation plus rapide des données tabulaires en images, tout en offrant une meilleure performance prédictive lorsqu'utilisé avec des réseaux de neurones convolutifs.

Zandavi et al. ont introduit *Fotomics* [89], une méthode pour convertir les données

omiques de cellules uniques en images, utilisant la transformation de Fourier pour faciliter la classification des types cellulaires via un CNN. Cette approche commence par le prétraitement des données omiques pour la normalisation, suivie par l'application de la transformation de Fourier rapide (FFT) qui transforme ces données en vecteurs de fréquences. Ces vecteurs sont ensuite cartographiés dans un espace bidimensionnel, où chaque molécule est positionnée selon les composantes réelles et imaginaires issues de la FFT. Une optimisation de l'espace est effectuée pour maximiser l'utilisation de la zone d'image, suivie d'une rotation pour aligner l'image de façon optimale pour l'application du CNN. Enfin, ces coordonnées sont converties en une grille de pixels, créant une image où l'intensité de chaque pixel reflète les niveaux d'expression des gènes correspondants. Ces images sont ensuite utilisées pour entraîner un CNN, permettant une classification efficace des types de cellules basée sur les motifs visuels détectés dans les images.

Enfin, Damri et al. ont introduit FC-Viz [82], une méthode qui transforme les données tabulaires en images en plusieurs étapes intégrées. Après avoir regroupé les caractéristiques similaires en clusters en fonction de leur corrélation, FC-Viz organise ces clusters sur une grille 2D, permettant aux caractéristiques qui partagent des interactions ou des corrélations fortes d'être placées à proximité les unes des autres. Des techniques de réduction de dimension sont utilisées au sein de chaque cluster pour optimiser l'espace. Ensuite, les caractéristiques des différents clusters sont transformées en pixels dans l'image finale, où l'intensité de chaque pixel reflète la valeur de la caractéristique qu'il représente. Cette transformation en pixels maintient l'intégrité des relations entre les caractéristiques, permettant ainsi au CNN de mieux exploiter ces informations pour la classification.

2.3.2 Contexte de notre étude

Notre étude se concentre sur une approche qui utilise une transformation minimale des données pour les adapter aux entrées des réseaux de neurones convolutifs, sans changer fondamentalement leur nature. Plutôt que de recourir à des méthodes complexes de transposition de données séquentielles en structures 2D, notre objectif est de normaliser simplement les données séquentielles pour qu'elles puissent être traitées comme des images où l'intensité de chaque pixel reflète la valeur de la caractéristique qu'il représente (cf. Section 2.5.1). Cela donne lieu à des images abstraites, où chaque pixel correspond à une caractéristique des données originales, sans représenter des objets visuellement reconnaissables (cf. images B et C de la Figure 2.1 qui encodent des informations médicales textuelles dans des pixels).

L'objectif de nos travaux est d'évaluer les avantages de l'utilisation des réseaux de neurones convolutifs pour l'extraction de caractéristiques à partir de données séquentielles initialement tabulaires. En convertissant ces données en 2 dimensions et en employant un CNN pour extraire les caractéristiques, suivi d'un réseau de neurones récurrent (RNN) pour prendre en compte la temporalité, nous visons à déterminer si les CNN, typiquement utilisés pour l'analyse d'images, peuvent atteindre ou dépasser les performances des RNN ou des Transformers sur des données séquentielles transformées de manière minimale.

Dans le contexte du traitement de données à haute dimension destinées à des tâches de classification, il est courant d'opter pour des méthodes de réduction de dimension (cf.

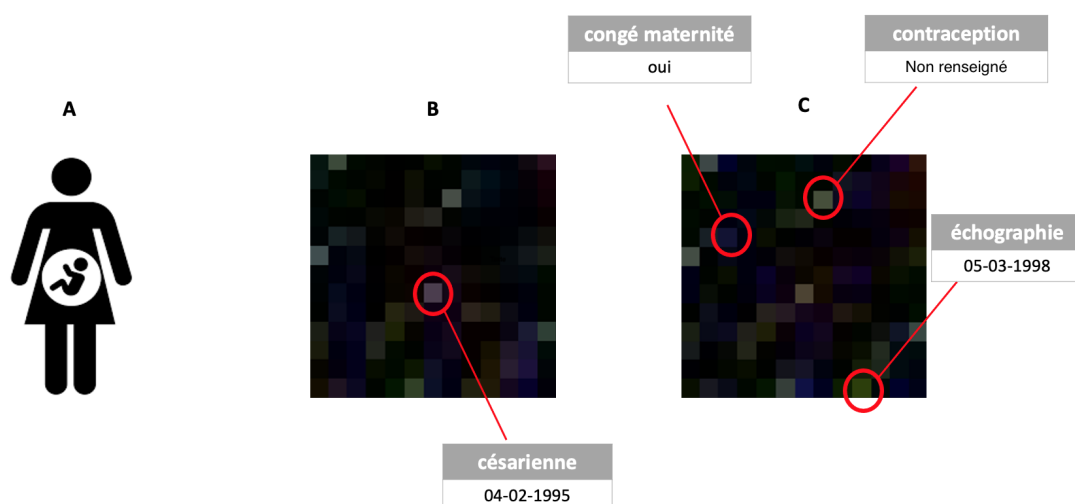


FIGURE 2.1 – Trois images qui correspondent à la classe "a été enceinte". L'image A est figurative et représente une femme enceinte par le dessin de sa silhouette. Les images B et C sont abstraites et représentent une grossesse par des pixels correspondant à des événements liés au suivi de la grossesse.

Section 2.2). Bien que ces techniques permettent d'éliminer les variables superflues, réduire la redondance et atténuer le bruit, elles peuvent également entraîner une certaine perte d'informations. Face à cette problématique, notre approche consiste à transposer des données tabulaires à haute dimension en images afin de les traiter par des réseaux de neurones convolutifs, donnant lieu à des images dites abstraites, où chaque pixel encode une information spécifique. Avant de poursuivre, nous souhaitons vérifier la viabilité de l'utilisation des CNN pour la tâche de classification d'images en présence de bruit, où les classes ne sont pas aisément identifiables, qu'elles soient représentées de manière figurative ou abstraite (cf. Figure 2.1). À cet égard, nous avons mené deux études empiriques : la première explore les limites d'un CNN pour classifier des objets "figuratifs" (visuellement identifiables) malgré un fort niveau de bruit ; la seconde évalue l'efficacité d'un CNN pour la classification de classes abstraites, non identifiables visuellement, également perturbées par du bruit. Ces études préliminaires sont essentielles pour confirmer notre hypothèse selon laquelle les CNN peuvent fournir de bonnes performances pour la tâche de classification d'images où l'information à identifier n'est pas visuellement reconnaissable, que ce soit parce qu'elle est représentée de façon abstraite ou parce qu'elle est fortement bruitée.

2.4 Études des limites des CNN sur des images bruitées

Depuis plusieurs années, les modèles du domaine de la vision par ordinateur montrent de très bonnes performances sur des images représentant des objets identifiables visuellement. De multiples architectures basées sur les CNN (cf. Section 1.1) ont vu le jour, comparant leurs performances notamment sur des jeux de données communs comme *ImageNet* [14] ou COCO [15], qui représentent des classes identifiables à l'oeil nu comme des objets du

quotidien ou des animaux. Nous définissons ces images comme étant figuratives.

Traditionnellement, les modèles de classification sont entraînés sur des images où l'objet à identifier occupe une grande partie de l'image. Nous nous questionnons quant aux limites des CNN lorsqu'ils sont appliqués sur des images dont l'objet à identifier est peu représenté, voire segmenté et dispersé à travers l'image. De plus, comme discuté dans la partie 2.3.2, nous aimerions connaître les limites de l'application des CNN sur des images abstraites, où chaque pixel encode une information spécifique.

Pour répondre à ces questions, nous menons deux études empiriques pour explorer les limites des CNN dans des contextes de bruit. La première étude, détaillée dans la Section 2.4.1, examine le comportement de l'apprentissage d'un CNN sur la tâche de classification d'objets figuratifs en présence de bruit. Dans ce contexte, nous montrons que l'objet doit représenter à minima 0,45% de la surface totale pour que le modèle arrive à apprendre. Nous observons également que l'objet peut être fragmenté sans impacter négativement la performance du modèle. La seconde étude, détaillée dans la Section 2.4.2, examine les performances d'un CNN sur des images abstraites, telles que des QR codes générés à partir de chaînes de caractères contenant une constante à prédire. Nous ajoutons divers types de bruit et comparons les performances du CNN à celles d'un décodeur déterministe conçu pour lire les QR codes. Les résultats montrent que le CNN prédit la constante avec une efficacité au moins équivalente à celle du décodeur. Ces études confirment que les CNN sont efficaces sur des images bruitées, que les données à identifier soient de nature figurative ou abstraite.

2.4.1 Étude de l'information résiduelle nécessaire pour la classification d'objets figuratifs

Nous cherchons à déterminer les caractéristiques intrinsèques nécessaires pour qu'un objet figuratif soit correctement identifié par un modèle de classification. Plus précisément, nous explorons les capacités d'un CNN à classer des images où l'objet cible n'occupe qu'une partie de l'image. Nous souhaitons également comprendre si un CNN peut intégrer efficacement des informations dispersées à travers l'image pour identifier un objet segmenté. Pour cela, nous entraînons un modèle sur une tâche simple, la classification de logos (*Twitter* ou *Facebook*, choisis arbitrairement), en manipulant divers paramètres tels que la position, le niveau de bruit, et la segmentation de l'objet dans l'image.

La méthode proposée pour la construction du jeu de données, l'ajout de bruit et l'entraînement du modèle de classification sont détaillés dans la Section 2.4.1.1. Les résultats de nos expérimentations sont présentés dans la Section 2.4.1.2 et la conclusion ainsi que les perspectives sont discutées dans la Section 2.4.1.3.

2.4.1.1 Méthode

Nous souhaitons générer des images où l'information à identifier n'occupe qu'une partie de l'espace de l'image. Pour ce faire, nous créons des QR codes (*Quick-Response Codes*) [90], qui encodent une chaîne de caractères en bits représentés par une grille de carrés, souvent

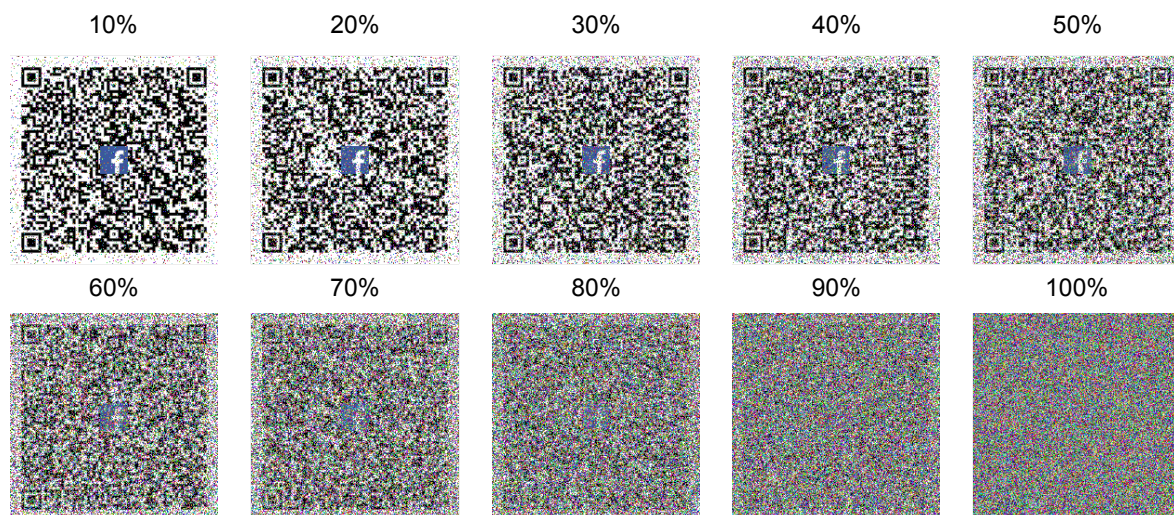


FIGURE 2.2 – Représentation d’une image à classer dont une surface, un pourcentage de pixels variant de 10% à 100%, est bruitée par un changement de couleur tirée aléatoirement.

noirs ou blancs (cf. guide de génération et de décodage des QR codes dans l’annexe B). Nous considérons ces QR codes comme du bruit de fond, sur lesquels nous superposons des formes simples à identifier, les logos de *Facebook* ou *Twitter* (voir Figure 2.3) et ajoutons du bruit. Ces logos, positionnés sur le QR code, constituent l’élément à identifier par notre modèle de classification, nous détaillons les différentes configurations du logo testées dans cette partie.

Générations des images et ajout du bruit : Afin d’assurer la bonne transmission de l’information, les QR codes sont générés avec un système de correction d’erreur appelé système *Reed-Solomon* [91]. Ce système permet de corriger les erreurs dues à la transmission imparfaite d’un message en répliquant l’information un certain nombre de fois à une certaine distance. Il existe 4 niveaux de correction d’erreur associés à un pourcentage théorique de la surface pouvant être bruitée sans influencer les performances du décodeur. Ces niveaux sont : L (7%), M (15%), Q (25%) et H (30%). Dans cette étude, nous générons des QR codes avec une correction d’erreur de niveau H, c’est-à-dire environ 30% de redondance. Le logo n’est pas pris en compte car il est ajouté à l’image après la génération du QR code.

Nous générons des QR codes à partir de chaînes de 150 caractères choisis aléatoirement et plaçons le logo *Twitter* ou *Facebook* par dessus. Les QR codes ont une taille de 224×224 pixels et les logos ont une taille de 30×30 pixels.

Nous bruitons les images en tirant aléatoirement les pixels qui seront modifiés, avec un nombre total correspondant au pourcentage de la surface de l’image à bruite. Le pourcentage de pixels modifiés varie de 10% à 100% de la surface totale de l’image, avec un pas de 10 (cf. Figure 2.2).

Le tirage aléatoire des pixels à bruite suit une distribution uniforme. Chaque pixel correspond à un tuple de position (p_1, p_2) avec $p_1, p_2 \in [0, 223]$. Une couleur choisie aléatoirement est attribuée à chaque pixel. Une couleur est définie par un triplet (c_1, c_2, c_3) avec $c_1, c_2, c_3 \in [0, 255]$ et le nombre de couleurs possibles est de 256^3 avec une probabilité



FIGURE 2.3 – Exemples de QR codes avec le logo de *Facebook* ou de *Twitter* superposé.



FIGURE 2.4 – Deux images contenant le logo *Facebook* segmenté. Les segments sont séparés par une distance de *Manhattan*, 50 pixels pour l'image de gauche et 190 pixels pour celle de droite.

de $\frac{1}{256^3}$.

Position et représentation du logo dans l'image : nous souhaitons définir les limites d'un CNN sur la tâche de classification sur des images où l'objet à identifier n'a pas une position fixe. Pour cela, nous testons différentes configurations :

- Logo dont la position est fixée au centre de l'image.
- Logo dont la position est aléatoire dans l'image.

De plus, nous souhaitons étudier l'impact de la segmentation de l'information sur les performances du modèle. Pour cela, nous fragmentons les logos en deux et positionnons les fragments de manière aléatoire ou non dans l'image de manière à respecter une distance, définie au préalable (voir Figure 2.4). Nous utilisons la distance de *Manhattan* [92], et la faisons varier de 50 à 190 pixels, en garantissant que les deux fragments soient dans l'image. Dans un repère 2D, les barycentres des fragments sont représentés par leurs coordonnées (x_1, y_1) et (x_2, y_2) . La distance entre ces barycentres est donnée par :

$$d_{Manhattan} = |x_1 - x_2| + |y_1 - y_2| \quad (2.1)$$

Pour chaque configuration, nous appliquons le processus d'ajout de bruit décrit précédemment et nous identifions le pourcentage de bruit maximum tel que le modèle arrive à identifier le logo. Autrement-dit, le pourcentage de bruit maximum qu'il est possible d'ajouter aux images, avant que le modèle ne fasse de sous-apprentissage et n'arrive plus à extraire les caractéristiques liées à l'objet.

Identification du logo : pour tester les limites d'un modèle de classification basé sur une architecture convolutive simple, nous entraînons le modèle LeNet5 [18] avec la fonction de perte de l'entropie croisée et l'optimiseur *Adam* [93] avec un taux d'apprentissage de 10^{-4} . Les entraînements se font sur 100 *epochs* avec un *batch* de taille 5. La taille du *batch* est déterminée par les ressources computationnelles à notre disposition.

Pour éviter le surapprentissage, nous faisons de l'augmentation de données [6]. Plus précisément, nous effectuons 3 rotations de l'image (90° , 180° , 270°). Les jeux de données

Position	Logo entier		Logo fragmenté	
	fixe	aléatoire	fixe	aléatoire
Bruit maximum	90%	70%	80%	60%
Exactitude	96%	97%	100%	100%

TABLE 2.1 – Résultats du pourcentage maximal de bruit tel que le modèle arrive à apprendre, pour des logos entiers ou fragmentés dont la position est fixe ou aléatoire.

d'entraînement contiennent 1 200 images par classe (2 400 au total), dont 20% sont utilisés pour la validation. Les jeux de données de test, auxquels aucune augmentation de données n'a été appliquée, sont représentés par 300 images, soit 150 par classe.

2.4.1.2 Étude expérimentale

Nous évaluons les performances du modèle avec la mesure de l'exactitude (*accuracy*). L'exactitude est le ratio entre le nombre d'observations bien classées par le modèle et le nombre total d'observations à classer.

La Table 2.1 présente, pour chaque configuration du logo, l'exactitude du modèle de classification associée au niveau maximal de bruit applicable aux images. Ce niveau maximal correspond au seuil au-delà duquel l'information nécessaire à la classification devient indiscernable pour le modèle, c'est-à-dire lorsque ses performances en termes d'exactitude deviennent équivalentes à celles d'un choix aléatoire (50%). Les résultats montrent que la position du logo a un impact sur les performances du modèle. En effet, lorsque le logo est entier et fixe, le modèle tolère un niveau de bruit de 90% contre 70% avec position aléatoire, avec une exactitude presque égale (96% et 97%). Lorsque le logo est fragmenté avec une position fixe, le modèle tolère plus de bruit (80% contre 60% en position aléatoire) tout en maintenant une exactitude parfaite (100%). Concernant l'état du logo, qu'il soit entier ou fragmenté, nous constatons un impact modéré sur les performances du modèle. En particulier, le modèle montre une tolérance réduite de 10% au bruit pour le logo fragmenté comparé au logo entier, indépendamment de sa position. Cependant, il est difficile de tirer des conclusions définitives sur l'impact réel de la fragmentation du logo. En effet, la méthode d'ajout de bruit par paliers de 10% limite notre capacité à évaluer précisément ce seuil, car il est possible que le modèle reste performant à un niveau de bruit légèrement inférieur à celui testé. De plus, nous avons observé pendant les expérimentations que la distance entre les fragments n'avait aucun impact sur les résultats obtenus.

2.4.1.3 Extensions et discussions

L'étude expérimentale réalisée a pour objectif de déterminer les limites de la capacité d'un réseau de neurones convolutif à classer des images où l'objet à identifier occupe une petite portion de l'image totale. Notre approche consiste à ajouter du bruit aux images pour évaluer à partir de quel seuil de bruit le modèle ne parvient plus à identifier correctement

les caractéristiques de l'objet. Sachant que le logo a une taille de (30×30) dans une image de (224×224) , il représente 1,8% de l'image. En considérant que le modèle fonctionne avec en moyenne 75% de bruit, nous pouvons conclure que dans notre contexte expérimentale, l'information résiduelle nécessaire dans l'image est de $\frac{1.8}{100} \times (1 - \frac{75}{100}) = 0.0045 = 0,45\%$. De plus, nos expérimentations montrent que le modèle est aidé lorsque la position de l'objet est fixe, qu'il soit segmenté ou non.

Finalement, l'objectif de cette étude est de déterminer si, lors de la transformation de données à haute dimension en images, un CNN est capable d'extraire des informations pertinentes à la classification, même si ces dernières sont dispersées et représentent une petite partie de l'image. Pour tester cela, nous générons des images aléatoires sous forme de QR codes, auxquelles nous superposons un logo de *Twitter* ou *Facebook* dans différentes configurations. Nous observons les capacités du CNN à identifier ces logos en dépit du bruit présent. Cette approche nous permet de délimiter les capacités d'un CNN pour la classification d'objets figuratifs. Cependant, transformer des données tabulaires en images résulte en des images où chaque pixel "encode" une caractéristique spécifique. C'est pourquoi, dans la partie suivante, nous présentons une étude qui explore les limites d'un CNN appliqué sur des images abstraites, telles que des QR codes, où chaque pixel encode une information distincte.

2.4.2 Étude de l'impact du bruit sur la classification d'images abstraites

Les QR codes et leurs dérivés sont largement utilisés à des fins commerciales au quotidien, mais aussi pour récupérer des informations sécurisées, comme ce fut le cas avec le pass sanitaire utilisé dans le cadre de la lutte contre la COVID-19. Les informations encodées dans ce type de représentation ne sont pas identifiables à l'oeil nu. Nous définissons ces images comme étant abstraites. Dans ce travail, nous souhaitons définir les limites d'un modèle basé sur des couches convolutives pour la classification de QR codes.

Il existe des travaux combinant opérateurs classiques de traitement d'images et modèle d'apprentissage basés sur des réseaux de neurones pour prendre en compte les distorsions possibles sur les QR codes. En effet, la reconnaissance des QR codes est souvent confrontée à des problèmes de fluctuations irrégulières de l'arrière-plan, d'éclairages inadéquats et de distorsions en fonction du contexte d'acquisition de l'image. Pour faire face à ces problèmes, des travaux ont vu le jour, que ce soit en adoptant un algorithme de filtre médian adaptatif amélioré et une méthode de correction de distorsion de QR codes basée sur des réseaux de neurones [94], ou en proposant un algorithme qui localise et segmente les QR codes à l'aide d'un CNN [95] notamment pour résoudre d'éventuelles rotations et déformations liées à des scènes complexes [96]. Nous souhaitons voir le problème d'un autre point de vue et explorer la possibilité de décoder un QR code, ou une partie, par le biais d'un modèle d'apprentissage.

L'objectif de cette étude est de faire une analyse empirique permettant de définir les limites d'un modèle basé sur l'apprentissage profond pour la classification d'images dites abstraites. En particulier nous posons la question de savoir si un CNN, avec l'introduction d'un a priori sur l'information encodée via les exemples d'apprentissage, peut être plus

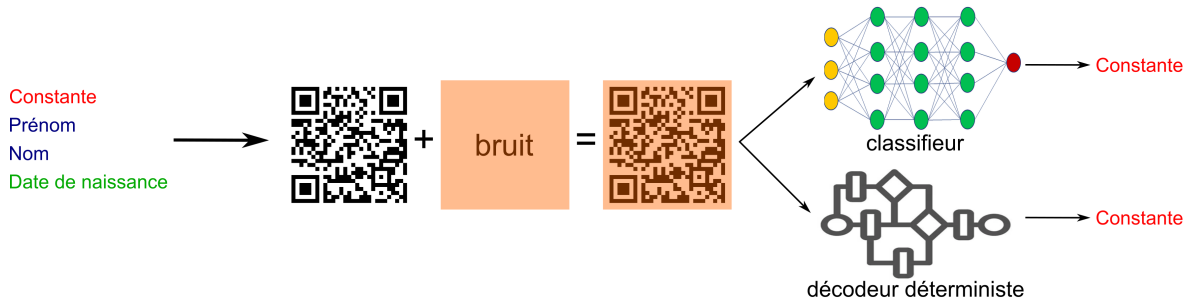


FIGURE 2.5 – Schéma des expérimentations. 1) Génération des QR codes contenant les informations obtenues lors du pass sanitaire européen. 2) Ajout de bruit. 3) Entraînement d'un classifieur sur les images bruitées. 4) Comparaison des performances du classifieur et celles d'un décodeur classique de QR codes pour la tâche de détection de la constante de validité du pass.

robuste qu'un modèle déterministe en présence de bruit. Pour cela, nous générons des données de synthèse, des QR codes contenant les informations obtenues lors du scan d'un pass sanitaire européen soit le prénom, le nom, la date de naissance et une constante relative à la validité du pass (valide ou invalide par exemple). Dans ce travail, nous cherchons à prédire cette constante. Nous traitons donc un problème de classification binaire. Enfin, nous ajoutons différents types de bruits aux QR codes pour comparer la robustesse d'un CNN à celle d'un décodeur de QR code classique déterministe. Les différentes étapes de l'expérimentation sont schématisées dans la Figure 2.5. La méthode de génération des images, l'ajout des différents bruits aux images, la distribution des jeux de données d'entraînement, de validation et de test ainsi que l'explication des méthodes de classification utilisées sont détaillés dans la Section 2.4.2.1. La Section 2.4.2.2 présente les résultats des expérimentations. Enfin, des extensions, pour comprendre les limites de nos expérimentations, et leurs résultats sont discutés dans la Section 2.4.2.3.

2.4.2.1 Méthode

Génération des QR codes : Nous générons des QR codes avec les différents niveaux de système de correction d'erreur (L,M,Q et H) à partir de chaînes de caractères de la forme :

`constante/Prénom/Nom/date_de_naissance`, avec :

- `constante` $\in \{ "approuv"; "invalide" \}$
- `Prénom` et `Nom` sont tirés aléatoirement avec la bibliothèque *names*¹
- `date_de_naissance` $\in [01/01/1921; 01/01/2021]$

Les QR codes sont générés avec la bibliothèque *qrcode*² en prenant en paramètres d'entrée la chaîne de caractères et le niveau de correction d'erreur. Finalement, afin de pouvoir comparer les performances du CNN avec celles du décodeur déterministe, la séquence "approuv" a été choisie comme constante à prédire/décoder, car les QR codes utilisent l'encodage *American Standard Code for Information Interchange* (ASCII)³, qui ne prend pas

1. <https://pypi.org/project/names/> 2. <https://pypi.org/project/qrcode/> 3. L'ASCII est un système d'encodage des caractères développé dans les années 1960. Il s'agit d'un code standardisé qui attribue un nombre entier unique à chaque caractère, permettant ainsi aux ordinateurs de stocker, traiter et communiquer du texte.



FIGURE 2.6 – QR codes dont un pourcentage de la surface totale est bruitée en inversant la valeur des pixels (un pixel noir devient blanc et inversement).

en charge les caractères accentués. Les résultats de l'expérience avec la constante "approuvé" sont discutés dans la Section 2.4.2.3.

Modèles de bruit : nous cherchons à déterminer le seuil de bruit à partir duquel le modèle de classification n'est plus capable d'identifier correctement les octets correspondant à chaque classe. Pour quantifier ce seuil, nous bruitons un pourcentage défini de la surface des QR codes en inversant la valeur des pixels (blanc→noir, noir→blanc) (cf. Figure 2.6). Pour faciliter l'analyse des résultats, comme le décodeur déterministe lit un QR code bit par bit, nous générons les QR codes tel que 1 bit corresponde à 1 pixel.

L'utilisation d'un décodeur classique étant courante au quotidien, nous souhaitons évaluer les performances sur des QR codes bruités de façon plus réaliste que par inversement de pixels. Inspirés par des travaux sur la dégradation de document [97], nous utilisons *ocrodeg*⁴, une bibliothèque développée par *NVIDIA* permettant de simuler différents bruits sur des documents numérisés [98]. Plus précisément, nous utilisons trois types de bruits :

- La distorsion aléatoire (Figure 2.7a), qui simule l'étalement de l'encre en fonction d'un paramètre *sigma*. Plus *sigma* est grand, plus les distorsions sont prononcées.
- La distorsion par enroulement (Figure 2.7b), qui simule le gondolage du papier en fonction d'un paramètre *mag*. Ce paramètre contrôle la magnitude de l'enroulement appliquée à l'image. Plus *mag* est grand, plus les distorsions sont importantes.
- La sélection de l'avant/arrière-plan (Figure 2.7c), qui simule des défauts de luminosité.

Lors de l'utilisation du décodeur classique sur les images bruitées avec l'approche de la sélection de l'avant/arrière-plan, le décodage est précédé par une étape de binarisation dont le seuil est déterminé par l'algorithme d'*Otsu*⁵ [99]. En effet, comme cette approche modifie l'intensité lumineuse des pixels en nuances de gris, le décodeur ne peut pas les décoder sans étape de binarisation préalable.

Jeu de données : nous entraînons un modèle de classification sur des QR codes auxquels sont ajoutés les différents modèles de bruit présentés précédemment. Le jeu de données

4. <https://github.com/NVlabs/ocrodeg> 5. L'algorithme d'*Otsu* est une technique de seuillage automatique utilisée en traitement d'images pour la binarisation. Il fonctionne en calculant l'histogramme des niveaux de gris de l'image, puis en recherchant le seuil optimal qui maximise la variance inter-classe entre deux groupes de pixels. Ce processus implique d'itérer sur tous les seuils possibles, en divisant l'histogramme en deux classes à chaque itération et en évaluant la séparation entre ces classes. Une fois le seuil optimal trouvé, l'image est binarisée, transformant tous les pixels en noir ou blanc. Cette méthode est particulièrement efficace pour séparer automatiquement les objets du fond dans une image, en s'adaptant à ses caractéristiques spécifiques.

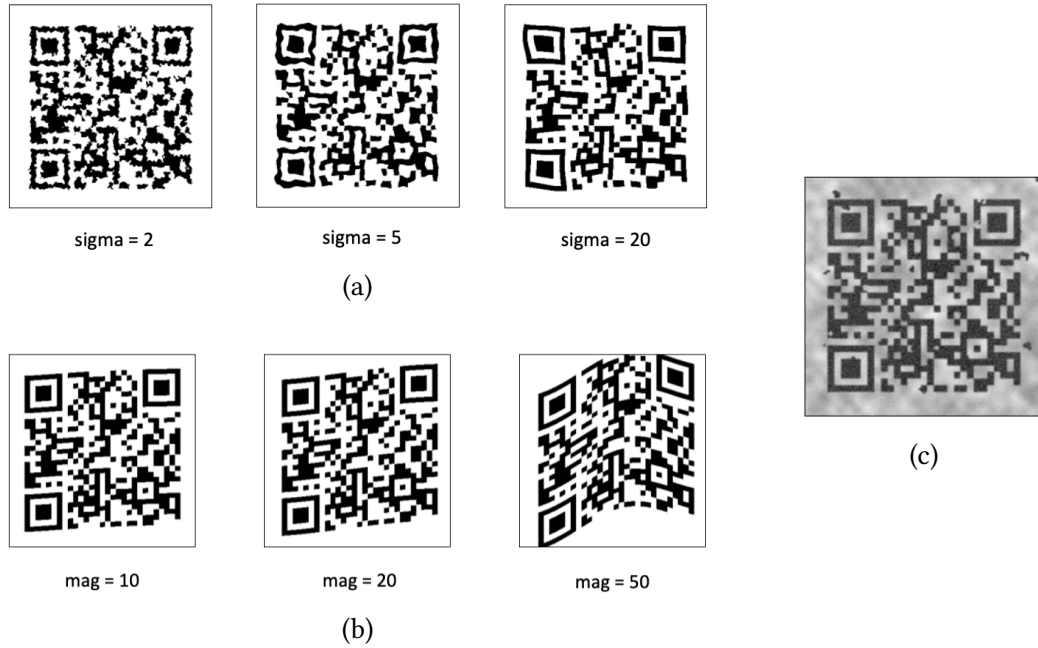


FIGURE 2.7 – Exemples de QR codes bruités utilisant différentes techniques de distorsion. (a) la distorsion aléatoire, qui simule l'étalement de l'encre en fonction du paramètre σ . (b) la distorsion par enroulement, qui simule le gondolage du papier en fonction du paramètre mag . (c) la sélection de l'avant/arrière-plan, qui simule une baisse de luminosité.

d'entraînement contient 7 920 observations par classe ("approuv", "invalid"), soit 15 840 observations au total. La distribution du jeu de données d'entraînement, dont 20% est utilisé pour la validation, est décrite dans la Figure 2.8. Pour les tests, nous générons plusieurs jeux de données de 2 000 observations chacun :

- jeu de données bruitées par inversed pixels de 0% à 100% par paliers de 1.
- jeu de données bruitées par une distorsion aléatoire avec $\sigma \in \{2; 5; 20\}$
- jeu de données bruitées par la fonction de distorsion par enroulement avec $\text{mag} \in \{10; 20; 50\}$
- jeu de données bruitées par la fonction de sélection de l'avant/arrière-plan.

Prédiction de la constante : l'objectif est de comparer la robustesse d'un modèle convolutif de classification d'images à celle d'un décodeur déterministe sur des QR codes bruités. Pour les expérimentations, nous utilisons un décodeur déterministe couramment utilisé [100, 101] de la bibliothèque *pyzbar*⁶ qui implémente les étapes classiques de décodage de QR codes (cf. guide de décodage d'un QR code dans l'annexe B).

Concernant l'architecture convolutive, nous choisissons d'entraîner le modèle ResNet18 [24] sur les QR codes bruités. Ce choix est motivé par le fait que ResNet18 est un bon compromis entre performance et taille du réseau. Le modèle est entraîné sur 100 *epochs* avec un batch de 1 024. Nous utilisons la fonction de perte de l'entropie croisée [102] et l'optimiseur *Adam* [93] paramétré par un taux d'apprentissage de 10^{-4} . Nous réduisons le

6. <https://pypi.org/project/pyzbar/>

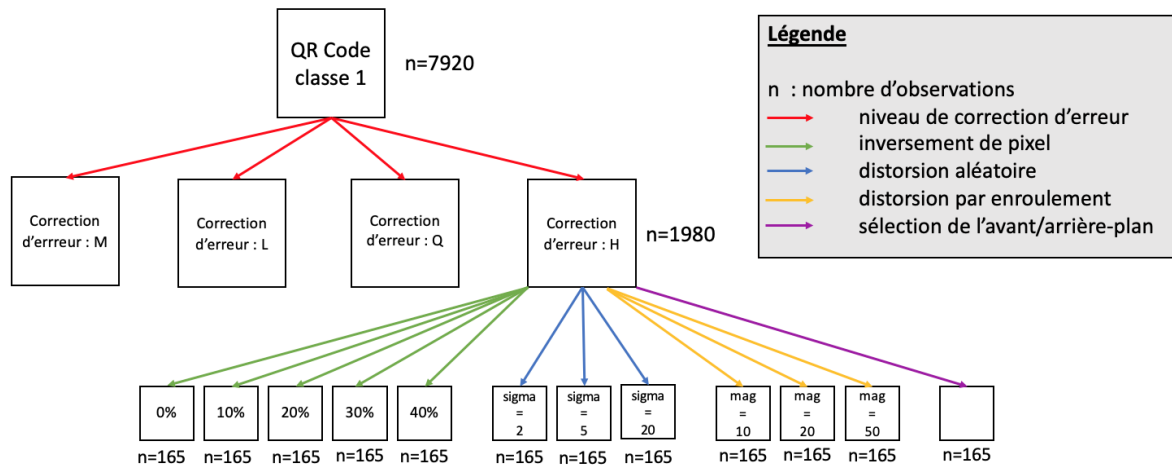


FIGURE 2.8 – Arbre représentatif de la distribution du jeu de données d'entraînement. Chaque classe est représentée par 7 920 observations dont 1 980 par correction d'erreur. Pour chaque correction d'erreur, 165 observations sont générées par type de bruit : inversement de pixels de 0% à 40% inclus, distorsion aléatoire pour $\sigma \in \{2; 5; 20\}$, distorsion par enroulement pour $mag \in \{10; 20; 50\}$ et sélection de l'avant/arrière-plan

taux d'apprentissage lorsque la perte sur le jeu de données de validation a cessé de diminuer depuis 5 *epochs*.

Nous entraînons le modèle sur des QR codes bruités de 0% à 40% par paliers de 10%. Nous faisons le choix de ne pas entraîner le modèle sur des QR codes bruités au delà de 50% car il s'agit du seuil de dégradation maximale avant la symétrie. En effet, un QR code bruité à 100% est finalement un QR code négatif sans bruit (cf. Figure 2.6).

2.4.2.2 Étude expérimentale

Nous évaluons les performances du classifieur et du décodeur déterministe avec la mesure de l'exactitude. Pour le décodeur, nous estimons qu'il détecte bien si la constante ("invalid" ou "approuv") est dans la chaîne de caractères du résultat.

Comparaison des performances des modèles sur des QR codes bruités aléatoirement :

la Figure 2.9 montre les comportements du classifieur ainsi que celles du décodeur sur des images bruitées par inversement de pixels de 0% à 100%. On remarque qu'à partir de 50% de bruit (correspondant à un QR code aléatoire), le classifieur a une *exactitude* proche de 50%, ce qui correspond effectivement à de l'aléatoire. Concernant le décodeur, ses performances sont nulles à partir de 6% de bruit. Ce résultat est dû au fait que le bruit ajouté peut se trouver dans les pilotes de positionnements. Les pilotes de positionnements sont les grands carrés aux trois coins (cf. Figure 14 de l'annexe B) du QR code qui déterminent sa position et son orientation. Pour décoder un QR code, la première étape faite par le décodeur consiste à détecter ses pilotes de positionnements. Donc si le décodeur n'arrive pas à les identifier, il ne peut pas considérer l'image comme un QR code et le décoder, ce qui n'est pas le cas

Bruit appliqué	Classifieur	Décodeur
distorsion aléatoire	99,5%	63,5%
enroulement	99,35%	54,3%
avant/arrière-plan	99,35%	97,5%

TABLE 2.2 – Comparaison des performances du classifieur et du décodeur en fonction du bruit utilisé durant la génération des QR codes. Pour la distorsion aléatoire et la distorsion par enroulement, nous présentons les performances moyennes obtenues.

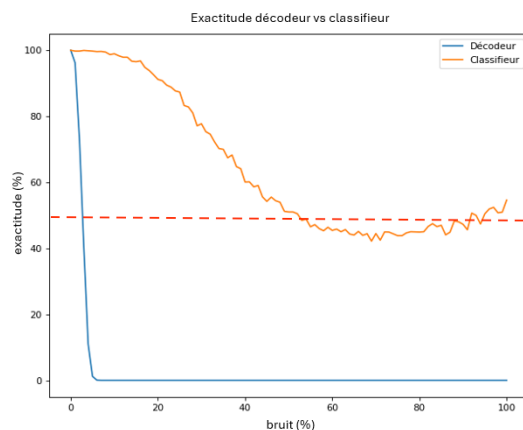


FIGURE 2.9 – Courbes d'exactitude du décodeur déterministe et du classifieur sur des QR codes bruités, par inversedement de pixel, de 0% à 100%. Une ligne rouge en pointillés située à exactitude=50% est incluse pour faciliter la visualisation.

du classifieur. Il est à noter que des expérimentations [103] ont montré que si les pilotes de positionnements ne sont pas bruités alors la surface maximale pouvant être bruitée correspond bien au niveau de correction d'erreur.

Comparaison des performances des modèles sur des QR codes en fonction du bruit réaliste appliqué : la Table 2.2 présente les performances des deux méthodes en fonction du bruit utilisé. Nous considérons ces bruits comme étant réalistes car ils correspondent à des défauts d'impression ou de capture de l'image. Nous observons que le classifieur arrive à prédire avec de bonnes performances quel que soit le type de bruit appliqué. En comparaison, le décodeur arrive à détecter presque deux tiers des QR codes bruités avec une distorsion aléatoire et seulement la moitié des QR codes bruités avec une distorsion par enroulement. Enfin, 97,5% des QR codes bruités avec la sélection de l'avant/arrière-plan sont détectés par le décodeur. Bien que pour toutes les expérimentations effectuées, le CNN détecte bien à minima tous les QR codes bien détectés par le décodeur déterministe, il est important de noter que ces résultats ne reflètent pas nécessairement ceux que pourraient avoir les décodeurs utilisés au quotidien, notamment ceux inclus dans les smartphones. Ces derniers sont généralement couplés à des techniques de pré-traitement permettant une meilleure lecture du QR code.

2.4.2.3 Extensions et discussions

Tests sur des QR codes encodant d'autres informations : afin de généraliser nos expérimentations, nous effectuons les mêmes tests que ceux décrits dans la Section 2.4.2.1 en encodant des chaînes de caractères correspondant à des URLs vers des comptes *Facebook* ou *Twitter* de la forme :

https ://**constante**/**Username** avec :

- **constante** = *Facebook* ou *Twitter*
- **Username** chaîne de $x \in [5; 20]$ caractères tirés aléatoirement.

La Figure 2.10 montre que les résultats sont comparables à ceux présentés auparavant et confirme la possibilité de généraliser notre approche.

Similarité syntaxique et positionnement : les informations obtenues lors du scan d'un pass sanitaire sont le nom, le prénom, la date de naissance et une constante validant ou non le pass. Nous avons généré des QR codes à partir de chaînes de caractères contenant ces informations et comparé les performances d'un CNN à celles d'un décodeur déterministe pour la classification de la constante. Afin d'analyser l'influence de la similarité syntaxique entre les mots, nous testons différentes constantes à prédire :

- valide / invalide
- true / false
- approuvé / invalide

De plus, la génération d'un QR code étant une suite d'étapes prédéfinies et n'incluant pas la notion d'aléatoire (le résultat de génération de QR codes à partir de deux séquences identiques est le même), si un caractère est toujours au même endroit dans la séquence, l'octet correspondant sera toujours au même endroit dans le QR code. Le CNN pourrait apprendre sur la position de l'octet à prédire et non pas sur sa représentation visuelle. Nous voulons savoir si la place de la constante à prédire a une influence sur les performances d'un modèle basé sur une architecture convolutive. Ainsi, contrairement aux expérimentations décrites précédemment, nous générons la chaîne de caractères avec le format :

Prénom/Nom/date_de_naissance/constante, où les variables *nom* et *prénom* ne sont pas de longueur fixe, ce qui assure une position variable de l'encodage de la constante à prédire.

La Figure 2.11 montre les performances de ResNet18 et celles du décodeur déterministe pour la prédiction de différentes constantes. La courbe bleu correspond aux performances du décodeur déterministe lorsqu'on lui demande de décoder "approuv". Celle orange représente le cas où le décodeur doit trouver "approuvé", ce qu'il n'arrive pas à faire à cause de la correspondance ASCII. Les courbes verte, rouge et violette correspondent respectivement aux performances de ResNet18 entraîné à détecter les classes "valide"/"invalide", "approuvé"/"invalide" et "true"/"false" lorsque la classe est placée en fin de séquence pour la génération du QR code. Enfin, la courbe marron représente les performances du classifieur entraîné à détecter les constantes "approuvé"/"invalide" lorsque qu'elles sont placées en début de séquence pour l'encodage vers un QR code.

Nous observons par ce graphique que la similitude entre les mots influe sur les performances du modèle. En effet, pour les séquences ayant le même format, celle qui amène les meilleurs résultats (courbe rouge) contient les mots "approuvé" ou "invalide", qui sont moins proches syntaxiquement que les autres. Nous savons que deux mots syntaxiquement proches génèrent des encodages QR code visuellement similaires. Ainsi, les résultats de ces expérimentations révèlent les limites des architectures convolutives dans la tâche de classification de motifs visuellement proches. De plus, nous pensons que la taille des mots n'a pas réellement d'impact sur les performances du modèle. En effet, plus les mots sont

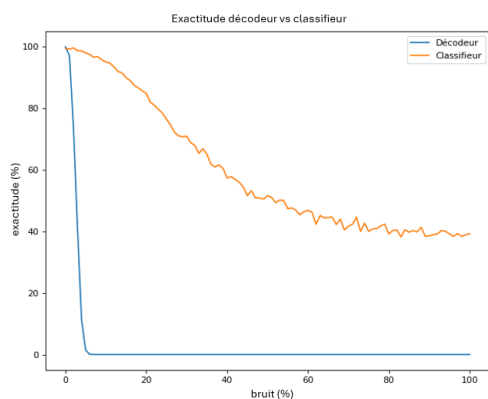


FIGURE 2.10 – Courbes de l’exactitude du décodeur déterministe et du classifieur sur des QR codes générés à partir de séquence de type <https://constante/username> avec constante = *Facebook* ou *Twitter*.

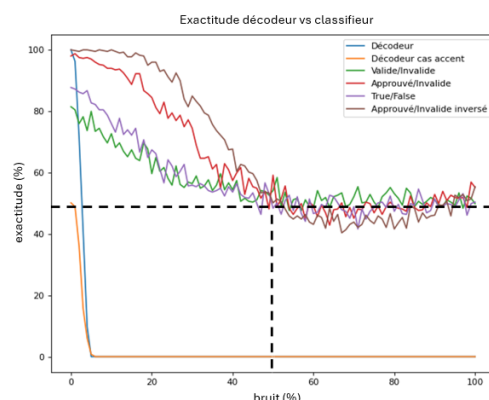


FIGURE 2.11 – Comparaison de l’inférence des modèles entraînés sur différentes classes en fonction du pourcentage de bruit par inversement de pixels (de 0% à 100%).

longs, plus ils seront représentés par plusieurs octets, ce qui rendra l’identification plus facile, sauf si les mots sont syntaxiquement proches. Dans ce cas, cela reviendrait au même que des petits mots syntaxiquement proches, dont il faudra chercher l’octet représentant la différence. Enfin, les expérimentations sur l’impact de la position des constantes "approuvé" et "invalide" dans la séquence montrent que le modèle est plus performant lorsqu’il est entraîné sur des séquences contenant la constante à une position fixe (courbe marron).

Les expérimentations effectuées montrent qu’un modèle de classification d’images basé sur une architecture convolutive peut être plus performant qu’un modèle déterministe dans le cas contraint de la détection d’une constante. Nos travaux ne constituent pas l’état de l’art pour la tâche de décodage des QR codes car le modèle de classification d’images n’a pas été entraîné pour détecter la totalité de la chaîne de caractères encodée mais pour localiser et prédire une information jugée importante dans le QR code. Cependant, ces expérimentations montrent qu’un modèle convolutif de classification couramment utilisé, ResNet18, peut obtenir de bonnes performances sur des images dites "abstraites" et bruitées.

2.4.3 Conclusion des études préliminaires

Nous avons mené deux études préliminaires à nos travaux de transposition de données tabulaires en images afin de vérifier la viabilité de l’application d’un CNN comme extracteur de caractéristiques sur des images non conventionnelles (i.e. non standard dans le domaine du traitement d’images). Ces images incluent des cas où l’information à identifier est peu représentée ou encodée. Nous avons testé différentes configurations de l’information à identifier pour déterminer les propriétés nécessaires à une bonne performance du modèle. En particulier, nous examinons l’impact de la position et de la segmentation de l’information. Les résultats des deux études montrent que les CNN arrivent à apprendre à partir d’images non conventionnelles, même en présence de bruit. Cependant, nos expérimentations mettent aussi

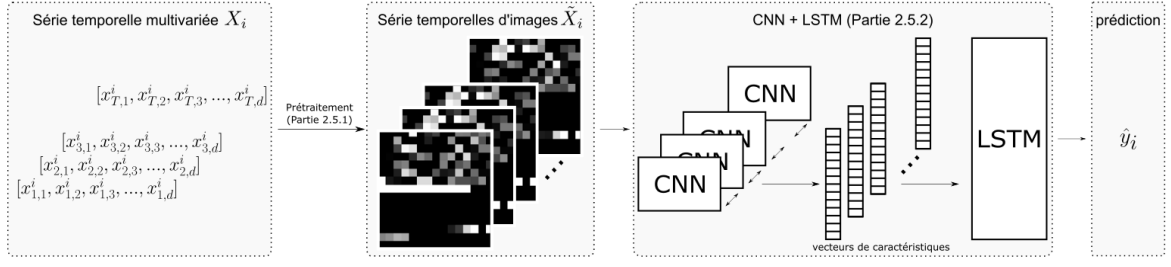


FIGURE 2.12 – Schéma de l'approche proposée. Tout d'abord, la série temporelle multivariée X_i est transformée en une série temporelle d'images $\tilde{X}_i$. Chaque image de cette série temporelle est ensuite transmise à un CNN unique pour extraire des vecteurs de caractéristiques. La série de vecteurs de caractéristiques est ensuite utilisée comme entrée d'un modèle LSTM pour obtenir une prédiction.

en évidence que ces modèles requièrent une position fixe de l'information au sein de l'image abstraite afin d'assurer des résultats optimaux. Nous tenons compte de ces observations dans le développement de notre méthode de transposition de données tabulaires en images, présentées dans la partie qui suit. Ainsi, nous veillons à ce que la valeur représentant une caractéristique soit toujours placée au même pixel sur les différentes images.

2.5 Transformation de données séquentielles en images et application d'un modèle hybride

Nous souhaitons pallier les défis posés par la grande dimensionnalité des données séquentielles en les convertissant en images et en appliquant un modèle hybride léger CNN-LSTM, où le CNN extrait les caractéristiques des images qui sont ensuite données en entrée d'un LSTM pour prendre en compte la temporalité. L'approche que nous proposons est décrite dans la Figure 2.12. Nous convertissons les données tabulaires en images de manière simple, où chaque image correspond à un pas de temps de la séquence et, au sein d'une image, chaque pixel correspond à une caractéristique définie. Nous comparons notre approche à trois modèles couramment utilisés pour traiter les séries temporelles et montrons que notre méthode est plus performante, notamment dans un cas d'usage de classification multi-classes sur des données de santé bruitées.

Dans cette partie nous présentons la méthodologie de l'approche proposée et l'architecture du modèle utilisé. Les expérimentations et les discussions liées aux résultats sont présentées respectivement dans les Sections 2.6 et 2.7.

2.5.1 Méthodologie

Nous considérons un ensemble de données $X = (X_i, y_i)_{i \in \llbracket 1; n \rrbracket}$ composé de n séries temporelles multivariées. Plus précisément, chaque X_i est une séquence $(x_1^i, \dots, x_T^i)$, où T est un entier représentant la longueur de la série. Chaque x_k^i , pour $1 \leq k \leq T$, est un

vecteur de $\mathbb{R}^d$, où $d \in \mathbb{N}$ est le nombre de variables mesurées à chaque pas de temps :

$$\forall i \in \llbracket 1, n \rrbracket, \forall k \in \llbracket 1, T \rrbracket, x_k^i = [x_{k,1}^i, \dots, x_{k,d}^i] \in \mathbb{R}^d \quad (2.2)$$

Chaque y_i est un vecteur $\in \llbracket 1, C \rrbracket$, où $C \in \mathbb{N}$ est le nombre de classes. Notre objectif est de concevoir un modèle permettant de prédire $\hat{y}_i \in \llbracket 1, C \rrbracket$.

Pour pallier le fléau de la dimension engendré par des données avec un grand nombre d'observations (c'est-à-dire un grand T) à haute dimension (c'est-à-dire un grand d), nous proposons de transformer chaque X_i en une série temporelle $\tilde{X}_i$ d'images de taille $m \times m$:

$$\tilde{X}_i = (\tilde{x}_1^i, \dots, \tilde{x}_T^i) \in \mathbb{R}^{T \times m \times m},$$

c'est à dire que chaque $\tilde{x}_k^i$ pour $1 \leq k \leq T$, est une matrice de taille $m \times m$.

Pour transformer le vecteur x_k^i dans $\mathbb{R}^d$ en une matrice $\tilde{x}_k^i$ dans $\mathbb{R}^{m \times m}$ nous appliquons les étapes suivantes :

- (i) poser $m = \lceil \sqrt{d} \rceil$,
- (ii) remplir x_k^i avec $(m^2 - d)$ zéros pour obtenir un vecteur dans $\mathbb{R}^{m^2}$,
- (iii) transposer x_k^i en une matrice de taille $m \times m$ en prenant les variables dans l'ordre d'apparition.

À titre d'exemple, les étapes de transposition pour $x_k^i = [x_{k,1}^i, x_{k,2}^i, x_{k,3}^i, x_{k,4}^i, x_{k,5}^i, x_{k,6}^i, x_{k,7}^i]$ sont :

- (i) on pose $m = \lceil \sqrt{7} \rceil = 3$
- (ii) $x_k^i = [x_{k,1}^i, x_{k,2}^i, x_{k,3}^i, x_{k,4}^i, x_{k,5}^i, x_{k,6}^i, x_{k,7}^i, 0, 0]$
- (iii) $\tilde{x}_k^i = \begin{pmatrix} x_{k,1}^i & x_{k,2}^i & x_{k,3}^i \\ x_{k,4}^i & x_{k,5}^i & x_{k,6}^i \\ x_{k,7}^i & 0 & 0 \end{pmatrix}$

2.5.2 Modèle de classification

Une fois les séquences transformées en séries temporelles d'images, nous entraînons un modèle de classification hybride *from scratch*. Le modèle est composé d'une première phase où un CNN agit comme un extracteur de caractéristiques, suivi d'une seconde phase où un LSTM analyse la temporalité des séquences. Le fonctionnement de ce modèle se déroule en deux étapes principales :

1. **Extraction des caractéristiques par le CNN** : chaque image de la série temporelle, représentée par une matrice de $\mathbb{R}^{m \times m}$, est traitée par le CNN. Ce dernier transforme l'image en un vecteur de caractéristiques dans $\mathbb{R}^N$, où N est une dimension déterminée par l'architecture du réseau. L'application du CNN à chaque image individuelle de la série temporelle $\tilde{X}_i \in \mathbb{R}^{T \times m \times m}$ génère ainsi une nouvelle série de vecteurs, notée $L \in \mathbb{R}^{T \times N}$. Cette série de vecteurs représente les informations essentielles extraites de chaque étape temporelle.

2. **Analyse temporelle par le LSTM** : la série de vecteurs L est ensuite donnée en entrée d'un LSTM, qui est conçu pour gérer les dépendances à long terme et comprendre la dynamique temporelle entre les vecteurs successifs (cf. Section 1.2). Le LSTM traite la série L , en tenant compte de l'ordre temporel des entrées, pour produire une prédiction finale $\hat{y}_i \in \llbracket 1, C \rrbracket$, où C représente le nombre de classes possibles.

Ce modèle hybride tire parti des forces du CNN pour une extraction efficace et précise des caractéristiques statiques à partir des images, et du LSTM pour intégrer ces informations sur l'ensemble de la séquence, permettant ainsi une compréhension approfondie des structures temporelles complexes.

2.6 Expérimentations

Pour évaluer notre méthode, nous la testons sur un ensemble de données constitué de séries temporelles à haute dimension. Plus précisément, nous utilisons un échantillon de données provenant du *Système National des Données de Santé* (SNDS). Les données du SNDS sont présentées dans l'annexe A de notre document. Les spécificités de l'échantillon sélectionné, ainsi que les techniques utilisées pour que sa mise en forme soit adaptée à nos besoins analytiques, sont expliquées dans la Section 2.6.1. La méthode de transposition des données en 2 dimensions, permettant l'application d'architectures convolutives, est décrite dans la Section 2.6.2. Enfin, l'implémentation du modèle hybride que nous employons, combinant un CNN et un LSTM, ainsi que la comparaison avec d'autres modèles existants, sont présentées en détail dans la Section 2.6.3.

2.6.1 Construction du jeu de données

Nous travaillons avec un échantillon du SNDS contenant les données des années 2014 et 2015. Plus précisément, l'échantillon contient 12 tables du *Datamart de Consommation Inter Régime* (DCIR), 6 tables du *Programme de Médicalisation des Systèmes d'Information - Médecine, Chirurgie et Obstétrique* (PMSI-MCO), et les 2 tables Bénéficiaires, qui nous permettant de chaîner les données. Nous agrégeons toutes les informations dans une seule base de données en joignant les tables à l'aide de l'identifiant du patient (cf. chaîne de traitement *SCALPEL-Flattening* dans l'annexe A). Dans la base de données finale, nous appelons chaque ligne une *observation* ou un *événement médical*. Chaque observation contient un identifiant de patient et est donc associée à un individu. Pour obtenir notre jeu de données X , nous regroupons les événements médicaux par identifiant de patient. Dans X , chaque élément X_i représente un patient et sa série temporelle de T événements médicaux qui sont des vecteurs de $\mathbb{R}^d$ (cf. équation 2.2), avec $T \in [0, 500]$. Autrement-dit, nous sélectionnons les individus dont l'historique médical comporte au maximum $T = 500$ événements entre 2014 et 2015. Le type des données est décrit dans la Section 2.6.2.

Nous entraînons notre modèle sur la tâche de classification de pathologies. Nous sélectionnons les individus présentant l'une des 6 pathologies suivantes, tout en veillant à maintenir un équilibre dans la représentation de chaque classe :

- (c_1) tumeurs,
- (c_2) maladies endocriniennes, nutritionnelles et métaboliques,
- (c_3) troubles mentaux et comportementaux,
- (c_4) maladies du système circulatoire,
- (c_5) maladies du système ostéo-articulaire, des muscles et du tissu conjonctif,
- (c_6) lésions traumatiques, empoisonnements et certaines autres conséquences de causes externes.

Pour faciliter la tâche d’entraînement, nous choisissons d’effectuer une classification multi-classes mono-label. Cela signifie que nous entraînons le modèle à reconnaître plusieurs pathologies (multi-classes) mais à n’en associer qu’une seule par individu (mono-label). Ainsi, nous sélectionnons les patients affectés par une seule pathologie.

Finalement, le jeu de données contient $n = 1000$ séries temporelles X_i d’événements médicaux avec une distribution équilibrée entre les $C = 6$ classes. Nous divisons le jeu de données en un ensemble d’apprentissage (65%), un ensemble de validation (15%) et un ensemble de test (20%). Nous utilisons un échantillonnage aléatoire tout en conservant des ensembles de données équilibrés. Chaque événement médical est décrit par 347 caractéristiques qui peuvent être des informations sur le patient ou des informations provenant des remboursements effectués par tous les régimes d’assurance maladie pour les soins en ville ou à l’hôpital. Ces caractéristiques peuvent être catégorielles ou continues. Nous expliquons comment nous traitons ces différences dans la partie suivante.

2.6.2 Des données médicales aux séries temporelles multivariées

Nous nettoyons les données en éliminant les doublons et en supprimant les colonnes contenant uniquement des valeurs manquantes. De plus, nous souhaitons effectuer une classification de pathologies dont le code associé, codé en CIM-10 (10^e révision de la *Classification Internationale des Maladies*⁷), se trouve dans une variable précise. Cependant, dans les données du SNDS, certaines variables sont des extensions de celle que nous voulons prédire et pourraient biaiser le modèle si elles étaient utilisées pour l’entraînement. Nous identifions donc ces variables à l’aide d’une expertise médicale pour les exclure des données.

Un événement médical est représenté par des caractéristiques à valeur réelle, telles que la taille et le poids du patient, et des caractéristiques à valeur catégorielle, comme le sexe du patient ou le code du type d’événement médical. Afin d’uniformiser les types de données, nous appliquons l’approche de *label encoding* sur les attributs catégoriels. Cette méthode convertit les valeurs catégorielles en valeurs entières discrètes, et pour éviter toute confusion avec les valeurs manquantes, nous incrémentons chaque valeur codée de 1. Ainsi, la plus petite valeur codée commence à 1, réservant la valeur 0 exclusivement pour les données manquantes. En effet, la gestion des données manquantes est cruciale dans notre jeu de données, où certaines informations peuvent être absentes, comme la date de décès pour les patients encore vivants. En attribuant systématiquement la valeur 0 aux entrées manquantes,

7. La Classification Internationale des Maladies est une classification médicale utilisée dans le monde entier à des fins épidémiologiques, de gestion de la santé et cliniques.

nous créons une distinction claire qui permet au modèle de reconnaître que le 0 indique une absence d'information.

Dans notre étude, chaque individu est associé à une série temporelle dont la taille T varie selon le nombre d'événements dans son parcours médical. Cependant, les modèles utilisés dans notre analyse ne sont pas conçus pour traiter des séquences de tailles différentes. Pour résoudre ce problème, nous appliquons une technique de *padding*, qui consiste à ajouter des observations contenant uniquement des zéros au début de chaque série temporelle. Cette méthode standardise la longueur de toutes les séries à un nombre fixe d'observations T , permettant ainsi à nos modèles de traiter uniformément les données.

Ainsi, après le nettoyage des données, la conversion des données catégorielles en numérique, et l'ajout de *padding* pour uniformiser les longueurs des séries temporelles, le jeu de données X , est représenté par des séries temporelles X_i de taille $T = 500$ et de dimension $d = 347$ (correspondant à $m = \lceil \sqrt{347} \rceil = 19$).

2.6.3 Modèles et paramétrisation

L'objectif de nos expérimentations est de faire de la classification de pathologies. Pour cela, nous proposons d'utiliser un modèle hybride composé d'un CNN comme extracteur de caractéristiques des images obtenues après la transformation des séries temporelles (cf. Section 2.5.1), suivi d'un LSTM, pour prendre en compte la temporalité des séries.

Les CNN sont traditionnellement utilisés pour traiter des images figuratives, souvent en couleur RVB (Rouge, Vert, Bleu) ou en niveaux de gris. Ces images sont typiquement normalisées dans un intervalle de 0 à 255, correspondant à l'intensité des pixels. Pour nous aligner sur cette utilisation conventionnelle et maximiser les capacités du CNN, nous normalisons les séries temporelles pour correspondre à la gamme des niveaux de gris. Autrement-dit, les images que nous donnons en entrée du CNN sont normalisées de telle sorte que $\tilde{X}_i \in \llbracket 0, 255 \rrbracket^{T \times m \times m}$. Plus précisément, nous normalisons les valeurs de chaque caractéristique de sorte que la valeur maximale soit associée à 255 et la minimale à 0. Les autres valeurs sont représentées par des nombres réels entre ces deux extrêmes, reflétant leur position relative dans cet intervalle. Cette normalisation, effectuée après les pré-traitements initiaux, permet au CNN de fonctionner dans des conditions optimales et de se rapprocher des performances qu'il aurait avec des images traditionnelles en niveaux de gris. Les images moyennes par classes sont présentées dans la Figure 19 de l'annexe C.

Concernant l'architecture, nous souhaitons qu'elle soit la plus légère possible pour pallier les problèmes de coûts computationnels liés au fléau de la dimension. Ainsi, notre architecture CNN, détaillée dans la Figure 2.13 se compose de trois couches convolutives 2D, une couche entièrement connectée qui correspond au vecteur des caractéristiques extraites, suivie d'une couche LSTM, d'une couche de *dropout* [5] et d'une couche entièrement connectée.

Nous comparons notre modèle à trois architectures conçues pour le traitement de séquences :

- 1DCNN-LSTM, ayant la même profondeur que le CNN-LSTM bidimensionnel, basé sur l'hypothèse qu'il pourra extraire un niveau d'information comparable à celui du

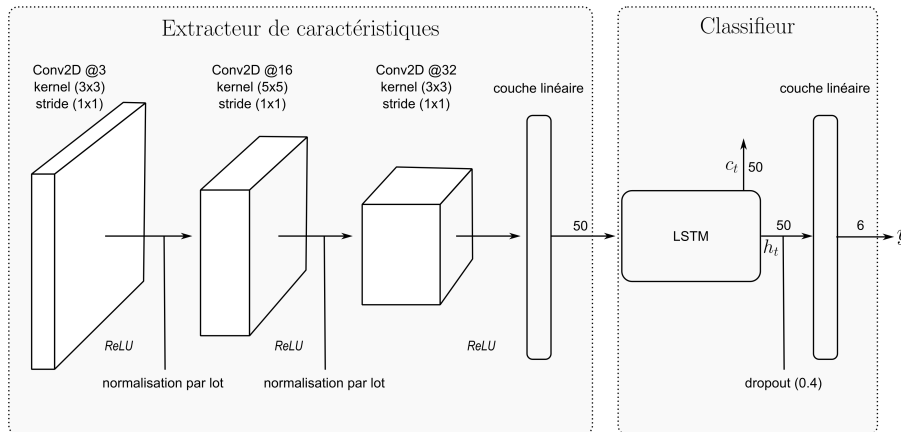


FIGURE 2.13 – Architecture du modèle CNN-LSTM utilisé. Le modèle se compose de trois couches convolutives 2D avec 3, 16 et 32 filtres ayant respectivement une taille de 3×3 , 5×5 et 3×3 . Une normalisation par lot [8] est placée après la première et la deuxième couche, et la fonction d'activation ReLU est utilisée pour chaque couche. Après ces couches convolutives, les caractéristiques extraites sont aplatées et passées à une couche entièrement connectée. Cette couche entièrement connectée a une sortie de taille 50, correspondant au vecteur final des caractéristiques extraites. L'architecture CNN est suivie par un LSTM avec un état caché (h_t) et un état de cellule (c_t) de taille 50. Le LSTM est suivi d'une couche de *dropout* avec une probabilité de 0,4 et d'une couche entièrement connectée avec une sortie de taille 6.

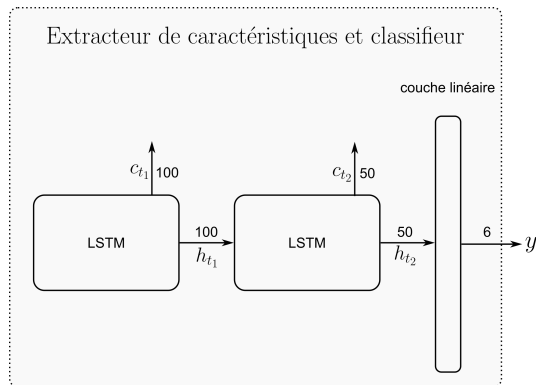


FIGURE 2.14 – Architecture du modèle LSTM utilisé. Le modèle comporte deux couches LSTM successives. La première couche LSTM a un état de cellule c_{t_1} et un état caché h_{t_1} , tous deux de taille 100. La deuxième couche LSTM a un état de cellule c_{t_2} et un état caché h_{t_2} , tous deux de taille 50. Ces couches LSTM sont suivies d'une couche entièrement connectée de taille 6 en sortie, correspondant au nombre de classes à prédire.

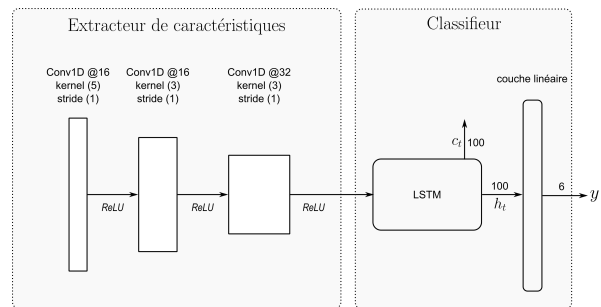


FIGURE 2.15 – Architecture du modèle 1DCNN-LSTM utilisé. Le modèle est composé de 3 couches convolutives 1D avec 16, 16 et 32 filtres ayant respectivement une taille de 5, 3 et 3. La fonction d'activation ReLU est utilisée pour chaque couche. La partie LSTM du modèle a un état caché (h_t) et un état de cellule (c_t) de taille 100 et une couche entièrement connectée avec une sortie de taille 6.

modèle à convolutions bidimensionnelles,

- LSTM [35] dont le nombre de paramètres est du même ordre de grandeur que celui du 1DCNN-LSTM,
- BERT [13] de la famille des Transformers.

Le LSTM, dont l’architecture est détaillée dans la Figure 2.14, comportent deux couches LSTM suivies d’une couche entièrement connectée. L’architecture du 1DCNN-LSTM, décrite dans la Figure 2.15, est composée de 3 couches convolutives à une dimension, suivies d’une couche LSTM et d’une couche entièrement connectée.

Le LSTM et le 1DCNN-LSTM sont entraînés *from scratch* tandis que nous affinons un BERT [39] pré-entraîné sur 800 millions de mots issus du jeu de données *BookCorpus* [104] et 2,5 milliards de mots issus de la version anglaise de *Wikipédia* (à l’exclusion des listes, des tableaux et des en-têtes).

Contrairement à la normalisation entre 0 et 255 appliquée aux images fournies en entrée du modèle CNN-LSTM, pour les modèles LSTM et 1DCNN-LSTM, nous utilisons la transformation logarithmique, $\log(x + 1)$, pour normaliser les entrées X_i . Cette méthode permet d’atténuer l’impact des valeurs extrêmes, facilitant ainsi l’apprentissage des modèles en réduisant les variations importantes de l’amplitude des données d’entrée. Concernant le modèle BERT, nous utilisons le même *tokenizer* que celui employé lors de son pré-entraînement. Cette cohérence est cruciale pour garantir que les tokens générés correspondent aux représentations vectorielles pré-apprises, maximisant ainsi l’efficacité du transfert d’apprentissage.

Finalement, nous entraînons tous les modèles pendant 500 *epochs* avec un *batch* de taille 128. Nous utilisons la fonction de perte de l’entropie croisée [102] et l’optimiseur de la descente de gradient stochastique avec un taux d’apprentissage de 0,01, qui est divisé par un facteur 10 lorsque la perte sur le jeu de validation a cessé de diminuer depuis 5 *epochs*.

2.7 Résultats et discussions

modèle	nombre de paramètres entraîna- bles	exactitude ⁸	temps d’entraînement
BERT	108M	21%	93min 19s
1DCNN-LSTM	260K	20%	61min 27s
LSTM	210K	19%	19min 16s
Modèle proposé (CNN-LSTM)	28K	46%	30min 21s

TABLE 2.3 – Performances et temps d’apprentissage des modèles de classification.

Les résultats présentés dans la Table 2.3 montrent que la tâche de classification des séries temporelles d’images avec le modèle CNN-LSTM atteint de meilleures performances

8. L’exactitude est utilisée pour avoir une vue d’ensemble des performances, bien que le rappel puisse être plus pertinent dans certains cas d’usage médicaux.

en terme d'exactitude (ratio entre le nombre d'observations bien classées et le nombre total d'observations) que les modèles prenant des données à une dimension en entrée. Les faibles performances de ces modèles pourraient s'expliquer par le problème de disparition du gradient [30], qui affecte particulièrement les réseaux récurrents lors du traitement de séries temporelles longues. Cependant, cette différence pourrait également être due au nombre important de paramètres de ces modèles par rapport à la taille de notre jeu de données d'entraînement, suggérant qu'un échantillon plus conséquent pourrait leur permettre de mieux converger. Malgré ces limitations, nos résultats suggèrent des avantages prometteurs de notre approche. En effet, le CNN-LSTM s'est montré plus efficace que le LSTM seul, indiquant que la méthode d'extraction de caractéristiques par un CNN avant la classification par le LSTM pourrait être plus adaptée que l'utilisation du LSTM seul en tant qu'extracteur de caractéristiques et classifieur. De plus, le CNN-LSTM avec des convolutions 2D obtient une exactitude supérieure de 26% par rapport au 1DCNN-LSTM, en deux fois moins de temps d'apprentissage. Ces résultats, bien qu'ils nécessitent une validation sur un jeu de données plus large, suggèrent l'intérêt de passer aux convolutions 2D pour des séquences de longue taille, permettant potentiellement une meilleure capture des caractéristiques spatiales et temporelles complexes, en moins de temps.

Concernant les performances des modèles par classe (cf. les matrices de confusion des différents modèles dans la Figure 2.16), on observe que le modèle CNN-LSTM montre une répartition plus équilibrée des prédictions entre les différentes classes (à l'exception de la première classe - tumeurs). Les modèles LSTM, 1DCNN-LSTM et BERT tendent à converger vers une ou deux classes dominantes, malgré un jeu de données équilibré. Si ce comportement pourrait à première vue suggérer un surapprentissage des modèles, il pourrait également être lié aux hyperparamètres d'apprentissage choisis. Ainsi, bien que notre approche semble mieux capturer les caractéristiques des différentes classes, une étude plus approfondie de l'influence des hyperparamètres sur la capacité des modèles à distinguer les classes serait nécessaire pour une comparaison plus robuste.

Le modèle que nous proposons, l'approche de transposition de données tabulaires en images, combinée à l'application d'un petit modèle hybride CNN-LSTM surpasse les modèles avec lesquels il est comparé. En effet, les modèles appliqués sur les données à une dimension ont des performances proches de l'aléatoire sur notre problème. Nous pensons que les modèles appliqués aux données brutes sont limités par la taille des séquences. En effet, plus les séquences sont longues, plus le modèle a des difficultés à extraire les corrélations entre les caractéristiques. La taille des séquences est également problématique pour l'utilisation des approches classiques de classification. En raison de leur complexité temporelle, nous n'avons pas pu appliquer les modèles *Time series forest (TSF)* [105] et *Bag-of-SFA-Symbols Ensemble (BOSSensemble)* [106] sur les données brutes.

Nous avons également testé différentes architectures du CNN-LSTM. Nous remarquons que l'ajout de couches de *pooling* fait baisser l'exactitude de 46% à 30%. L'information de nos images résidant dans chaque pixel, l'agrégation spatiale génère une perte d'informations et donc de performances du modèle.

Le modèle que nous proposons a été entraîné sur des données complexes et très bruitées. En effet, qu'il s'agisse de l'ajout de *padding*, des données manquantes ou encore de la présence de caractéristiques non corrélées à la détection de la pathologie, nous pensons atteindre la

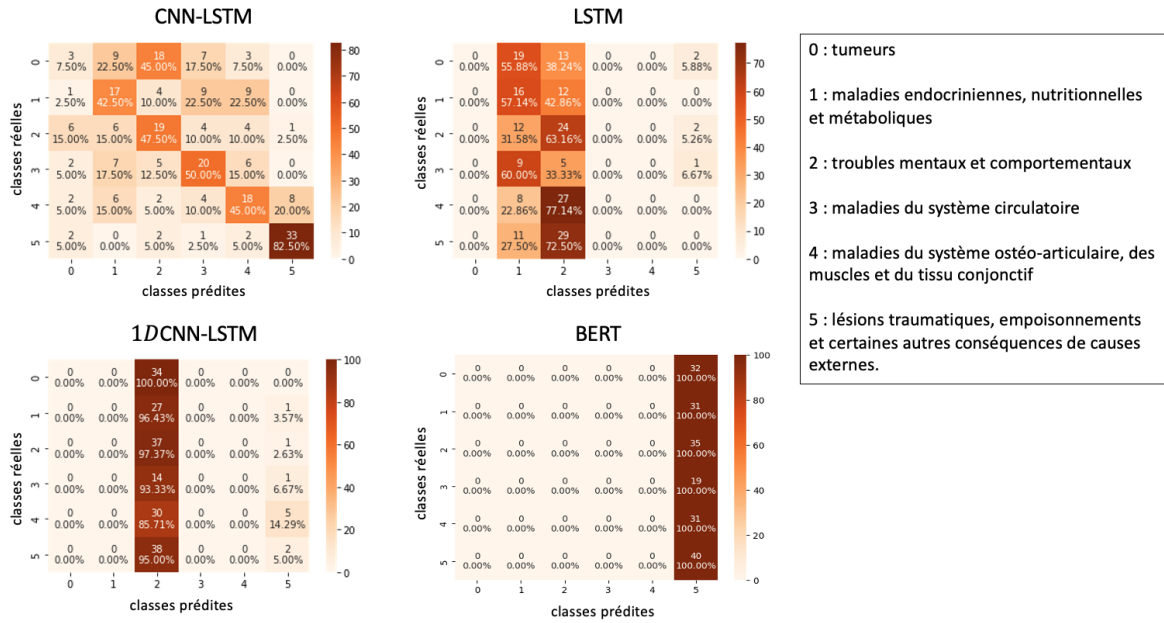


FIGURE 2.16 – Matrices de confusion de notre modèle (CNN-LSTM) et des modèles 1DCNN-LSTM, LSTM et BERT, pour la tâche de classification de 6 pathologies.

limite des performances de la tâche de classification de pathologies sur notre échantillon de données. Au vu des résultats obtenus, nous pensons que notre approche peut apporter une réelle plus-value sur de longues séries temporelles multivariées. En effet, les images que nous générons à partir du SNDS sont de taille (19×19) , ce qui est loin des dimensions spatiales prises en entrée des CNN modernes (par exemple, les réseaux pré-entraînés sur *ImageNet* prennent souvent des images de taille 224×224 en entrée). Cela signifie qu'il serait concevable de générer des images à partir de données à plus haute dimension avec notre approche, tout en ayant un coût computationnel raisonnable. De plus, nous pourrions envisager de décomposer certains champs comme la date, qui seraient alors représentés par plusieurs pixels, sans être contraint par la taille de la matrice.

2.8 Conclusion et perspectives

Dans cette étude, nous proposons une nouvelle approche pour la classification des séries temporelles en transformant les données à haute dimension en représentations basées sur des images. Il s'agit d'une contribution importante car elle nous permet de surmonter le *fléau de la dimension*, qui est un problème courant dans l'analyse des données à haute dimension. Les méthodes traditionnelles nécessitent souvent une réduction de dimension, mais cela peut entraîner une perte d'informations. En utilisant des représentations basées sur l'image, nous sommes en mesure de conserver toutes les informations contenues dans les données tout en appliquant de puissantes techniques de traitement d'images.

Une autre contribution clé de notre approche est l'utilisation d'une architecture petite et légère, que l'on peut entraîner en un temps raisonnable, ce qui en fait une solution pratique

pour les applications du monde réel. Nous proposons un modèle hybride qui combine les avantages des CNN et des LSTM. La transposition des données séquentielles tabulaires en séries temporelles d'images nous permet d'exploiter la capacité des CNN à extraire des caractéristiques pertinentes des images, qui peuvent ensuite être transmises au LSTM, capable de capturer les dépendances temporelles dans les données. Nous constatons qu'en utilisant cette approche hybride, notre modèle obtient des performances nettement meilleures que les modèles récurrents pour la classification des séries temporelles. Toutefois, une étude plus approfondie de l'impact des hyperparamètres d'apprentissage serait nécessaire pour s'assurer de la bonne convergence de l'ensemble des modèles et consolider ces résultats.

Dans nos travaux futurs, en plus d'étudier l'impact des hyperparamètres d'apprentissage pour valider la convergence des modèles, nous envisageons d'étendre notre méthode à des jeux de données présentant une dimensionnalité encore plus élevée. Cette évolution nous permettrait de générer des images résultant de la transformation des données qui se rapprochent des tailles standards généralement traitées par les CNN. Cette adaptation permettrait d'améliorer l'efficacité de l'apprentissage sur des données à très haute dimension en exploitant des architectures conçues pour des images de dimensions plus conséquentes, telles que celles couramment utilisées dans les applications de vision par ordinateur.

De plus, nous aimerions regrouper les variables selon leur nomenclature avant de procéder à la transformation en images. Cette stratégie vise à structurer les données de manière à refléter des regroupements logiques, enrichissant ainsi le contexte local de chaque pixel. En organisant les données de cette façon, nous pourrions tirer pleinement parti des capacités des CNN. En effet, cette approche permettrait aux CNN de mieux capturer les relations spatiales et contextuelles entre les différents groupes de variables, augmentant ainsi leur capacité à identifier des motifs pertinents dans des données de haute dimension.

Ces développements futurs devraient nous permettre de non seulement valider davantage l'efficacité de notre méthode, mais aussi d'explorer de nouvelles façons de traiter des ensembles de données complexes en palliant certaines limites liées au fléau de la dimension. Alors que les grands modèles de langage (LLM) dominent actuellement le traitement du texte, cherchant à capturer toujours plus de contexte et à apprendre à partir d'immenses volumes de données, notre approche prend une direction différente. Nous visons à utiliser moins de données d'entraînement et de ressources computationnelles, ouvrant ainsi la voie à des solutions efficaces pour des problèmes de haute dimensionnalité dans des contextes aux ressources contraintes.

Chapitre 3

Représentation des images en cartes de saillance pour la détection d'anomalies

3.1 Introduction aux défis liés à la détection d'anomalies

La détection des anomalies, également connue sous les noms de détection des valeurs aberrantes ou détection des nouveautés, fait référence au "problème de la recherche de motifs dans les données qui ne sont pas conformes au comportement attendu" [107]. Ce domaine de recherche, actif depuis les années 1960 [108], a gagné en importance au fil des décennies. Aujourd'hui, en raison de la demande croissante pour des applications variées telles que la gestion des risques, la conformité réglementaire, la sécurité globale, la surveillance financière, la santé et la télédétection, la détection d'anomalies devient cruciale dans divers champs disciplinaires incluant l'exploration de données, l'apprentissage automatique, les statistiques et la vision par ordinateur [109].

En vision par ordinateur, la détection d'anomalies consiste à identifier automatiquement des structures anormales dans des images. Cette tâche est complexe car elle nécessite la distinction entre ce qui est habituel et ce qui ne l'est pas, sans disposer d'une définition universellement acceptée de la norme applicable à toutes les images [110]. Par exemple, dans le cadre de l'analyse des images générées à partir des données du SNDS pour la classification des pathologies (cf. Chapitre 2), diverses anomalies peuvent être identifiées. Parmi celles-ci, on peut noter la prescription d'un médicament inadapté à une pathologie spécifique. À l'inverse, l'absence de prescription d'un traitement essentiel pour une maladie donnée constitue également une anomalie significative. Ces écarts par rapport aux pratiques médicales recommandées peuvent révéler des problèmes potentiels dans la prise en charge des patients. Ainsi, en les identifiant, il devient possible d'intervenir de manière ciblée, que ce soit par l'ajustement des protocoles de traitement ou la mise en place de systèmes d'alerte pour prévenir les erreurs de prescription.

Pour cette étude, nous choisissons de nous écarter de l'analyse des données du SNDS, dont la définition des anomalies aurait nécessité un laborieux travail d'annotation des parcours médicaux par des experts. À la place, nous nous concentrons sur un cas d'usage spécifique au

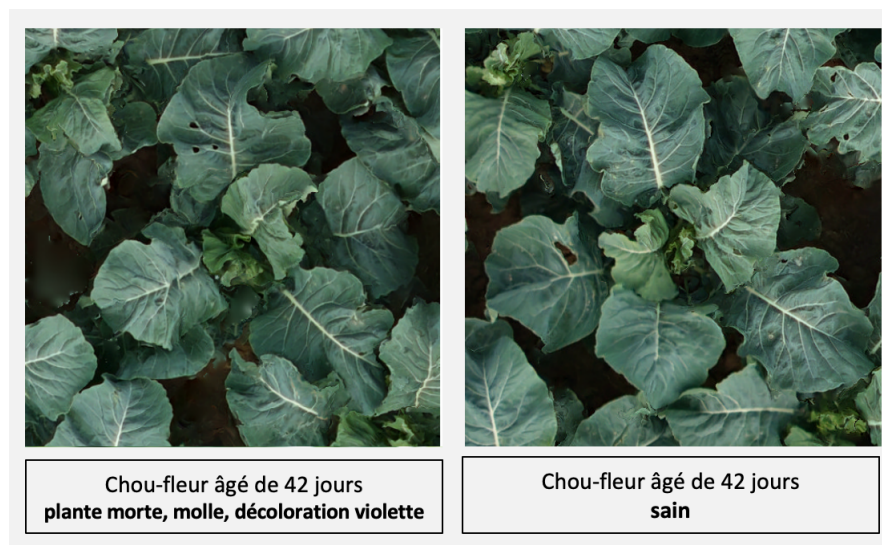


FIGURE 3.1 – Deux choux-fleurs âgés de 42 jours. Celui de gauche est anormal et celui de droite est normal

domaine de la télédétection : la détection des plantes malades. Ce choix est motivé par les défis sans précédent auxquels l'agriculture moderne doit faire face, notamment le changement climatique, l'augmentation de la population mondiale et la nécessité d'accroître la production alimentaire sur des surfaces cultivables restreintes. Dans ce contexte, une gestion efficace des maladies des plantes [111, 112] devient cruciale, car elle joue un rôle essentiel pour maintenir la productivité agricole, garantir la sécurité alimentaire et promouvoir la durabilité environnementale [113].

L'utilisation de l'apprentissage profond, et en particulier des réseaux de neurones convolutifs (voir Section 1.1), est de plus en plus courante pour la détection d'anomalies, mais elle est limitée par un problème majeur : le manque de données annotées [114]. En effet, dans le contexte de la détection d'anomalies, les instances anormales sont, par définition, celles qui s'éloignent significativement de la distribution typique des données. Cette nature divergente les rend moins fréquentes, ce qui résulte en une surreprésentation des données conformes à la distribution. Cette disproportion pose un risque significatif de surapprentissage [4], où le modèle devient extrêmement bien adapté aux données normales mais incapable de reconnaître efficacement les anomalies, qui sont sous-représentées.

En plus des difficultés liées au déséquilibre des classes normales et anormales, les anomalies peuvent être à peine perceptibles visuellement, ce qui complique la tâche de classification. La Figure 3.1 montre un exemple de deux choux-fleurs qui se ressemblent et qui ont été annotés par un expert. Celui de gauche présente de nombreux défauts ("plante morte", "molle" et "décoloration violette" ont été identifiés par un expert) tandis que celui de droite est sain. Enfin, les plantes normales et anormales peuvent présenter de fortes variations intra-classe qui sont difficiles à apprendre par un modèle supervisé et donc limiter sa généralisation.

Face aux défis posés par le surapprentissage et la détection de variations subtiles dans les données, nous proposons une approche qui se concentre sur l'identification des caractéristiques d'importance communes entre les données normales et évalue la proximité des

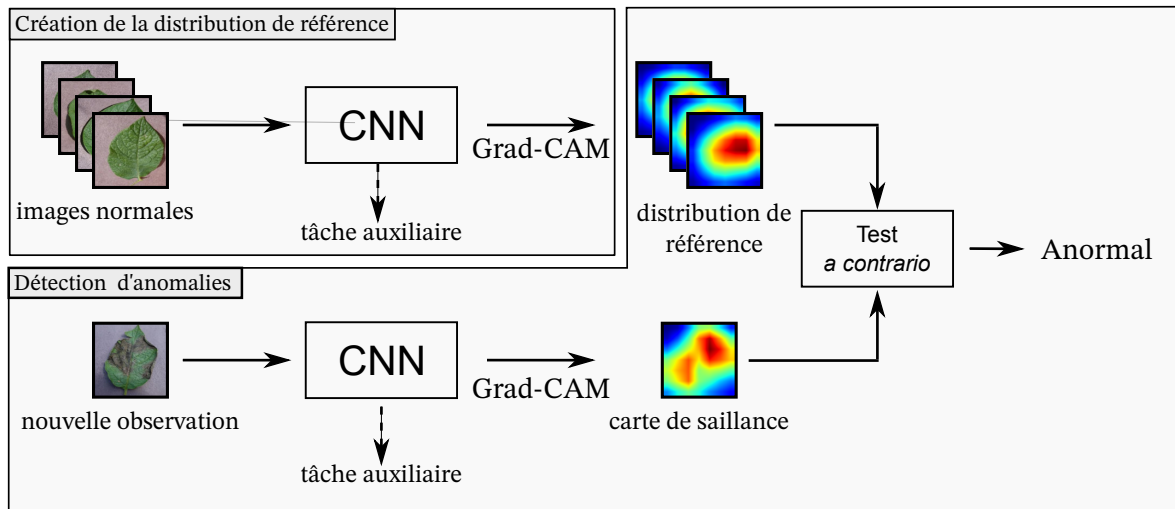


FIGURE 3.2 – Schéma de l’approche proposée pour la détection d’anomalies. Pour chaque nouvelle observation, nous obtenons la carte de saillance associée en appliquant Grad-CAM à un modèle entraîné sur une tâche auxiliaire. Nous comparons ensuite cette carte de saillance à la distribution de référence des cartes de saillance issues d’images normales en utilisant une approche *a contrario* pour déterminer si la nouvelle observation est normale ou non.

caractéristiques d’importance d’une nouvelle image avec la norme établie par les images normales. Notre étude est fondée sur l’hypothèse que les caractéristiques d’importance des données normales diffèrent de celles des données anormales.

Notre approche, schématisée dans la Figure 3.2, consiste d’abord à entraîner un modèle sur une tâche auxiliaire. Nous appliquons ensuite un modèle d’explicabilité sur le modèle pour récupérer des cartes de saillance, révélant les zones d’importance des images. À partir de ces résultats, nous établissons une distribution de référence représentative des caractéristiques d’importance des images normales. Finalement, pour définir si une nouvelle image est normale ou non, nous appliquons un test statistique *a contrario*. Plus précisément, nous calculons un score de normalité, reflétant à quel point les caractéristiques d’importance d’une nouvelle image suivent la norme établie par la distribution de référence. Enfin, nous comparons ce score à un seuil prédéfini pour déterminer si l’image est normale ou anormale.

Contributions

1. Nous proposons une méthode qui entraîne un modèle sur une tâche prétexte triviale pour obtenir des cartes de saillance, obtenues par un modèle d’explicabilité, révélant les zones d’importance des images. Sur cette base, nous développons une approche *a contrario* pour la détection d’anomalies.
2. Nous introduisons une formulation du problème de détection des anomalies des plantes qui ne nécessite **aucune connaissance préalable** des anomalies.
3. Nous démontrons que notre modèle surpasse les approches de l’état de l’art sur trois ensembles de données, ce qui atteste des capacités de **généralisation** de notre méthode.

Plan du chapitre

État de l’art (Section 3.2) : présentation des approches de détection d’anomalies, des méthodes utilisant une approche *a contrario*, et des techniques d’explicabilité. Chaque catégorie est abordée individuellement, avec une partie dédiée à leur application dans le domaine de la télédétection.

Détection d’anomalies par une approche *a contrario* (Section 3.3) : méthodologie de notre approche.

Expérimentations (Section 3.4) : présentation des expériences réalisées sur divers jeux de données de plantes et comparaison avec des modèles de référence dans le domaine de la détection d’anomalies.

Résultats et discussions (Section 3.5)

Conclusion et perspectives (Section 3.6)

3.2 État de l’art

Les méthodes de détection d’anomalies dans les images peuvent être divisées selon deux tâches [115] : la détection d’anomalies au niveau de l’image, pour définir si l’image est normale ou anormale, et la détection d’anomalies au niveau du pixel, utilisée pour localiser les zones anormales dans une image. Les travaux que nous présentons dans ce chapitre se concentrent sur la détection des anomalies au niveau de l’image. Nous présentons donc par la suite l’état de l’art des méthodes de cette catégorie (cf. Section 3.2.1), et non des approches de détection d’anomalies au niveau du pixel.

De plus, notre approche de détection d’anomalies repose sur une méthode *a contrario* appliquée à des cartes de caractéristiques obtenues par des techniques d’explicabilité. Nous introduisons le test *a contrario* dans la Section 3.2.2 et les approches d’explicabilité dans la Section 3.2.3.

3.2.1 Détection d’anomalies au niveau de l’image

Les approches de détection des anomalies au niveau de l’image peuvent être divisées en 3 catégories, les approches d’estimation de la densité, les approches de classification à une classe et les modèles auto-supervisés.

3.2.1.1 Approches d'estimation de la densité

L'idée principale des approches d'estimation est de construire un modèle statistique qui caractérise la distribution normale des données. Ce modèle sert ensuite de référence pour détecter les anomalies qui s'écartent significativement de cette distribution attendue. Ces approches comprennent d'une part les méthodes d'estimation non paramétriques, telles que la méthode d'estimation de densité par noyau [116] et la méthode des plus proches voisins *K-Nearest Neighbors* (KNN) [117], et d'autre part les méthodes d'estimation de distribution paramétrique, comme le mélange gaussien [118].

L'avantage principal de ces méthodes est leur capacité à capturer des déviations fines par rapport à la normale, permettant une détection plus nuancée que les approches basées sur de simples seuils. Cependant, l'estimation d'une densité de probabilité nécessite un grand nombre de données d'entraînement mais la capacité des modèles classiques à traiter de gros volumes de données sans perdre en performance est limitée.

Une alternative plus récente utilisant les auto-encodeurs variationnels [119] permet d'apprendre une représentation latente des données pour estimer leur distribution, mais fait face aux mêmes contraintes en termes de volume de données nécessaires.

3.2.1.2 Approches de classification à une classe

Les approches de classification à une classe, comme *Support Vector Data Description* (SVDD) [120] et *One-Class Support Vector Machines* (OCSVM) [121], se concentrent sur la catégorisation d'une classe unique. Elles visent à établir une frontière de décision autour de la classe cible (e.g. les images normales) dans l'espace des caractéristiques.

Avec l'avènement des techniques d'apprentissage profond, l'intérêt s'est progressivement déplacé vers l'intégration des CNN aux approches traditionnelles de classification à une classe. Oza et al. ont proposé *One-Class Convolutional Neural Network* (OCCNN) [122], un modèle spécifiquement entraîné pour la classification d'anomalies à une classe. Perera et al. ont proposé une méthode basée sur l'apprentissage par transfert, qui utilise un CNN pré-entraîné pour extraire des caractéristiques d'image discriminantes puis un KNN (*K-Nearest Neighbors*) pour construire le classifieur à une classe [123].

Bien que les modèles de classification à une classe soient efficaces pour identifier des anomalies en se basant uniquement sur les données normales, leur performance peut être limitée par la complexité des données. En effet, le fléau de la dimension [3] affecte particulièrement les modèles de classification à une classe lorsqu'ils sont appliqués à des données de haute dimension, comme les images. Par "haute dimension", on entend un espace de données où le nombre de caractéristiques est élevé, souvent bien supérieur au nombre d'échantillons. Les images sont considérées comme des données à haute dimension parce que chaque pixel dans une image peut représenter une variable distincte. Ainsi, à mesure que la dimension des données augmente, il peut devenir difficile de distinguer les anomalies des données normales sans une augmentation substantielle de la quantité de données d'entraînement. En effet, la dilution de la densité des données et la difficulté à capturer les structures pertinentes nécessitent davantage de données d'entraînement pour garantir une bonne généralisation.

Une manière d’aborder ces limitations consiste à apprendre des représentations discriminantes des données, soit à travers des tâches prétextes, soit en apprenant à reconstruire les caractéristiques normales des données. C’est le principe des modèles auto-supervisés, présentés dans la partie suivante.

3.2.1.3 Modèles auto-supervisés

Les méthodes auto-supervisées de détection d’anomalies reposent sur l’idée de former des modèles pour apprendre à réaliser des tâches spécifiques (appelées tâches prétextes ou auxiliaires) sans avoir besoin de données étiquetées pour l’apprentissage. Ces tâches prétextes sont conçues de manière à ce que le modèle apprenne des caractéristiques importantes et discriminantes sur des données normales, permettant ainsi au modèle de détecter des anomalies lorsque des données s’écartent significativement de ces caractéristiques apprises. En effet, si les données de test diffèrent significativement des représentations normales apprises (par exemple, si elles ne suivent pas les modèles ou les prédictions basées sur la tâche prétexte), elles sont alors considérées comme anormales. En résumé, ce processus exploite la capacité du modèle à identifier des caractéristiques normales et à détecter les écarts comme indications potentielles d’anomalies.

Les différents modèles de détection d’anomalies auto-supervisés se distinguent principalement par les tâches prétextes utilisées pour l’entraînement, les architectures de réseau, et les stratégies d’évaluation des anomalies [124]. Par exemple, Golan et al. utilisent la rotation d’images comme tâche prétexte, entraînant le modèle à reconnaître des orientations [125]. Tandis que Li et al. créent des anomalies en découpant et collant des segments d’image, apprenant ainsi à identifier des configurations visuelles atypiques grâce à leur modèle *CutPaste* [126].

L’arrivée de l’apprentissage contrastif [127] représente une avancée significative dans l’apprentissage auto-supervisé pour la détection d’anomalies [128]. Cette méthode améliore la représentation des données en rapprochant, sans supervision, les exemples positifs d’un même échantillon tout en éloignant les exemples négatifs, permettant une distinction plus nette entre les données normales et anormales. Les modèles basés sur l’apprentissage contrastif, comme CSI [128] et SSD (*Self-Supervised outlier Detection*) [129], obtiennent d’excellentes performances en apprenant des représentations robustes qui facilitent la détection précise d’anomalies.

Parmi les approches auto-supervisées, les méthodes basées sur la reconstruction constituent une approche fondamentale qui repose sur l’entraînement d’un modèle à reconstruire des données normales, permettant ainsi d’identifier comme anomalies les données que le modèle ne parvient pas à bien reconstruire. Leur fonctionnement repose sur la compression de l’image dans un espace de faible dimension, suivie de sa reconstruction. L’hypothèse sous-jacente est que les images normales peuvent être reconstruites avec peu d’erreurs, tandis que les images anormales présentent des erreurs de reconstruction plus importantes. Les auto-encodeurs variationnels [119] sont fréquemment utilisés pour cette tâche car ils sont capables de séparer les informations redondantes des informations essentielles grâce à une couche cachée de faible dimensionnalité. Cette couche, souvent identifiée comme l’espace latent, représente les données en un espace de dimension réduite, forçant ainsi le

modèle à apprendre les distributions les plus significatives des données originales, tout en écartant les détails redondants ou bruités.

Des avancées ont été faites pour améliorer la détection d'anomalies en exploitant la distribution des données dans l'espace latent, en augmentant la difficulté de la reconstruction pour les images anormales, et en utilisant des approches basées sur les réseaux antagonistes génératifs (GANs) [130], comme *AnoGAN* [131], pour mieux distinguer les images normales des anormales. Certaines méthodes combinent la reconstruction d'image et l'entraînement antagoniste pour améliorer à la fois la performance et l'efficacité de la détection d'anomalies [132].

Bien que les approches par reconstruction offrent une méthode valable pour la détection d'anomalies, elles rencontrent plusieurs limites, notamment une forte dépendance vis-à-vis des données d'entraînement et de la fonction de coût, une réduction potentielle de l'efficacité en présence d'anomalies difficiles à détecter ou de données non représentatives, ainsi qu'une complexité qui peut dégrader l'interprétation des résultats.

3.2.1.4 Travaux sur la détection des maladies des plantes

Dans le domaine de la télédétection, la détection des maladies des plantes représente un défi important [133]. Pour résoudre ce problème, une multitude d'études ont vu le jour, exploitant des modèles couramment utilisés dans le domaine de la vision par ordinateur. Ces études comprennent l'utilisation d'architectures convolutives [134], de systèmes de classification [135], ainsi que de techniques dédiées à d'autres tâches spécifiques. Par exemple, Fuentes et al. utilisent des modèles de détection d'objets pour détecter et localiser les infestations de parasites et les maladies affectant les tomates [136]. Wang et al. utilisent un modèle de segmentation d'instance [137] pour identifier les pathologies sur les feuilles de maïs [138]. Des approches non supervisées ont également été explorées, comme *AnoLeaf* [139], qui utilise un réseau génératif pour l'inpainting afin de détecter les maladies des plantes. Ces travaux illustrent comment les avancées en apprentissage profond, et plus spécifiquement en vision par ordinateur, peuvent être appliquées efficacement à la détection d'anomalies dans les images de plantes.

3.2.2 Test *a contrario*

Lorsque le problème de la détection d'anomalies est tel que nous ne disposons pas de beaucoup de données annotées sur la présence d'anomalies, il est difficile d'utiliser des modèles d'apprentissage nécessitant beaucoup d'exemples. Dans ce cas, les approches *a contrario* [140, 141] peuvent être efficaces. Le principe de ces approches est de concevoir un modèle de référence, considéré comme une hypothèse nulle représentant la norme. La probabilité qu'une nouvelle observation appartienne au modèle de référence est alors calculée. Après seuillage de cette probabilité, on peut décider si cet échantillon est normal ou anormal. Agnès Desolneux définit les étapes d'un test *a contrario* comme suit [12] :

1. Définir une hypothèse nulle ($\mathcal{H}_0$) : l'hypothèse nulle est une distribution de probabilité sur des éléments spécifiques. Elle représente une situation de référence dans laquelle

aucun phénomène d'intérêt particulier n'est supposé se produire. Dans le contexte de l'imagerie, cela pourrait signifier une distribution aléatoire ou uniforme des éléments (comme des points ou des motifs) dans une image.

2. Définir un seuil ε : cette valeur est utilisée pour déterminer à quel point un événement observé doit être improbable sous l'hypothèse nulle pour être considéré comme significatif.
3. Observer un événement géométrique (E) dans une image : cet événement est une configuration particulière d'éléments détectée dans l'image. L'événement pourrait être, par exemple, un alignement spécifique de points ou une forme particulière.
4. Calculer la probabilité de E sous $\mathcal{H}_0$: cette étape consiste à évaluer à quel point l'événement observé (E) est probable si l'hypothèse nulle était vraie. Autrement dit, on calcule la probabilité de voir une telle configuration si les éléments étaient répartis selon la distribution définie par $\mathcal{H}_0$.
5. Comparer cette probabilité à un seuil : le seuil est déterminé de manière à garantir qu'en moyenne, il y aurait moins de ε détections dans une image où l'élément serait distribué selon $\mathcal{H}_0$. Si la probabilité de l'événement E est inférieure à ce seuil, cela signifie que l'observation E est suffisamment improbable sous l'hypothèse nulle pour être considérée comme significative.
6. Déclarer E comme étant ε -significatif : si l'événement satisfait le critère ci-dessus, il est déclaré comme " ε -significatif", ce qui signifie qu'il est peu probable qu'il soit dû au hasard ou à une distribution conforme à l'hypothèse nulle.

Cette approche a prouvé son efficacité, notamment pour la localisation d'anomalies dans les images. Tailanian et al. proposent une approche *a contrario* multi-échelle, entièrement non supervisée, pour la détection d'anomalies [142]. La méthode proposée décompose l'image en différentes échelles, puis extrait des caractéristiques significatives à l'aide de l'*Analyse en Composantes Principales* (ACP) [143] sur des patches autour de chaque pixel, ainsi que des caractéristiques obtenues par un ResNet pré-entraîné. La clé de cette approche est l'utilisation de l'analyse *a contrario* pour calculer le "Nombre de Fausses Alarmes" (NFA), qui évalue la probabilité qu'une caractéristique donnée soit anormale sous l'hypothèse nulle. En fusionnant les résultats obtenus à différentes échelles, la méthode produit une carte d'anomalies précise, permettant de détecter efficacement les défauts dans une grande variété de contextes, y compris pour des applications industrielles comme l'inspection de la qualité du cuir dans l'industrie automobile.

Dans le contexte de l'analyse d'images complexes, où les variations intra-classe peuvent être particulièrement prononcées, l'établissement d'une distribution de référence, représentative de la normalité, constitue un défi majeur. En effet, les différences subtiles au sein d'une même catégorie d'images (par exemple, des images de plantes saines présentant des variations dues à l'âge, à la lumière, à l'humidité, etc.), compliquent la tâche d'identifier ce qui définit une "norme" dans les données. La complexité de certaines images rend difficile la distinction entre les variations normales et les anomalies significatives, un prérequis pour la mise en œuvre efficace d'une approche *a contrario*, qui repose sur la comparaison des observations à une hypothèse nulle bien définie.

Face à cette difficulté, l'utilisation des modèles d'explicabilité émerge comme une solution prometteuse. Ces modèles ont pour but de rendre les décisions des modèles d'apprentissage

plus transparentes et compréhensibles pour les humains. En appliquant des techniques d'explicabilité à des modèles entraînés pour faire de la classification d'images par exemple, il est possible d'identifier les régions d'une image qui ont contribué le plus significativement à la prédiction. Cette compréhension approfondie des régions d'intérêt peut alors guider la construction d'une distribution de référence plus précise pour l'approche *a contrario*. En effet, certains modèles d'explicabilité permettent d'extraire les caractéristiques d'importance d'une image par le biais d'une carte de saillance. Ainsi, la représentation visuelle de ces zones peut être utilisée comme base pour la construction de la distribution de référence, en mettant en avant les caractéristiques essentielles des différentes classes.

3.2.3 Explicabilité

Bien qu'il existe de nombreuses approches dans le domaine de l'explicabilité [11], celles qui fournissent une carte de saillance, mettant en évidence les zones d'importance de l'image pour la prédiction se répartissent en 3 catégories [144] :

- Les méthodes basées sur la perturbation, qui modifient les caractéristiques d'entrée et observent l'effet sur la sortie.
- Les méthodes basées sur les gradients, qui utilisent les dérivées de la sortie du modèle par rapport à ses entrées pour évaluer l'importance de chaque caractéristique.
- les méthodes contrastives, qui cherchent à expliquer les prédictions en fournissant des raisons pour lesquelles certaines prédictions ont été faites plutôt que d'autres, offrant une perspective comparative.

3.2.3.1 Méthodes basées sur la perturbation

Les méthodes basées sur la perturbation expliquent comment les prédictions du réseau changent lorsqu'un ou plusieurs pixels d'entrée sont perturbés. Les cartes de sensibilité à l'occlusion [20], sont les premiers travaux sur les modèles d'explicabilité basés sur la perturbation. L'idée est de trouver les zones d'importance de l'image en remplaçant répétitivement différents patches de l'image d'entrée par un carré gris uni et en observant l'influence sur la prédiction. Ribeiro et al. ont proposé *Local Interpretable Model-agnostic Explanations* (LIME) [145], une technique d'explication applicable à tous réseaux de neurones, qui génère un modèle interprétable, fidèle localement, au modèle de prédiction. LIME perturbe aléatoirement les données d'entrée du modèle et observe les changements dans les prédictions pour comprendre les variables d'importance. Zintgraf et al. ont proposé une approche probabiliste pour expliquer les prédictions [146]. Ils utilisent un échantillonnage conditionnel pour estimer la pertinence d'un patch en calculant la différence dans les prédictions du modèle avec et sans le patch d'entrée. Fong et al. ont introduit une approche d'explication basée sur l'édition d'image agnostique du modèle, où la méthode optimise un masque pour expliquer les prédictions du réseau pour la classe cible [147]. Ils ont ensuite généralisé cela en introduisant les "perturbations extrêmes", désignant une perturbation ayant l'effet maximum sur les prédictions du réseau pour une classe, parmi toutes les perturbations d'une zone donnée. Enfin, Petsiuk et al. ont proposé *Randomized Input Sampling for Explanation* (RISE)

[148], une technique pour générer des cartes de saillance pixel par pixel, en appliquant aléatoirement des masques sur l'image d'entrée.

Bien que les méthodes basées sur la perturbation ne nécessitent pas d'accès aux paramètres du modèle et requièrent seulement des passes en avant, elles sont souvent trop coûteuses en calcul et leurs résultats sont spécifiques à un sous-ensemble et à une modification des caractéristiques d'entrée, ne se généralisant pas à toutes les perturbations possibles.

3.2.3.2 Méthodes basées sur le gradient

Les méthodes basées sur le gradient exploitent la propagation avant et la rétropropagation dans les réseaux de neurones pour évaluer l'influence de chaque pixel d'entrée sur la prédiction finale. Le processus consiste à prédire d'abord l'étiquette de classe, puis à utiliser cette prédiction pour rétropropager l'erreur et estimer les attributions d'entrée, aboutissant à la création de cartes de saillance [149].

Le premier modèle d'explicabilité de cette catégorie a été proposé par l'architecture DeconvNet [20]. Cette approche vise à visualiser les stimuli qui activent les cartes de caractéristiques dans les couches convolutionnelles. Pour ce faire, DeconvNet utilise des techniques comme ReLU et le *max pooling* afin de reconstruire l'image d'entrée. Cette reconstruction permet d'identifier les zones de l'image originale qui sont principalement responsables des activations dans le réseau, mettant ainsi en évidence les parties de l'image que le modèle considère comme les plus pertinentes pour sa décision.

Comme cette approche est limitée aux architectures à convolutions, avec ReLU comme fonction d'activation, Simonyan et al. ont proposé les cartes de saillance, applicables à n'importe quelle couche d'un réseau [150]. Cette méthode utilise la valeur absolue de la dérivée partielle du neurone de sortie cible par rapport aux caractéristiques d'entrée, pour trouver les caractéristiques d'une image qui affectent le plus la sortie avec une perturbation minimale.

Inspirés par DeconvNet et les cartes de saillance, Springenberg et al. ont proposé l'algorithme de rétropropagation guidée, qui améliore la visualisation des contributions en ajoutant un signal de guidage des couches supérieures lors de la rétropropagation [151]. Cette approche permet une meilleure localisation des caractéristiques sans nécessiter de commutateurs spécifiques pour suivre le chemin des activations.

Toujours dans l'idée de générer des cartes de saillance, *Global Average Pooling* (GAP) [152] simplifie la structure du réseau et renforce son interprétabilité, en utilisant la moyenne spatiale des cartes de saillance pour la prédiction directe des catégories, agissant comme un régulateur structurel.

Les méthodes telles que la rétropropagation guidée et les réseaux déconvolutionnels sont limitées par le fait qu'elles ignorent les gradients négatifs, ce qui les empêche de mettre en évidence les entrées réduisant la probabilité d'une prédiction spécifique. Pour pallier ce problème, Zhou et al. ont développé la méthode de Carte d'Activation de Classe (CAM) [153]. Cette approche utilise une couche GAP juste avant la couche de sortie, permettant

ainsi de régulariser le réseau, de maintenir sa capacité de localisation jusqu'à la dernière couche convolutionnelle, et d'identifier les régions discriminantes pour chaque classe en une seule propagation avant. Cependant, CAM est limité par la nécessité que la couche GAP soit directement suivie par la couche de sortie, sans couches entièrement connectées intermédiaires, sans quoi l'absence de poids de classe attribués à chaque unité d'activation l'empêche de fonctionner.

Pour surmonter cette contrainte, Selvaraju et al. ont proposé Grad-CAM [154], une méthode qui généralise CAM en utilisant les gradients pour pondérer l'importance de chaque carte de caractéristiques. Grad-CAM fonctionne en effectuant une propagation avant et une rétropropagation partielle pour chaque image, calculant la moyenne des gradients des logits¹ de classe par rapport aux cartes d'activation de la dernière couche convolutionnelle. Ces gradients moyens servent ensuite de poids pour combiner les cartes de caractéristiques, créant ainsi une carte de saillance. Enfin, une fonction ReLU est appliquée pour ne conserver que les caractéristiques contribuant positivement à la prédiction de la classe ciblée. Cette approche permet de générer des cartes de saillance pour diverses architectures de réseaux convolutifs, sans nécessiter de modification structurelle ou de ré-entraînement.

Plusieurs améliorations de Grad-CAM ont vu le jour, comme Guided Grad-CAM [154], qui fusionne la rétropropagation guidée et Grad-CAM pour produire des visualisations à la fois fines et discriminatives de classe. Grad-CAM ++ [155], qui généralise Grad-CAM pour de meilleures explications des images avec multiples occurrences d'objets est applicable à des données spatio-temporelles. Cette approche affine les cartes de saillance en pondérant les gradients des pixels en fonction de leur importance permettant ainsi une localisation plus précise des objets et une meilleure prise en compte des contributions positives et négatives des pixels. L'approche *Integrated Gradients* (IG) [156], quant à elle, calcule l'importance des entrées en variant progressivement l'entrée du réseau et en accumulant les gradients. Enfin, afin d'aborder les limitations liées à l'annulation des gradients négatifs pendant la rétropropagation, Wang et al. ont proposé Score-CAM [157], une méthode qui évalue l'importance des cartes d'activation sans utiliser de gradients, en se concentrant sur la contribution globale des caractéristiques.

Les méthodes classiques de visualisation basées sur les gradients peuvent être trompeuses à cause de discontinuités ou d'une incapacité à détecter des contributions négatives. DeepLIFT [158] aborde ces problèmes en comparant la sortie du modèle à une référence prédéfinie, permettant ainsi d'analyser l'impact de chaque entrée, même en l'absence de gradient. DeepLIFT se décline en deux variantes, *Re-scale* et *RevealCancel*, pour distinguer les effets positifs et négatifs. Afin de réduire la volatilité des gradients, SmoothGrad [159] utilise un noyau gaussien pour lisser les gradients, rendant les cartes de saillance plus facilement interprétables. Enfin, Srinivas et al. ont développé FullGrad [160], une méthode évaluant à la fois l'impact local des pixels individuels et l'impact global des groupes de pixels, offrant une vue d'ensemble plus précise de la contribution des caractéristiques sur la prédiction faite par le modèle.

1. Les logits sont les sorties brutes de la dernière couche d'un réseau de neurones avant l'application de la fonction d'activation finale, représentant les scores non normalisés pour chaque classe.

3.2.3.3 Méthodes contrastives

Les méthodes contrastives se distinguent des méthodes présentées jusqu'ici car elles expliquent non seulement pourquoi un modèle a prédit une certaine classe, mais aussi pourquoi il n'en a pas prédit une autre. Cette double perspective se rapproche de la façon dont les humains prennent des décisions et fournissent des explications.

La méthode d'explication contrastive initiale, *Contrastive Explanation Method* (CEM) [161], a pour objectif de fournir une double explication de la prédiction faite par n'importe quelle architecture profonde. CEM cherche à expliquer les "positifs pertinents", qui correspondent aux variables qui ont permis au modèle de faire sa prédiction, et les "négatifs pertinents", qui correspondent aux variables qui expliquent pourquoi le modèle n'a pas fait d'autres prédictions. Les "positifs pertinents" se réfèrent à des caractéristiques dans les données d'entrée qui, par leur présence, sont essentielles pour que le modèle arrive à une certaine prédiction. Afin d'identifier ces caractéristiques, le CEM utilise une technique d'optimisation pour trouver la version la plus simplifiée de l'entrée qui, tout en étant suffisante pour mener à la même prédiction que l'entrée originale, conserve les caractéristiques cruciales. À l'opposé, les "négatifs pertinents" sont des caractéristiques dont l'absence est nécessaire pour que le modèle fasse sa prédiction. En d'autres termes, si ces caractéristiques étaient présentes, elles pourraient conduire à une prédiction différente. Pour les trouver, CEM cherche à introduire une perturbation interprétable dans les données d'entrée de manière à ce que leur suppression mène à la plus grande baisse de confiance dans la prédiction de la classe actuelle par le modèle. Cela montre quelles caractéristiques, si elles étaient incluses, auraient empêché le modèle de faire la prédiction qu'il a faite. Le cœur de CEM réside dans son processus d'optimisation. Ce processus ajuste les données d'entrée en cherchant une "perturbation interprétable", autrement dit, une modification des données d'entrée qui est significative du point de vue de l'explication mais qui reste minimale. En optimisant une fonction de coût qui minimise la modification des données tout en maximisant l'impact sur la prédiction, il crée une version modifiée de l'entrée. Cette modification, lorsqu'elle est appliquée ou retirée, montre clairement quels aspects de l'entrée influencent la décision du modèle vers ou contre une prédiction spécifique.

Une autre approche, *Contrastive Deep Explanation* (CDeepEx) [162], se distingue de CEM par la nécessité d'accéder aux paramètres du modèle et à son espace latent mais aussi car elle se concentre principalement sur l'identification de ce qui pourrait être considéré comme l'équivalent des "négatifs pertinents" dans le cadre des explications fournies par le modèle CEM. En d'autres termes, CDeepEx cherche à expliquer pourquoi une certaine prédiction n'a pas été faite, en explorant l'espace des modifications minimales nécessaires pour qu'une entrée donnée soit classée différemment. Ce faisant, elle révèle implicitement les limites de la décision du modèle et les conditions sous lesquelles une prédiction différente serait faite, ce qui éclaire sur les caractéristiques dont l'absence a permis la prédiction actuelle. Pour ce faire, CDeepEx apprend un réseau antagoniste en utilisant le même jeu de données que celui sur lequel le classificateur original a été entraîné. Le réseau antagoniste est entraîné pour identifier et exploiter les faiblesses du modèle original, dans le but de comprendre quelles modifications sur les données d'entrée pourraient conduire à une classe différente de celle prédite. Pour une entrée donnée et une classe spécifique que le modèle n'a pas prédite, CDeepEx localise cette entrée dans l'espace latent et cherche la modification minimale

nécessaire pour modifier la classification du modèle vers la classe cible. Cela implique de trouver un point dans l'espace latent tel que, si l'entrée était modifiée pour se rapprocher de ce point, elle serait associée à une classe non prédite. Le résultat de CDeepEx est une modification de l'entrée originale qui rend la prédiction "maximalement ambiguë" entre la classe prédite et la classe non prédite. Cela signifie que l'explication fournie montre comment légèrement changer l'entrée pour qu'elle soit sur le point d'être classée différemment.

Enfin, Prabhushankar et al. ont suggéré d'intégrer les explications contrastives aux méthodes d'attribution basées sur le gradient déjà existantes [163]. Pour un modèle profond, ils comparent la classe prédite à une classe non prédite, en utilisant la dérivée de la perte entre ces deux classes comme base de contraste. Cette analyse est rétropropagée dans le réseau, et les explications sont visualisées via Grad-CAM sous forme de cartes de saillance. Bien que cette méthode nécessite un accès au modèle de prédiction, elle est notablement plus rapide que les méthodes antérieures telles que CEM et CDeepEx, car elle ne requiert qu'une seule rétropropagation et non l'optimisation d'une fonction objectif.

3.2.3.4 Explicabilité pour la télédétection

De nombreux projets liés à l'agriculture intègrent des modèles d'explicabilité dans leur chaîne de traitement. Kim et al. détectent les symptômes du mildiou² de l'oignon [164] en utilisant un modèle faiblement supervisé³ entraîné à partir d'annotations au niveau de l'image, puis localisent les symptômes par seuillage de la carte d'activation du modèle obtenue par CAM.

Kim et al. détectent les maladies des plantes à l'aide d'un modèle non supervisé, combinant des observations avec des flots normalisants, une carte de saillance visuelle et des encodages de position [165].

Bhattarai et al. proposent CountNet [166], un modèle faiblement supervisé qui compte les fruits et les fleurs dans des environnements non structurés. Le modèle apprend à partir d'annotations au niveau de l'image avec le nombre d'objets en entrée, sans spécifier explicitement la nature et l'emplacement de l'objet. Les auteurs utilisent deux modèles d'explicabilité, Score-CAM et l'algorithme de rétropropagation guidée, pour démontrer que le réseau examine effectivement les caractéristiques des fleurs/fruits pour les compter.

Kim et al. forment un modèle de classification pour distinguer trois zones de culture, puis utilisent les cartes de saillance obtenues par CAM pour segmenter l'image et récupérer ces informations sous forme d'annotations, afin d'automatiser une moissonneuse [167].

Enfin, Kierdorf et al. utilisent Grad-CAM [154], OSM [20], et LIME [145] pour récupérer des cartes de saillance dans l'inférence d'un modèle entraîné pour détecter si un chou-fleur est prêt à être récolté ou non [168]. Ils appliquent ensuite une approche de *clustering*⁴ sur

2. Le mildiou est une maladie des plantes causée par des champignons, qui affecte principalement les feuilles, provoquant des taches et parfois un duvet blanc. 3. Un modèle faiblement supervisé apprend d'abord sur peu de données étiquetées ou sur des données dont les annotations ne sont pas parfaitement adaptées à la tâche visée, puis utilise ce savoir pour analyser et exploiter une grande quantité de données non étiquetées.

4. Le clustering est une méthode d'apprentissage non supervisé visant à partitionner un ensemble de données en sous-groupes homogènes, appelés *clusters*, en optimisant une mesure de similarité intra-cluster et de dissimilarité inter-clusters.

les cartes de saillance pour comprendre les caractéristiques d'importance propres à chaque classe. Cette approche est utilisée pour améliorer la performance du modèle et comme aide à la décision pour les agriculteurs qui récoltent des choux-fleurs.

Inspirés par les travaux qui utilisent les cartes de saillance pour comprendre les zones d'intérêt qui différencient les classes d'images, nous proposons une approche, détaillée dans la partie suivante, qui consiste à détecter les anomalies en appliquant un test *a contrario* sur des cartes de saillance. Ces cartes sont obtenues à partir d'un modèle d'explicabilité appliqué à un réseau entraîné sur une tâche prétexte. Autrement dit, nous proposons d'utiliser les cartes de saillance comme données d'entrée d'un modèle de détection d'anomalies.

3.3 Détection d'anomalies par une approche *a contrario*

Nous supposons que lorsqu'un modèle est entraîné sur une tâche et qu'il obtient de bons résultats, il est capable d'extraire les caractéristiques d'importance des images pour faire sa prédiction. Ainsi, si une frontière visuelle existe entre des images normales et anormales, alors les caractéristiques d'importance de ces deux types d'images ne devraient pas être les mêmes. Par conséquent, en utilisant un modèle d'explicabilité pour récupérer les cartes de saillance liées aux prédictions d'un modèle en inférence, qui mettent en évidence les zones importantes des images, nous devrions visuellement observer une différence entre les cartes de saillance des images normales et anormales, révélant ainsi les caractéristiques distinctes de chaque type d'image.

À partir de ce postulat, nous proposons une méthode pour la détection d'anomalies qui se positionne à l'intersection des approches auto-supervisées et des approches d'estimation de la densité. Notre méthode s'appuie sur les caractéristiques d'importance révélées par des cartes de saillance obtenues par un modèle d'explicabilité, appliqué à un modèle entraîné sur une tâche prétexte. À partir de ces cartes de saillance, nous adoptons une approche *a contrario* pour évaluer la probabilité qu'une image soit normale, en comparant les caractéristiques d'importance observées à une distribution de référence, construite à partir des normes identifiées au sein des données. Cette combinaison permet de bénéficier de la richesse des représentations apprises de manière auto-supervisée, tout en appliquant une analyse statistique à la phase de détection, faisant de notre méthode une proposition permettant d'identifier les écarts subtils entre les données normales et anormales.

3.3.1 Définition du problème

Soit x une image carrée de taille $(L \times L)$ pixels. Notre objectif est de détecter si x est une anomalie en utilisant une approche *a contrario* basée sur des cartes de saillance, obtenues par un modèle d'explicabilité. La Figure 3.3 montre les différentes étapes de notre méthode. Notre approche de détection d'anomalies consiste à calculer une distribution de référence à partir des cartes de saillance d'images normales, puis à calculer un score pour chaque nouvelle image, représentant la proximité avec la distribution de référence (voir Section 3.3.2). Nous appliquons ensuite un test *a contrario* à ce score pour déterminer si l'image est normale ou

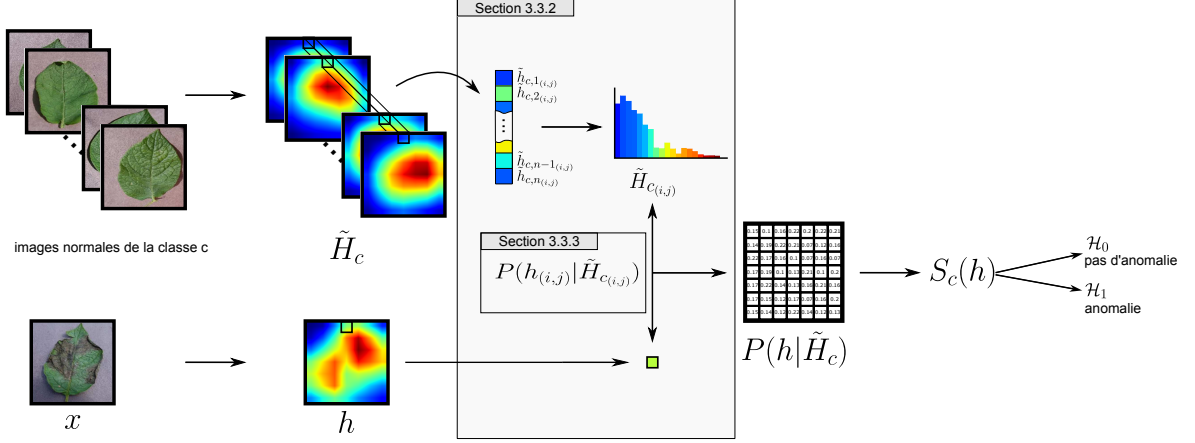


FIGURE 3.3 – Schéma de la méthodologie de notre approche. Étant donné un ensemble d'images normales de la classe c , nous récupérons les cartes de saillance associées pour établir la distribution de référence $\tilde{H}_c$. Pour une nouvelle image x de la classe c , nous récupérons la carte de saillance h associée et calculons, pour chaque pixel, la probabilité qu'il suive la distribution de référence. Nous moyennons ensuite ces probabilités pour obtenir le score de normalité $S_c(h)$, qui sera comparé à un seuil pour savoir dans quelle hypothèse du test *a contrario* l'image se trouve.

anormale.

L'approche *a contrario* que nous proposons est basée sur deux hypothèses :

- hypothèse nulle ($\mathcal{H}_0$) : "L'image est normale". Cette hypothèse suppose que les pixels d'une image se conforment à la distribution établie pour les images normales.
- hypothèse alternative ($\mathcal{H}_1$) : "L'image est anormale". Cette hypothèse correspond au cas où la distribution des pixels de l'image s'écarte de celle des images normales, indiquant une possible anomalie.

Pour chaque image, nous calculons un score de normalité qui, par comparaison à un seuil défini, détermine si l'image appartient à l'hypothèse nulle ($\mathcal{H}_0$), c'est-à-dire qu'elle suit la distribution typique des images normales - ou si elle relève de l'hypothèse alternative ($\mathcal{H}_1$) et présente donc des caractéristiques anormales.

3.3.2 Calcul d'un score de normalité

Pour récupérer les cartes de saillance, nous entraînons un modèle prédictif sur une tâche auxiliaire spécifique au jeu de données (voir Section 3.4.2.1). Nous proposons d'appliquer un modèle d'explicabilité (cf. Section 3.2.3), Grad-CAM [154], qui met en évidence les zones d'importance de l'image à travers des cartes de saillance, indiquant les régions qui ont contribué à la prédiction faite par le modèle. Grad-CAM a été choisi pour son bon compromis entre complexité et interprétabilité, offrant une mise en œuvre relativement simple tout en produisant des explications visuelles claires et accessibles.

Après l'application de Grad-CAM nous obtenons la carte de saillance h qui est une matrice $h_{(i,j)}$ pour $i, j \in [1, L]^2$. Cette carte de saillance fournit des informations sur les

zones d'importance dans l'image pour la tâche auxiliaire. Notre hypothèse est que ces zones sont différentes pour les observations normales et anormales.

Nous souhaitons calculer un score pour déterminer si une nouvelle image x est normale en estimant la probabilité que chaque pixel de sa carte de saillance h soit tiré de la distribution des pixels (voir Section 3.3.3) obtenue à partir des cartes de saillance d'images normales de la même classe. Notons que, indépendamment de la tâche prétexte choisie, qui doit être supervisée et donc nécessiter des étiquettes, nous utilisons ces étiquettes pour définir des classes spécifiques. Pour chaque classe, nous construisons ensuite des distributions de référence dédiées.

Ainsi, pour une image de classe c , pour chaque pixel $h_{(i,j)}$, nous calculons la probabilité qu'il soit tiré de la distribution de référence notée :

$$\tilde{H}_{c_{(i,j)}} = [\tilde{h}_{c,1_{(i,j)}}, \tilde{h}_{c,2_{(i,j)}}, \tilde{h}_{c,3_{(i,j)}}, \dots, \tilde{h}_{c,n_{(i,j)}}], \quad (3.1)$$

qui représente les valeurs des pixels $\tilde{h}_{c_{(i,j)}}$ des n cartes de saillance obtenues à partir des images normales de la classe c . Ainsi, cette distribution de référence représente l'hypothèse nulle $\mathcal{H}_0$.

Nous calculons $S_c(h)$, le score indiquant la probabilité que h , la carte de saillance d'une image de la classe c , suive la distribution de référence de cette même classe. Le calcul de $S_c(h)$ se fait en moyennant les probabilités que les pixels $h_{(i,j)}$ suivent la distribution des pixels aux positions i, j des images normales de la classe c :

$$S_c(h) = \frac{\sum_{i,j=1}^L P(h_{(i,j)} | \tilde{H}_{c_{(i,j)}})}{L^2}. \quad (3.2)$$

Ainsi, $S_c(h)$ représente le score de normalité de la carte de saillance h . Ce score $S_c(h)$ peut ensuite être seuillé pour déterminer si l'image x suit l'hypothèse nulle $\mathcal{H}_0$ ou l'hypothèse alternative $\mathcal{H}_1$.

Notons que le score $S_c(h)$ n'est pas une probabilité, mais plutôt une moyenne arithmétique de probabilités, car nous ne pouvons pas supposer que les pixels sont indépendants les uns des autres. En effet, les pixels d'une image dite "figurative", c'est-à-dire qui représente quelque chose d'identifiable visuellement, sont spatialement corrélés, chaque pixel étant influencé par les pixels adjacents pour former des structures cohérentes comme les contours, les textures et les objets.

3.3.3 Probabilité d'appartenir à la distribution de référence

Pour calculer $S_c(h)$ (Équation 3.2), nous devons calculer la probabilité qu'un pixel d'une image appartienne à la distribution de référence du pixel à la même position dans les images normales de la même classe : $P(h_{(i,j)} | \tilde{H}_{c_{(i,j)}})$.

Nous estimons cette probabilité à partir de la distribution de référence des cartes de saillance $\tilde{H}_{c_{(i,j)}}$. À partir de la carte de saillance h d'une nouvelle observation x , nous calculons $P(h_{(i,j)} | \tilde{H}_{c_{(i,j)}})$, la probabilité estimée que le point $h_{(i,j)}$ soit tiré de la distribution de $\tilde{H}_{c_{(i,j)}}$ à l'aide du processus suivant :

- a. **Calcul de l'histogramme** : Soit b le nombre de cases uniformément réparties dans l'intervalle $[0,1]$ (qui est la plage de valeurs des cartes de saillance). Pour $1 \leq z \leq b$, B_z représente la z -ième case, c'est à dire $B_z = [\frac{z-1}{b}, \frac{z}{b}[$ pour $1 \leq z \leq b-1$, et $B_b = [\frac{b-1}{b}, 1]$.

Soit d_z la densité de probabilité dans la case B_z :

$$d_z = \frac{b}{n} \sum_{k=1}^n I_{B_z}(\tilde{h}_{c,k(i,j)}) \quad (3.3)$$

où n est le nombre total d'observations et $I_{B_z}(\tilde{h}_{c,k(i,j)})$ est une fonction qui vaut 1 si $\tilde{h}_{c,k(i,j)}$ se trouve dans l'emplacement B_z , et 0 sinon.

- b. **Estimation de la probabilité** :

Soit z l'indice de la case dans laquelle tombe le point $h_{(i,j)} : z = \lfloor h_{(i,j)} \times b \rfloor + 1$. La probabilité estimée que le point $h_{(i,j)}$ appartienne à la distribution $\tilde{H}_{c(i,j)}$ est alors $P(h_{(i,j)} | \tilde{H}_{c(i,j)}) = d_z \cdot \frac{1}{b}$

Dans la partie suivante, nous détaillons comment nous appliquons notre méthode à un cas d'étude de détection de plantes malades. Nous présentons les jeux de données utilisés, les tâches auxiliaires choisies ainsi que les spécificités liées à l'application de notre approche sur les différents jeux de données.

3.4 Expérimentations

3.4.1 Jeux de données

Pour évaluer les performances de la méthode proposée, nous réalisons une évaluation sur trois jeux de données de plantes, décrits ci-dessous.

3.4.1.1 Jeu de données *GrowliFlower*

Le jeu de données *GrowliFlower* [169] a été développé par un groupe de chercheurs de l'Université de Bonn, en Allemagne, et du *Forschungszentrum Jülich GmbH*, en collaboration avec *JB Hyperspectral Devices GmbH*. Ce projet vise à soutenir l'innovation dans les domaines de l'agriculture de précision et de la télédétection, en se concentrant particulièrement sur la culture du chou-fleur. Créé pour répondre aux besoins d'automatisation et d'optimisation de la gestion des cultures, *GrowliFlower* est un ensemble de données géoréférencées et basé sur des séries temporelles d'images UAV (*Unmanned Aerial Vehicle*). Il propose des orthophotos RVB (Rouge, Vert, Bleu) et multispectrales de haute résolution capturées sur deux champs de chou-fleur pendant les années 2020 et 2021. La résolution au sol pour les orthophotos RVB du champ 1 est de 1,65 mm/px, respectivement pour la largeur et la hauteur des pixels. La résolution au sol pour le champ 2 est de 3,10 mm/px. De plus, le jeu de données contient des orthophotos multispectrales pour le champ 2 avec une résolution au sol de 2,5 cm par largeur et longueur de pixel.

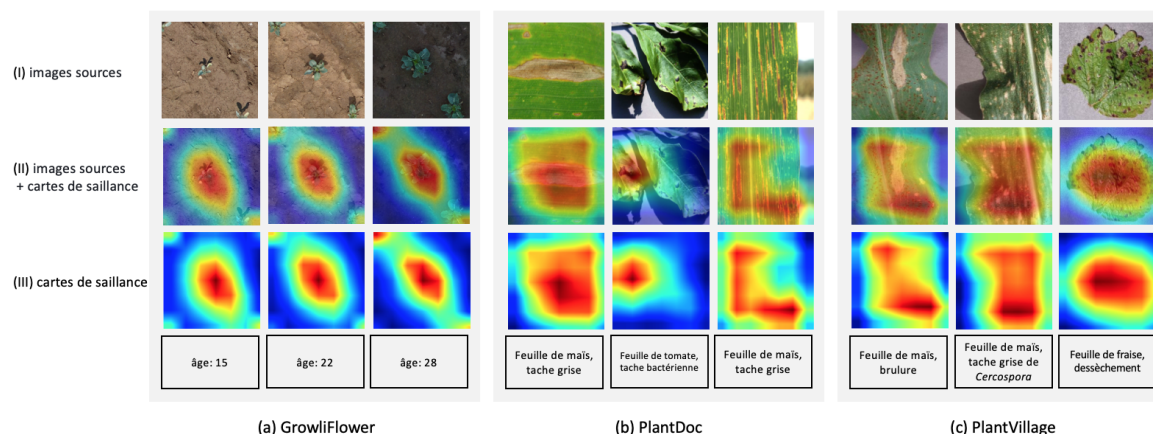


FIGURE 3.4 – Exemples d’images des jeux de données *GrowliFlower*, *PlantDoc* et *PlantVillage*, utilisés dans notre étude. (a) images d’origine, (b) superposition des images d’origine et des cartes de saillance obtenues par Grad-CAM, (c) cartes de saillance obtenues par Grad-CAM.

Le jeu de données inclut environ 14 000 coordonnées de plantes individuelles et des informations détaillées sur les traits phénotypiques de centaines de plantes. De plus, il offre des annotations précises pour la segmentation d’instances de feuilles et de plantes, ainsi que des images pré- et post-défoliation pour évaluer la taille des têtes de chou-fleur. *GrowliFlower* est conçu pour faciliter le développement des modèles d’apprentissage pour l’analyse de la croissance des plantes et la dérivation de traits phénotypiques, contribuant ainsi à l’avancement des technologies agricoles intelligentes.

Dans notre étude, nous utilisons un sous-ensemble du jeu de données, appelé *GrowliflowerR*, qui contient des orthophotos RVB et des caractéristiques phénotypiques collectées sur 740 plantes, y compris des informations sur le fait que les choux-fleurs soient affectés par une maladie ou un défaut. Les données in-situ proposent la définition de 19 défauts (voir Table 3.1), mais en pratique les défauts présentent des variantes, expliquées littéralement dans les fichiers correspondant à chaque champ. Pour les besoins de l’étude, puisque notre objectif est de détecter les anomalies mais pas de les caractériser, nous considérons tous les défauts comme un type d’anomalie (voir quelques exemples d’anomalies dans les Figures 3.1 et 3.5).

3.4.1.2 Jeu de données *PlantDoc*

Le jeu de données *PlantDoc* [170] a été développé par des chercheurs de l’*Indian Institute of Technology Gandhinagar* afin de combler le manque de données disponibles pour la détection des maladies des plantes hors des environnements de laboratoire, permettant une application plus réaliste des modèles de vision par ordinateur. Ce jeu de données contient 2 598 images de feuilles prises dans leur environnement naturel, de 13 espèces de plantes différentes et comprend 27 catégories, incluant des feuilles saines (en bleu dans la Table 3.2) et des feuilles ayant diverses maladies (en noir dans la Table 3.2). Les images ont été recueillies via du *scrapping*⁵ sur des moteurs de recherche d’images, principalement sur *Google Images*

5. Le *scrapping* (ou *web scrapping*) est une technique d’extraction automatisée de données à partir de sites web.

Observations	
— Plante	— Sans feuilles
— Pas de plante	— Feuilles anciennes
— Plante couchée	— Feuilles jaunâtres
— Plante entière	— Feuilles rougeâtres
— 2 plantes	— Feuilles vert pâle
— Bourgeon aveugle	— Feuilles vert jaune pâle
— Planté trop profondément	— Croissance retardée avec de nombreux rejets
— Pucerons présents	— Dommages aux feuilles causés par le tracteur
— Pucés de charbon présentes	
— Mouches présentes	
— Feuille(s)	

TABLE 3.1 – Observations in-situ des traits phénotypiques de développement et de stress des plantes du jeu de données *GrowliFlower*.

et *Ecosia*, puis annotées pour la classification des maladies. Cette approche permet de traiter des images avec des arrière-plans naturels et des conditions d'éclairage variables, reflétant les conditions réelles des agriculteurs, en particulier en Inde où beaucoup utilisent des appareils mobiles bas de gamme.

3.4.1.3 Jeu de données *PlantVillage*

Le jeu de données *PlantVillage* [171] a été développé par des chercheurs de l'université de Pennsylvanie et de l'École Polytechnique Fédérale de Lausanne (EPFL) pour améliorer le diagnostic des maladies des plantes à travers l'apprentissage automatique et le *crowdsourcing*⁶. Les images de *PlantVillage* ont été prises majoritairement dans des stations de recherche expérimentale aux États-Unis, utilisant un protocole standardisé où chaque feuille était placée sur un fond gris ou noir et photographiée sous différents angles pour capturer une gamme complète de caractéristiques visuelles sous diverses conditions de lumière. Le jeu de données a été conçu pour être évolutif et s'enrichir au fil du temps. Au moment où nous avons récupéré le jeu de données, celui ci comprenait 54 305 images de feuilles appartenant à 14 types de plantes différents, incluant des feuilles saines (en bleu dans la Table 3.3) et des feuilles avec diverses maladies (en noir dans la Table 3.3).

Notons que les jeux de données *PlantDoc* et *PlantVillage* ont été développés pour la détection des maladies, qui dans notre cas sont considérées comme des anomalies. Par conséquent, ces jeux de données incluent une majorité d'images présentant une anomalie. En effet, *PlantDoc* et *PlantVillage* contiennent respectivement 67% et 72% d'images contenant

6. Le *crowdsourcing* est une pratique qui consiste à faire appel à un grand nombre de personnes, généralement via Internet, pour accomplir des tâches, résoudre des problèmes ou contribuer à un projet.

Classes du jeu de données <i>PlantDoc</i>	
— Pomme - Saine	— Pomme de terre - Mildiou
— Pomme - Rouille	— Framboise - Saine
— Pomme - Tavelure	— Soja - Saine
— Poivron - Saine	— Courge - Oïdium
— Poivron - Tache bactérienne	— Fraise - Saine
— Myrtille - Saine	— Tomate - Saine
— Cerise - Saine	— Tomate - Tache bactérienne
— Maïs - Brûlure	— Tomate - Alternariose
— Maïs - Tache grise	— Tomate - Mildiou
— Maïs - Rouille	— Tomate - Moisissure
— Raisin - Saine	— Tomate - Virus de la mosaïque
— Raisin - Pourriture noire	— Tomate - Septoriose
— Pêche - Saine	— Tomate - Virus jaune
— Pomme de terre - Alternariose	

TABLE 3.2 – Liste des classes du jeu de données *PlantDoc*. En bleu les classes normales, en noir les anormales. Pour chaque tuple, le premier élément correspond au fruit ou au légume de la feuille concernée, et le deuxième élément à l'état de la feuille.

une anomalie tandis que *GrowliFlower* en contient 20%.

Finalement, comme pour le jeu de données *GrowliFlower*, pour chaque classe des jeux de données *PlantDoc* et *PlantVillage*, nous considérons toutes les variantes d'anomalies comme une anomalie.

3.4.2 Application du modèle proposé

Pour détecter des images anormales, nous entraînons un modèle sur une tâche auxiliaire. À partir de la prédiction, nous récupérons les cartes de saillance obtenues avec un modèle d'explicabilité (Grad-CAM [154]), mettant en évidence les zones d'importance dans l'image. Nous travaillons ensuite avec ces cartes de saillance et appliquons la méthode décrite dans la Section 3.3. Dans cette partie, nous décrivons les tâches auxiliaires spécifiques aux jeux de données et donnons les détails expérimentaux pour l'application de notre modèle.

3.4.2.1 Tâche Auxiliaire

Pour le jeu de données *GrowliFlower*, nous choisissons comme tâche auxiliaire la prédiction de l'âge des choux-fleurs (voir la Figure 3.4 (a) qui représente une série d'images

Classes de feuilles du jeu de données <i>PlantVillage</i>	
— Pomme - Saine	— Pomme de terre - Saine
— Pomme - Tavelure du pommier	— Pomme de terre - Alternariose
— Pomme - Pourriture noire	— Pomme de terre - Mildiou
— Pomme - Rouille du pommier	— Framboise - Saine
— Myrtille - Saine	— Soja - Saine
— Cerise (y compris griotte) - Saine	— Courge - Oïdium
— Cerise (y compris griotte) - Oïdium	— Fraise - Saine
— Maïs - Saine	— Fraise - Brûlure des feuilles
— Maïs - Tache grise de Cercospora	— Tomate - Saine
— Maïs - Rouille commune	— Tomate - Tache bactérienne
— Maïs - Helminthosporiose	— Tomate - Alternariose
— Raisin - Saine	— Tomate - Mildiou
— Raisin - Pourriture noire	— Tomate - Moisissure des feuilles
— Raisin - Esca (Noir de l'esca)	— Tomate - Septoriose
— Raisin - Brûlure des feuilles (Tache d'Isariopsis)	— Tomate - Acariens tétranyques (Tétranyque tisserand)
— Orange - Huanglongbing (Verdissement des agrumes)	— Tomate - Tache cible
— Pêche - Saine	— Tomate - Virus de la jaunisse de la tomate
— Pêche - Tache bactérienne	— Tomate - Virus de la mosaïque de la tomate
— Poivron - Tache bactérienne	
— Poivron - Saine	

TABLE 3.3 – Liste des classes du jeu de données *PlantVillage*. En bleu les classes normales, en noir les anormales. Pour chaque tuple, le premier élément correspond au fruit ou au légume de la feuille concernée, et le deuxième élément à l'état de la feuille.

avec les âges associés), défini par la différence en jours entre la date à laquelle la photo a été prise et la date de plantation du chou-fleur (de 1 à 93 jours). Nous avons divisé aléatoirement le jeu de données en ensembles *Entraînement*, *Validation* et *Test*, contenant respectivement 60%, 30% et 10% des séries temporelles d'images de chou-fleur.

Pour le jeu de données *PlantDoc*, la tâche auxiliaire consiste à classer les images par type de feuille. Les classes possibles de feuilles sont listées dans la Figure 3.2 mais nous regroupons les feuilles de même espèce (du même fruit/légume) dans une même classe, ce qui revient à un total de 13 classes. Le jeu de données est divisé afin de former le modèle sur

7. La fonction de perte utilisée pour *GrowliFlower* est différente de celle utilisée pour *PlantDoc* et *PlantVillage* car la tâche d'entraînement est différente (régression vs classification)

Jeu de données	<i>GrowliFlower</i>	<i>PlantDoc</i>	<i>PlantVillage</i>
Tâche	Régression	Classification	Classification
Modèle	ResNet18	ResNet18	ResNet18
# époques	200	100	100
Taux d'apprentissage	0.1	0.01	0.001
Fonction d'optimisation	SGD	SGD	SGD
Fonction de perte ⁷	Huber	Entropie croisée	Entropie croisée

TABLE 3.4 – Paramétrisation des modèles d'apprentissage pour la tâche auxiliaire en fonction du jeu de données.

70% des images, de le valider sur 20% et de le tester sur 10%.

Comme pour le jeu de données *PlantDoc*, la tâche auxiliaire utilisée pour le jeu de données ***PlantVillage*** est la classification de type de feuille. Les classes sont les mêmes que pour le jeu de données *PlantDoc*, avec l'ajout de la classe "orange". Nous divisons le jeu de données en ensembles d'entraînement, de validation et de test avec respectivement 65%, 15% et 20% du jeu de données.

Pour les trois jeux de données, nous affinons un ResNet18 pré-entraîné sur *ImageNet*, conformément à la méthode employée par les auteurs du jeu de données *PlantDoc* [170], qui suivent la même méthode pour améliorer les performances.

Nous évaluons les modèles de classification de feuilles (entraînés sur *PlantDoc* et *PlantVillage*) avec la mesure de l'exactitude (*accuracy*), qui représente la proportion de prédictions correctes parmi toutes les prédictions faites par le modèle. Le modèle de régression, qui correspond à la prédiction de l'âge des choux-fleurs du jeu de données *GrowliFlower*, est évalué à l'aide de l'erreur quadratique moyenne (*MSE*) et le coefficient de détermination (R^2). *MSE* calcule la moyenne des écarts au carré entre les valeurs prédites et les valeurs réelles, et R^2 évalue la proportion de la variance dans les données réelles expliquée par le modèle, indiquant dans quelle mesure le modèle s'adapte aux données observées. Les formules pour ces mesures sont les suivantes :

$$\text{Exactitude} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}} \quad (3.4)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.5)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (3.6)$$

où y_i est la valeur réelle, $\hat{y}_i$ est la valeur prédite, $\bar{y}$ est la moyenne des valeurs réelles, et n est le nombre total d'observations.

La Table 3.4 détaille les paramètres d'entraînement utilisés pour chaque modèle sur sa

tâche auxiliaire. Ces paramètres ont été choisis pour maximiser les performances sur le jeu de données de validation spécifique à chaque modèle. Les performances des modèles sur les jeux de test correspondants sont présentées dans la Table 3.5.

3.4.2.2 Détection d'anomalies

Une fois le modèle entraîné sur la tâche auxiliaire, nous appliquons Grad-CAM pour obtenir les cartes de saillance associées aux prédictions. Ces cartes de saillance, superposées aux images originales, sont montrées dans la II^e rangée de la Figure 3.4 et seules dans la III^e rangée de la Figure 3.4.

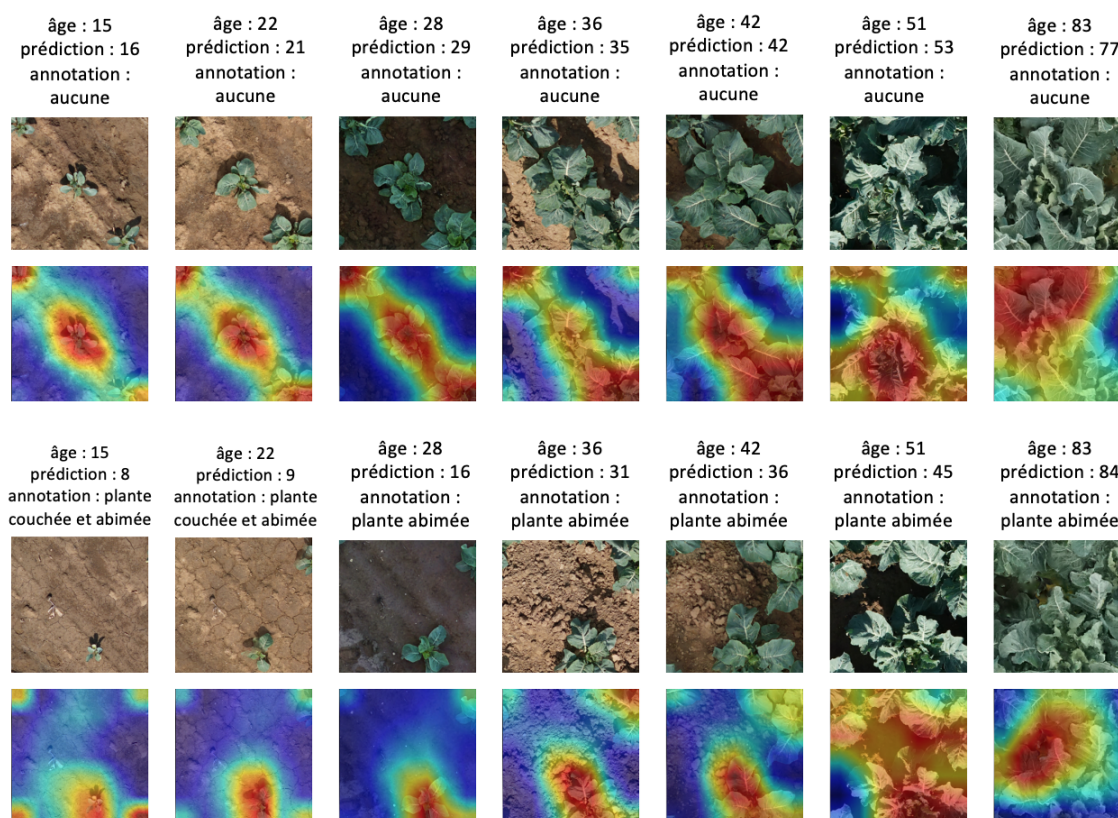


FIGURE 3.5 – Exemples de séries temporelles d'images de deux choux-fleurs du jeu de données *GrowliFlower*, superposées, ou non, avec les cartes de saillance. Pour chaque pas de temps, sont indiqués : l'âge du chou-fleur, l'âge prédit par le modèle et les annotations associées indiquant la présence/nature d'une anomalie. "Aucune" signifie qu'il n'y a pas d'anomalie associée à la plante.

Nous recueillons toutes les observations normales des jeux de données d'entraînement et de validation pour définir nos distributions de référence. Pour les jeux de données *PlantDoc* et *PlantVillage*, nous testons le modèle sur les données de test.

Pour le jeu de données *GrowliFlower*, qui n'inclut pas beaucoup de données anormales, nous testons notre approche sur un jeu de données comprenant les données de test ainsi que toutes les données anormales des jeux de données d'entraînement et de validation (c'est-à-dire les données non utilisées pour la construction de la distribution de référence). De plus,

Jeu de données	<i>GrowliFlower</i>	<i>PlantDoc</i>	<i>PlantVillage</i>
Tâche auxiliaire	Régression	Classification	Classification
Performances	$MSE = 30.76$ $R^2 = 0.9$	$exactitude = 0.78$	$exactitude = 0.99$

TABLE 3.5 – Performances des modèles d’apprentissage pour la tâche auxiliaire en fonction du jeu de données.

comme les données de *GrowliFlower* sont des séries temporelles d’images, nous calculons un score de normalité par série, en moyennant les scores $S_c(h)$ des images individuelles à différents pas de temps. Notons que dans ce jeu de données, il n’y a qu’une seule classe de plante (le chou-fleur). Dans ce cas, la classe est considérée comme l’âge de la plante (c’est-à-dire le pas de temps dans la série temporelle). Nous calculons donc une distribution de référence par âge.

3.5 Résultats et discussions

3.5.1 Tâche auxiliaire

L’idée de notre méthode est d’utiliser des cartes de saillance comme données d’entrée pour l’étude *a contrario* de détection d’anomalies. Pour cela, le prérequis est d’avoir un modèle de prédiction suffisamment performant sur sa tâche auxiliaire pour exploiter sa capacité à extraire les caractéristiques d’importance des images.

La Table 3.5 montre que le modèle de classification des feuilles, entraîné sur le jeu de données *PlantDoc*, est moins performant que celui entraîné sur le jeu de données *PlantVillage*. Cela est probablement dû au fait que contrairement aux images du jeu de données *PlantDoc*, qui représentent les feuilles dans des conditions réelles (dans des champs), les images du jeu de données *PlantVillage* ont été prises en laboratoire. Par conséquent, le modèle n’a pas besoin d’apprendre le contexte dans lequel se trouve la feuille, mais peut se concentrer uniquement sur sa forme, sa couleur et sa texture, ce qui peut faciliter la tâche de classification.

Pour le jeu de données *GrowliFlower*, nous travaillons avec des séries temporelles d’images. Par conséquent, nous avons d’abord essayé d’entraîner un modèle hybride (ResNet18 suivi d’un LSTM [35]) pour la tâche auxiliaire (prédiction de l’âge). Nous supposons qu’un tel modèle permettrait de prendre en compte la temporalité de l’évolution du chou-fleur à partir des vecteurs de caractéristiques obtenus par ResNet18. Cependant, malgré la prise en compte explicite des données séquentielles, il sous-performait dans notre contexte. Comme les séquences d’âge sont presque toujours identiques, le modèle apprenait sur la base de ces séquences plutôt que des images elles-mêmes, résultant en une très faible performance (MSE élevée).

3.5.2 Comparaison avec d'autres méthodes

Nous comparons notre approche avec deux autres approches de détection d'anomalies, **f-AnoGAN** [172] et **One-Class Support Vector Machines (OCSVM)** [121].

f-AnoGAN est un modèle basé sur la reconstruction auto-supervisée, qui apprend à reconstruire des images normales. Il détecte les anomalies en calculant un score combinant l'erreur résiduelle des caractéristiques du discriminateur et l'erreur de reconstruction de l'image. Ainsi, une image se voit attribuer un score élevé si elle diverge significativement de la reconstruction attendue, suggérant qu'il s'agit d'une anomalie. Les observations sont ensuite classées comme anormales si leur score dépasse un seuil défini.

OCSVM quant à lui est un modèle non supervisé de la famille de classification à une classe, qui apprend la frontière des points de données normaux et identifie comme anomalies les données situées en dehors de cette frontière.

Nous entraînons le modèle f-AnoGAN avec les mêmes paramètres que ceux choisis par les auteurs de l'article original [172]. Pour OCSVM, après avoir comparé différents paramètres, nous avons choisi d'utiliser un noyau gaussien avec les paramètres $\gamma = 0.001$ et $\nu = 0.03$.

Pour le jeu de données *GrowliFlower*, nous agrégeons les résultats obtenus à chaque pas de temps de la série d'images pour obtenir un seul résultat par série. En effet, le modèle f-AnoGAN calcule un score par image, donc nous moyennons les scores pour obtenir un score par chou-fleur. Concernant l'application de OCSVM, comme ce modèle ne donne pas un score de normalité, mais classe les données selon leur position relative à une frontière de décision, nous prenons la classe la plus fréquemment attribuée aux différentes étapes temporelles de l'évolution d'un chou-fleur.

Nous présentons les résultats quantitatifs dans la Table 3.6. Nous évaluons les modèles de détection d'anomalies avec des mesures d'évaluation basées sur la précision et le rappel dans un contexte de classification binaire, où les **vrais positifs (VP)** correspondent aux instances correctement identifiées comme positives par le modèle ; les **faux positifs (FP)** sont les instances incorrectement identifiées comme positives alors qu'elles sont négatives ; et les **faux négatifs (FN)** sont les instances positives qui ont été incorrectement classées comme négatives par le modèle. La **précision** et le **rappel** sont définis comme suit :

$$\text{Précision} = \frac{VP}{VP + FP} \quad (3.7)$$

$$\text{Rappel} = \frac{VP}{VP + FN} \quad (3.8)$$

Sur la base de ces définitions, nous utilisons les métriques suivantes pour évaluer les modèles :

- Le **score F1**, qui est la moyenne harmonique de la précision et du rappel, calculé par la formule :

$$F1 = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

Jeu de données	Modèle	AUC	Score $F1$	Précision	Rappel
<i>GrowliFlower</i>	Modèle proposé	0.76	0.85	0.74	1.0
	f-AnoGAN	0.63	0.85	0.74	1.0
	OCSVM	X	0.72	0.69	0.76
<i>PlantDoc</i>	Modèle proposé	0.75	0.8	0.67	0.99
	f-AnoGAN	0.61	0.76	0.62	1.0
	OCSVM	X	0.55	1.0	0.38
<i>PlantVillage</i>	Modèle proposé	0.79	0.8	0.68	0.99
	f-AnoGAN	0.68	0.84	0.72	1.0
	OCSVM	X	0.84	0.99	0.73

TABLE 3.6 – Résultats de la détection d’anomalies. Nous rapportons l’aire sous la courbe (AUC), le score $F1$, la précision et le rappel de chaque modèle sur chaque jeu de données utilisé.

- L’aire sous la courbe précision/rappel (AUC), qui représente la capacité du modèle à distinguer entre les classes positives et négatives à différents seuils de classification.

La Figure 3.6 montre les courbes précision/rappel de notre approche et de f-AnoGAN. Nous avons volontairement omis les résultats de OCSVM sur ces courbes car la frontière de décision qu’il permet d’obtenir ne peut pas être ajustée pour équilibrer la précision et le rappel. Ainsi, la précision et le rappel des résultats obtenus avec OCSVM représentent les coordonnées d’un point, présentés dans la Table 3.6.

La Figure 3.6 montre que notre modèle atteint une précision parfaite pour des valeurs de rappel allant jusqu’à 10% sur le jeu de données *GrowliFlower* et 18% sur le jeu de données *PlantDoc*, permettant l’identification précise d’une partie des anomalies. Autrement dit, il existe un petit seuil en dessous duquel toutes les images sont correctement identifiées comme anormales. En contraste, le modèle f-AnoGAN attribue systématiquement des scores bas. Il est à noter que cette distribution des scores ne permet pas de faire une distinction claire entre les images normales et anormales, entraînant une baisse de la précision et du rappel avec l’augmentation du seuil.

De plus, la Table 3.6 montre que le score $F1$ de notre modèle est identique à celui de f-AnoGAN sur le jeu de données *GrowliFlower*, mais est supérieur à ceux de f-AnoGAN et OCSVM sur le jeu de données *PlantDoc*. Le meilleur score $F1$ est obtenu pour une valeur de rappel proche ou égale à 1. Cependant, notre approche présente une valeur d’AUC supérieure à celle de f-AnoGAN sur les trois jeux de données, une performance attribuable à la haute précision obtenue pour de faibles valeurs de rappel.

Les résultats obtenus par OCSVM montrent une performance inférieure en termes de précision et de rappel pour les jeux de données *GrowliFlower* et *PlantDoc*. En effet, le modèle classe toutes les images du jeu de données *PlantDoc* comme anormales. Ces observations soulignent l’avantage d’un seuil ajustable, particulièrement dans des contextes où les anomalies sont difficiles à distinguer visuellement.

Enfin, pour le jeu de données *PlantVillage*, notre modèle est moins performant que

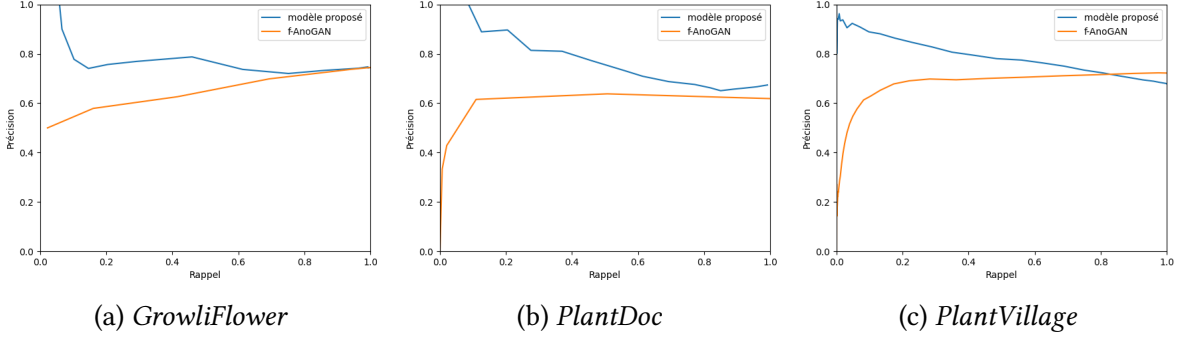


FIGURE 3.6 – Courbes de précision/rappel de notre modèle et de f-AnoGAN sur les jeux de données *GrowliFlower*, *PlantDoc* et *PlantVillage*.

f-AnoGAN et OCSVM en termes de score $F1$. Nous pensons que cela est dû au fait que dans ce jeu de données, les images de feuilles ne sont pas prises dans leur environnement naturel tandis que la valeur ajoutée de notre modèle est d'être moins sensible aux variations liées à l'environnement que les autres modèles. En effet, dans la Figure 3.5, on remarque que les zones d'importance, révélées par les cartes de saillance, suivent bien la position des plantes. Ainsi, bien que le modèle entraîné sur la tâche auxiliaire montre de bonnes performances sur le jeu de données *PlantVillage*, la faible variabilité dans la position des feuilles et le contexte uniforme dans lequel les images sont prises résultent en une faible variabilité dans les cartes de saillance générées, pénalisant notre approche de détection d'anomalies.

3.5.3 Limitations de l'approche proposée

Notre étude est basée sur l'hypothèse que les caractéristiques d'importance ne sont pas les mêmes pour les images normales et anormales, et que cette information peut être révélée par les cartes de saillance obtenues par Grad-CAM (voir les cartes de saillance d'un chou-fleur sain dans la première série d'images de la Figure 3.5 et celles d'un chou-fleur avec anomalies dans la deuxième série). Lorsque nous travaillons avec les séries temporelles d'images du jeu de données *GrowliFlower*, nous calculons un score pour chaque chou-fleur afin de déterminer si il est normal. Pour ce faire, nous moyennons les scores $S_t(h)$ pour t dans $[1, T]$, avec T la taille de la série, représentant dans quelle mesure l'image à l'âge t suit la distribution de référence des images saines au même âge. Ainsi, dans un cas exagéré, si $S_t(h) = 1$, l'image est normale, et si $S_t(h) = 0$, elle est anormale. Étant donné que nous calculons le score final ($\frac{\sum_{t=1}^T S_t(h)}{T}$) comme une moyenne des scores à chaque pas de temps, et qu'il est possible pour un chou-fleur d'être anormal à seulement un pas de temps, la valeur du score peut être proche de celle d'un chou-fleur normal. Ainsi, notre approche est plus adaptée pour détecter des choux-fleurs qui présentent des anomalies à plusieurs pas de temps.

De plus, les expériences menées sur différents jeux de données, dont deux composés de feuilles de fruits/légumes, montrent que la valeur ajoutée de notre modèle est de détecter des anomalies liées à la position. En effet, le score de normalité, basé sur la probabilité que la carte de saillance d'une nouvelle observation suive la distribution des cartes de saillance des

images normales, sera d'autant plus significatif si les zones d'importance de l'image sont positionnées différemment que dans la distribution de référence. Notre méthode est donc particulièrement efficace avec des images d'objets capturés dans leur environnement naturel mais elle est moins pertinente lorsqu'appliquée sur des images prises dans un environnement uniforme.

Enfin, les résultats montrent qu'il est possible de fixer un petit seuil en dessous duquel toutes les observations sont correctement identifiées comme anormales (précision = 1). Ce seuil, ajustable en fonction des besoins, peut être utilisé comme un indicateur par les agriculteurs. De plus, la superposition des cartes de saillance sur les images originales, combinée à des informations telles que l'âge prédit par le modèle et l'âge réel (voir la Figure 3.5), peut être utilisée par les agriculteurs pour analyser les différents facteurs liés aux anomalies. En comprenant quelles zones de l'image ont permis au modèle de faire sa prédiction, qu'elle soit correcte ou non, les agriculteurs peuvent identifier plus rapidement les caractéristiques importantes d'une plante, ce qui les aide à détecter et diagnostiquer les maladies plus efficacement.

3.6 Conclusion et perspectives

Dans cette étude, nous proposons un modèle de support à la décision pour la détection d'anomalies et démontrons son efficacité sur un jeu de données de séries temporelles d'images de choux-fleurs et deux jeux de données de feuilles de fruits et légumes. Le modèle proposé exploite les informations spatiales apprises lors d'une tâche auxiliaire (dans notre cas, la prédiction de l'âge des plantes et la classification des types de feuilles) et ne nécessite aucune hypothèse préalable sur la présence d'anomalies dans les données. Le seul prérequis est d'avoir un échantillon de données normales. Notre méthodologie est générique par rapport à la méthode d'extraction des cartes de saillance et la tâche auxiliaire. Dans nos travaux futurs, nous aimerions tester d'autres approches d'explicabilité pour évaluer leur impact sur le calcul du score et, par conséquent, sur la détection d'anomalies.

Dans l'ensemble de données *GrowliFlower*, qui contient des séries temporelles d'images, les anomalies associées aux choux-fleurs ne présentent pas de continuité temporelle. Une anomalie peut être présente à un instant t et ensuite disparaître, tout comme une nouvelle anomalie peut apparaître à l'instant $t + \Delta t$. Par conséquent, bien que notre travail actuel traite de séries temporelles d'images, en raison de ce manque d'anomalies continues, nous traitons chaque image indépendamment. Nous n'avons observé aucune continuité dans les données in-situ indiquant des anomalies liées aux choux-fleurs. Dans nos recherches futures, nous aimerions travailler avec des jeux de données qui présentent des anomalies continues à partir d'un certain pas de temps t et exploiter l'aspect temporel des images. Plus précisément, nous souhaitons explorer des approches telles que BFAST [173], qui nous permettrait de déterminer à partir de quel moment une anomalie se manifeste. Ce faisant, nous visons à améliorer la détection et la gestion des anomalies pour aider les agriculteurs à surveiller et protéger leurs cultures.

Chapitre 4

Évaluation des poids d'auto-attention d'un Transformer par représentation sous forme de graphe

4.1 Introduction des défis liés à la validation de l'apprentissage des Transformers

Depuis leur introduction [13], les modèles Transformers [38] (cf. Section 1.2.2) sont reconnus comme l'état de l'art pour le traitement des données séquentielles [174], principalement en raison de leur mécanisme d'auto-attention. Ce mécanisme capture efficacement les relations entre les différents éléments (tokens) d'une séquence tout en minimisant le problème de la disparition du gradient [30] souvent rencontré avec les architectures récurrentes.

Parallèlement à leur ascension, le besoin d'explicabilité des réseaux de neurones, et plus précisément des Transformers, s'est intensifié, en raison de leur réputation de "boîtes noires" [175]. Dans ce contexte, les modèles d'explicabilité, applicables à divers types de réseaux de neurones (cf. Section 3.2.3), permettent de déterminer précisément comment les entrées impactent les sorties. Il est important de noter la distinction entre l'explicabilité et l'interprétabilité. L'interprétabilité se concentre sur la conception de modèles qui sont intrinsèquement compréhensibles, tandis que l'explicabilité tente de fournir des explications a posteriori pour des modèles boîte noire existants [9].

Le besoin d'explicabilité des modèles est particulièrement cruciale dans des domaines sensibles comme le médical, où les décisions prises par les modèles peuvent avoir un impact profond sur la vie des individus [9].

Bien qu'il soit inexact de prétendre que le mécanisme d'attention dans les Transformers puisse directement servir d'explication [176, 177], il peut néanmoins fournir des interprétations plausibles et significatives [178, 179]. C'est pourquoi de nombreux travaux ont émergé, utilisant l'auto-attention pour tenter d'améliorer l'explicabilité des modèles [175]. Cepen-

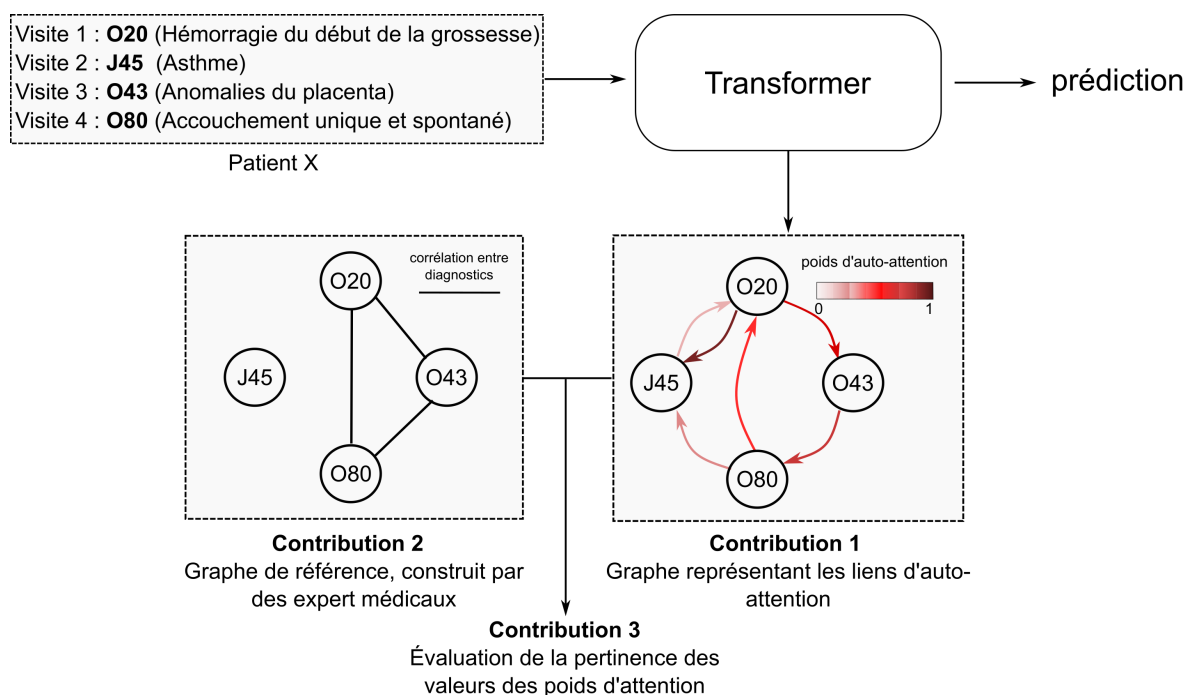


FIGURE 4.1 – Schéma de l'approche proposée. Étant donné une séquence de diagnostics que l'on peut retrouver dans le SNDS (cf. annexe A) donnée en entrée d'un Transformer, nous extrayons les poids d'auto-attention appris par le modèle entre chaque paire de diagnostics et les représentons sous forme de graphe, où chaque nœud est un diagnostic et chaque arc qui relie le nœud A au nœud B représente le poids d'auto-attention que le nœud A accorde au nœud B. Nous évaluons ce graphe par rapport à un graphe représentant l'expertise médicale afin d'évaluer la pertinence des poids d'auto-attention appris par le modèle.

dant, ces méthodes ne permettent pas de valider directement la qualité de l'apprentissage en comparant l'explication fournie par les modèles à une vérité terrain.

Dans ce contexte, nous proposons une approche, schématisée dans la Figure 4.1, pour évaluer l'apprentissage des Transformers en examinant les liens d'auto-attention qu'ils ont appris. Notre méthode consiste à représenter ces liens sous forme de graphe et à évaluer leur pertinence par rapport à une vérité terrain. Plus précisément, nous évaluons de façon automatisée les arcs du graphe représentant les liens d'auto-attention en les comparant à une référence établie, elle aussi représentée sous forme de graphe. Notre approche permet d'obtenir un score d'évaluation qui indique si le modèle Transformer accorde la bonne importance aux liens validés par l'expertise.

Dans notre étude, nous nous concentrons sur un cas d'usage spécifique lié à l'évaluation de modèles Transformers appliqués sur des données de dossiers de santé électroniques (*Electronic Health Record* (EHR)). Plus précisément, nous utilisons un modèle entraîné pour la tâche de prédiction de diagnostic, en se basant sur le parcours médical d'un individu. Ce cas d'usage est particulièrement adapté à notre approche car, dans la pratique médicale, il existe non seulement des règles de décision établies mais également l'expérience clinique qui guide les médecins dans le diagnostic des maladies. Ainsi, notre approche permet d'évaluer à quel point le modèle apprend et donne de l'importance aux mêmes règles que celles utilisées

par les médecins.

L'objectif principal de cette étude est de déterminer si un modèle entraîné pour prédire des diagnostics est capable d'établir des liens pertinents entre les diagnostics passés d'un patient, conformément à une expertise médicale. Autrement dit, nous souhaitons vérifier si le modèle apprend des corrélations entre les diagnostics qui correspondent aux schémas et aux connaissances médicales établies. Cette approche permet non seulement d'évaluer l'efficacité du modèle à reproduire des processus de décision médicale réels mais aussi d'appréhender dans quelle mesure les mécanismes d'attention utilisés dans les Transformers peuvent être fiables et utiles dans des contextes où les décisions ont des implications critiques pour la santé des patients.

Contributions

1. Nous proposons une méthode qui représente **sous forme de graphe** les **interactions et les poids d'auto-attention entre les tokens** appris par un Transformer.
2. Nous proposons une approche pour **modéliser l'expertise médicale sous la forme d'un graphe**.
3. Nous introduisons une nouvelle **métrique pour évaluer la pertinence des connexions** apprises par le modèle en les comparant à une référence établie.
4. Nous **validons notre approche sur deux cas d'usage médicaux**, visant à évaluer les interactions entre les diagnostics sur lesquels le modèle s'appuie pour prédire le diagnostic de la prochaine visite à l'hôpital.

Plan du chapitre

État de l'art (Section 4.2) : présentation des travaux appliquant des Transformers sur des dossiers de santé électroniques et des travaux utilisant les poids d'auto-attention pour améliorer l'explicabilité des Transformers.

Évaluation de l'apprentissage d'un Transformer (Section 4.3) : méthodologie de notre approche d'évaluation de l'apprentissage du mécanisme d'auto-attention d'un Transformer, en le représentant sous forme de graphe.

Expérimentations (Section 4.4) : application de notre approche sur un modèle Behrt [180] entraîné sur les données du SNDS (cf. annexe A) pour prédire la prochaine visite à l'hôpital.

Résultats et discussions (Section 4.5)

Conclusion et perspectives (Section 4.6)

4.2 État de l’art

4.2.1 Transformers appliqués aux dossiers de santé électroniques

Introduits à l’origine pour la tâche de traduction automatique [13], les Transformers sont désormais couramment utilisés pour traiter tout type de données séquentielles, y compris les dossiers de santé électroniques (EHR) [181]. Les EHR contiennent une multitude de données complexes. Cela inclut des codes CIM (*Classification Internationale des Maladies*¹) pour les diagnostics et les médications, ainsi que des informations démographiques et des historiques de traitement, qui sont enregistrés à chaque visite du patient. La complexité de ces données provient non seulement de leur volume et de leur diversité, mais aussi de leur structuration en séquences temporelles, qui capturent les cycles de diagnostic, de traitement et de suivi des patients, incluant les réadmissions potentielles.

Au cours des dernières années, de nombreux travaux ont exploré l’application des Transformers sur les EHR [181]. Ces recherches se répartissent en trois catégories principales :

- L’apprentissage de la structure des EHR.
- La prédiction de diagnostics futurs.
- Le couplage des données des EHR avec d’autres types de données médicales.

Apprentissage de la structure des EHR

Choi et al. ont proposé le modèle *Graph Convolution Transformer* (GCT), qui intègre des techniques de convolution de graphes dans l’architecture des Transformers [182]. Les convolutions de graphes permettent de modéliser les relations entre les différents codes CIM, exploitant ainsi la structure sous-jacente des données, telles que les liens entre diagnostics et traitements. Ces relations sont ensuite utilisées pour orienter les mécanismes d’attention dans le Transformer, facilitant une compréhension contextuelle approfondie des séquences de traitement et de diagnostic.

Shang et al. ont proposé G-BERT [183], un modèle qui utilise des réseaux neuronaux de graphes (GNN) pour apprendre les représentations vectorielles des relations et structures ontologiques médicales des codes CIM. Ces représentations vectorielles sont par la suite intégrées dans un Transformer, entraîné pour anticiper les futurs besoins médicamenteux des patients. Plus précisément, les GNN capturent la complexité des interactions entre les codes, tandis que le Transformer les utilise pour émettre des recommandations de traitement précises. G-BERT a la capacité d’exploiter les données des patients ayant uniquement une visite dans leur parcours, grâce à une phase de pré-entraînement non supervisé, ce qui améliore ses performances en recommandation de médicaments.

Enfin, Peng et al. ont introduit GRAM [184], un modèle qui utilise un mécanisme d’attention basé sur les graphes pour générer des représentations vectorielles des codes médicaux,

1. La Classification Internationale des Maladies est une classification médicale utilisée dans le monde entier à des fins épidémiologiques, de gestion de la santé et cliniques.

qui illustre leurs relations hiérarchiques et contextuelles. L’attention multi-têtes (cf. Section 1.2.2) analyse comment ces relations impactent les traitements et diagnostics au fil du temps. Ce modèle se révèle particulièrement efficace pour décrypter les subtilités des parcours de soins des patients, offrant une représentation graphique riche des interactions entre différents aspects des EHR.

Prédiction de diagnostics futurs

Li et al. ont développé Behrt [180], une adaptation du modèle BERT [39] pour les dossiers de santé électroniques (EHR). Behrt utilise un Transformer bidirectionnel [39] pour apprendre des représentations contextuelles des séquences de données médicales. Le modèle est pré-entraîné sur un large corpus d’EHR en utilisant une tâche de prédiction de tokens masqués, où il apprend à prédire des codes de diagnostic masqués aléatoirement. Cette phase de pré-entraînement permet à Behrt d’acquérir une compréhension générale des relations entre les différents concepts médicaux. Après le pré-entraînement, Behrt est affiné sur des tâches spécifiques, notamment la prédiction de diagnostics futurs. Le modèle intègre non seulement les séquences de codes de diagnostic, mais aussi des informations sur l’âge du patient, la position temporelle des événements, et la distinction entre les différentes visites. Cette approche permet à Behrt de capturer des dépendances complexes et à long terme dans les trajectoires de santé des patients. Dans nos travaux, nous utilisons une version de Behrt entraînée sur les données du *Système National des Données de Santé* (SNDS) (cf. Section 4.4).

Li et al. ont étendu leurs travaux et présenté Hi-BEHRT [185], qui utilise une structure hiérarchique pour traiter plus efficacement les longues séquences de données médicales.

Rasmy et al. ont proposé Med-BERT [186], qui modifie la tâche de pré-entraînement de BERT [39] pour inclure la prédiction de la durée de séjour et utilise une combinaison de codes CIM-9 et CIM-10² pour prédire le diabète et l’insuffisance cardiaque. L’utilisation d’un grand vocabulaire combinant CIM-9 et CIM-10 rend Med-BERT particulièrement avantageux en termes de transférabilité et d’applicabilité à divers contextes cliniques.

Luo et al. ont proposé HiTANet (*Hierarchical Time-aware Attention Network*) [187], qui incorpore un vecteur temporel pour représenter le temps écoulé entre les visites consécutives, combiné à l’encodage de la visite initiale pour prédire les diagnostics futurs sur trois cohortes de maladies : la broncho-pneumopathie chronique obstructive, l’insuffisance cardiaque et la maladie rénale. L’intégration explicite de l’information temporelle dans HiTANet lui confère un avantage significatif pour capturer l’évolution des maladies au fil du temps, améliorant ainsi la précision des prédictions.

Enfin, Ren et al. ont introduit RAPT (*RepresentAtion by Pre-training time-aware Transformer*) [188], un modèle qui intègre un vecteur de durée et des tâches de pré-entraînement supplémentaires telles que la prédiction de similarité et la vérification de la pertinence pour résoudre les problèmes de données insuffisantes, d’incomplétude et de séquences courtes

2. Les CIM-9 et CIM-10 sont deux versions différentes de la classification des maladies. Proposée en 1979, la CIM-9 comprend 14 000 codes couvrant les diagnostics et les procédures. Les codes sont principalement numériques et relativement généraux. Adoptée en 1990 et mise en œuvre dans de nombreux pays au début des années 2000, la CIM-10 est beaucoup plus détaillée, avec environ 70 000 codes diagnostiques. Elle fournit une description beaucoup plus précise des maladies et de leurs symptômes.

des EHR. RAPT est évalué pour prédire les issues de grossesse, les périodes à risque, ainsi que les diagnostics de diabète et d'hypertension pendant la grossesse.

Couplage de données médicales

Parmi les travaux qui couplent les données des EHR à d'autres données médicales afin d'utiliser des modèles multimodaux, Darabi et al. ont introduit TAPER (*Time-Aware Patient EHR Representation*) [189], un modèle qui utilise des Transformers distincts pour générer différentes représentations vectorielles des codes cliniques (CIM, médicaments et procédures) et des notes cliniques. Ces représentations sont ensuite fusionnées pour former une "représentation du patient", qui est utilisée pour prédire les diagnostics futurs, les procédures, la durée de séjour à l'hôpital, les réadmissions et la mortalité. De la même façon, Luo et al. utilisent BERT [39] pour créer des représentations communes des données médicales pour la prédiction de la toux chronique [190], et Meng et al. pour prédire un épisode dépressif [191].

Concernant les architectures multimodales, Xu et al. ont introduit l'architecture MUFASA [192], qui optimise simultanément les architectures spécifiques aux modalités et les stratégies de fusion en utilisant des techniques de recherche d'architecture neuronale³ (NAS). MUFASA recherche plusieurs architectures indépendantes pour chaque modalité de données et utilise un modèle de fusion pour combiner les sorties de ces modalités, améliorant ainsi les performances prédictives sur les données EHR.

Enfin, Zhang et al. propose un cadre d'apprentissage supervisé des représentations des patients [193], qui utilise les données EHR multimodales pour comprendre et représenter de manière complète les conditions de santé des patients. Leur approche maximise l'information mutuelle entre les différentes représentations des modalités tout en minimisant la fonction de perte spécifique à la tâche d'apprentissage. Ainsi, les représentations apprises conservent l'information pertinente à la tâche, restent relativement invariantes entre les modalités, sont robustes au bruit et préservent les caractéristiques individuelles des patients.

4.2.2 Explicabilité des Transformers basée sur l'auto-attention

Le besoin d'expliquer les prédictions faites par des réseaux neuronaux profonds, souvent considérés comme des "boîtes noires", s'accroît constamment, notamment pour l'application des modèles dans des secteurs qui ont un impact direct sur la vie humaine, comme la santé [9]. Les principales approches du domaine de l'explicabilité (*Explainable Artificial Intelligence* (XAI)) reposent sur différents concepts tels que la perturbation ou le gradient (voir Section 3.2.3). L'arrivée des Transformers a ouvert un nouveau champ de recherche dans ce domaine car ils se distinguent des autres approches d'apprentissage profond par l'utilisation du mécanisme d'attention [13]. Ce dernier permet d'apprendre et de pondérer les interactions entre les différents éléments d'une séquence en entrée. Des outils de visualisation tels que *bertviz* [194] pour examiner les interactions entre les différents éléments

3. Les techniques de recherche d'architecture neuronale sont des méthodes automatisées visant à identifier la structure optimale d'un réseau de neurones pour une tâche donnée.

d'entrée du modèle via les poids d'auto-attention sont couramment utilisés. Cependant, ces dernières années, une multitude de travaux utilisant les poids d'auto-attention pour expliquer les modèles basés sur les Transformers ont vu le jour. Nous en présentons une sélection dans cette partie.

Abnar et Zuidema ont introduit deux nouvelles méthodes pour analyser et comprendre le flux d'information dans les modèles Transformer à travers les poids d'auto-attention [195]. Leur recherche se concentre sur la manière dont l'information se propage à travers les différentes couches du Transformer, et comment cela peut être quantifié pour mieux interpréter les modèles. Plus précisément, les auteurs ont développé deux méthodes, *Attention Rollout* et *Attention Flow*, conçues pour surmonter les limitations des mécanismes d'attention traditionnels, qui deviennent de moins en moins interprétables dans les couches supérieures du modèle.

Attention Rollout calcule comment l'attention de toutes les couches précédentes contribue à l'auto-attention dans une couche donnée. Elle assume que l'identité des tokens d'entrée se combine linéairement à travers les couches en utilisant les poids d'auto-attention. L'idée est de "dérouler" ces poids pour mieux visualiser comment l'information de chaque token est préservée ou diluée à travers le réseau.

Attention Flow est une méthode qui modélise le mécanisme d'auto-attention d'un Transformer comme un réseau de flux. Dans ce réseau, chaque connexion entre éléments a une capacité déterminée par le poids d'auto-attention original. La méthode utilise un algorithme de flux maximal pour identifier le chemin le plus influent par lequel l'auto-attention se propage des couches intermédiaires vers les tokens d'entrée, offrant une autre manière de comprendre comment les attentions sont distribuées à travers le modèle.

Dans leur expérimentations, les auteurs appliquent leurs méthodes à la tâche de prédiction du nombre de verbes, c'est-à-dire la prédiction de la singularité ou de la pluralité d'un verbe dans une phrase, lorsque l'entrée est la phrase jusqu'à la position du verbe. Les résultats montrent que leurs méthodes offrent une meilleure corrélation avec les scores d'importance obtenus par des méthodes d'ablation ou qui reposent sur l'étude des gradients par rapport à l'auto-attention brute. Cela suggère que les deux méthodes proposées sont plus fiables pour interpréter les poids d'auto-attention dans les Transformers.

Les techniques *Attention Rollout* et *Attention Flow* sont couramment utilisées pour analyser les mécanismes d'attention [175], mais leur complexité peut augmenter significativement avec la profondeur du réseau. Face à ce défi, la recherche explore d'autres méthodes basées sur l'auto-attention pour expliquer le fonctionnement des Transformers.

Par exemple, Renz et al. introduisent un score de pertinence pour évaluer l'importance de chaque objet dans le cadre de la planification de la conduite autonome [196]. Le score est calculé en additionnant les poids à travers toutes les couches et têtes d'attention pour chaque objet. Cette méthode permet de déterminer quels objets sont les plus critiques pour la prise de décision. De plus, ils introduisent le "*Relative Filtered Driving Score*" (RFDS) pour évaluer l'efficacité du modèle à identifier ces objets pertinents. Le RFDS compare les performances de conduite du modèle avec celle d'un algorithme expert de référence. Cet expert est un système basé sur des règles qui a accès à la position réelle de tous les véhicules, ainsi qu'à des informations privilégiées comme leurs actions et dynamiques futures. L'expert utilise ces

données pour déterminer la position future de tous les véhicules, estimer les intersections potentielles, et ainsi planifier une trajectoire sûre en évitant la plupart des collisions.

En comparant les performances de leur modèle avec cet expert complet, les auteurs peuvent évaluer la capacité de leur approche à identifier les objets les plus pertinents pour la conduite, rendant ainsi le processus de décision plus transparent et explicable.

Dans la même idée, Feng et al. agrègent, en sommant, les poids d'auto-attention de toutes les têtes d'attention dans toutes les couches pour attribuer un score d'importance à chaque caractéristique des dossiers de santé électroniques [197]. Ces scores d'auto-attention sont utilisés pour expliquer les prédictions en identifiant les caractéristiques les plus influentes pour la tâche de prédiction de la mortalité. Leurs expérimentations révèlent que les scores d'auto-attention offrent une fidélité comparable, voire supérieure, aux méthodes d'explicabilité génériques comme SHAP (*SHapley Additive exPlanations*) [198]. Ces scores sont plus simples à calculer et directement issus du processus de décision du modèle.

Trisedya et al. introduisent un score de pertinence pour évaluer l'importance de chaque attribut et de ses voisins dans un modèle d'alignement de graphes de connaissances [199]. Plus précisément, les poids d'auto-attention sont utilisés pour attribuer une importance relative à chaque attribut et voisin en fonction de leur contribution à l'alignement des entités entre différents graphes de connaissances. Leur approche, nommée i-Align, est appliquée dans le contexte de l'enrichissement et de la fusion de graphes de connaissances, où l'objectif est de trouver des correspondances entre les entités de différents graphes pour les fusionner en un graphe plus complet. La principale plus-value de leur méthode réside dans sa capacité à fournir des explications interprétables pour chaque prédiction d'alignement, tout en maintenant des performances élevées. Ces explications prennent la forme d'un ensemble des attributs et des voisins les plus influents, permettant aux experts de vérifier la pertinence des alignements proposés et ainsi de maintenir la qualité du graphe de connaissances résultant lors du processus de fusion.

Kim et al. ont introduit le cadre xRAC (*Read, Attend, and Code*) qui vise à améliorer l'explicabilité des prédictions de codes médicaux à partir de notes cliniques [200]. Ils proposent deux méthodes, **xRAC-ATTN**, qui utilise les scores d'auto-attention pour identifier les parties pertinentes du texte clinique, et **xRAC-KD**, qui utilise la distillation des connaissances⁴ pour créer des modèles plus simples et interprétables. L'utilisation de xRAC-ATTN permet de fournir des explications textuelles qui justifient les codes prédits, en montrant quelles parties du texte clinique ont influencé la décision du modèle.

Clauwaert et al. examinent les poids d'auto-attention des différentes têtes d'un Transformer entraîné pour repérer les sites de démarrage de la transcription (*Transcription Start Site* (TSS)) dans l'ADN (acide désoxyribonucléique) des bactéries [201]. Ces sites marquent le début de la transcription de l'ADN en ARN (acide ribonucléique). Ils démontrent que les différentes têtes d'attention du modèle peuvent se spécialiser de manière automatique

4. La distillation de connaissances est une technique utilisée en apprentissage automatique où un modèle complexe (appelé modèle enseignant) transfère ses connaissances à un modèle plus simple (appelé modèle étudiant). L'objectif est de conserver les performances du modèle complexe tout en bénéficiant de la simplicité et de l'efficacité computationnelle du modèle plus simple. Cette méthode est particulièrement utile lorsque l'on souhaite déployer des modèles sur des dispositifs avec des ressources limitées ou lorsque l'on souhaite améliorer l'interprétabilité des modèles.

pour détecter des motifs spécifiques dans l'ADN, comme les sites de liaison des facteurs de transcription (protéines qui régulent l'expression des gènes). En calculant les scores d'auto-attention, ils identifient quelles parties de l'ADN sont considérées comme cruciales par le modèle pour prédire les TSS. Cette analyse leur permet de comprendre comment le modèle effectue ses prédictions et quelles séquences d'ADN sont les plus pertinentes dans ce contexte.

Sebbaq et al. ont développé MTBERT-Attention [202], une approche pour la classification cognitive de textes en utilisant un modèle multi-tâches avec une couche de co-attention. Cette architecture permet au modèle de traiter simultanément plusieurs tâches d'apprentissage, optimisant ainsi la performance en tirant parti des synergies entre les tâches. La couche de co-attention permet au modèle de se concentrer sur les éléments pertinents de différentes sources de données, améliorant la précision de la classification. En plus d'augmenter la performance, les auteurs explorent les poids d'auto-attention pour rendre le modèle explicable, permettant de visualiser et de comprendre quels aspects du texte influencent le plus les décisions de classification. Pour ce faire, ils utilisent une couche d'attention supplémentaire et calculent des scores d'auto-attention pour chaque mot. Ces scores sont ensuite utilisés pour créer un classifieur simple qui permet de comparer les prédictions basées sur l'auto-attention avec celles du modèle original.

Chen et al. abordent l'explicabilité des recommandations séquentielles en modifiant le mécanisme d'auto-attention utilisé par les Transformers. Le mécanisme d'auto-attention classique [13] utilise la fonction Softmax et attribue donc toujours des valeurs positives. Cette caractéristique est problématique car elle ne permet pas de véritablement ignorer les éléments non pertinents, ce qui peut conduire à du bruit dans le modèle et réduire son interprétabilité. En effet, même les éléments les moins importants reçoivent une attention non nulle, ce qui peut fausser l'interprétation de leur importance réelle dans la prise de décision du modèle. Pour éviter cela, les auteurs introduisent le modèle *Probabilistic Masked Attention Network* (PMAN) [203], qui implémente un masque probabiliste et utilise la fonction d'activation ReLU au carré pour produire des scores d'auto-attention qui peuvent être exactement zéro pour les éléments jugés non pertinents. Cette approche permet non seulement de filtrer les interactions bruitées mais aussi d'améliorer l'explicabilité du modèle en offrant une visualisation claire et directe de l'impact de chaque élément sur la recommandation finale. L'apprentissage du masque est intégré dans un processus d'optimisation qui ajuste dynamiquement les attentions pour maximiser la pertinence des prédictions, rendant les décisions du modèle plus transparentes et compréhensibles pour les utilisateurs.

Schwenke et Atzmueller explorent l'explicabilité des Transformers dans le contexte spécifique de la classification de séries temporelles [204]. Leur approche consiste à utiliser l'agrégation symbolique approximative pour transformer les données de séries temporelles en symboles plus interprétables pour l'humain. Plus précisément, ils discrétisent les données en cinq symboles (très bas, bas, moyen, haut, très haut), réduisant ainsi la complexité tout en préservant la capacité de discrimination. Ils exploitent ensuite l'auto-attention du Transformer pour identifier les points de données significatifs qui mettent en évidence les parties importantes des données pour la classification. Cette méthode de visualisation permet aux utilisateurs de mieux comprendre où le modèle se concentre dans les séries temporelles et pourquoi certaines décisions sont prises, contribuant ainsi à une meilleure interprétabilité

locale des actions du modèle. En outre, ils valident cette approche en utilisant à la fois des données synthétiques et réelles pour démontrer l'efficacité de leur méthode en termes de préservation des informations cruciales tout en simplifiant la visualisation pour l'utilisateur.

Enfin, Bacco et al. présentent deux méthodes utilisant des Transformers pour l'analyse de sentiment explicative [205]. Ils utilisent la *summarisation extractive*⁵ pour créer des résumés qui expliquent les décisions d'analyse de sentiment du modèle. Le premier modèle utilise une attention hiérarchique pour évaluer l'importance des phrases, tandis que le second applique un Transformer à chaque phrase pour évaluer sa pertinence sentimentale. Les performances de ces modèles sont testées sur le jeu de données de critiques de films IMDB⁶, mesurant la précision de la classification des sentiments et la qualité des explications via les résumés générés.

4.2.3 Positionnement de notre approche

Les approches utilisant les poids d'auto-attention des modèles Transformers ont souvent pour objectif de visualiser ou d'estimer la pertinence des liens entre les éléments d'entrée, permettant ainsi aux experts d'utiliser ces explications pour vérifier l'exactitude des modèles. Notre travail se distingue en visant à utiliser les poids d'auto-attention pour valider la pertinence des liens appris en référence à une connaissance établie. Nous proposons une méthode qui représente sous forme de graphe les liens d'auto-attention entre les différents éléments d'une séquence d'entrée. Nous développons un score d'évaluation de ce graphe par rapport à une expertise établie, aussi représentée sous forme de graphe, permettant de quantifier la pertinence des liens appris par le modèle. Notre approche offre la possibilité d'évaluer des modèles type Transformer, en intégrant la validation des connaissances dans le processus d'interprétation des mécanismes d'attention.

4.3 Évaluation de l'apprentissage d'un Transformer

La méthodologie de notre approche, illustrée dans la Figure 4.2, se décompose en plusieurs étapes. Premièrement, nous récupérons les poids d'auto-attention appris par un Transformer, représentant l'attention que chaque token accorde aux autres tokens d'une même séquence. Ces poids sont utilisés pour créer une matrice d'attention globale (cf. Section 4.3.1). Bien que notre méthode soit applicable à n'importe quelle couche d'attention, nous choisissons de nous concentrer sur la dernière couche du modèle. Ce choix est motivé par l'hypothèse que la dernière couche d'attention capture les interactions entre tokens les plus pertinentes pour la prédiction, représentant ainsi l'état le plus abouti du traitement de l'information. Parallèlement, nous identifions les tokens d'importance d'un vocabulaire prédéfini en analysant les gradients. Ces informations sont ensuite utilisées pour générer un graphe (Section 4.3.2) qui représente les liens d'auto-attention entre les tokens d'importance

5. La *summarisation extractive* consiste à sélectionner et extraire directement des phrases ou des segments du texte original considérés comme les plus importants ou représentatifs. Le résultat est un résumé composé de morceaux du texte original, sans reformulation. 6. <https://developer.imdb.com/non-commercial-datasets/>

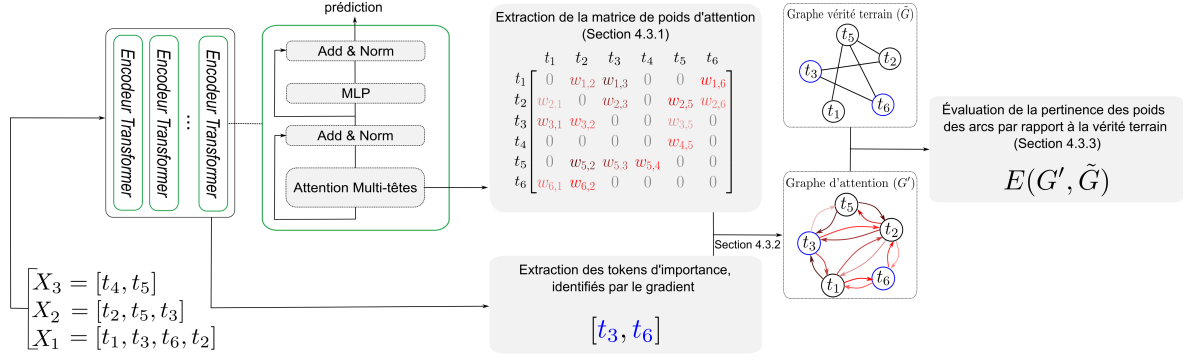


FIGURE 4.2 – Étapes de la méthodologie proposée : à partir des séquences données en entrée d'un Transformer, nous identifions les tokens d'importance par analyse des gradients. Une matrice d'attention globale est construite dans laquelle chaque token attribue un poids (noté w) à tous les autres tokens d'un vocabulaire global, à condition que les deux tokens coexistent dans la même séquence. Ensuite, nous extrayons un sous graphe d'auto-attention G' en partant des tokens d'importance (noeuds bleus) et en ajoutant tous les tokens (noeuds noirs) auxquels ils sont liés dans la matrice d'attention globale. Un arc reliant deux noeuds dans G' est pondéré par le poids d'auto-attention que le noeud source accorde au noeud cible. Nous évaluons la pertinence des poids d'auto-attention appris par le modèle à l'aide de la fonction d'évaluation E , qui compare le graphe d'attention G' au graphe représentant la vérité terrain, noté $\tilde{G}$

et ceux auxquels ils sont connectés dans la matrice d'attention globale. Enfin, nous évaluons ce graphe (Section 4.3.3) en le comparant à une référence établie, également représentée sous forme de graphe.

4.3.1 Création de la matrice d'attention globale

Soit $T = \{t_1, t_2, \dots, t_V\}$ un ensemble (ou vocabulaire) de V tokens distincts. Nous considérons un ensemble de données étiquetées $X = (X_i, y_i)_{i \in \llbracket 1; N \rrbracket}$, composé de N séquences X_i , chacune associée à une étiquette y_i . Plus précisément, chaque X_i est une séquence de z_i tokens $(x_1, \dots, x_{z_i})$, où $z_i \in \mathbb{N}$ et $\forall j \in \llbracket 1, z_i \rrbracket, x_j \in T$. Nous passons chaque séquence X_i à un Transformer entraîné pour prédire $\hat{y}_i$. Si la prédiction du modèle $\hat{y}_i$ correspond à l'étiquette réelle y_i , nous récupérons la matrice d'attention (de la dernière couche) A_i de taille $z_i \times z_i$ pendant le traitement de X_i par le modèle :

$$A_i = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,z_i} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,z_i} \\ \vdots & \vdots & \ddots & \vdots \\ a_{z_i,1} & a_{z_i,2} & \cdots & a_{z_i,z_i} \end{bmatrix} \quad (4.1)$$

où chaque élément $a_{m,n}$ représente l'auto-attention que le token x_m dans la séquence X_i donne au token x_n dans la même séquence. Plus précisément, dans une couche de Transformer, l'auto-attention $a_{m,n}$ est calculée en projetant x_m en un vecteur de requête $\mathbf{Q}_m = W^Q x_m$ et x_n en un vecteur de clé $\mathbf{K}_n = W^K x_n$, où W^K et W^Q représentent les matrices de poids apprises. Ces vecteurs de requête et de clé sont utilisés pour calculer un

score d'auto-attention brut via un produit scalaire, qui est ensuite divisé par la dimension des vecteurs de clé (la taille des représentations vectorielles utilisées pour représenter chaque token dans l'espace des clés). Enfin, ce score normalisé est passé par une fonction *softmax* qui le convertit en probabilités, donnant ainsi :

$$a_{m,n} = \frac{e^{\left(\frac{\mathbf{Q}_m^\top \cdot \mathbf{K}_n}{\sqrt{d_k}}\right)}}{\sum_{l=1}^{z_i} e^{\left(\frac{\mathbf{Q}_m^\top \cdot \mathbf{K}_l}{\sqrt{d_k}}\right)}} \quad (4.2)$$

où z_i est le nombre total de tokens dans la séquence X_i et d_k est la taille du vecteur de clé $\mathbf{K}_n$. Il est à noter que l'auto-attention $a_{m,n}$ dans la matrice A_i est calculée comme la moyenne de l'auto-attention que le token x_m donne au token x_n dans toutes les têtes d'attention de la couche sélectionnée.

Une fois les poids d'auto-attention récupérés pour chaque paire dans chaque séquence, nous agrégeons ces poids dans une matrice d'attention globale G de taille $(V \times V)$ où V est la taille du vocabulaire de tokens T . G peut être interprétée comme une matrice d'adjacence pondérée, représentant les relations d'auto-attention entre tous les tokens du vocabulaire. Cette matrice répertorie les valeurs des poids d'auto-attention de toutes les paires possibles de tokens dans le vocabulaire global. Plus précisément, G synthétise l'attention que chaque token t_m accorde à chaque autre token t_n à travers toutes les séquences analysées. Pour chaque paire de tokens (t_m, t_n) , nous identifions toutes les occurrences de ces tokens dans les différentes séquences et accumulons les valeurs d'auto-attention correspondantes pour ces tokens à partir des matrices A_i associées à chaque séquence. Nous collectons ces valeurs dans un ensemble S_{mn} :

$$S_{mn} = \{a_{m',n'} \mid x_{m'} = t_m, x_{n'} = t_n, m, n \in \{1, \dots, V\}^2, m', n' \in \{1 \dots, z_i\}^2\} \quad (4.3)$$

Chaque élément $G[m, n]$ de la matrice est ensuite calculé en prenant la médiane de ces valeurs de l'ensemble S_{mn} :

$$G[m, n] = \text{Médiane}(S_{mn}) \quad (4.4)$$

Il est à noter que nous choisissons de prendre la valeur médiane pour éviter les valeurs aberrantes potentielles. Finalement, $G[m, n]$ capture le niveau d'auto-attention que chaque token t_m accorde à chaque autre token t_n à travers toutes les séquences, reflétant la tendance centrale de l'intensité d'interaction entre chaque paire de tokens dans l'ensemble de données.

4.3.2 Génération du graphe d'auto-attention

Nous souhaitons construire un graphe d'auto-attention G' qui représente les interactions entre certains tokens telles qu'apprises par le modèle. En d'autres termes, nous extrayons un sous-graphe de la matrice d'attention globale, en sélectionnant les relations entre un ensemble choisi de tokens (nœuds). Nous expliquons dans cette partie comment nous sélectionnons les nœuds du graphe.

Pour chaque séquence X_i où le modèle prédit correctement l'étiquette $\hat{y}_i$, nous récupérons le gradient $\Delta(x_j)$ de chaque token x_j dans X_i en effectuant une passe de rétropropagation. Les gradients nous permettent de mesurer l'influence de chaque token sur la prédiction. Sachant que le modèle fait une bonne prédiction, des gradients avec des valeurs plus élevées indiquent une contribution plus significative à la décision du modèle [150]. Pour chaque token du vocabulaire T qui apparaît dans au moins une séquence correctement prédite, nous calculons l'importance médiane des gradients associés à ce token. Puis nous sélectionnons un nombre prédéfini g des tokens les plus importants (cf. Figure 4.2) en fonction de la valeur médiane des gradients, identifiant ainsi les tokens qui influencent de manière la plus constante et significative les prédictions correctes du modèle.

Nous utilisons ces g tokens comme nœuds initiaux de G' . Ensuite, nous étendons cet ensemble initial en ajoutant tout token t_n du vocabulaire T , pour lequel l'auto-attention $G[m, n]$ est non nulle, où le token t_m appartient à l'ensemble initial g . Ainsi, l'ensemble des nœuds dans G' inclut les tokens g initiaux et tous les tokens qui leur sont directement liés, avec leurs poids d'auto-attention médians associés.

Enfin, nous construisons un graphe orienté où chaque paire (t_m, t_n) parmi les nœuds de G' est connectée par une arc (t_m, t_n) si $G[m, n]$ est non nul. Les arcs sont pondérés par les valeurs correspondantes dans G , qui quantifient l'intensité de l'auto-attention que t_m accorde à t_n (voir exemple Figure 4.2).

4.3.3 Évaluation de l'auto-attention apprise par un Transformer

Nous souhaitons évaluer la qualité des liens d'auto-attention entre les tokens appris par un Transformer. Pour ce faire, nous comparons ces connexions à une référence établie, représentée par le graphe $\tilde{G}$. L'évaluation est réalisée en calculant la différence entre la proportion pondérée des arcs de G' communs à $\tilde{G}$ et la proportion pondérée des arcs de G' non communs à $\tilde{G}$. Concernant $\tilde{G}$, le seul prérequis est qu'il soit représenté sous forme de graphe. Il peut être orienté ou non et nous considérons qu'il n'est pas pondéré car il est difficile de transposer la pondération par l'auto-attention à un cas d'usage réel. Dans la Section 4.4.4, nous détaillons le protocole que nous avons développé pour générer le graphe de référence utilisé dans nos expérimentations.

La fonction d'évaluation, que nous appelons $E \in [-1, 1]$, est calculée comme suit :

$$E(G', \tilde{G}) = w_{in} - w_{out} \quad (4.5)$$

où :

- $w_{in} = \frac{\sum_{(m,n) \in G' \cap \tilde{G}} w_{mn}}{\sum_{(m,n) \in G'} w_{mn}} \in [0, 1]$, est la proportion pondérée des arcs dans le graphe d'auto-attention G' qui sont également présents dans le graphe de référence $\tilde{G}$, c'est-à-dire le nombre d'arcs de G' inclus dans $\tilde{G}$ pondérés par leur poids et normalisé par le nombre pondéré d'arcs dans G' .
- $w_{out} = \frac{\sum_{(m,n) \in G', (m,n) \notin \tilde{G}} w_{mn}}{\sum_{(m,n) \in G'} w_{mn}} \in [0, 1]$, est la proportion pondérée des arcs dans G' qui ne sont pas confirmés par $\tilde{G}$.

La fonction d'évaluation E mesure l'alignement entre le graphe d'auto-attention G' et le graphe de référence $\tilde{G}$. E produit des valeurs interprétables, illustrées par les cas extrêmes suivants :

Valeur maximale : Tous les arcs de G' sont inclus dans $\tilde{G}$ donc $E(G', \tilde{G}) = 1$ car $w_{in} = 1$ et $w_{out} = 0$.

Valeur minimale : Aucun arc de G' n'est inclus dans $\tilde{G}$ donc $E(G', \tilde{G}) = -1$ car $w_{in} = 0$ et $w_{out} = 1$.

Prédominance des correspondances fortes : Un nombre équivalent d'arcs de G' inclus et non inclus dans $\tilde{G}$, et $w_{in} > w_{out}$ donc $0 < E(G', \tilde{G}) < 1$.

Prédominance des correspondances faibles : Un nombre équivalent d'arcs de G' inclus et non inclus dans $\tilde{G}$, et $w_{in} < w_{out}$ donc $-1 < E(G', \tilde{G}) < 0$.

4.4 Expérimentations

Nous évaluons notre méthode à l'aide d'un modèle basé sur l'attention, entraîné par le *Lab Santé* de la *Direction de la Recherche, des Études, de l'Évaluation et des Statistiques* (Drees⁷). Pour tester la validité des graphes générés à partir des liens d'auto-attention pondérés, nous avons mis en place un protocole de création de graphes de référence en collaboration avec les experts médicaux du département.

4.4.1 Modèle fondé sur des mécanismes d'attention

Nous souhaitons étudier l'apprentissage d'un modèle basé sur des mécanismes d'attention, spécifiquement adapté à l'analyse des données médicales. Pour nos expériences, nous utilisons un modèle suivant l'architecture de Behrt [180], entraîné sur la tâche de prédiction des diagnostics des patients lors de futures visites à l'hôpital, basée sur une séquence historique de visites.

4.4.1.1 Données

Les données sur lesquelles le modèle a été entraîné proviennent des tables MCO (*Médecine, Chirurgie, Obstétrique*) de la base PMSI (*Programme de Médicalisation des Systèmes d'Information*) du SNDS (*Système National des Données de Santé*) (voir annexe A). Chaque visite à l'hôpital est caractérisée par un ensemble de diagnostics, comprenant un diagnostic

7. <https://drees.solidarites-sante.gouv.fr>

échantillon	visites	diagnostics	nombre d'exemples
tâche 1	> 2	> 3	14,59 millions
tâche 2	> 4	non renseigné	5,94 millions

TABLE 4.1 – Spécificités des échantillons utilisés pour les deux tâches d’apprentissage du modèle Behrt. La tâche 1 correspond à la prédiction de mots masqués, la tâche 2 correspond à l’affinage du modèle pré-entraîné sur la tâche 1.

principal et, le cas échéant, un diagnostic relié, ainsi que plusieurs diagnostics associés qui enrichissent le contexte du diagnostic principal.

4.4.1.2 Tâche d’entraînement

Le modèle est entraîné sur une tâche de classification multi-classes et multi-étiquettes. Cela signifie que l’étiquette à prédire peut contenir plusieurs classes différentes. Les classes sont représentées par 2 053 diagnostics codés selon la 10ème révision de la *Classification Internationale des Maladies* (CIM-10), qui est la liste des codes classant les maladies et les problèmes médicaux.

Le processus d’entraînement du modèle est divisé en deux phases distinctes pour affiner les capacités prédictives du modèle et le spécialiser pour répondre aux exigences spécifiques du domaine médical :

1. **Prédiction de mots masqués**, où le modèle apprend à identifier et reconstituer les éléments cachés dans les données d’entraînement.
2. **Affinage du modèle pré-entraîné** pour l’adapter spécifiquement à la tâche de prédiction des diagnostics pour les futures visites à l’hôpital.

Pour les deux tâches, le modèle est entraîné sur un jeu de données constitué à partir de deux échantillons : un échantillon aléatoire de 4% des données SNDS couvrant la période de 2008 à 2017, et un échantillon considérant toutes les données SNDS de 2018 à 2021. En fonction de la tâche, une condition sur les données est appliquée pour sélectionner les individus ayant eu un certain nombre de visites et/ou un minimum de diagnostics dans leur historique médical, afin de garantir que l’échantillon d’entraînement contienne des parcours médicaux suffisamment diversifiés et ainsi éviter le surapprentissage. Les spécificités des conditions de filtrage des échantillons et le nombre d’exemples utilisés pour les entraînements liés aux deux tâches sont résumés dans la Table 4.1. Pour nos expérimentations, nous utilisons le jeu de données de test de la deuxième tâche d’apprentissage, comprenant les historiques médicaux de 1,19 millions d’individus.

4.4.2 Cas d’usage

Nous appliquons notre méthode pendant la phase d’inférence⁸ du modèle, en récupérant les historiques médicaux des individus pour lesquels le modèle a correctement prédit la

8. L’inférence est l’utilisation d’un modèle déjà entraîné pour faire une prédiction sur de nouvelles données.

prochaine visite, c'est-à-dire que le diagnostic à prédire est dans le top 2 des prédictions. Nous nous assurons que l'autre diagnostic présent dans le top 2 est contextuellement lié au diagnostic prédit, renforçant ainsi la pertinence de notre interprétation des gradients d'importance (cf. Section 4.3.2). Nous travaillons sur deux cas d'usage fréquents et distincts, basés sur différents échantillons des données de test. Ces cas d'usage concernent la prédiction de diagnostics incidents, ce qui signifie que le diagnostic à prédire n'apparaît pas dans la séquence d'entrée. Cette approche permet d'analyser précisément l'influence des diagnostics antérieurs sur les prédictions.

4.4.2.1 Cas d'usage de l'accouchement

La classe *accouchement* inclut les codes CIM-10 suivants :

- (O80) accouchement spontané unique
- (O81) accouchement unique par forceps et ventouse
- (O82) accouchement unique par césarienne
- (O83) autres accouchements assistés uniques
- (O84) accouchements multiples

Nous sélectionnons les individus pour lesquels la visite à prédire contient exactement un code de la classe *accouchement*, qui n'est pas dans la séquence que nous donnons en entrée au modèle. Autrement-dit, nous sélectionnons les individus dont la visite à prédire contient un des 5 diagnostics de la classe *accouchement*, et dont l'historique médical ne contient pas ce même diagnostic.

L'échantillon initial comprend les historiques médicaux de 16 032 individus. Cependant, en raison de contraintes de ressources, nous procédons à un tirage aléatoire de 2 000 individus au sein de cet échantillon pour appliquer notre approche. Dans l'échantillon utilisé, le modèle parvient à prédire correctement le code lié à l'accouchement pour 190 individus, qui constituent ainsi notre échantillon final pour l'analyse de ce cas d'usage.

4.4.2.2 Cas d'usage des maladies hypertensives

La classe des *maladies hypertensives* est définie par les codes CIM-10 suivants :

- (I10) hypertension essentielle
- (I11) cardiopathie hypertensive
- (I12) néphropathie hypertensive
- (I13) cardionéphropathie hypertensive
- (I15) hypertension secondaire

Nous sélectionnons les individus pour lesquels la visite à prédire contient le diagnostic I10 et aucun code de la classe des *maladies hypertensives* n'est dans la séquence correspondant à l'historique médical, ce qui correspond à 34 601 exemples. Comme pour le cas de l'accouchement, nous sélectionnons aléatoirement 2 000 exemples. Dans ce cas, le modèle prédit correctement le code I10 pour 514 individus, formant ainsi notre échantillon final pour le cas d'usage des *maladies hypertensives*.

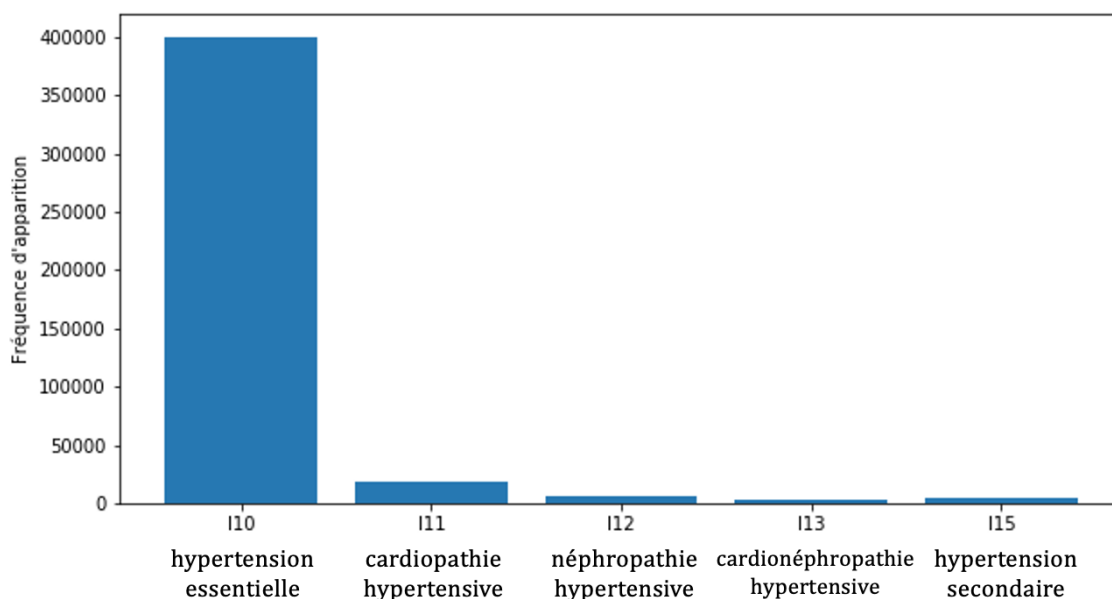


FIGURE 4.3 – Fréquence d’apparition des différents diagnostics de la classe *maladies hypertensives*, dans l’échantillon global contenant les historiques médicaux de 1,19 millions d’individus.

4.4.2.3 Remarques

Contrairement au cas de l’accouchement, où nous savons intuitivement qu’un accouchement n’influence pas médicalement un autre accouchement, le cas des maladies hypertensives est plus complexe car nous avons moins d’informations sur la causalité des diagnostics de cette classe entre eux. C’est pourquoi nous choisissons de nous focaliser sur un seul diagnostic, I10, qui est le plus représenté dans l’échantillon (cf. Figure 4.3). Nous sélectionnons des individus pour lesquels aucun autre diagnostic de cette classe n’est présent dans leur historique médical afin que les résultats soient comparables à ceux du cas d’usage de l’accouchement, car la présence d’autres diagnostics pourrait influencer l’apparition de I10 et biaiser les résultats.

4.4.3 Génération du graphe d’auto-attention G' à partir de Behrt en inférence

Pour chaque cas d’usage, la création du graphe d’auto-attention G' commence par la récupération des matrices d’auto-attention A_i pour chaque individu X_i , extraites de la dernière couche du modèle Behrt. À partir de ces matrices, nous construisons la matrice d’attention globale G de dimensions $(2\,053 \times 2\,053)$, où 2 053 représente le nombre de diagnostics possibles (la taille V du vocabulaire T dans la Section 4.3.1).

Parallèlement, nous récupérons le gradient associé à chaque diagnostic de la séquence représentant l’historique médical d’un individu, que nous donnons en entrée au modèle. Cela nous permet d’identifier, pour chaque individu, quels diagnostics ont le plus influencé

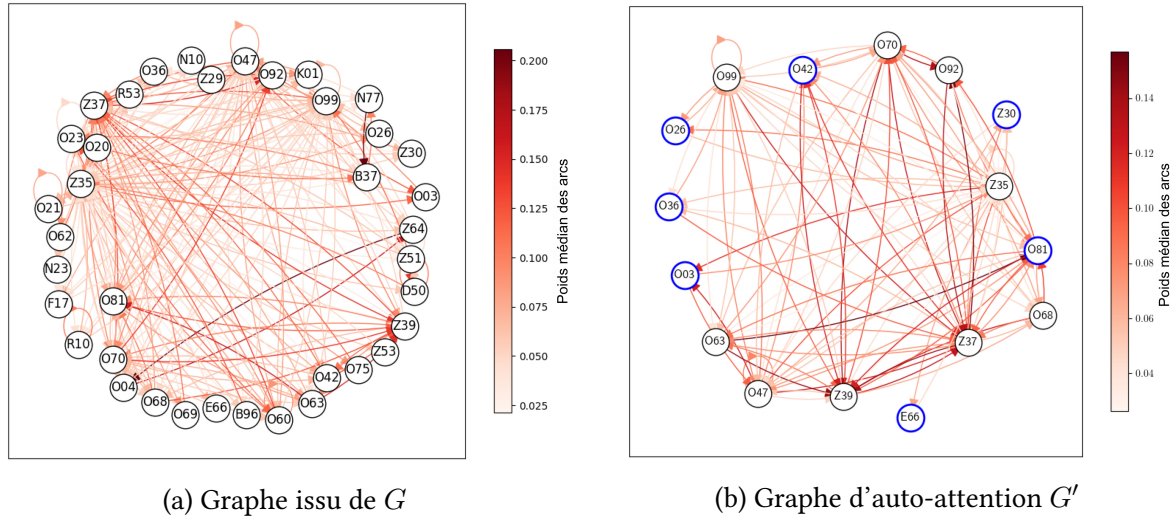


FIGURE 4.4 – Graphes du cas d'usage de l'accouchement. (a) graphe issu de la matrice d'attention globale G , (b) sous-graphe d'auto-attention G' généré à partir des diagnostics d'importance (en bleu) et ceux à qui ils sont liés dans G . Les libellés des codes sont listés dans la Table 4 de l'annexe D.

la prédiction. Nous calculons ensuite la valeur médiane des gradients par diagnostic, que nous trions par ordre décroissant pour identifier les valeurs les plus significatives, comme spécifié dans la Section 4.3.2.

Pour les deux cas d'usage et afin de préserver la confidentialité, nous nous limitons à l'étude des diagnostics et des paires de diagnostics présents dans au moins cinq parcours médicaux distincts. Cette méthode permet de maintenir l'anonymat tout en préservant la pertinence analytique. Par conséquent, certains diagnostics sélectionnés parmi les g diagnostics d'importance peuvent ne pas apparaître dans les graphes si leurs connexions avec d'autres diagnostics ne répondent pas à ce critère. De même, une connexion entre deux diagnostics ne sera pas visible si la paire n'apparaît pas dans au moins cinq séquences distinctes. Ainsi, pour les deux cas d'usage, le nombre de diagnostics significatifs identifiés par la valeur médiane du gradient est choisi arbitrairement de manière à ce que le graphe généré contienne suffisamment de liens.

4.4.3.1 Cas d'usage de l'accouchement

Nous sélectionnons les 10 diagnostics les plus influents selon la valeur médiane de leur gradient. À partir de ces diagnostics, nous extrayons le sous graphe d'auto-attention G' qui relie les nœuds identifiés comme d'importance aux autres nœuds associés dans la matrice d'attention globale, ainsi que les liens entre les nœuds ajoutés (les nœuds qui ne font pas parti de l'ensemble des diagnostics d'importance). Le résultat, illustré dans la Figure 4.4b, est un graphe où les nœuds d'importance ont une bordure bleu tandis que ceux auxquels ils sont liés dans la matrice d'attention globale ont une bordure noire. De plus, le graphe résultant est dirigé et pondéré. Autrement-dit, un arc allant du nœud A au nœud B est pondéré par l'auto-attention médiane que A accorde à B, le poids étant représenté par la couleur de l'arc.

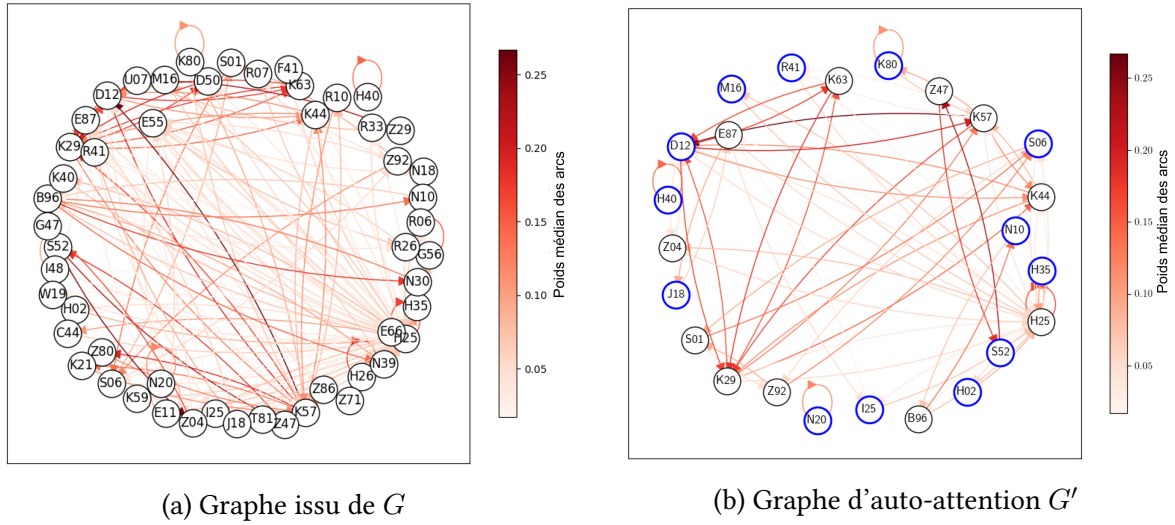


FIGURE 4.5 – Graphes du cas d'usage de l'hypertension. (a) graphe issu de la matrice d'attention globale G , (b) graphe d'auto-attention G' généré à partir des diagnostics d'importance (en bleu) et ceux à qui ils sont liés dans G . Les libellés des codes sont listés dans la Table 4 de l'annexe D.

4.4.3.2 Cas d'usage des maladies hypertensives

Nous identifions les 46 diagnostics les plus significatifs selon la valeur de leur gradient médian. Comme pour le cas d'usage de l'accouchement, nous extrayons le sous-graphe d'auto-attention G' en partant des noeuds représentant les diagnostics d'importance et en ajoutant tous les noeuds auxquels ils sont liés dans la matrice d'attention globale. Les arcs du graphe sont pondérés selon les valeurs des poids d'auto-attention issues de la matrice G . Le graphe pour ce cas d'usage est présenté dans la Figure 4.5b.

4.4.4 Création du graphe de référence $\tilde{G}$

Pour interpréter les liens entre les diagnostics appris par le modèle Behrt à travers les mécanismes d'attention, nous souhaitons les comparer avec une vérité terrain, qui se traduit par une expertise médicale. L'attention, bien que significative dans notre modèle, n'a pas de correspondance directe et évidente dans le contexte médical. Ainsi, afin de valider ou d'invalidier ces liens, nous avons conçu un protocole destiné aux professionnels de santé, impliquant la création d'un graphe de relations entre diagnostics. Ce graphe est conçu pour être non orienté et non pondéré.

Le protocole repose sur deux listes de diagnostics : la première, appelée "**gradients**", contient les diagnostics que nous avons identifiés comme étant influents sur la prédiction. La seconde, appelée "**autres**", inclut les diagnostics qui sont liés aux diagnostics "gradients" selon la matrice d'attention globale. Pour produire un graphe qui reflète fidèlement la réalité clinique des deux cas d'usage présentés, nous avons fait appel à l'expertise de deux médecins de la Drees pour former des clusters. Nous leur avons spécifiquement demandé de lier chaque diagnostic de la liste "autres" à un ou plusieurs diagnostics de la liste "gradients",

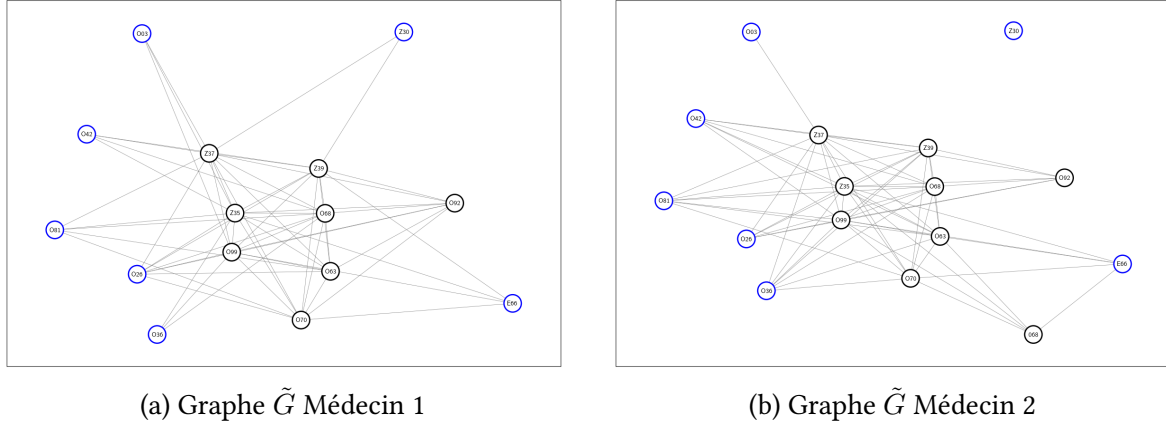


FIGURE 4.6 – Graphes issus des expertises médicales pour le cas d’usage de l’accouchement. Les deux graphes sont générés en demandant aux médecins de relier les diagnostics noirs à un ou plusieurs diagnostics bleus si il existe une corrélation contextuelle entre eux. Les graphes (a) et (b) sont respectivement les résultats du protocole effectué avec les médecins 1 et 2.

en tenant compte de l’existence d’une corrélation contextuelle entre eux. Cette corrélation peut concerner des aspects tels que la comorbidité, la causalité, l’impact sur le traitement, l’implication clinique ou la fréquence de codage, qui désigne la récurrence avec laquelle certains diagnostics sont enregistrés ensemble dans les dossiers médicaux.

Les graphes issus des deux expertises médicales sont présentés dans la Figure 4.6 pour le cas d’usage de l’accouchement et dans la Figure 4.7 pour le cas d’usage des maladies hypertensives. Lorsqu’un nœud est isolé, cela signifie que le médecin n’a trouvé aucune corrélation avec d’autres diagnostics de la liste avec lesquels il devait le lier.

4.4.5 Évaluation de G' par rapport à $\tilde{G}$

Nous évaluons les poids d’auto-attention du graphe G' par rapport à la vérité terrain $\tilde{G}$ en calculant $E(G', \tilde{G})$ (cf. équation 4.5). Comme G' est orienté et que $\tilde{G}$ ne l’est pas, nous considérons les arêtes⁹ de $\tilde{G}$ comme des arcs¹⁰ bidirectionnels. L’objectif est d’analyser G' selon différents seuils qui déterminent les arcs pris en compte. Plus précisément, nous visons à déterminer s’il existe un seuil d’auto-attention au-delà duquel les arcs de G' en commun avec $\tilde{G}$ sont correctement identifiés par le modèle comme étant significatifs, c’est-à-dire que les poids de ces arcs sont supérieurs à ceux des arcs non inclus. Pour chaque seuil établi, nous ne considérons que les arcs de G' dont le poids dépasse ce seuil. Nous retirons ensuite les nœuds isolés de G' et conservons les mêmes nœuds dans $\tilde{G}$. Les résultats de cette évaluation, en fonction des seuils choisis, sont présentés dans la Section 4.5.

9. Arête : Une connexion non orientée entre deux noeuds d’un graphe. 10. Arc : Une connexion orientée (avec une direction) entre deux noeuds d’un graphe.

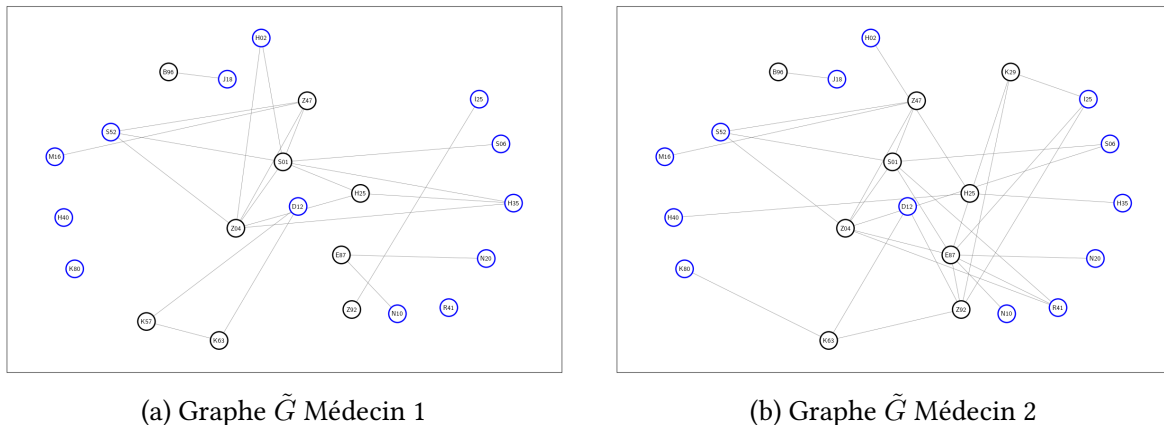


FIGURE 4.7 – Graphes issus des expertises médicales pour le cas d'usage de l'hypertension. Les deux graphes sont générés en demandant aux médecins de relier les diagnostics noirs à un ou plusieurs diagnostics bleus si il existe une corrélation contextuelle entre eux. Les graphes (a) et (b) sont respectivement les résultats du protocole effectué avec les médecins 1 et 2.

4.5 Résultats et discussions

Les courbes d'évolution du score d'évaluation de G' par rapport aux graphes de référence en fonction d'un seuil sont montrées pour chaque cas d'usage dans la Figure 4.8. La similarité des tendances des courbes pour chaque cas d'usage démontre la cohérence des références établies par les médecins, reflétant ainsi une concordance des avis médicaux.

Pour le cas de l'accouchement (cf. Figure 4.8a), nous observons qu'au-dessus d'un seuil de 0,13 ou 0,15, selon le graphe de référence, le score atteint est égal à 1. Cela indique que tous les arcs de G' sont inclus dans $\tilde{G}$, confirmant l'existence d'un seuil au-delà duquel les arcs correspondent précisément à la vérité terrain et démontrant que le modèle attribue correctement des poids d'auto-attention élevés aux arcs pertinents. Avant ce seuil, le score reste positif mais ne dépasse pas 0,25 suggérant que, bien que certains arcs ne soient pas inclus dans le graphe de référence, leur influence est relativement mineure.

En revanche, dans le cas des maladies hypertensives (voir Figure 4.8b), les courbes montrent une décroissance constante, ce qui signifie que plus le seuil de sélection est élevé, moins il y a d'arcs en commun avec le graphe de référence, ou leur poids est inférieur à ceux exclus. Ces résultats indiquent que, dans ce cas d'usage, le modèle ne parvient pas à accorder suffisamment d'attention aux paires de diagnostics validées par l'expertise médicale.

Notre approche permet une évaluation plus granulaire de l'apprentissage du modèle que ce qui serait possible avec des mesures globales telles que la précision ou le rappel. En effet, le cadre que nous proposons facilite la visualisation et l'évaluation des liens d'auto-attention appris par le modèle, et permet d'ajuster l'apprentissage en fonction de ces observations. Dans nos expérimentations, bien que le modèle n'atteigne pas un taux de rappel élevé - diagnostiquant correctement 190 individus sur 2 000 dans le cas de l'accouchement et 514 sur 2 000 dans le cas de l'hypertension - notre approche est appliquée pour déterminer si ces résultats sont la conséquence d'un surapprentissage. Les analyses révèlent que le modèle établit des liens entre diagnostics plus pertinents pour l'accouchement que pour l'hypertension,

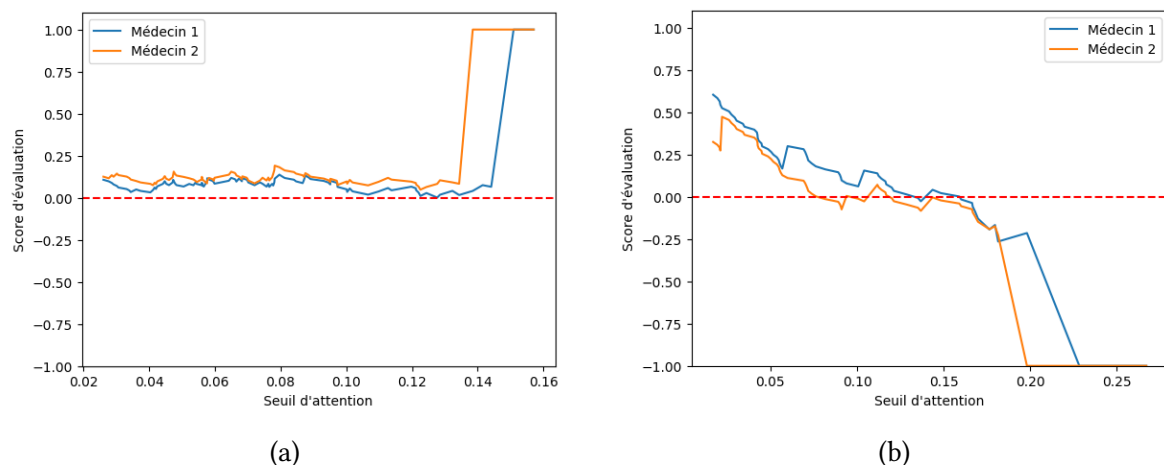


FIGURE 4.8 – Courbes illustrant l'évolution du score d'évaluation en fonction du seuil d'auto-attention utilisé pour sélectionner les arcs dans le graphe d'auto-attention G' . (a) correspond au cas d'usage de l'accouchement, et (b) au cas d'usage des maladies hypertensives. Dans les deux figures, la courbe bleue représente l'évaluation du graphe par rapport à l'expertise médicale du premier médecin, tandis que la courbe orange représente l'évaluation par rapport au deuxième médecin. Une ligne rouge en pointillés située à score=0 est incluse pour faciliter la visualisation des scores d'évaluation devenant négatifs.

ce qui est surprenant compte tenu du meilleur taux de rappel observé pour l'hypertension. Cette anomalie est interprétée comme étant due à la complexité du cas de l'hypertension, qui présente une grande variété de parcours pouvant conduire à des prédictions correctes. Cette interprétation est cohérente avec les graphes des experts médicaux, qui montrent une plus grande concordance dans l'avis des médecins pour l'accouchement (Figure 4.6) que pour l'hypertension (Figure 4.7). Ces observations soulignent la valeur ajoutée de notre approche, dont l'analyse crée un lien direct entre les données d'entraînement utilisées et la performance obtenue, permettant un ajustement potentiel de l'échantillon utilisé.

Dans nos expériences, nous avons développé un protocole pour créer un graphe de référence, qui nous permet d'évaluer le graphe d'auto-attention. Les courbes présentées dans la Figure 4.8 révèlent une observation importante : pour un même cas d'usage, les courbes d'évaluation suivent des tendances très similaires, et ce indépendamment du médecin ayant établi la référence. Cette cohérence entre les graphes de référence établis par différents experts médicaux pour un même cas suggère qu'ils sont globalement d'accord et que notre protocole réussit à établir une vérité terrain cohérente et représentative de l'expertise médicale générale. Cette constatation renforce ainsi la validité et la fiabilité de notre approche d'évaluation, démontrant sa capacité à capturer les connaissances médicales partagées de manière robuste.

Finalement, la méthode proposée offre une évaluation précise de l'apprentissage du modèle basé sur les mécanismes d'attention, ce qui est fondamental dans le domaine médical. En effet, comprendre comment le modèle apprend est crucial pour pouvoir utiliser ses prédictions afin de prendre des décisions éclairées.

4.6 Conclusion et Perspectives

La méthode que nous proposons permet d'évaluer automatiquement les liens entre les tokens appris par un modèle basé sur l'attention, en utilisant l'architecture Behrt, entraînée par le *Lab Santé* (Drees) sur les données du SNDS. Nous analysons l'apprentissage du modèle sur une tâche pouvant être évaluée par un expert : la prédiction d'un diagnostic. Plus précisément, le modèle que nous utilisons est entraîné pour prédire le diagnostic lié à la prochaine visite à l'hôpital à partir d'une séquence de visites antérieures, qui ne contient pas le diagnostic à prédire, que nous appelons un diagnostic incident.

Pour générer les graphes d'auto-attention, nous sélectionnons d'abord les tokens qui ont influencé la prédiction faite par le modèle. Ensuite, nous les relierons aux tokens auxquels ils sont connectés par des poids d'auto-attention dans la matrice d'attention globale. Cette approche nous permet de filtrer les tokens et les liens pour ne conserver que ceux qui sont directement corrélés à la prédiction. En partant des tokens qui influencent la prédiction et en examinant leurs connexions, nous pouvons étudier les tokens liés à la prédiction de manière plus précise. De plus, ce filtrage permet d'obtenir des graphes plus légers et plus lisibles, avec moins de nœuds et d'arcs. Les Figures 4.4a et 4.5a montrent les graphes représentant toutes les interactions entre tokens avant le filtrage, tandis que les Figures 4.4b et 4.5b présentent les graphes après filtrage. Dans nos travaux futurs, nous envisageons d'explorer une approche alternative pour la génération des graphes d'auto-attention. Au lieu de sélectionner arbitrairement le nombre de tokens d'importance pour initialiser le graphe, nous aimerions extraire les composantes faiblement connexes¹¹ après application d'un seuil sur les arcs du graphe complet issu de la matrice d'attention globale G . Cette méthode permettrait une identification plus systématique des groupes de tokens interconnectés avec un poids d'auto-attention élevé (en fonction du seuil) dans la structure d'attention du modèle. Ensuite, l'objectif serait d'étudier si ces composantes faiblement connexes se retrouvent sous forme de composantes connexes¹² dans un graphe de référence incluant tous les diagnostics de la matrice globale. Cette comparaison permettrait d'apprécier de façon plus générale les liens d'auto-attention appris par le modèle. Cependant, cela nécessiterait un travail chronophage pour les experts médicaux, qui devraient créer un graphe de référence plus complet que ceux issus de notre approche actuelle.

Concernant l'extraction des poids d'auto-attention, nos expérimentations actuelles se concentrent sur ceux de la dernière couche du modèle. Par la suite, nous aimerions concevoir une méthode pour intégrer l'auto-attention de toutes les couches, afin d'obtenir une évaluation plus complète du processus d'apprentissage du modèle. De plus, le poids d'auto-attention entre deux tokens est calculé en faisant la moyenne des poids de toutes les têtes d'attention de la dernière couche. Dans nos travaux futurs, nous aimerions analyser chaque tête d'attention individuellement afin d'examiner si les poids associés peuvent être évalués séparément. Enfin, nous agrégeons les poids d'auto-attention entre deux tokens en calculant la médiane de ces poids dans toutes les séquences confondues. Par la suite, nous aimerions étudier l'impact du choix de l'opération d'agrégation.

11. Composante faiblement connexe (graphe orienté) : sous-ensemble maximal de nœuds tel que pour chaque paire de nœuds u et v , il existe un chemin entre u et v dans le graphe non orienté sous-jacent (c'est-à-dire le graphe obtenu en ignorant l'orientation des arcs). 12. Composante connexe (graphe non orienté) : sous-ensemble maximal de nœuds tel que pour chaque paire de nœuds, il existe un chemin les reliant.

Nous appliquons notre méthode d'évaluation sur deux cas d'usage : la prédiction d'un diagnostic signalant un accouchement et un cas révélant un diagnostic d'hypertension essentielle. Pour ce faire, nous évaluons le graphe généré à partir des informations extraites lors de la phase d'inférence du modèle Behrt, en le comparant à un graphe de référence. Cette évaluation consiste à calculer la différence entre la proportion pondérée des arcs communs au graphe d'auto-attention et au graphe de référence, et ceux qui ne sont pas inclus, c'est-à-dire ceux non validés par l'expertise médicale. Nous ajustons le seuil au-dessus duquel les arcs sont considérés dans le graphe d'auto-attention G' pour déterminer si, au-delà d'un certain seuil, les arcs inclus dans le graphe de référence $\tilde{G}$ ont un poids plus significatif que ceux exclus. Cela nous permet de déterminer si un seuil spécifique produit un graphe qui se rapproche le plus de la vérité terrain.

Nous avons développé une méthode pour créer un graphe de référence à partir d'un protocole simple à mettre en œuvre (cf. Section 4.4.4). Dans nos travaux, deux experts médicaux ont été sollicités pour créer ces graphes de référence. Bien que nous les ayons analysés individuellement, une approche future pourrait consister à les unifier pour avoir un unique graphe de référence. De plus, le protocole d'établissement du graphe de référence ne prend pas en compte la pondération des arêtes, car il n'y a pas de correspondance directe avec l'auto-attention dans le contexte médical. À l'avenir, nous aimerions développer une méthode pour pondérer les arêtes établies par l'expertise médicale, permettant ainsi une évaluation plus précise de la distribution d'auto-attention attribuée par le modèle.

Enfin, les expériences menées sur deux cas d'usage distincts (*hypertension* et *accouchement*) ont révélé des résultats contre-intuitifs : le modèle montrant une meilleure performance prédictive (*hypertension*) établie des liens diagnostiques moins pertinents que dans le cas où il semble a priori moins performant (*accouchement*). Cette observation souligne l'importance d'une analyse détaillée des relations apprises, au-delà des mesures d'évaluation globales comme le rappel. Notre approche, en permettant une analyse automatique mais approfondie de l'apprentissage du modèle, met en lumière des biais liés aux données d'entraînement qui ne sont pas détectables avec des métriques classiques. Ainsi, cette méthode facilite non seulement la compréhension des prédictions du modèle, mais renforce également la confiance qu'on peut leur accorder, offrant une perspective d'évaluation plus nuancée et informative.

Cette étude revêt un caractère exploratoire. Dans le futur, nous envisageons d'étendre cette approche à un éventail plus large de pathologies. Cela nous permettrait non seulement de tester la robustesse de notre méthode, mais aussi d'explorer les potentiels chevauchements entre différentes conditions médicales. Ces investigations pourraient révéler des liens inattendus entre pathologies et enrichir notre compréhension des parcours de soins complexes.

Conclusion et Perspectives

Les travaux réalisés dans cette thèse cherchent à répondre à la question de recherche suivante : **"Comment la représentation de l'information influence t-elle l'atteinte d'un objectif donné?"** Pour répondre à cette question, nous nous sommes intéressés à trois objectifs dans différents contextes de modélisation :

Objectif 1 : pallier le fléau de la dimension [3] lors de l'application des réseaux de neurones sur des données à haute dimension.

Objectif 2 : identifier les images contenant des anomalies complexes à détecter.

Objectif 3 : évaluer les liens d'auto-attention appris par un modèle Tranformer [13].

Les études liées aux trois objectifs définis sont présentées dans ce manuscrit sous forme de trois chapitres distincts. Les travaux sur le premier objectif (Chapitre *Représentation des données tabulaires en images pour pallier le fléau de la dimension*) sont précédés d'études préliminaires sur les limites des réseaux convolutifs (CNN) sur des images bruitées. Les expériences réalisées montrent que les CNN sont non seulement robustes au bruit, mais aussi performants pour la tâche de classification d'images abstraites.

Études des limites des CNN sur des images abstraites et/ou bruitées

- La première étude préliminaire a été présentée en 2021 lors du *8th International Workshop on Big Data Analytic Technology for Bioinformatics and Health Informatics @IEEE Big Data* [206].
- La deuxième étude préliminaire a été présentée en 2022 lors du congrès *Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP)* [207].

Les résultats des études préliminaires nous ont encouragés à poursuivre l'idée de transposer les données multidimensionnelles en images, résultant en des images abstraites, pour l'application de CNN. Cette approche, appliquée à un échantillon des données du *Système National des Données de Santé* (SNDS) (cf. annexe A), permet d'obtenir de meilleurs résultats de classification de pathologies avec un temps computationnel moindre en utilisant un modèle hybride CNN-LSTM, comparé aux approches traditionnellement utilisées pour traiter des séries temporelles, telles que le LSTM [35], le 1DCNN-LSTM et un modèle Transformer, Bert [13]. Ainsi, l'approche proposée se positionne comme une nouvelle méthode permettant de pallier le fléau de la dimension tout en exploitant l'exhaustivité des caractéristiques complexes des données.

Cependant, pour améliorer les résultats obtenus et rendre viable la solution proposée, nous souhaitons trouver une approche permettant d'organiser les caractéristiques afin d'ajouter du contexte à chaque pixel. Ainsi, nous pourrions tirer pleinement parti des capacités des CNN à capturer les relations spatiales et contextuelles des motifs d'une image.

Transformation de données tabulaires à haute dimension en images

Ces travaux ont été présentés en 2023 au colloque ORASIS [208] et à la conférence *IEEE International Conference on Image Processing (ICIP)* [209].

Les travaux répondant au deuxième objectif de la thèse, présentés dans le Chapitre *Représentation des images en cartes de saillance pour la détection d'anomalies*, consistent à détecter des anomalies à partir des zones d'importance identifiées dans les images. Plus précisément, notre approche consiste à appliquer un modèle d'explicabilité sur un modèle préalablement entraîné sur une tâche prétexte, générant ainsi des cartes de saillance qui révèlent les caractéristiques importantes des images, puis à utiliser un test *a contrario* [12] sur ces cartes pour détecter les anomalies. Nous comparons notre méthode avec deux autres approches de détection d'anomalies : *f-AnoGAN* [172] et *One-Class Support Vector Machines (OCSVM)* [121].

Les expérimentations ont été réalisées sur différents jeux de données de feuilles/plantes, l'objectif étant de déterminer quelles feuilles/plantes étaient malades. Les résultats montrent que notre approche est plus sensible aux variations de position de l'objet dont on doit détecter l'anomalie que les modèles comparés. Ces résultats sont encourageants pour l'utilisation de notre approche dans un contexte où nous avons un *a priori* sur la position dans laquelle l'objet d'intérêt devrait être placé.

Cependant, notre approche actuelle permet d'obtenir un score de normalité que pour une image à un instant t . À l'avenir, nous souhaitons appliquer notre méthode pour détecter les anomalies de manière dynamique. Autrement dit, étant donné une séquence d'images, déterminer à partir de quel moment une anomalie apparaît, en prenant en compte l'état de l'objet d'intérêt aux pas de temps précédents.

Détection d'anomalies à partir des cartes de saillance

Ces travaux ont été présentés en 2023 au *workshop on Machine Vision for Earth Observation and Environment Monitoring (MVEO) @BMVC* [210] et en poster ^a lors du *8th workshop on Computer Vision in Plant Phenotyping and Agriculture (CVPPA) @ICCV*. De plus, une extension de l'article présenté à MVEO a été publiée dans le journal *International Journal of Applied Earth Observation and Geoinformation* d'Elsevier [211].

a. <https://cvppa2023.github.io/assets/12.pdf>

Enfin, les travaux répondant au troisième objectif (Chapitre *Évaluation des poids d'auto-attention d'un Transformer par représentation sous forme de graphe*) proposent une approche permettant de valider l'apprentissage d'un modèle de type Transformer en représentant ses poids d'auto-attention sous forme de graphe pour les comparer à une référence établie.

Autrement dit, l'approche proposée permet d'évaluer les interactions entre caractéristiques sur lesquelles le modèle accorde de l'importance lors de la phase d'inférence.

Les résultats de ces expérimentations, appliquées à un cas d'usage médical sur un échantillon du SNDS, montrent que le score d'évaluation n'est pas tout à fait en adéquation avec les mesures de performances traditionnellement utilisées, telles que le rappel, qui calcule la proportion de vrais positifs correctement identifiés par rapport au nombre total de vrais positifs. En effet, pour un cas d'usage où le taux de rappel est élevé, notre approche révèle que les interactions apprises par le modèle sont moins pertinentes que pour un cas d'usage avec un rappel plus faible. Ces résultats montrent l'importance et les enjeux d'évaluer les liens d'auto-attention appris par le modèle.

En effet, il arrive que l'analyse des mécanismes internes d'apprentissage ne corresponde pas aux performances observées en sortie du modèle. Par exemple, dans d'autres types de modèles où l'on applique des approches d'explicabilité (cf. Section 3.2.3) pour identifier les caractéristiques déterminantes dans la prédiction, on peut constater que même avec d'excellentes mesures de performance, le modèle peut se focaliser sur des caractéristiques qui sont en réalité des biais. Dans le cas du traitement d'images, par exemple, le modèle peut se concentrer sur des éléments contextuels récurrents plutôt que sur les caractéristiques réellement pertinentes pour la tâche.

Les modèles Transformer ont la particularité d'utiliser des mécanismes d'attention qui permettent de capturer dynamiquement les relations contextuelles entre les différents éléments d'une séquence d'entrée. Nos expérimentations montrent l'importance cruciale d'évaluer l'apprentissage de ces mécanismes, notamment dans des domaines sensibles comme le médical, où les décisions peuvent avoir des conséquences significatives [9]. Plus largement, nos expérimentations ouvrent la voie vers la recherche d'une métrique qui évalue à la fois les sorties du modèle et les mécanismes internes appris.

Évaluation des liens d'auto-attention appris par un Transformer

Ces travaux seront présentés en 2025 aux conférences *The 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)* et Extraction et Gestion des Connaissances (EGC).

Les trois chapitres de ce manuscrit s'articulent autour d'une question centrale tout en abordant des objectifs distincts. Les deux premières études se concentrent sur la représentation des données en entrée du modèle, que ce soit pour permettre l'application d'un certain type d'architecture ou pour révéler les caractéristiques de haut niveau. La deuxième et la troisième études se penchent sur l'apprentissage des modèles. La deuxième étude explore son utilisation pour une nouvelle tâche, tandis que la troisième étude vise à valider le processus d'apprentissage lui-même. Par la suite, nous aimerions combiner les approches proposées durant cette thèse pour, par exemple, proposer une méthode permettant de pallier le fléau de la dimension en transposant les données à haute dimension en images, puis d'appliquer un modèle Transformer sur ces images pour une tâche donnée, tout en ayant la capacité d'évaluer l'apprentissage du modèle grâce aux poids d'auto-attention. Parallèlement, nous pourrions détecter des anomalies dans les données grâce à l'approche *a contrario*, basée sur

les cartes de saillance.

Si nous transposons ces perspectives aux données du SNDS, nous aimerions tester l'approche de détection d'anomalies sur les images générées par notre méthode. Cela permettrait d'identifier les caractéristiques médicales sur lesquelles le modèle se focalise pour faire sa prédiction, et de repérer les caractéristiques qui dévient de cette norme, indiquant une potentielle anomalie dans les prescriptions médicales. De plus, après avoir regroupé les pixels de nos images par nomenclatures pour donner du contexte aux différentes zones d'une image, nous pourrions utiliser un *Vision Transformer* [34]. Ce dernier a la capacité d'utiliser des mécanismes d'auto-attention, et nous pourrions les évaluer pour voir si l'apprentissage dévie des expertises médicales. Toutes ces perspectives ont pour objectif d'enrichir l'utilisation des modèles d'apprentissage profond sur des données complexes comme le SNDS, dans des contextes d'utilisation sensibles, qui peuvent impacter la vie des individus.

Finalement, les travaux réalisés pendant cette thèse explorent la possibilité de changer la représentation de l'information pour optimiser les performances des modèles d'apprentissage profond et les valider, notamment lorsqu'ils sont appliqués à des données complexes, que ce soit par leurs caractéristiques ou leurs dimensions. Les résultats obtenus montrent que la solution pour pallier certains défis liés à l'apprentissage profond n'est pas toujours la modification des architectures utilisées, mais peut être l'adaptation de la représentation de l'information, car celle-ci a un impact significatif sur l'atteinte des objectifs définis.

Publications

Conférences/Workshops internationaux

- ★ Rebecca Leygonie, Guillaume Vimont, *Empirical analysis of the residual information needed for image classification.*, 8th International Workshop on Big Data Analytic Technology for Bioinformatics and Health Informatics, IEEE International Conference on Big Data, (pp. 2813-2817), 2021.
- ★ Rebecca Leygonie, Sylvain Lobry, Laurent Wendling, *An a contrario approach for plant disease detection*, Workshop on Machine Vision for Earth Observation and Environment Monitoring (MVEO), The British Machine Vision Conference (BMVC), 2023.
- ★ Rebecca Leygonie, Sylvain Lobry, Guillaume Vimont, Laurent Wendling, *Transforming Multidimensional Data into Images to Overcome the Curse of Dimensionality.*, IEEE International Conference on Image Processing (ICIP), (pp. 700-704), 2023.
- ★ **(Accepté)** Rebecca Leygonie, Sylvain Lobry, Laurent Wendling, *Evaluating Transformers Learning by Representing Attention Weights as a Graph*, The 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), 2025.

Journal international

- ★ Rebecca Leygonie, Sylvain Lobry, Laurent Wendling, *Can we detect plant diseases without prior knowledge of their existence ?*, International Journal of Applied Earth Observation and Geoinformation, Elsevier.

Congrès/Colloques nationaux

- ★ Rebecca Leygonie, Sylvain Lobry, Laurent Wendling, *Apprentissage profond pour la classification de QR Codes bruités.*, Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP), 2022.
- ★ Rebecca Leygonie, Sylvain Lobry, Guillaume Vimont, Laurent Wendling, *Transposition de données multidimensionnelles en images pour pallier le fléau de la dimension.*, ORASIS, 2023.
- ★ **(Accepté)** Rebecca Leygonie, Sylvain Lobry, Laurent Wendling, *Représentation de l'auto-attention par graphe pour évaluer les Transformers*, Extraction et Gestion des Connaissances (EGC), 2025.

Bibliographie

- [1] Seref SAGIROGLU et Duygu SINANC. « Big data : A review ». In : *International Conference on Collaboration Technologies and Systems (CTS)*. Institute of Electrical et Electronics Engineers (IEEE). 2013, p. 42-47.
- [2] Ian GOODFELLOW, Yoshua BENGIO et Aaron COURVILLE. *Deep learning*. Massachusetts Institute of Technology (MIT) press, 2016.
- [3] Richard BELLMAN. « Adaptive Control Processes - A Guided Tour ». In : *Princeton Legacy Library* 2045 (1961), p. 1-258.
- [4] Douglas M HAWKINS. « The Problem of Overfitting ». In : *Journal of Chemical Information and Computer Sciences* 44.1 (2004), p. 1-12.
- [5] Nitish SRIVASTAVA, Geoffrey HINTON, Alex KRIZHEVSKY, Ilya SUTSKEVER et Ruslan SALAKHUTDINOV. « Dropout : A Simple Way to Prevent Neural Networks from Overfitting ». In : *Journal of Machine Learning Research (JMLR)* 15.1 (2014), p. 1929-1958.
- [6] Connor SHORTEN et Taghi M KHOSHGOFTAAR. « A survey on Image Data Augmentation for Deep Learning ». In : *Journal of Big Data* 6.1 (2019), p. 1-48.
- [7] Sinno Jialin PAN et Qiang YANG. « A Survey on Transfer Learning ». In : *Transactions on Knowledge and Data Engineering (TKDE)* 22.10 (2009), p. 1345-1359.
- [8] Sergey IOFFE et Christian SZEGEDY. « Batch normalization : Accelerating deep network training by reducing internal covariate shift ». In : *International Conference on Machine Learning (ICML)*. Proceedings of Machine Learning Research (PMLR). 2015, p. 448-456.
- [9] Cynthia RUDIN. « Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead ». In : *Nature Machine Intelligence* 1.5 (2019), p. 206-215.
- [10] J. Ross QUINLAN. « Induction of Decision Trees ». In : *Machine Learning* 1 (1986), p. 81-106.
- [11] Alejandro Barredo ARRIETA, Natalia DIÁZ-RODRÍGUEZ, Javier DEL SER, Adrien BENNETOT, Siham TABIK, Alberto BARBADO, Salvador GARCÍA, Sergio GIL-LÓPEZ, Daniel MOLINA,

-
- Richard BENJAMINS, Raja CHATILA et Francisco HERRERA. « Explainable Artificial Intelligence (XAI) : Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI ». In : *Information Fusion* 58 (2020), p. 82-115.
- [12] Agnès DESOLNEUX. « When the a contrario approach becomes generative ». In : *International Journal of Computer Vision (IJCV)* 116.1 (2016), p. 46-65.
- [13] Ashish VASWANI, Noam SHAZEER, Niki PARMAR, Jakob USZKOREIT, Llion JONES, Aidan N GOMEZ, Łukasz KAISER et Illia POLOSUKHIN. « Attention Is All You Need ». In : *31st International Conference on Neural Information Processing Systems (NIPS)* (2017), p. 6000-6010.
- [14] Jia DENG, Wei DONG, Richard SOCHER, Li-Jia LI, Kai LI et Li FEI-FEI. « ImageNet : A Large-Scale Hierarchical Image Database ». In : *Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2009, p. 248-255.
- [15] Tsung-Yi LIN, Michael MAIRE, Serge BELONGIE, James HAYS, Pietro PERONA, Deva RAMANAN, Piotr DOLLÁR et C Lawrence ZITNICK. « Microsoft COCO : Common Objects in Context ». In : *13th European Conference on Computer Vision (ECCV)*. Springer. 2014, p. 740-755.
- [16] Kuniyiko FUKUSHIMA. « Neocognitron : A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position ». In : *Biological cybernetics* 36.4 (1980), p. 193-202.
- [17] Yann LECUN, Bernhard BOSER, John S DENKER, Donnie HENDERSON, Richard E HOWARD, Wayne HUBBARD et Lawrence D JACKEL. « Backpropagation Applied to Handwritten Zip Code Recognition ». In : *Neural computation* 1.4 (1989), p. 541-551.
- [18] Yann LECUN, Léon BOTTOU, Yoshua BENGIO et Patrick HAFFNER. « Gradient-Based Learning Applied to Document Recognition ». In : *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)* 86.11 (1998), p. 2278-2324.
- [19] Alex KRIZHEVSKY, Ilya SUTSKEVER et Geoffrey E HINTON. « ImageNet Classification with Deep Convolutional Neural Networks ». In : *Advances in Neural Information Processing Systems (NIPS)* 25 (2012).
- [20] Matthew D ZEILER et Rob FERGUS. « Visualizing and Understanding Convolutional Networks ». In : *13th European Conference on Computer Vision (ECCV)*. Springer. 2014, p. 818-833.
- [21] Karen SIMONYAN et Andrew ZISSERMAN. « Very deep convolutional networks for large-scale image recognition ». In : *3rd International Conference on Learning Representations (ICLR)*. 2015.
- [22] Christian SZEGEDY, Wei LIU, Yangqing JIA, Pierre SERMANET, Scott REED, Dragomir ANGUELOV, Dumitru ERHAN, Vincent VANHOUCKE et Andrew RABINOVICH. « Going

- Deeper with Convolutions ». In : *Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2015, p. 1-9.
- [23] Rupesh Kumar SRIVASTAVA, Klaus GREFF et Jürgen SCHMIDHUBER. « Training Very Deep Networks ». In : *Advances in Neural Information Processing Systems (NIPS)* 28 (2015).
- [24] Kaiming HE, Xiangyu ZHANG, Shaoqing REN et Jian SUN. « Deep Residual Learning for Image Recognition ». In : *Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2016, p. 770-778.
- [25] Gao HUANG, Zhuang LIU, Laurens VAN DER MAATEN et Kilian Q WEINBERGER. « Densely Connected Convolutional Networks ». In : *Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2017, p. 4700-4708.
- [26] Andrew G HOWARD, Menglong ZHU, Bo CHEN, Dmitry KALENICHENKO, Weijun WANG, Tobias WEYAND, Marco ANDREETTO et Hartwig ADAM. « MobileNets : Efficient convolutional neural networks for mobile vision applications ». In : *arXiv preprint arXiv :1704.04861* (2017).
- [27] Mingxing TAN et Quoc LE. « EfficientNet : Rethinking Model Scaling for Convolutional Neural Networks ». In : *International Conference on Machine Learning (ICML)*. Proceedings of Machine Learning Research (PLMR). 2019, p. 6105-6114.
- [28] David E RUMELHART, Geoffrey E HINTON et Ronald J WILLIAMS. « Learning representations by back-propagating errors ». In : *Nature* 323.6088 (1986), p. 533-536.
- [29] Olga RUSSAKOVSKY, Jia DENG, Hao SU, Jonathan KRAUSE, Sanjeev SATHEESH, Sean MA, Zhiheng HUANG, Andrej KARPATHY, Aditya KHOSLA, Michael BERNSTEIN, Alexander C. BERG et Li FEI-FEI. « ImageNet Large Scale Visual Recognition Challenge ». In : *International Journal of Computer Vision (IJCV)* 115.3 (2015), p. 211-252.
- [30] Yoshua BENGIO, Patrice SIMARD et Paolo FRASCONI. « Learning Long-Term Dependencies with Gradient Descent is Dificult ». In : *Transactions on Neural Networks* 5.2 (1994), p. 157-166.
- [31] Mark SANDLER, Andrew HOWARD, Menglong ZHU, Andrey ZHMOGINOV et Liang-Chieh CHEN. « MobileNetV2 : Inverted Residuals and Linear Bottlenecks ». In : *Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2018, p. 4510-4520.

-
- [32] Jie HU, Li SHEN et Gang SUN. « Squeeze-and-Excitation Networks ». In : *Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2018, p. 7132-7141.
- [33] Prajit RAMACHANDRAN, Barret ZOPH et Quoc V LE. « Searching for Activation Functions ». In : *arXiv preprint arXiv :1710.05941* (2017).
- [34] Alexey DOSOVITSKIY, Lucas BEYER, Alexander KOLESNIKOV, Dirk WEISSENBORN, Xiaohua ZHAI, Thomas UNTERTHINER, Mostafa DEHGHANI, Matthias MINDERER, Georg HEIGOLD, Sylvain GELLY, Jakob USZKOREIT et Neil HOULSBY. « An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale ». In : *International Conference on Learning Representations (ICLR)*. 2020.
- [35] Sepp HOCHREITER et Jürgen SCHMIDHUBER. « Long Short-Term Memory ». In : *Neural computation* 9.8 (1997), p. 1735-1780.
- [36] Alex GRAVES et Jürgen SCHMIDHUBER. « Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures ». In : *Neural networks* 18.5-6 (2005), p. 602-610.
- [37] Kyunghyun CHO, Bart van MERRIËNBOER, Çağlar GÜLÇEHRE, Dzmitry BAHDANAU, Fethi BOUGARES, Holger SCHWENK et Yoshua BENGIO. « Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation ». In : *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, p. 1724-1734.
- [38] Tianyang LIN, Yuxin WANG, Xiangyang LIU et Xipeng QIU. « A survey of transformers ». In : *AI open* 3 (2022), p. 111-132.
- [39] Jacob DEVLIN, Ming-Wei CHANG, Kenton LEE et Kristina TOUTANOVA. « BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding ». In : *North American Chapter of the Association for Computational Linguistics (ACL)*. 2019.
- [40] Zhilin YANG, Zihang DAI, Yiming YANG, Jaime CARBONELL, Russ R SALAKHUTDINOV et Quoc V LE. « XLNet : Generalized Autoregressive Pretraining for Language Understanding ». In : *Advances in Neural Information Processing Systems (NIPS)* 32 (2019).
- [41] Yinhan LIU, Myle OTT, Naman GOYAL, Jingfei DU, Mandar JOSHI, Danqi CHEN, Omer LEVY, Mike LEWIS, Luke ZETTLEMOYER et Veselin STOYANOV. « Roberta : A robustly optimized bert pretraining approach ». In : *arXiv preprint arXiv :1907.11692* (2019).
- [42] Victor SANH, Lysandre DEBUT, Julien CHAUMOND et Thomas WOLF. « DistilBERT, a distilled version of BERT : smaller, faster, cheaper and lighter ». In : *33th Conference on Neural Information Processing Systems (NIPS)*. 2019.
- [43] Colin RAFFEL, Noam SHAZEER, Adam ROBERTS, Katherine LEE, Sharan NARANG, Michael MATENA, Yanqi ZHOU, Wei LI et Peter J LIU. « Exploring the Limits of

- Transfer Learning with a Unified Text-to-Text Transformer ». In : *Journal of Machine Learning Research (JMLR)* 21.140 (2020), p. 1-67.
- [44] Alec RADFORD, Karthik NARASIMHAN, Tim SALIMANS et Ilya SUTSKEVER. « Improving language understanding by generative pre-training ». In : *OpenAI* (2018).
- [45] Alec RADFORD, Jeffrey WU, Rewon CHILD, David LUAN, Dario AMODEI et Ilya SUTSKEVER. « Language Models are Unsupervised Multitask Learners ». In : *OpenAI blog* 1.8 (2019), p. 9.
- [46] Tom BROWN, Benjamin MANN, Nick RYDER, Melanie SUBBIAH, Jared D KAPLAN, Prafulla DHARIWAL, Arvind NEELAKANTAN, Pranav SHYAM, Girish SASTRY, Amanda ASKELL, Sandhini AGARWAL, Ariel HERBERT-VOSS, Gretchen KRUEGER, Tom HENIGHAN, Rewon CHILD, Aditya RAMESH, Daniel ZIEGLER, Jeffrey WU, Clemens WINTER, Chris HESSE, Mark CHEN, Eric SIGLER, Mateusz LITWIN, Scott GRAY, Benjamin CHESSE, Jack CLARK, Christopher BERNER, Sam MCCANDLISH, Alec RADFORD, Ilya SUTSKEVER et Dario AMODEI. « Language Models are Few-Shot Learners ». In : *Advances in Neural Information Processing Systems (NIPS)*. T. 33. Curran Associates, Inc., 2020, p. 1877-1901.
- [47] Josh ACHIAM, Steven ADLER, Sandhini AGARWAL, Lama AHMAD, Ilge AKKAYA, Floren-
cia Leoni ALEMAN, Diogo ALMEIDA, Janko ALTENSCHMIDT, Sam ALTMAN, Shyamal
ANADKAT et al. « Gpt-4 technical report ». In : *Open AI* (2023).
- [48] Shuiwang JI, Wei XU, Ming YANG et Kai YU. « 3D Convolutional Neural Networks
for Human Action Recognition ». In : *Transactions on Pattern Analysis and Machine
Intelligence (TPAMI)* 35.1 (2012), p. 221-231.
- [49] Jeffrey DONAHUE, Lisa ANNE HENDRICKS, Sergio GUADARRAMA, Marcus ROHRBACH,
Subhashini VENUGOPALAN, Kate SAENKO et Trevor DARRELL. « Long-term Recurrent
Convolutional Networks for Visual Recognition and Description ». In : *Conference on
Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical et Electronics
Engineers/Computer Vision Foundation (IEEE/CVF). 2015, p. 2625-2634.
- [50] Xingjian SHI, Zhourong CHEN, Hao WANG, Dit-Yan YEUNG, Wai-Kin WONG et Wang-
chun WOO. « Convolutional LSTM Network : A Machine Learning Approach for
Precipitation Nowcasting ». In : *Advances in Neural Information Processing Systems
(NIPS)* 28 (2015).
- [51] Nicolas BALLAS, Li YAO, Christopher J PAL et Aaron COURVILLE. « Delving Deeper
into Convolutional Networks for Learning Video Representations ». In : *International
Conference on Learning Representations (ICLR)*. 2016.
- [52] Karen SIMONYAN et Andrew ZISSERMAN. « Two-Stream Convolutional Networks
for Action Recognition in Videos ». In : *Advances in Neural Information Processing
Systems (NIPS)* 27 (2014).

-
- [53] Limin WANG, Yuanjun XIONG, Zhe WANG, Yu QIAO, Dahua LIN, Xiaoou TANG et Luc VAN GOOL. « Temporal Segment Networks : Towards Good Practices for Deep Action Recognition ». In : *European Conference on Computer Vision (ECCV)*. Springer. 2016, p. 20-36.
 - [54] Gedas BERTASIUS, Heng WANG et Lorenzo TORRESANI. « Is Space-Time Attention All You Need for Video Understanding ? » In : *International Conference on Machine Learning (ICML)*. T. 2. 2021, p. 4.
 - [55] Anurag ARNAB, Mostafa DEHGHANI, Georg HEIGOLD, Chen SUN, Mario LUČIĆ et Cordelia SCHMID. « ViViT : A Video Vision Transformer ». In : *International Conference on Computer Vision (ICCV)*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2021, p. 6836-6846.
 - [56] Michail TARASIOU, Erik CHAVEZ et Stefanos ZAFEIRIOU. « ViTs for SITS : Vision Transformers for Satellite Image Time Series ». In : *Institute of Electrical and Electronics Engineers/Computer Vision Foundation (IEEE/CVF) Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, p. 10418-10428.
 - [57] Laurens VAN DER MAATEN, Eric POSTMA et Jaap VAN DEN HERIK. « Dimensionality Reduction : A Comparative ». In : *Journal of Machine Learning Research (JMLR)* 10.66-71 (2009), p. 13.
 - [58] Isabelle GUYON, Jason WESTON, Stephen BARNHILL et Vladimir VAPNIK. « Gene Selection for Cancer Classification using Support Vector Machines ». In : *Machine Learning* 46 (2002), p. 389-422.
 - [59] Ronald Aylmer FISHER. « Statistical methods for research workers ». In : *Breakthroughs in statistics : Methodology and distribution*. Springer, 1970, p. 66-70.
 - [60] Karl PEARSON. « X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling ». In : *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50.302 (1900), p. 157-175.
 - [61] Karl PEARSON. *On the Theory of Contingency and Its Relation to Association and Normal Correlation*. 2013.
 - [62] Harold HOTELLING. « Analysis of a complex of statistical variables into principal components. » In : *Journal of Educational Psychology* 24.6 (1933), p. 417.
 - [63] Ronald A FISHER. « The use of multiple measurements in taxonomic problems ». In : *Annals of Eugenics* 7.2 (1936), p. 179-188.
 - [64] B ANS, J HÉRAULT et C JUTTEN. « Architectures neuromimétiques adaptatives : Détection de primitives ». In : *Cognitiva* 85 (1985), p. 593-597.
 - [65] Charles SPEARMAN. « General Intelligence, Objectively Determined and Measured. » In : *American Journal of Psychology* 15 (1904), p. 201-293.

- [66] Gene H GOLUB et Christian REINSCH. « Singular Value Decomposition and Least Squares Solutions ». In : *Handbook for Automatic Computation : Volume II : Linear Algebra*. Springer, 1971, p. 134-151.
- [67] Mark A KRAMER. « Nonlinear Principal Component Analysis Using Autoassociative Neural Networks ». In : *American Institute of Chemical Engineers (AIChE) journal* 37.2 (1991), p. 233-243.
- [68] Mohsena ASHRAF, Farzana ANOWAR, Jahanggir H SETU, Atiqul I CHOWDHURY, Esh-tiak AHMED, Ashraful ISLAM et Abdullah AL-MAMUN. « A Survey on Dimensionality Reduction Techniques for Time-Series Data ». In : *Institute of Electrical and Electronics Engineers (IEEE) Access* 11 (2023), p. 42909-42923.
- [69] Bernhard SCHÖLKOPF, Alexander SMOLA et Klaus-Robert MÜLLER. « Nonlinear Component Analysis as a Kernel Eigenvalue Problem ». In : *Neural Computation* 10.5 (1998), p. 1299-1319.
- [70] Mikhail BELKIN et Partha NIYOGI. « Laplacian Eigenmaps for Dimensionality Reduction and Data Representation ». In : *Neural Computation* 15.6 (2003), p. 1373-1396.
- [71] Xiaofei HE et Partha NIYOGI. « Locality Preserving Projections ». In : *Advances in Neural Information Processing Systems (NIPS)* 16 (2003).
- [72] Sam T ROWEIS et Lawrence K SAUL. « Nonlinear Dimensionality Reduction by Locally Linear Embedding ». In : *Science* 290.5500 (2000), p. 2323-2326.
- [73] Joshua B TENENBAUM, Vin de SILVA et John C LANGFORD. « A Global Geometric Framework for Nonlinear Dimensionality Reduction ». In : *Science* 290.5500 (2000), p. 2319-2323.
- [74] Kilian Q WEINBERGER, Fei SHA et Lawrence K SAUL. « Learning a Kernel Matrix for Nonlinear Dimensionality Reduction ». In : *21th International Conference on Machine Learning (ICML)*. 2004, p. 106.
- [75] Ronald R COIFMAN et Stéphane LAFON. « Diffusion maps ». In : *Applied and Computational Harmonic Analysis* 21.1 (2006), p. 5-30.
- [76] Hagit SHATKAY. « The Fourier Transform-A Primer ». In : *Brown University* (1995), p. 1-20.
- [77] Tim EDWARDS. « Discrete Wavelet Transforms : Theory and Implementation ». In : *Universidad de 1991* (1991), p. 28-35.
- [78] Tomaso POGGIO, Hrushikesh MHASKAR, Lorenzo ROSASCO, Brando MIRANDA et Qianli LIAO. « Why and When Can Deep—but Not Shallow—Networks Avoid the Curse of Dimensionality : a Review ». In : *International Journal of Automation and Computing* 14.5 (2017), p. 503-519.

-
- [79] Zhuang LIU, Mingjie SUN, Tinghui ZHOU, Gao HUANG et Trevor DARRELL. « Re-thinking the Value of Network Pruning ». In : *International Conference on Learning Representations (ICLR)*. 2019.
 - [80] Kenneth ABEND. « How Convolutional Neural Networks Defy the Curse of Dimensionality : Deep Learning Explained ». In : *Authorea Preprints* (2023).
 - [81] Vamshi C MADALA, Shivkumar CHANDRASEKARAN et Jason BUNK. « CNNs Avoid Curse of Dimensionality by Learning on Patches ». In : *Open Journal of Signal Processing* 4 (2023), p. 233-241.
 - [82] Amit DAMRI, Mark LAST et Niv COHEN. « Towards efficient image-based representation of tabular data ». In : *Neural Computing and Applications* 36.2 (2024), p. 1023-1043.
 - [83] Huimei HAN, Ying LI et Xingquan ZHU. « Convolutional neural network learning for generic data classification ». In : *Information Sciences* 477 (2019), p. 448-465.
 - [84] Alok SHARMA, Edwin VANS, Daichi SHIGEMIZU, Keith A BOROEVICH et Tatsuhiko TSUNODA. « DeepInsight : A methodology to transform a non-image data to an image for convolution neural network architecture ». In : *Scientific Reports* 9.1 (2019), p. 11399.
 - [85] Laurens van der MAATEN et Geoffrey HINTON. « Visualizing Data using t-SNE ». In : *Journal of Machine Learning Research (JMLR)* 9 (2008), p. 2579-2605.
 - [86] Omid BAZGIR, Ruibo ZHANG, Saugato Rahman DHRUBA, Raziur RAHMAN, Souparno GHOSH et Ranadip PAL. « Representation of features as images with neighborhood dependencies for compatibility with convolutional neural networks ». In : *Nature Communications* 11.1 (2020), p. 4391.
 - [87] Caryl E RUSBULT et Isabella M ZEMBRODT. « Responses to Dissatisfaction in Romantic Involvements : A Multidimensional Scaling Analysis ». In : *Journal of Experimental Social Psychology* 19.3 (1983), p. 274-293.
 - [88] Yitan ZHU, Thomas BRETTIN, Fangfang XIA, Alexander PARTIN, Maulik SHUKLA, Hyunseung YOO, Yvonne A EVRARD, James H DOROSHOW et Rick L STEVENS. « Converting tabular data into images for deep learning with convolutional neural networks ». In : *Scientific Reports* 11.1 (2021), p. 11325.
 - [89] Seid Miad ZANDAVI, Derong LIU, Vera CHUNG, Ali ANAISSI et Fatemeh VAFAEE. « Fotomics : fourier transform-based omics imagification for deep learning-based cell-identity mapping using single-cell omics profiles ». In : *Artificial Intelligence Review* 56.7 (2023), p. 7263-7278.
 - [90] Tan Jin SOON. « QR code ». In : *synthesis Journal* 2008 (2008), p. 59-78.
 - [91] Irving S REED et Gustave SOLOMON. « Polynomial Codes Over Certain Finite Fields ». In : *Journal of the Society for Industrial and Applied Mathematics (SIAM)* 8.2 (1960), p. 300-304.

- [92] Stephen M STIGLER. *The history of statistics*. Harvard University Press, 1990.
- [93] Diederik KINGMA et Jimmy BA. « Adam : A Method for Stochastic Optimization ». In : *International Conference on Learning Representations (ICLR)*. 2015.
- [94] Lina HUO, Jianxing ZHU, Pradeep Kumar SINGH et Pljonkin Anton PAVLOVICH. « Research on QR image code recognition system based on artificial intelligence algorithm ». In : *Journal of Intelligent Systems* 30.1 (2021), p. 855-867.
- [95] Tzu-Han CHOU, Chuan-Sheng HO et Yan-Fu KUO. « QR code detection using convolutional neural networks ». In : *International Conference on Advanced Robotics and Intelligent Systems (ARIS)*. Institute of Electrical et Electronics Engineers (IEEE). 2015, p. 1-5.
- [96] Jinbo PENG, Song YUAN et Xin YUAN. « QR Code Detection with Faster-RCNN Based on FPN ». In : *International Conference on Artificial Intelligence and Security (ICAIS)*. Springer. 2020, p. 434-443.
- [97] Tapas KANUNGO, Robert M HARALICK et Ihsin PHILLIPS. « Global and Local Document Degradation Models ». In : *2nd International Conference on Document Analysis and Recognition (ICDAR)*. Institute of Electrical et Electronics Engineers (IEEE). 1993, p. 730-734.
- [98] Vijaya Kumar Bajjer RAMANNA, Syed Saqib BUKHARI et Andreas DENGEL. « Document Image Dewarping using Deep Learning. » In : *International Conference on Pattern Recognition Applications and Methods (ICPRAM)*. 2019, p. 524-531.
- [99] Nobuyuki OTSU. « A threshold selection method from gray-level histograms ». In : *Automatica* 11.285-296 (1975), p. 23-27.
- [100] Mushfika Sharmin RAHMAN, Atiqul Islam CHOWDHURY, KM Tawsik ZAWAD, Tasnim Mashrur MAHEE, Rifat AHMED et Nazmus SAKIB. « An Approach to Facilitate Business System by Multiple Barcode Detection using Faster RCNN ». In : *International Journal of Applied Information Systems* 12.26 (2019), p. 10-15.
- [101] Wendy Cahya KURNIAWAN, Hiroshi OKUMURA et Anik Nur HANDAYANI. « An Improvement on QR Code Limit Angle Detection using Convolution Neural Network ». In : *International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*. T. 6. 2019, p. 234-238.
- [102] Reuven Y RUBINSTEIN et Dirk P KROESE. *The cross-entropy method : a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*. T. 133. Springer, 2004.
- [103] Wilson HUON. *QR Error Correction Helps and Hinders Scanning*. <https://huonw.github.io/blog/2021/09/qr-error-correction/>. 2021.
- [104] Yukun ZHU, Ryan KIROS, Rich ZEMEL, Ruslan SALAKHUTDINOV, Raquel URTASUN, Antonio TORRALBA et Sanja FIDLER. « Aligning Books and Movies : Towards Story-

-
- like Visual Explanations by Watching Movies and Reading Books ». In : *International Conference on Computer Vision (ICCV)*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2015, p. 19-27.
- [105] Houtao DENG, George RUNGER, Eugene TUV et Martyanov VLADIMIR. « A Time Series Forest for Classification and Feature Extraction ». In : *Information Sciences* 239 (2013), p. 142-153.
- [106] Patrick SCHÄFER. « The BOSS is concerned with time series classification in the presence of noise ». In : *Data Mining and Knowledge Discovery* 29.6 (2015), p. 1505-1530.
- [107] Varun CHANDOLA, Arindam BANERJEE et Vipin KUMAR. « Anomaly Detection : A Survey ». In : *Association for Computing Machinery (ACM) computing surveys (CSUR)* 41.3 (2009), p. 1-58.
- [108] Frank E GRUBBS. « Procedures for Detecting Outlying Observations in Samples ». In : *Technometrics* 11.1 (1969), p. 1-21.
- [109] Guansong PANG, Chunhua SHEN, Longbing CAO et Anton Van Den HENGEL. « Deep Learning for Anomaly Detection : A Review ». In : *Association for Computing Machinery (ACM) Computing Surveys (CSUR)* 54.2 (2021), p. 1-38.
- [110] Thibaud EHRET, Axel DAVY, Jean-Michel MOREL et Mauricio DELBRACIO. « Image Anomalies : A Review and Synthesis of Detection Methods ». In : *Journal of Mathematical Imaging and Vision* 61 (2019), p. 710-743.
- [111] William E FRY. *Principles of Plant Disease Management*. Academic Press, 2012.
- [112] Jan C ZADOKS et Richard D SCHEIN. « Epidemiology and plant disease management. » In : *Epidemiology and Plant Disease Management* (1979).
- [113] Serge SAVARY, Andrea FICKE, Jean-Noël AUBERTOT et Clayton HOLLIER. « Crop losses due to diseases and their implications for global food production losses and food security ». In : *Food security* 4.4 (2012), p. 519-537.
- [114] Jiajia LI, Dong CHEN, Xinda QI, Zhaojian LI, Yanbo HUANG, Daniel MORRIS et Xiaobo TAN. « Label-Efficient Learning in Agriculture : A Comprehensive Review ». In : *Computers and Electronics in Agriculture* 215 (2023), p. 108412.
- [115] Jie YANG, Ruijie XU, Zhiquan QI et Yong SHI. « Visual anomaly detection for images : A systematic survey ». In : *Procedia computer science* 199 (2022), p. 471-478.
- [116] Murray ROSENBLATT. « Remarks on Some Nonparametric Estimates of a Density Function ». In : *The annals of mathematical statistics* (1956), p. 832-837.
- [117] Evelyn FIX. *Discriminatory analysis : Nonparametric discrimination : Consistency properties*. T. 1. United States Air Force (USAF) School of Aviation Medicine, 1985.

- [118] Christopher M BISHOP. « Pattern Recognition and Machine Learning ». In : *Springer google schola* 2 (2006), p. 1122-1128.
- [119] Diederik P KINGMA et Max WELLING. « Auto-Encoding Variational Bayes ». In : *stat* 1050 (2014), p. 1.
- [120] David MJ TAX et Robert PW DUIN. « Support Vector Data Description ». In : *Machine Learning* 54 (2004), p. 45-66.
- [121] Bernhard SCHÖLKOPF, John C PLATT, John SHAWE-TAYLOR, Alex J SMOLA et Robert C WILLIAMSON. « Estimating the Support of a High-Dimensional Distribution ». In : *Neural Computation* 13.7 (2001), p. 1443-1471.
- [122] Poojan OZA et Vishal M PATEL. « One-Class Convolutional Neural Network ». In : *Signal Processing Letters* 26.2 (2018), p. 277-281.
- [123] Pramuditha PERERA et Vishal M PATEL. « Learning Deep Features for One-Class Classification ». In : *Transactions on Image Processing (TIP)* 28.11 (2019), p. 5450-5463.
- [124] Hadi HOJJATI, Thi Kieu Khanh Ho et Narges ARMANFARD. « Self-Supervised Anomaly Detection : A Survey and Outlook ». In : *arXiv preprint arXiv :2205.05173* (2022).
- [125] Izhak GOLAN et Ran EL-YANIV. « Deep Anomaly Detection Using Geometric Transformations ». In : *Advances in Neural Information Processing Systems (NIPS)* 31 (2018).
- [126] Chun-Liang LI, Kihyuk SOHN, Jinsung YOON et Tomas PFISTER. « CutPaste : Self-Supervised Learning for Anomaly Detection and Localization ». In : *Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2021, p. 9664-9674.
- [127] Ting CHEN, Simon KORNBLITH, Mohammad NOROUZI et Geoffrey HINTON. « A Simple Framework for Contrastive Learning of Visual Representations ». In : *International Conference on Machine Learning (ICML)*. Proceedings of Machine Learning Research (PMLR). 2020, p. 1597-1607.
- [128] Jihoon TACK, Sangwoo MO, Jongheon JEONG et Jinwoo SHIN. « CSI : Novelty Detection via Contrastive Learning on Distributionally Shifted Instances ». In : *Advances in Neural Information Processing Systems (NIPS)* 33 (2020), p. 11839-11852.
- [129] Vikash SEHWAG, Mung CHIANG et Prateek MITTAL. « SSD : A Unified Framework for Self-Supervised Outlier Detection ». In : *9th International Conference on Learning Representations (ICLR)*. 2021.
- [130] Ian GOODFELLOW, Jean POUGET-ABADIE, Mehdi MIRZA, Bing XU, David WARDEFARLEY, Sherjil OZAIR, Aaron COURVILLE et Yoshua BENGIO. « Generative Adversarial Nets ». In : *Advances in Neural Information Processing Systems (NIPS)* 27 (2014).
- [131] Thomas SCHLEGL, Philipp SEEBÖCK, Sebastian M WALDSTEIN, Ursula SCHMIDT-ERFURTH et Georg LANGS. « Unsupervised Anomaly Detection with Generative Adversarial

-
- Networks to Guide Marker Discovery ». In : *International Conference on Information Processing in Medical Imaging (IPMI)*. Springer. 2017, p. 146-157.
- [132] Mohammad SABOKROU, Mohammad KHALOOEI, Mahmood FATHY et Ehsan ADELI. « Adversarially Learned One-Class Classifier for Novelty Detection ». In : *Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2018, p. 3379-3388.
- [133] Federico MARTINELLI, Riccardo SCALENGHE, Salvatore DAVINO, Stefano PANNO, Giuseppe SCUDERI, Paolo RUISI, Paolo VILLA, Daniela STROPPIANA, Mirco BOSCHETTI, Luiz R. GOULART, Cristina E. DAVIS et Abhaya M. DANDEKAR. « Advanced methods of plant disease detection. A review ». In : *Agronomy for Sustainable Development* 35 (2015), p. 1-25.
- [134] Hiteshwari SABROL et Satish KUMAR. « Recent Studies of Image and Soft Computing Techniques for Plant Disease Recognition and Classification ». In : *International Journal of Computer Applications (IJCA)* 126.1 (2015).
- [135] Jinzhu LU, Lijuan TAN et Huanyu JIANG. « Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification ». In : *Agriculture* 11.8 (2021), p. 707.
- [136] Alvaro FUENTES, Sook YOON, Sang Cheol KIM et Dong Sun PARK. « A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition ». In : *Sensors* 17.9 (2017), p. 2022.
- [137] Evan SHELHAMER, Jonathan LONG et Trevor DARRELL. « Fully Convolutional Networks for Semantic Segmentation ». In : *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39.4 (2017), p. 640-651.
- [138] Zhen WANG et Shanwen ZHANG. « Segmentation of Corn Leaf Disease Based on Fully Convolution Neural Network ». In : *Academic Journal of Computing & Information Science* 1 (2018), p. 9-18.
- [139] Swati BHUGRA, Vinay KAUSHIK, Amit GUPTA, Brejesh LALL et Santanu CHAUDHURY. « AnoLeaf : Unsupervised Leaf Disease Segmentation via Structurally Robust Generative Inpainting ». In : *Winter Conference on Applications of Computer Vision (WACV)*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2023, p. 6415-6424.
- [140] Agnes DESOLNEUX, Lionel MOISAN et Jean-Michel MOREL. *From Gestalt Theory to Image Analysis : A Probabilistic Approach*. T. 34. Springer Science & Business Media, 2007.
- [141] David LOWE. *Perceptual Organization and Visual Recognition*. T. 5. Springer Science & Business Media, 2012.

- [142] Matias TAILANIAN, Pablo MUSÉ et Álvaro PARDO. « A Multi-Scale A Contrario method for Unsupervised Image Anomaly Detection ». In : *20th International Conference on Machine Learning and Applications (ICMLA)*. Institute of Electrical et Electronics Engineers (IEEE). 2021, p. 179-184.
- [143] Karl PEARSON. « LIII. On lines and planes of closest fit to systems of points in space ». In : *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 2.11 (1901), p. 559-572.
- [144] Kumar ABHISHEK et Deeksha KAMATH. « Attribution-based XAI Methods in Computer Vision : A Review ». In : *arXiv preprint arXiv :2211.14736* (2022).
- [145] Marco Tulio RIBEIRO, Sameer SINGH et Carlos GUESTRIN. « “Why Should I Trust You ?” Explaining the Predictions of Any Classifier ». In : *Proceedings of the 22nd The Association for Computing Machinery’s Special Interest Group on Knowledge Discovery and Data Mining (ACM SIGKDD) International Conference on Knowledge Discovery and Data Mining*. 2016, p. 1135-1144.
- [146] Luisa M ZINTGRAF, Taco S COHEN, Tameem ADEL et Max WELLING. « Visualizing Deep Neural Network Decisions : Prediction Difference Analysis ». In : *International Conference on Learning Representations (ICLR)*. 2022.
- [147] Ruth C FONG et Andrea VEDALDI. « Interpretable Explanations of Black Boxes by Meaningful Perturbation ». In : *International Conference on Computer Vision (ICCV)*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2017, p. 3429-3437.
- [148] Vitali PETSIUK, Abir DAS et Kate SAENKO. « RISE : Randomized Input Sampling for Explanation of Black-box Models ». In : *arXiv preprint arXiv :1806.07421* (2018).
- [149] Marco ANCONA, Enea CEOLINI, Cengiz ÖZTIRELI et Markus GROSS. « Gradient-Based Attribution Methods ». In : *Explainable AI : Interpreting, explaining and visualizing deep learning* (2019), p. 169-191.
- [150] Karen SIMONYAN, Andrea VEDALDI et Andrew ZISSERMAN. « Deep Inside Convolutional Networks : Visualising Image Classification Models and Saliency Maps ». In : *International Conference on Learning Representations (ICLR)*. 2014.
- [151] J SPRINGENBERG, Alexey DOSOVITSKIY, Thomas BROX et M RIEDMILLER. « Striving for Simplicity : The All Convolutional Net ». In : *International Conference on Learning Representations (ICLR) (workshop track)*. 2015.
- [152] Min LIN, Qiang CHEN et Shuicheng YAN. « Network in network ». In : *arXiv preprint arXiv :1312.4400* (2013).
- [153] Bolei ZHOU, Aditya KHOSLA, Agata LAPEDRIZA, Aude OLIVA et Antonio TORRALBA. « Learning Deep Features for Discriminative Localization ». In : *Conference on Com-*

-
- puter Vision and Pattern Recognition (CVPR)*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2016, p. 2921-2929.
- [154] Ramprasaath R SELVARAJU, Michael COGSWELL, Abhishek DAS, Ramakrishna VEDANTAM, Devi PARIKH et Dhruv BATRA. « Grad-CAM : Visual Explanations from Deep Networks via Gradient-based Localization ». In : *International Conference on Computer Vision (ICCV)*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2017, p. 618-626.
- [155] Aditya CHATTOPADHAY, Anirban SARKAR, Prantik HOWLADER et Vineeth N BALASUBRAMANIAN. « Grad-CAM++ : Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks ». In : *Winter conference on Applications of Computer Vision (WACV)*. Institute of Electrical et Electronics Engineers (IEEE). 2018, p. 839-847.
- [156] Mukund SUNDARARAJAN, Ankur TALY et Qiqi YAN. « Axiomatic Attribution for Deep Networks ». In : *International Conference on Machine Learning (ICML)*. Proceedings of Machine Learning Research (PMLR). 2017, p. 3319-3328.
- [157] Haofan WANG, Zifan WANG, Mengnan DU, Fan YANG, Zijian ZHANG, Sirui DING, Piotr MARDZIEL et Xia HU. « Score-CAM : Score-Weighted Visual Explanations for Convolutional Neural Networks ». In : *Conference on Computer Vision and Pattern Recognition (CVPR) workshops*. Institute of Electrical et Electronics Engineers/Computer Vision Foundation (IEEE/CVF). 2020, p. 24-25.
- [158] Avanti SHRIKUMAR, Peyton GREENSIDE et Anshul KUNDAJE. « Learning Important Features Through Propagating Activation Differences ». In : *International Conference on Machine Learning (ICML)*. Proceedings of Machine Learning Research (PMLR). 2017, p. 3145-3153.
- [159] Daniel SMILKOV, Nikhil THORAT, Been KIM, Fernanda VIÉGAS et Martin WATTENBERG. « SmoothGrad : removing noise by adding noise ». In : *arXiv preprint arXiv :1706.03825* (2017).
- [160] Suraj SRINIVAS et François FLEURET. « Full-Gradient Representation for Neural Network Visualization ». In : *Advances in Neural Information Processing Systems (NIPS)* 32 (2019).
- [161] Amit DHURANDHAR, Pin-Yu CHEN, Ronny LUSS, Chun-Chen TU, Paishun TING, Karthikeyan SHANMUGAM et Payel DAS. « Explanations based on the Missing : Towards Contrastive Explanations with Pertinent Negatives ». In : *Advances in Neural Information Processing Systems (NIPS)* 31 (2018).
- [162] Amir FEGHAHATI, Christian R SHELTON, Michael J PAZZANI et Kevin TANG. « CDeepEx : Contrastive Deep Explanations ». In : *European Conference on Artificial Intelligence (ECAI)*. IOS Press, 2020, p. 1143-1151.

- [163] Mohit PRABHUSHANKAR, Gukyeong KWON, Dogancan TEMEL et Ghassan ALREGIB. « Contrastive explanations in neural networks ». In : *International Conference on Image Processing (ICIP)*. Institute of Electrical et Electronics Engineers (IEEE). 2020, p. 3289-3293.
- [164] Wan-Soo KIM, Dae-Hyun LEE et Yong-Joo KIM. « Machine vision-based automatic disease symptom detection of onion downy mildew ». In : *Computers and Electronics in Agriculture* 168 (2020), p. 105099.
- [165] Taejoo KIM, Hyeongjun KIM, Kyeonghoon BAIK et Yookyung CHOI. « Instance-Aware Plant Disease Detection by Utilizing Saliency Map and Self-Supervised Pre-Training ». In : *Agriculture* 12.8 (2022), p. 1084.
- [166] Uddhav BHATTARAI et Manoj KARKEE. « A weakly-supervised approach for flower/fruit counting in apple orchards ». In : *Computers in Industry* 138 (2022), p. 103635.
- [167] Wan-Soo KIM, Dae-Hyun LEE, Taehyeong KIM, Hyunggun KIM, Taeyong SIM et Yong-Joo KIM. « Weakly Supervised Crop Area Segmentation for an Autonomous Combine Harvester ». In : *Sensors* 21.14 (2021), p. 4801.
- [168] Jana KIERDORF et Ribana ROSCHER. « Reliability Scores From Saliency Map Clusters for Improved Image-Based Harvest-Readiness Prediction in Cauliflower ». In : *Geoscience and Remote Sensing Letters* (2023).
- [169] Jana KIERDORF, Laura Verena JUNKER-FROHN, Mike DELANEY, Mariele Donoso OLAVE, Andreas BURKART, Hannah JAENICKE, Onno MULLER, Uwe RASCHER et Ribana ROSCHER. « GrowliFlower : An image time-series dataset for GROWth analysis of cauLIFLOWER ». In : *Journal of Field Robotics* 40.2 (2023), p. 173-192.
- [170] Davinder SINGH, Naman JAIN, Pranjali JAIN, Pratik KAYAL, Sudhakar KUMAWAT et Nipun BATRA. « PlantDoc : A Dataset for Visual Plant Disease Detection ». In : *Proceedings of the Association for Computing Machinery (ACM) India Special Interest Group On Knowledge Discovery and Data Mining (IKDD) International Conference on Data Science and Management of Data (CoDS-COMAD) (7th ACM IKDD CoDS and 25th COMAD)*. 2020, p. 249-253.
- [171] David HUGHES et Marcel SALATHÉ. « An open access repository of images on plant health to enable the development of mobile disease diagnostics ». In : *arXiv preprint arXiv :1511.08060* (2015).
- [172] Thomas SCHLEGL, Philipp SEEBÖCK, Sebastian M WALDSTEIN, Georg LANGS et Ursula SCHMIDT-ERFURTH. « f-AnoGAN : Fast Unsupervised Anomaly Detection with Generative Adversarial Networks ». In : *Medical Image Analysis* 54 (2019), p. 30-44.
- [173] Jan VERBESELT, Rob HYNDMAN, Glenn NEWNHAM et Darius CULVENOR. « Detecting trend and seasonal changes in satellite image time series ». In : *Remote sensing of Environment* 114.1 (2010), p. 106-115.

-
- [174] Qingsong WEN, Tian ZHOU, Chaoli ZHANG, Weiqi CHEN, Ziqing MA, Junchi YAN et Liang SUN. « Transformers in Time Series : A Survey ». In : *32nd International Joint Conference on Artificial Intelligence (IJCAI)*. 2023, p. 6778-6786.
- [175] Paolo FANTOZZI et Maurizio NALDI. « The Explainability of Transformers : Current Status and Directions ». In : *Computers* 13.4 (2024), p. 92.
- [176] Sarthak JAIN et Byron C WALLACE. « Attention is not Explanation ». In : *Conference of the North American Chapter of the Association for Computational Linguistics (ACL) : Human Language Technologies*. T. 1. 2019, p. 3543-3556.
- [177] Danish PRUTHI, Mansi GUPTA, Bhuwan DHINGRA, Graham NEUBIG et Zachary C LIPTON. « Learning to Deceive with Attention-Based Explanations ». In : *58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2020, p. 4782-4793.
- [178] Sarah WIEGREFE et Yuval PINTER. « Attention is not not Explanation ». In : *Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, (EMNLP-IJCNLP)*. Association for Computational Linguistics (ACL). 2019, p. 11-20.
- [179] Shikhar VASHISHTH, Shyam UPADHYAY, Gaurav Singh TOMAR et Manaal FARUQUI. « Attention Interpretability Across NLP Tasks ». In : *arXiv preprint arXiv :1909.11218* (2019).
- [180] Yikuan LI, Shishir RAO, José Roberto Ayala SOLARES, Abdelaali HASSAINE, Rema RAMAKRISHNAN, Dexter CANOY, Yajie ZHU, Kazem RAHIMI et Gholamreza SALIMI-KHORSHIDI. « BEHRT : Transformer for Electronic Health Records ». In : *Scientific Reports* 10.1 (2020), p. 7155.
- [181] Subhash NERELLA, Sabyasachi BANDYOPADHYAY, Jiaqing ZHANG, Miguel CONTRERAS, Scott SIEGEL, Aysegul BUMIN, Brandon SILVA, Jessica SENA, Benjamin SHICKEL, Azra BIHORAC, Kia KHEZELI et Parisa RASHIDI. « Transformers in Healthcare : A Survey ». In : *arXiv preprint arXiv :2307.00067* (2023).
- [182] Edward CHOI, Zhen XU, Yujia LI, Michael DUSENBERRY, Gerardo FLORES, Emily XUE et Andrew DAI. « Learning the Graphical Structure of Electronic Health Records with Graph Convolutional Transformer ». In : *The Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*. T. 34. 01. 2020, p. 606-613.
- [183] Junyuan SHANG, Tengfei MA, Cao XIAO et Jimeng SUN. « Pre-training of Graph Augmented Transformers for Medication Recommendation ». In : *International Joint Conference on Artificial Intelligence (IJCAI)*. 2019.
- [184] Xueping PENG, Guodong LONG, Tao SHEN, Sen WANG et Jing JIANG. « Sequential Diagnosis Prediction with Transformer and Ontological Representation ». In : *In-*

- ternational Conference on Data Mining (ICDM). Institute of Electrical et Electronics Engineers (IEEE). 2021, p. 489-498.
- [185] Yikuan LI, Mohammad MAMOUEI, Gholamreza SALIMI-KHORSHIDI, Shishir RAO, Abdelaali HASSAINE, Dexter CANOY, Thomas LUKASIEWICZ et Kazem RAHIMI. « Hi-BEHRT : Hierarchical Transformer-Based Model for Accurate Prediction of Clinical Events Using Multimodal Longitudinal Electronic Health Records ». In : *Journal of Biomedical and Health Informatics* 27.2 (2022), p. 1106-1117.
- [186] Laila RASMY, Yang XIANG, Ziqian XIE, Cui TAO et Degui ZHI. « Med-BERT : pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction ». In : *NPJ Digital Medicine* 4.1 (2021), p. 86.
- [187] Junyu LUO, Muchao YE, Cao XIAO et Fenglong MA. « HiTANet : Hierarchical Time-Aware Attention Networks for Risk Prediction on Electronic Health Records ». In : *Proceedings of the 26th The Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining (ACM SIGKDD) International Conference on Knowledge Discovery and Data Mining*. 2020, p. 647-656.
- [188] Houxing REN, Jingyuan WANG, Wayne Xin ZHAO et Ning WU. « RAPT : Pre-training of Time-Aware Transformer for Learning Robust Healthcare Representation ». In : *Proceedings of the 27th The Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining (ACM SIGKDD) International Conference on Knowledge Discovery and Data Mining*. 2021, p. 3503-3511.
- [189] Sajad DARABI, Mohammad KACHUEE, Shayan FAZELI et Majid SARRAFZADEH. « TAPER : Time-Aware Patient EHR Representation ». In : *Journal of Biomedical and Health Informatics* 24.11 (2020), p. 3268-3275.
- [190] Xiao LUO, Priyanka GANDHI, Zuoyi ZHANG, Wei SHAO, Zhi HAN, Vasu CHANDRASEKARAN, Vladimir TURZHITSKY, Vishal BALI, Anna R. ROBERTS, Megan METZGER, Jarod BAKER, Carmen LA ROSA, Jessica WEAVER, Paul DEXTER et Kun HUANG. « Applying interpretable deep learning models to identify chronic cough patients using EHR data ». In : *Computer Methods and Programs in Biomedicine* 210 (2021), p. 106395.
- [191] Yiwen MENG, William SPEIER, Michael K ONG et Corey W ARNOLD. « Bidirectional Representation Learning From Transformers Using Multimodal Electronic Health Record Data to Predict Depression ». In : *Journal of Biomedical and Health Informatics* 25.8 (2021), p. 3121-3129.
- [192] Zhen XU, David R So et Andrew M DAI. « MUFASA : Multimodal Fusion Architecture Search for Electronic Health Records ». In : *The Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*. T. 35. 12. 2021, p. 10532-10540.
- [193] Xianli ZHANG, Buyue QIAN, Yang LI, Yang LIU, Xi CHEN, Chong GUAN et Chen LI. « Learning robust patient representations from multi-modal electronic health

-
- records : a supervised deep learning approach ». In : *International Conference on Data Mining (SDM)*. Society for Industrial et Applied Mathematics (SIAM). 2021, p. 585-593.
- [194] Jesse VIG. « A Multiscale Visualization of Attention in the Transformer Model ». In : *57th Annual Meeting of the Association for Computational Linguistics (ACL) : System Demonstrations*. 2019, p. 37-42.
- [195] Samira ABNAR et Willem ZUIDEMA. « Quantifying Attention Flow in Transformers ». In : *58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2020, p. 4190-4197.
- [196] Katrin RENZ, Kashyap CHITTA, Otniel-Bogdan MERCEA, A KOEPKE, Zeynep AKATA et Andreas GEIGER. « PlanT : Explainable Planning Transformers via Object-Level Representations ». In : *6th Annual Conference on Robot Learning*. MLResearchPress. 2022, p. 459-470.
- [197] Qizhang FENG, Jiayi YUAN, Forhan Bin EMDAD, Karim HANNA, Xia HU et Zhe HE. « Can Attention Be Used to Explain EHR-Based Mortality Prediction Tasks : A Case Study on Hemorrhagic Stroke ». In : *14th Association for Computing Machinery (ACM) International Conference on Bioinformatics, Computational Biology, and Health Informatics*. 2023, p. 1-6.
- [198] Wilson E MARCÍLIO et Danilo M ELER. « From explanations to feature selection : assessing SHAP values as feature selection mechanism ». In : *33rd Conference on Graphics, Patterns and Images (SIBGRAPI)*. Institute of Electrical et Electronics Engineers (IEEE). 2020, p. 340-347.
- [199] Bayu Distiawan TRISEDYA, Flora D SALIM, Jeffrey CHAN, Damiano SPINA, Falk SCHOLER et Mark SANDERSON. « i-Align : an interpretable knowledge graph alignment model ». In : *Data Mining and Knowledge Discovery 37.6 (2023)*, p. 2494-2516.
- [200] Byung-Hak KIM, Zhongfen DENG, S Yu PHILIP et Varun GANAPATHI. « Can Current Explainability Help Provide References in Clinical Notes to Support Humans Annotate Medical Codes ? » In : *13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*. 2022, p. 26-34.
- [201] Jim CLAUWAERT, Gerben MENSCHAERT et Willem WAEGEMAN. « Explainability in transformer models for functional genomics ». In : *Briefings in Bioinformatics 22.5 (2021)*.
- [202] Hanane SEBBAQ et Nour-eddine EL FADDOULI. « MTBERT-Attention : An Explainable BERT Model based on Multi-Task Learning for Cognitive Text Classification ». In : *Scientific African 21 (2023)*, e01799.
- [203] Huiyuan CHEN, Kaixiong ZHOU, Zhimeng JIANG, Chin-Chia Michael YEH, Xiaoting LI, Menghai PAN, Yan ZHENG, Xia HU et Hao YANG. « Probabilistic Masked Attention

- Networks for Explainable Sequential Recommendation. » In : *International Joint Conference on Artificial Intelligence (IJCAI)*. 2023, p. 2068-2076.
- [204] Leonid SCHWENKE et Martin ATZMUELLER. « Show Me What You're Looking For : Visualizing Abstracted Transformer Attention for Enhancing Their Local Interpretability on Time Series Data ». In : *The International Florida Artificial Intelligence Research Society (FLAIRS) Conference*. T. 34. 2021.
- [205] Luca BACCO, Andrea CIMINO, Felice DELL'ORLETTA et Mario MERONE. « Extractive Summarization for Explainable Sentiment Analysis using Transformers. » In : *DeepOntoNLP/X-SENTIMENT Extended Semantic Web Conference (ESWC)*. 2021, p. 62-73.
- [206] Rebecca LEYGONIE et Guillaume VIMONT. « Empirical analysis of the residual information needed for image classification ». In : *8th International Workshop on Big Data Analytic Technology for Bioinformatics and Health, Informatics in International Conference on Big Data*. Institute of Electrical et Electronics Engineers (IEEE). 2021, p. 2813-2817.
- [207] Rebecca LEYGONIE, Sylvain LOBRY et Laurent WENDLING. « Apprentissage profond pour la classification de QR Codes bruités ». In : *Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP)* (2022).
- [208] Rebecca LEYGONIE, Sylvain LOBRY, Guillaume VIMONT et Laurent WENDLING. « Transposition de données multidimensionnelles en images pour pallier le fléau de la dimension ». In : *ORASIS 2023*. 2023.
- [209] Rebecca LEYGONIE, Sylvain LOBRY, Guillaume VIMONT et Laurent WENDLING. « Transforming Multidimensional Data into Images to Overcome the Curse of Dimensionality ». In : *International Conference on Image Processing (ICIP)*. Institute of Electrical et Electronics Engineers (IEEE). 2023, p. 700-704.
- [210] Rebecca LEYGONIE, Sylvain LOBRY et Laurent WENDLING. « An a contrario approach for plant disease detection ». In : *Workshop on Machine Vision for Earth Observation (MVEO) at the 34th British Machine Vision Conference (BMVC)*. 2023.
- [211] Rebecca LEYGONIE, Sylvain LOBRY et Laurent WENDLING. « Can we detect plant diseases without prior knowledge of their existence? » In : *International Journal of Applied Earth Observation and Geoinformation* 134 (2024), p. 104192.
- [212] Emmanuel BACRY, Stéphane GAIFFAS, Fanny LEROY, Maryan MOREL, Dinh-Phong NGUYEN, Youcef SEBIAT et Dian SUN. « SCALPEL3 : a scalable open-source library for healthcare claims databases ». In : *International Journal of Medical Informatics* 141 (2020), p. 104203.

Annexes

A Données du Système National des Données de Santé

Le *Système National des Données de Santé*¹³ (SNDS) est un entrepôt de données médico-administratives pseudonymisées créé en 2016 dans la continuité d'un entrepôt précédent. Géré par la *Caisse Nationale de l'Assurance Maladie* (CNAM), le SNDS est conçu pour améliorer l'analyse et la gestion de la santé publique en mettant à disposition des données pour les études, les recherches et les évaluations d'intérêt public.

Le SNDS est utilisé pour faire de la veille, étudier ou évaluer des sujets tels que la santé, l'offre de soins et les dépenses, la mise en œuvre des politiques de santé, l'activité des professionnels de santé, la sécurité sanitaires, l'innovation dans les domaines de la santé et de la prise en charge médico-sociale, la surveillance, et la sécurité sanitaire. Il est en revanche interdit d'exploiter le SNDS pour la promotion des produits de santé auprès des professionnels de santé et des établissements de santé ainsi que pour la modification des contrats d'assurance.

Le SNDS contient plus de 3 000 variables, et un flux annuel de 1,2 milliards de feuilles de soins, 11 millions de séjours hospitaliers, 500 millions d'actes et 450 To de données. Le SNDS intègre des données de santé couvrant l'ensemble de la population française et contenant l'ensemble des soins présentés au remboursement, ce qui en fait l'une des bases de données de santé les plus complètes au monde (voir Table 2, qui liste le type d'informations contenues ou non dans le SNDS).

Le SNDS contient des informations provenant de diverses sources, telles que les hôpitaux, les médecins, les pharmacies et les assurances maladie. Plus précisément, le SNDS rassemble et met à disposition des informations de santé collectées par des organismes publics. Il intègre des données provenant des sources suivantes :

- le *Système National d'Information Inter Régimes de l'Assurance Maladie* (SNIIRAM), qui regroupe les données de l'assurance maladie ;
- le *Programme de Médicalisation des Systèmes d'Information* (PMSI), qui couvre les données des hôpitaux et autres établissements de santé ;
- la base du *Centre d'épidémiologie sur les causes médicales de décès* (CépiDC) de l'*Institut National de la Santé et de la Recherche Médicale*¹⁴ (INSERM), qui concerne les données statistiques relatives aux causes de décès.

13. <https://www.snds.gouv.fr/SNDS/Accueil> 14. <https://www.inserm.fr/en/home/>

Dans le futur, lorsqu'elles seront constituées, le SNDS intégrera les données « médico-sociales » des maisons départementales des personnes handicapées (données de la *Caisse Nationale de Solidarité pour l'Autonomie*) et un échantillon représentatif des données de remboursement par bénéficiaires transmis par les mutuelles.

Existe dans le SNDS	N'existe pas dans le SNDS
– Dépenses et remboursements (prestations en soins de ville, en établissements de santé, et montants associés) – Consommations de soins de ville (consultations et actes techniques) – Prescriptions (médicaments) – Dispositifs médicaux (aides techniques) – Autres prestations (cures, transports...) – Soins hospitaliers (hors séances) – Séjours hospitaliers (et diagnostics) – Indemnités journalières (maladie, accidents du travail et maladies professionnelles, maternité) et invalidité – Affections de Longue Durée (ALD) – Informations génériques sur les patients (âge, sexe, commune, département de résidence, couverture maladie universelle complémentaire, aide à la complémentaire santé, date, commune et causes médicales de décès, médecin traitant) – Informations sur les professionnels de santé (spécialité, mode d'exercice, sexe, âge, département d'implantation)	– Résultats d'examen clinique (imagerie, données biologiques) ou paraclinique (tabagisme, tension, IMC) – Motifs de consultation – Notion des facteurs de risque (tabac, alcool, nutrition, etc.) – Information sur les médicaments délivrés au cours des séjours hospitaliers (hors médicaments en sus¹⁵) – Données sociales (ou très peu)

TABLE 2 – Contenu du SNDS. La colonne de gauche liste les informations qu'on peut trouver dans le SNDS et celle de droite les informations qu'on ne peut pas trouver dans le SNDS.

Pour protéger l'identité des patients et garantir la confidentialité des données, chaque patient est identifié dans l'ensemble du SNDS par un pseudonyme. Ce pseudonyme est généré à partir du *Numéro d'Inscription au Répertoire* (NIR) (connu sous le nom de numéro de sécurité sociale), à l'aide d'un procédé cryptographique reposant sur des fonctions de hachage, appelé FOIN (Fonction d'Occultation des Identifiants Nominatifs).

15. Les médicaments en sus sont des traitements coûteux et innovants, facturés séparément du forfait hospitalier global, permettant ainsi leur traçabilité dans le SNDS et un remboursement spécifique par l'Assurance Maladie.

Les données du SNDS sont conservées pendant une période totale de 20 ans, puis archivées pour une durée supplémentaire de 10 ans. Cette durée de conservation prolongée est un compromis entre le besoin de disposer de données historiques pour la recherche et la santé publique, et les considérations de protection de la vie privée et de gestion des données.

L'accès aux données et leur analyse ne peuvent se faire que dans un cadre d'hébergement très restrictif, conforme au référentiel de sécurité du SNDS¹⁶, afin de garantir la traçabilité des accès et des traitements, ainsi que la confidentialité et l'intégrité des données.

Bases du SNDS

Le SNDS se décline en différentes bases, incluant des **bases de données individuelles** qui sont quasi-exhaustives sur l'ensemble de la population française, un **échantillon représentatif des personnes protégées**, c'est à dire couvertes par le système de sécurité sociale français, et **des bases thématiques avec des données agrégées**.

Les données alimentent les différentes bases du SNDS par divers circuits. Elles peuvent transiter par les bases de données des régimes d'assurance maladie pour le SNIIRAM, incluant les soins remboursés par les *Caisses Primaires d'Assurance Maladie* via la carte vitale ou des feuilles de soins papier pour les soins de ville, les bordereaux de facturation des cliniques privées, et les actes et consultations externes des hôpitaux publics. Pour les données d'activité hospitalière, l'*Agence Technique de l'Information sur l'Hospitalisation*¹⁷ (ATIH) gère le PMSI, qui couvre les activités de médecine, chirurgie, obstétrique, soins de suite et de réadaptation, hospitalisation à domicile et psychiatrie. Enfin, les données sur les causes médicales de décès sont recueillies via le circuit mairie-INSEE-INSERM, où les mairies enregistrent les décès, l'INSEE centralise les informations démographiques, et l'INSERM traite les causes médicales de décès.

Bases des données individuelles

Les données individuelles, dont la jointure est schématisée dans la Figure 9, comprennent trois sources : le *Programme de Médicalisation des Systèmes d'Information* (PMSI), le *Datamart de Consommation Inter Régime* (DCIR) et le *Centre d'épidémiologie sur les causes médicales de décès* (CépiDC).

Le PMSI recense l'activité des hôpitaux publics et privés afin d'évaluer et de rémunérer l'activité des établissements, de répartir les ressources en fonction des besoins quantifiés et de réaliser des études de santé publique et épidémiologiques à travers quatre secteurs distincts :

- Médecine, Chirurgie, Obstétrique (MCO), qui sont des hospitalisations de courte durée.
- Soins de Suite et de Réadaptation (SSR).
- Hospitalisation À Domicile (HAD).
- Psychiatrie (RIM-P).

16. <https://www.legifrance.gouv.fr/eli/arrete/2017/3/22/AFSE1705146A/jo/texte> 17. <https://www.atih.sante.fr>

Le DCIR contient l'ensemble des soins de ville remboursés pour les bénéficiaires de l'Assurance Maladie. Il permet de réaliser des études sur la consommation de soins des bénéficiaires et les pratiques des professionnels de santé.

Enfin, le CépiDC fournit des informations sur les causes de décès. En plus de la date de décès, il permet de connaître la cause initiale ainsi que l'ensemble des causes qui ont contribué au décès du bénéficiaire.

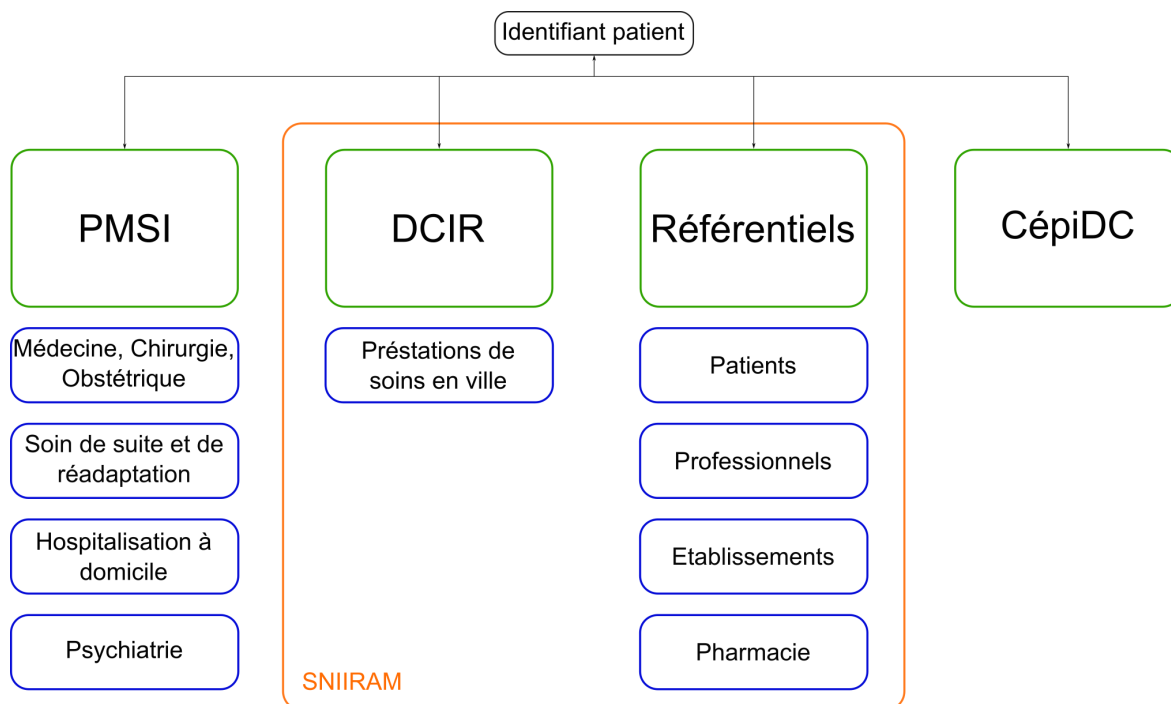


FIGURE 9 – Schéma simplifié de l'architecture des bases de données individuelles. Les tables du *Programme de Médicalisation des Systèmes d'Information* (PMSI), du *Datamart de Consommation Inter Régime* (DCIR) et du *Centre d'épidémiologie sur les causes médicales de décès* (CépiDC) peuvent être jointes grâce à un identifiant patient.

Échantillon représentatif des personnes protégées

L'*Échantillon Généraliste des Bénéficiaires* (EGB) est un échantillon représentatif de la population couverte par le système de sécurité sociale français, sélectionné au 1/97ème par sexe et par âge. Cela signifie qu'il inclut environ une personne sur 97 de cette population (un peu plus de 1%), assurant ainsi une représentation proportionnelle de l'ensemble des bénéficiaires.

L'EGB permet de réaliser des études longitudinales, c'est-à-dire des études qui suivent les mêmes individus sur une longue période. Cela est possible car l'EGB inclut à la fois les personnes qui utilisent fréquemment les services de santé et celles qui les utilisent peu ou pas du tout. Cette caractéristique le distingue du DCIR qui ne comprend que les données des personnes ayant effectivement consommé des soins. Plus précisément, l'EGB contient les données de consommation des bénéficiaires du DCIR ainsi que celles des PMSI MCO et

HAD.

Bases thématiques avec des données agrégées

Les bases thématiques avec des données agrégées, appelées *datamarts BO*, permettent de regrouper les données des bases DCIR et PMSI à un niveau global. Elles se déclinent en deux catégories : les données agrégées des dépenses et les données individuelles de l'offre de soins. Ces *datamarts BO* fournissent une vue d'ensemble des dépenses de santé et de l'offre de soins, facilitant ainsi les analyses et les études sur le système de santé.

Les données agrégées des dépenses concernent le suivi des dépenses, incluant des requêtes prédéfinies pour surveiller les dépenses de santé, les médicaments en sus délivrés à l'hôpital, les analyses de biologie (type d'analyse, date, etc.), la délivrance de médicaments par les pharmacies, et les données de cadrage, qui incluent des informations sur les professionnels de santé et les bénéficiaires des soins.

Les données individuelles de l'offre de soins concernent les prestations réalisées par des praticiens libéraux, avec des requêtes prédéfinies pour des analyses spécifiques, les dispositifs médicaux fournis aux patients, les actes médicaux classés selon une classification commune, les centres de santé et les établissements de santé privés.

Pré-traitements du SNDS

Les données du SNDS suivent un schéma relationnel, c'est-à-dire qu'elles sont organisées en tables qui sont reliées entre elles par des clés (des attributs). Chaque table représente une entité spécifique, et les relations entre ces tables permettent de structurer et de naviguer efficacement à travers les informations de santé. Une visualisation de la structure du SNDS est disponible en ligne ¹⁸. Elle permet notamment de faire des recherches par tables, variables et nomenclatures.

Afin de faciliter l'utilisation de ces données, Bacry et al. ont introduit SCALPEL [212], une chaîne de traitement pour transformer le SNDS, avec pour objectif d'améliorer la comparabilité et la reproductibilité des études en standardisant le nettoyage des données. SCALPEL est composée de deux méthodes distinctes, *SCALPEL-Flattening* et *SCALPEL-Extraction*, qui peuvent être utilisées consécutivement.

SCALPEL-Flattening

SCALPEL-Flattening ¹⁹ simplifie la structure du SNDS en l'aplatissant, ce qui permet de réduire la complexité de la structure et d'éviter les jointures coûteuses. Pour ce faire, *SCALPEL-Flattening* propose une méthode d'aplatissement pour la base de données DCIR et une pour la base de données PMSI.

18. <https://health-data-hub.shinyapps.io/dico-snds/>19. <https://github.com/X-DataInitiative/SCALPEL-Flattening>

Dans le DCIR, chaque ligne des tables à joindre correspond à une seule ligne dans la table principale. En conséquence, des jointures externes gauches successives sont utilisées sur une table principale avec les mêmes clés de jointure (voir Figure 10). Cette méthode consiste à conserver toutes les lignes de la table principale tout en y ajoutant les données correspondantes des autres tables lorsqu'elles existent.

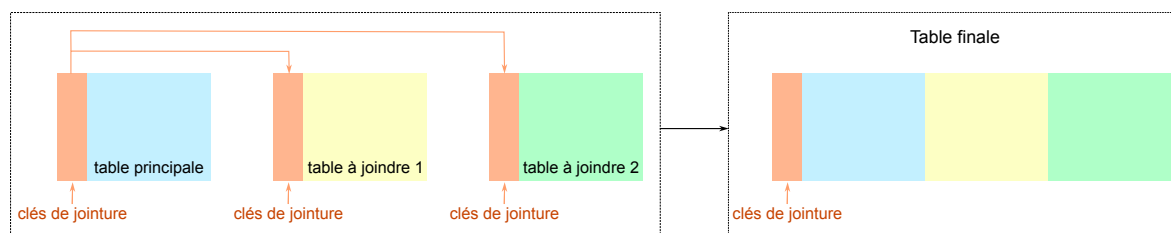


FIGURE 10 – Schéma de jointure de trois tables du DCIR (une principale et deux tables à joindre), par la méthode *SCALPEL-Flattening*.

Dans la base PMSI, la table principale contient des séjours hospitaliers et les tables à joindre contiennent des actes ou des informations sur ces actes. Par conséquent, plusieurs lignes provenant de différentes tables peuvent être associées à un même séjour hospitalier, ce qui signifie qu'il peut y avoir plusieurs lignes à joindre pour chaque ligne de la table principale. De plus, la table patient rassemble des informations sur le patient du séjour hospitalier, avec une correspondance exacte d'une ligne pour chaque ligne de la table principale. Pour répondre à ces contraintes, *SCALPEL-Flattening* utilise une méthode d'aplatissement des données du PMSI en évitant les jointures successives utilisées pour le DCIR, qui pourraient entraîner un produit cartésien de toutes les lignes des tables d'entrée, prenant ainsi beaucoup de temps et produisant des tables aplaties volumineuses avec des informations redondantes.

La première étape consiste en la création d'une table *centrale* en effectuant une jointure externe gauche entre la table principale et la table patient (cf. Figure 11). Ensuite, les autres tables sont jointes séparément à la table *centrale* et les résultats sont concaténés (cf. Figure 12). Le résultat est une table ayant autant de lignes que la somme de toutes les lignes des tables à joindre, incluant toutes les colonnes des tables d'entrée. Seules les informations de la table *centrale* sont répétées, autant de fois que nécessaire. Cette méthode minimise la redondance mais entraîne la présence de nombreux "blocs" vides (cf. Figure 12).

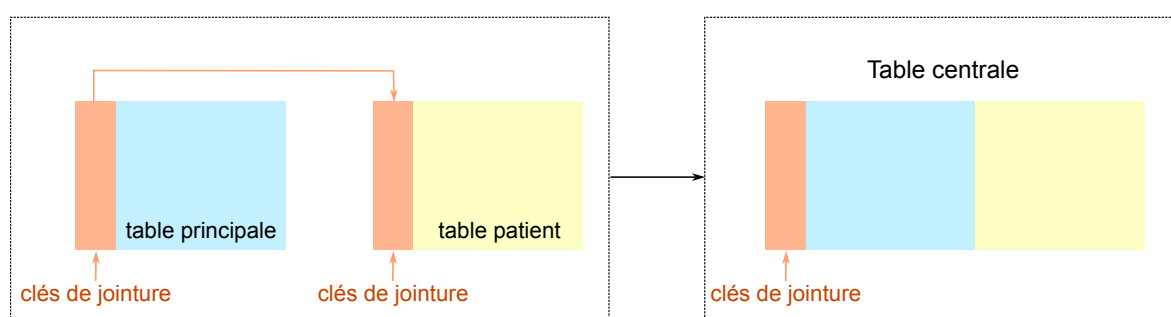


FIGURE 11 – Schéma de jointure des tables principales et patient du PMSI par la méthode *SCALPEL-Flattening*. Le résultat de cette jointure est la table *centrale*.

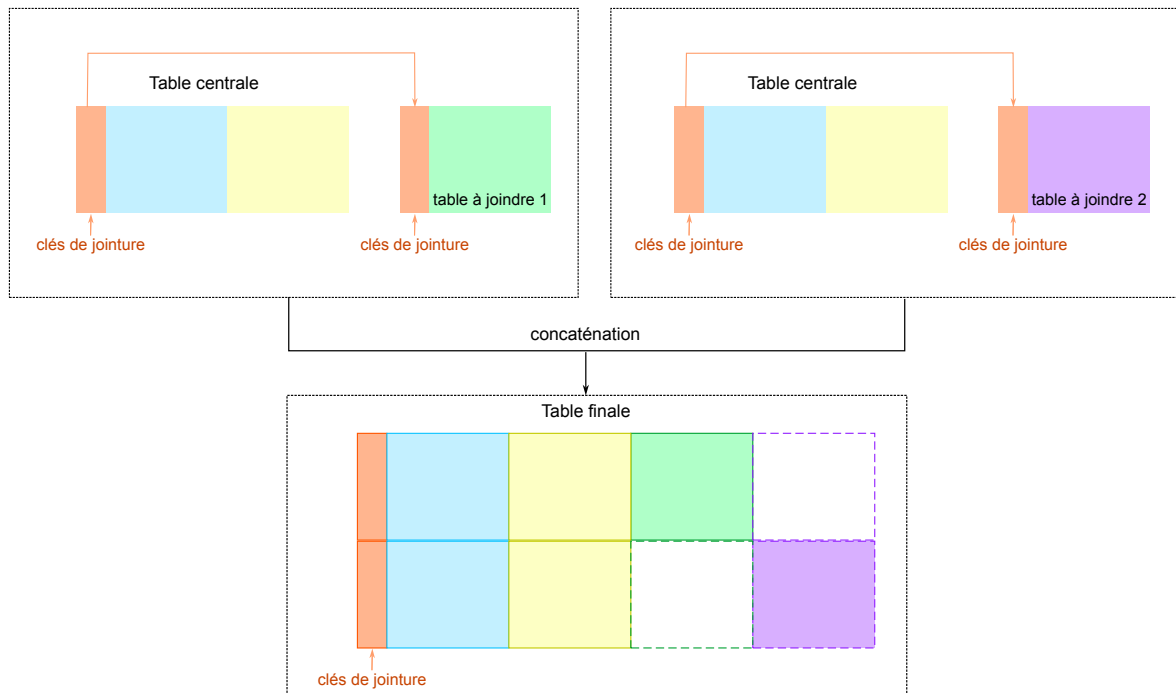


FIGURE 12 – Schéma de jointure de la table *centrale* (résultat de la jointure illustrée dans la Figure 11) du PMSI et des autres tables par la méthode *SCALPEL-Flattening*.

Dans certains cas, la jointure pour la construction de la table centrale nécessite des clés différentes de celles des jointures suivantes. Dans ce cas, l'ordre des jointures reste le même (d'abord avec la table patient, puis avec les autres tables) en utilisant des clés différentes pour la création de la table *centrale*.

Une fois les données aplaties par l'approche *SCALPEL-Flattening*, il est possible d'extraire des tables d'événements avec la méthode *SCALPEL-Extraction*.

SCALPEL-Extraction

L'objectif de *SCALPEL-Extraction*²⁰ est de transformer rapidement et efficacement les données brutes du SNDS en informations médicales structurées et prêtes à l'emploi pour faciliter les analyses et études médicales. Pour ce faire, la première étape de *SCALPEL-Extraction* consiste à lire les données aplaties résultant de l'exécution de *SCALPEL-Flattening* sur les données brutes du SNDS. *SCALPEL-Extraction* introduit deux mécanismes principaux : l'extraction et la transformation des données.

Les **extracteurs** sont responsables d'identifier les colonnes pertinentes dans les tables sources, filtrer les valeurs nulles, extraire des événements bruts (par exemple, dispensations de médicaments, diagnostics, actes médicaux), associer chaque ligne des tables sources à un ou plusieurs événements, et standardiser les données extraites selon un schéma cohérent et exploitable.

20. <https://github.com/X-DataInitiative/SCALPEL-Extraction>

Les **transformateurs**, quant à eux, ont pour rôle de traiter les événements bruts extraits, les convertir en événements plus complexes et informatifs, et combiner plusieurs événements pour créer des concepts médicaux détaillés. Par exemple, les transformateurs peuvent calculer les périodes d'exposition aux médicaments à partir des données de dispensation. Ils peuvent également générer des informations sur les périodes de suivi, les expositions à des molécules spécifiques, ou des résultats cliniques.

Cette approche en deux étapes permet une manipulation structurée et une analyse approfondie des données de santé.

Finalement, chaque concept extrait a une structure [patient, événement] où le patient est représenté par un identifiant unique (patientID), un genre, une date de naissance et éventuellement une date de décès. Les événements associés peuvent être ponctuels (comme un acte médical) ou continus (comme une hospitalisation) et sont liés à un patient.

Cas d'utilisation du SNDS

Le Système National des Données de Santé offre de nombreux cas d'utilisation pour la recherche en santé publique et l'amélioration du système de soins en France. Cette base de données exhaustive permet de mener des études épidémiologiques à grande échelle, d'évaluer l'efficacité et la sécurité des médicaments en vie réelle, et de surveiller les tendances de santé au niveau national. Le SNDS est particulièrement utile pour analyser les parcours de soins des patients, identifier les facteurs de risque de maladies, et évaluer l'impact des politiques de santé. Il permet également de réaliser des études médico-économiques, d'optimiser l'allocation des ressources de santé, et de détecter les signaux de pharmacovigilance. Les chercheurs utilisent le SNDS pour étudier les maladies rares, analyser les disparités géographiques d'accès aux soins, et évaluer la qualité des pratiques médicales. Enfin, le SNDS joue un rôle crucial dans la planification sanitaire, la prévention des épidémies, et l'amélioration continue de la qualité des soins en France.

B Guide de décodage d'un QR code

Dans cette partie, nous présentons les caractéristiques d'un QR code généré à partir de caractères alphanumériques et nous expliquons par un exemple les étapes de décodage.

Caractéristiques d'un QR code

Il existe plusieurs versions de QR code, dépendant du nombre de caractères encodés. La version d'un QR code indique sa taille. Les versions vont de **1** (21×21) à **40** (177×177) pour un nombre de caractères alphanumériques allant de 10 à 4 296.

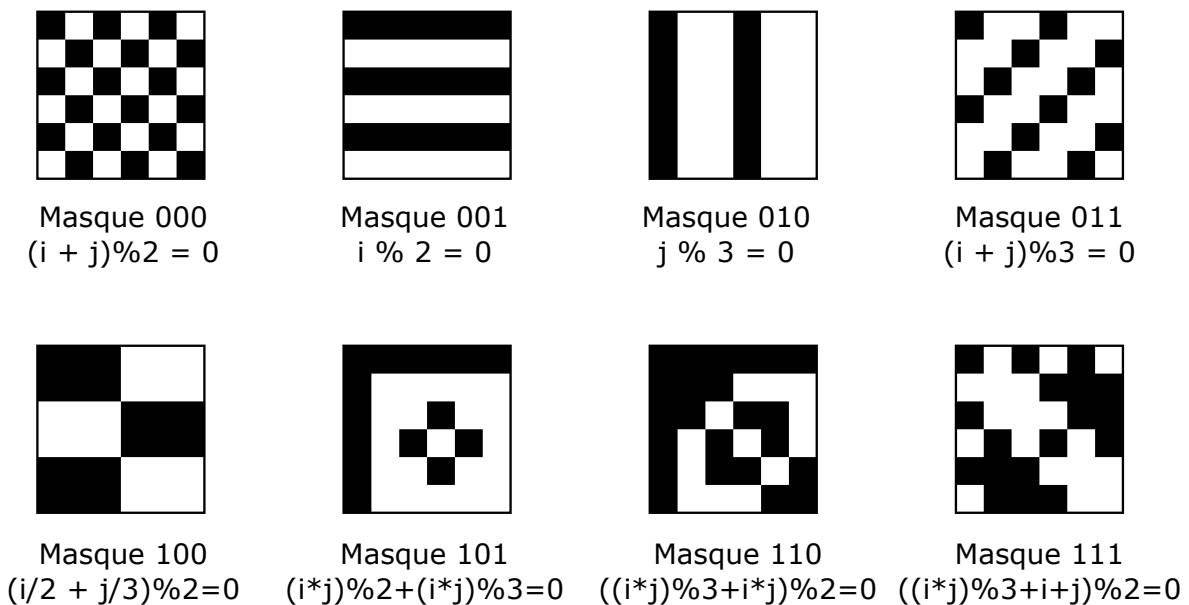


FIGURE 13 – Différents masques de QR code avec leurs règles d'inversion associées. Les variables i et j correspondent respectivement aux coordonnées des positions horizontale et verticale du module (bit).

Lors de la génération du QR code, pour qu'il n'y ait pas de zones noires ou blanches trop importantes, un masque est appliqué. Il existe 8 types de masque, détaillés dans la Figure 13. Chaque masque suit une règle spécifique basée sur la position (coordonnées horizontale et verticale) de chaque module (bit) dans le QR code. La règle détermine si le module doit être inversé ou non. Par exemple, pour le masque '000', la règle est $(\text{ligne} + \text{colonne}) \% 2 = 0$. Si cette condition est vraie pour un module donné, sa couleur est inversée (de blanc à noir ou de noir à blanc). Ce processus est appliqué systématiquement à tous les modules du QR code, créant ainsi un motif qui rompt les grandes zones uniformes et améliore la lisibilité du code par les scanners. Le masque optimal parmi les huit est choisi automatiquement lors de la génération finale du QR code pour assurer la meilleure répartition possible des modules noirs et blancs.

Un QR code est composé de plusieurs parties (octets) pour aider les décodeurs à le lire (cf. Figure 14) : les **pilotes de positionnements**, qui permettent au décodeur de détecter la

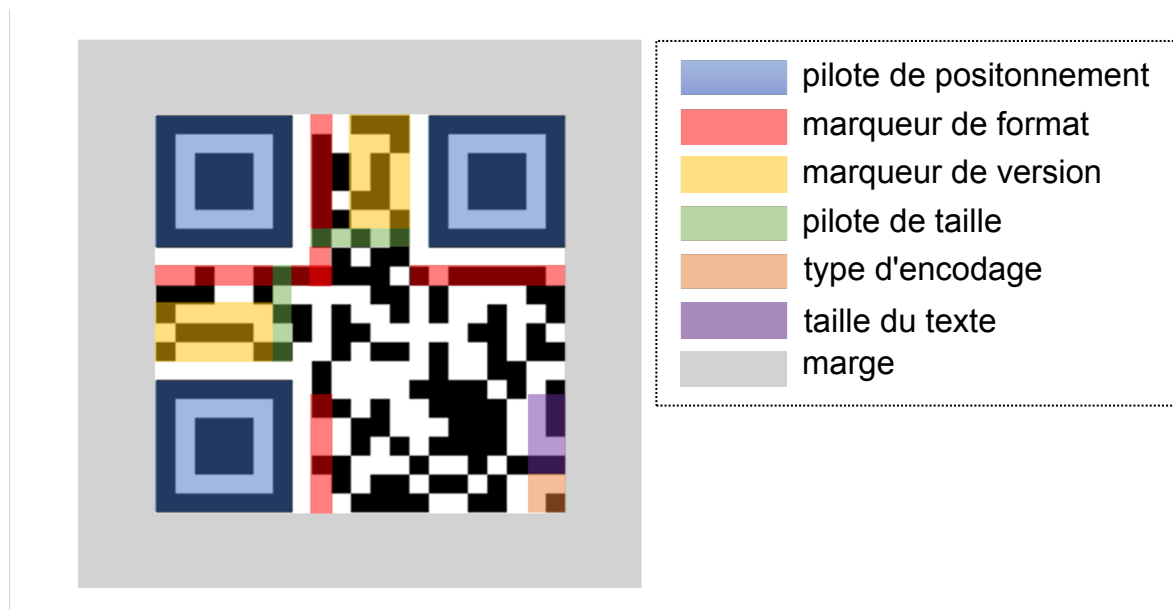


FIGURE 14 – Schéma des différentes parties d'un QR code

présence du QR code et son orientation ; les **marqueurs de version**, indiquant la version utilisée pour la génération du QR ; les **pilotes de taille**, permettant de détecter la dimension de la matrice de données ; la **marge**, permettant de distinguer le code de ce qui l'entoure ; les **marqueurs de format**, contenant les informations concernant la correction d'erreur et le numéro du masque utilisé ; le **type d'encodage utilisé** et la **taille du texte**. Le reste du QR code correspond à l'information encodée.

Dans la partie suivante, nous montrons les étapes de décodage du QR code de la Figure 14.

Décodage d'un QR code

Pour décoder un QR code, la première étape est de décoder le format (taux de correction d'erreurs et numéro du masque à appliquer). Le format est contenu sur 5 bits + 10 bits de redondance, soit 15 bits numérotés de 0 à 14 à deux endroits dans le QR code (cf. Figure 15). Pour le QR code que nous décodons, le format est : **001001110111110**.

Durant la génération du QR code, un masque est appliqué pour éviter que les 15 bits de formats soient identiques. Ce masque est le même pour toute génération de QR code et vaut : **101010000010010**.

Pour appliquer ou enlever ce masque au format, l'opération XOR ($\oplus$) est utilisée. Le résultat de l'opération XOR entre le format et le masque est :

$$\begin{array}{r}
 001001110111110 \\
 \oplus 101010000010010 \\
 \hline
 100011110101100
 \end{array}$$

Ainsi, le format « démasqué » est : **100011110101100**. Dans ce format, le taux de correction

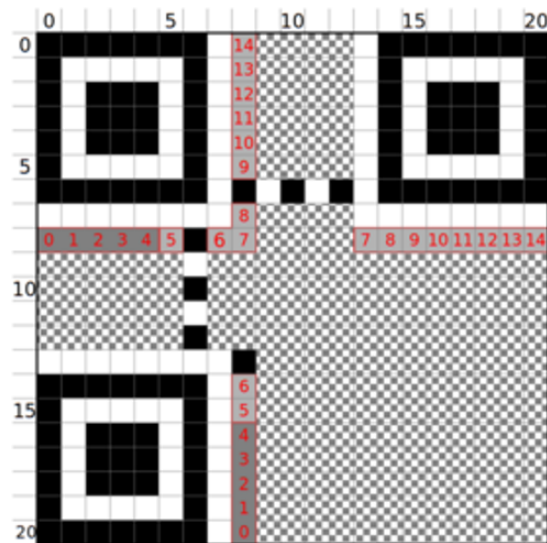


FIGURE 15 – Sens de lecture des bits correspondant au format du QR code.

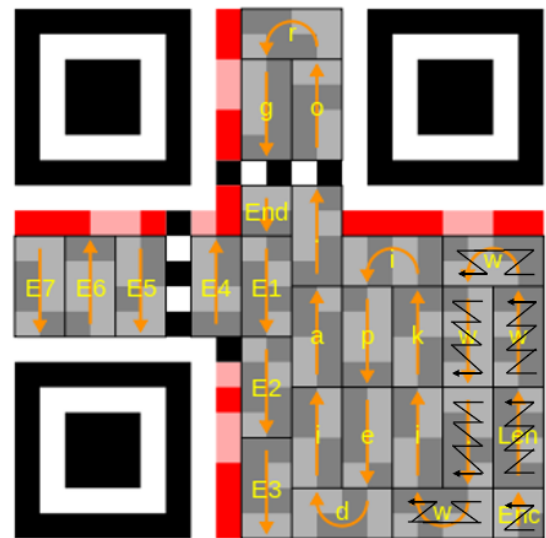


FIGURE 16 – Sens de lecture des octets d'un QR code.

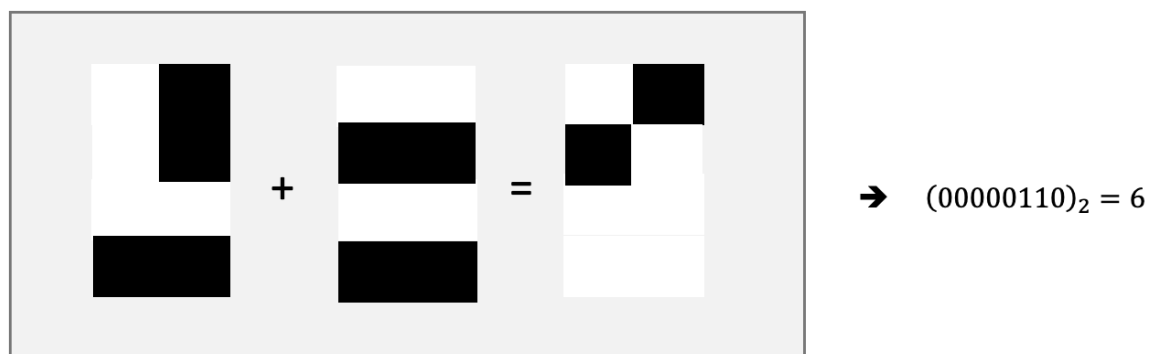


FIGURE 17 – Déchiffrement du nombre de caractères à décoder.

est représenté par les bits situés aux positions 0 et 1, tandis que le numéro du masque est représenté par les bits situés aux positions 2 à 4.

Pour décoder le QR code, nous devons récupérer le numéro du masque. Celui-ci est le 001. Nous devons également récupérer le nombre de caractères contenus dans le message. Pour cela, il faut déchiffrer l'octet correspondant en y appliquant le masque. Le masque 001 consiste à inverser la valeur des bits une ligne sur 2 (cf. les lignes noires dans le masque 001 de la Figure 13). Le déchiffrement de l'octet correspondant indique que le nombre de caractères à décoder est 6 (cf. Figure 17).

L'ordre de lecture des octets est commun à tous les QR codes (cf. Figure 16). Les positions des octets correspondants aux 6 caractères à décoder sont représentés dans la Figure 18.

Finalement, en appliquant le masque 001 aux valeurs des octets et en utilisant la table de conversion des chiffres binaires en caractères²¹, le décodage des octets produit les résultats

21. <https://www.ibm.com/docs/de/aix/7.1?topic=adapters-ascii-decimal-hexadecimal-octal-binary-conversion-table>

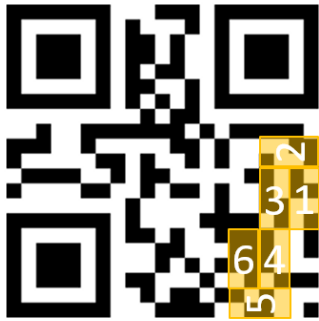


FIGURE 18 – Positions des octets à décoder.

Octet	Sans masque	Avec masque	Valeur ASCII
1	10001101	01000001	A
2	10110001	01110010	r
3	01011110	01101101	m
4	01010110	01100101	e
5	01011001	01100101	e
6	10111111	01110011	s

TABLE 3 – Décodage des octets avec l’ajout du masque pour obtenir les valeurs ASCII correspondantes

présentés dans la Table 3. La chaîne de caractères décodée est : "Armees".

C Exemples d'images obtenues après transformations des données du SNDS

Dans le Chapitre 2, nous explorons la représentation des données tabulaires sous forme d'images comme solution au fléau de la dimension, en nous concentrant sur le cas d'usage de la classification des pathologies. La Figure 19 montre le négatif des images moyennes pour chaque classe, obtenues après la transposition des données tabulaires en images. Cette visualisation nous permet d'identifier les pixels, correspondant aux caractéristiques des données, qui présentent des variations significatives entre les classes ou, au contraire, qui restent constants.

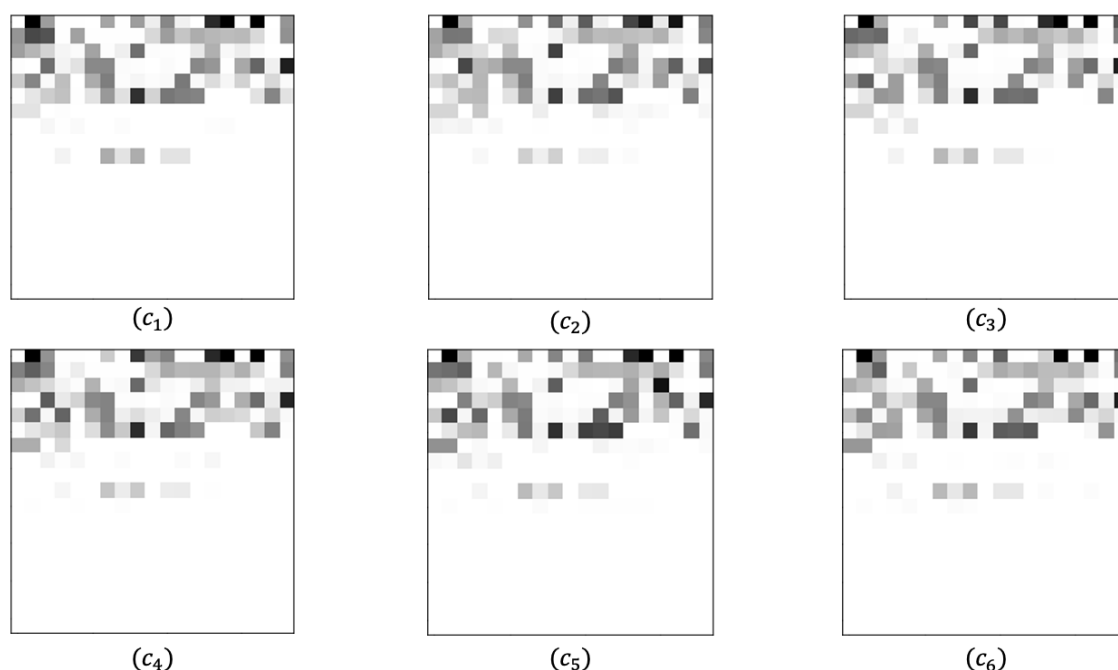


FIGURE 19 – Négatif des images moyenne de chaque classe. Où (c_1) : tumeurs; (c_2) : maladies endocriniennes, nutritionnelles et métaboliques; (c_3) : troubles mentaux et comportementaux, (c_4) : maladies du système circulatoire, (c_5) : maladies du système ostéo-articulaire, des muscles et du tissu conjonctif et (c_6) : lésions traumatiques, empoisonnements et certaines autres conséquences de causes externes.

D Codes CIM-10

Dans le Chapitre 4 sur l'évaluation des liens d'auto-attention appris par un Transformer, nous travaillons sur un cas d'usage d'étude des interactions entre diagnostics sur lesquelles le modèle s'appuie pour prédire le diagnostic de la prochaine visite. Les données que nous étudions sont issues du Système National des Données de Santé (SNDS) et sont codées d'après la 10ème édition de la Classification Internationale des Maladies (CIM-10). La Table 4 donne les noms associés aux codes CIM-10 présentés dans les Figures 4.4, 4.5, 4.6 et 4.7.

TABLE 4 – Liste (non exhaustive) des codes CIM10 et leur signification

Code CIM10	Signification
B37	Candidose
B96	Autres agents bactériens précisés, cause de maladies classées dans d'autres chapitres
C44	Autres tumeurs malignes de la peau
D12	Tumeur bénigne du côlon, du rectum, de l'anus et du canal anal
D50	Anémie par carence en fer
E11	Diabète sucré de type 2
E55	Carence en vitamine D
E66	Obésité et surpoids
E87	Autres déséquilibres hydro-électrolytiques et acido-basiques
F17	Troubles mentaux et du comportement liés à l'utilisation de tabac
F41	Autres troubles anxieux
G47	Troubles du sommeil
G56	Mononévrite du membre supérieur
H02	Autres affections des paupières
H25	Cataracte sénile
H26	Autres cataractes
H35	Autres affections rétinienne
H40	Glaucome
I25	Cardiopathie ischémique chronique
I48	Fibrillation et flutter auriculaires
J18	Pneumopathie à micro-organisme non précisé
K01	Dents incluses et enclavées
K21	Reflux gastro-oesophagien
K29	Gastrite et duodénite
K40	Hernie inguinale
K44	Hernie diaphragmatique
K57	Diverticulose de l'intestin
K59	Autres troubles fonctionnels de l'intestin
K63	Autres maladies de l'intestin
K80	Cholélithiase
M16	Coxarthrose [arthrose de la hanche]
N10	Néphrite tubulo-interstitielle aiguë
N18	Maladie rénale chronique
Suite sur la page suivante	

Table 4 – suite

Code CIM10	Signification
N20	Calcul du rein et de l'uretère
N23	Colique néphrétique, sans précision
N30	Cystite
N39	Autres affections de l'appareil urinaire
N77	Ulcération et inflammation vulvo-vaginales au cours de maladies classées ailleurs
O03	Avortement spontané
O04	Avortement médical
O20	Hémorragie du début de la grossesse
O21	Vomissements incoercibles au cours de la grossesse
O23	Infections de l'appareil génito-urinaire au cours de la grossesse
O26	Soins maternels pour d'autres affections liées principalement à la grossesse
O36	Soins maternels pour d'autres affections connues ou présumées du fœtus
O42	Rupture prématurée des membranes
O47	Faux travail
O60	Travail et accouchement prématurés
O62	Anomalies de la contraction utérine et de la dilatation du col
O63	Travail prolongé
O68	Travail et accouchement compliqués d'une détresse fœtale
O69	Travail et accouchement compliqués d'anomalies du cordon ombilical
O70	Déchirure obstétricale du périnée
O75	Autres complications du travail et de l'accouchement, non classées ailleurs
O81	Accouchement unique par forceps et ventouse
O92	Autres mastopathies et anomalies de la lactation associées à l'accouchement
O99	Autres maladies de la mère classées ailleurs mais compliquant la grossesse, l'accouchement et la puerpéralité
R06	Anomalies de la respiration
R07	Douleur au niveau de la gorge et du thorax
R10	Douleur abdominale et pelvienne
R26	Anomalies de la démarche et de la motilité
R33	Rétention d'urine
R41	Autres symptômes et signes relatifs aux fonctions cognitives et à la conscience
R53	Malaise et fatigue
S01	Plaie ouverte de la tête
S06	Lésion traumatique intracrânienne
S52	Fracture de l'avant-bras
T81	Complications d'actes à visée diagnostique et thérapeutique, non classées ailleurs
U07	Codes pour usage urgent
W19	Chute, sans précision
Z04	Examen et mise en observation pour d'autres raisons
Suite sur la page suivante	

Table 4 – suite

Code CIM10	Signification
Z29	Nécessité d'autres mesures prophylactiques
Z30	Prise en charge d'une contraception
Z35	Surveillance d'une grossesse à haut risque
Z37	Résultat de l'accouchement
Z39	Soins et examens du post-partum
Z47	Autres soins de contrôle orthopédiques
Z51	Autres soins médicaux
Z53	Sujets ayant recours aux services de santé pour des actes médicaux spécifiques, non effectués
Z64	Difficultés liées à certaines situations psychosociales
Z71	Sujets en contact avec les services de santé pour d'autres conseils et avis médicaux, non classés ailleurs
Z80	Antécédents familiaux de tumeur maligne
Z86	Antécédents personnels de certaines autres maladies
Z92	Antécédents personnels de traitement médical