Université Paris 8 - Vincennes Saint-Denis

École Doctorale nº 224 - Cognition, Langage, Interaction

Unité de recherche - Laboratoire d'Intelligence Artificielle et Sémantique des Données (LIASD)

Thèse de doctorat en Informatique

Discipline : Informatique

Graphes hétérogènes de connaissances pour la détection des Fake News multimodales

---

Thèse de doctorat présentée par

Gaël MAREC

Sous la direction de :

Nédra MELLOULI

Date de soutenance : 09/12/2025

Jury de thèse :

Lynda TAMINE LECHANI, Professeure Université Paul Sabatier Toulouse (Rapportrice)
Vincent CLAVEAU, Chargé de recherche CNRS, Université de Rennes (Rapporteur)
Nicolas JOUANDEAU, Professeur Université Paris 8 (Examinateur)
Nicolas TRAVERS, Professeur, HDR ESILV (Examinateur)
Eric FILIOL, HDR Thales, Head of Discipline CYBER, DATA & ALGO (Examinateur)

Résumé

Ce manuscrit de thèse présente une approche automatique de détection de *fake news* multimodales enrichie par des graphes de connaissances. Dans une première partie nous explicitons le contexte de massification de l'information dans lequel s'inscrit cette thèse. Nous y présentons les principales typologies d'approches de détection (basées sur le contenu, la source, ou le contexte). Dans notre état de l'art, nous présentons les approches basées sur le contenu, en nous concentrant sur les modalités textuelles et visuelles, puis celles enrichies par des connaissances. Nous mettons ainsi en évidence l'absence, dans la littérature, d'une méthode de détection combinant multimodalité, explicabilité et intégration de connaissances.

Dans la deuxième partie, nous exposons les prérequis formels et applicatifs du modèle. Nous y établissons une méthode de détection et localisation des transformations d'images pour la détection et l'explicabilité de la modalité visuelle. Nous y proposons des outils qui s'appuient sur le formalisme des graphes de connaissances et sur la compréhension sémantique des grands modèles de langage pour structurer l'information, qu'elle soit visuelle ou textuelle. Nous y expliquons aussi les approches essentielles pour exploiter les graphes à travers des représentations vectorielles apprises (ex. TransE, R-GCN).

Enfin, dans la troisième partie, nous détaillons notre système complet de détection, qui intègre un module de fusion multimodale fondé sur un mécanisme de pondération des signaux. Celui-ci permet de pondérer chaque signal (texte, image, connaissances) et de produire un score de probabilité prédictif explicable par les contributions de chacune des modalités.

Mots-clés : Intelligence artificielle, Désinformation, Vision par ordinateur, Représentation des connaissances, Traitement automatique du langage naturel

Abstract

This doctoral dissertation presents an automatic approach for the detection of multimodal *fake news*, enriched by knowledge graphs. In the first part, we explain the context of the massification of information in which this thesis is situated. We present the main types of detection approaches (based on content, source, or context). In our state of the art, we present content-based approaches, focusing on textual and visual modalities, then those enriched by knowledge. We thus highlight the absence, in the literature, of a detection method combining multimodality, explainability, and integration of knowledge.

In the second part, we present the formal and applicative prerequisites of the model. We establish a method for detecting and localizing image transformations for the detection and explainability of the visual modality. We also propose tools based on the formalism of knowledge graphs and on the semantic understanding of large language models to structure information, whether visual or textual. We also explain the essential approaches for exploiting graphs through learned vector representations (e.g., TransE, R-GCN).

Finally, in the third part, we detail our complete detection system, which integrates a multimodal fusion module based on a signal-weighting mechanism. This module makes it possible to weight each signal (text, image, knowledge) and to produce a probability score that is explainable by the contributions of each modality.

Keywords : Artificial intelligence, Disinformation, Computer vision, Knowledge re-

presentation, Natural language processing

Table des matières

I Contexte et Fondamentaux **7**

1 Introduction **8**
1.1 Contexte Général 8
1.1.1 Les réseaux sociaux, diffuseurs d'une désinformation massive 8
1.1.2 Fausses nouvelles : Un bref historique 9
1.1.3 Qu'est-ce qu'une fake news ? 9
1.1.4 Effets sur la société 11
1.1.5 Importance du contexte multimodal dans la détection de la désinformation 12
1.1.6 Les fake news sur les réseaux sociaux 12
1.2 Problématique et Motivations 13
1.3 Objectifs de la thèse 14
1.4 Présentation des contributions 15
1.4.1 Publications issues de cette thèse 15
1.5 Organisation du manuscrit 16

2 Etat de l'Art et Revue de la Littérature **17**
2.1 Fake News : Définition et Causes 17
2.1.1 Définition 17
2.1.2 Compréhension des causes 18
2.2 Panorama des approches existantes pour la détection 20
2.2.1 Typologie des approches selon le niveau d'intervention 20
2.2.2 Signal d'entrée pour la détection : contenu, source ou propagation ? 21
2.3 Jeux de Données pour la Détection des Fake News 24
2.3.1 Datasets Textuels 24
2.3.2 Datasets Exploitant le Contexte Social 26
2.3.3 Datasets Visuels 27
2.3.4 Datasets Multimodaux : Texte et Image 28
2.4 Exploitation du contenu visuel pour la détection des fake news 30
2.4.1 Les features forensiques 31
2.4.2 Les features sémantiques 31
2.4.3 Les features statistiques 32
2.4.4 Les features contextuels 33
2.4.5 Exemple de modèle : MVNN (Qi et al., 2019) 34
2.4.6 Benchmark des approches de détection visuelles 36
2.5 Exploitation du contenu textuel 36
2.5.1 Features linguistiques explicites 36
2.5.2 Réseaux de neurones profonds pour l'analyse textuelle 37

2.5.3 Présentation d'un modèle (Wani et al. 2021) . . . 38
2.5.4 Benchmark des approches de détection textuelles . . . 39
2.6 Approches Knowledge-based . . . 40
2.6.1 Méthodes basées sur les graphes de connaissances . . . 40
2.6.2 Méthodes basées sur les modèles de langage préentraînés (LLMs) . 42
2.6.3 Exemple de modèle : DEAP-FAKED (Mayank et al., 2021) . . . 42
2.6.4 Benchmark des approches Knowledge-based . . . 44
2.7 Exploitation de Contenu Multimodal : textuel et visuel . . . 45
2.7.1 Présentation d'un modèle : MVAE (Khattar et al. 2019) . . . 45
2.7.2 Benchmark des approches de détection multimodales . . . 46
2.8 Conclusion de l'état de l'art . . . 47
2.8.1 Limites des comparaisons quantitatives . . . 48
2.8.2 Tableau de synthèse qualitative . . . 48

II Fondements Formels et Applicatifs pour le Modèle de Détection 49

3 Fondamentaux des Graphes de Connaissances 51
3.1 Introduction . . . 51
3.1.1 Définition . . . 51
3.1.2 Distinction entre ontologies et graphes de connaissances . . . 53
3.2 Les graphes de connaissances pour le traitement des fake news . . . 53
3.2.1 Exploration des stratégies d'intégration des graphes de connaissances 54
3.3 Embedding de graphes de connaissances . . . 55
3.3.1 Modèles à distance translationnelle . . . 56
3.3.2 Modèles à matching sémantique . . . 58
3.4 Apprentissage des représentations . . . 60
3.4.1 Hypothèse de monde ouvert . . . 60
3.4.2 Hypothèse de monde fermé . . . 60
3.5 Embedding de graphes, au delà des faits . . . 61
3.5.1 Type des entités . . . 61
3.5.2 Chemin de relations . . . 61
3.5.3 Description textuelle . . . 61
3.6 Réseaux de neurones profonds pour l'embedding des graphes de connaissances 62
3.6.1 Des CNNs aux GCNs . . . 62
3.6.2 Le GCN classique . . . 64
3.6.3 Relational Graph Convolutional Networks (R-GCN) . . . 65
3.6.4 Pooling de graphes . . . 65

4 Application, LLMs pour la Génération de Graphes de Scènes 67
4.1 Introduction . . . 67
4.2 Travaux Connexes . . . 68
4.2.1 Génération historique des graphes de scènes . . . 69
4.2.2 Intégration de connaissances a priori . . . 69
4.2.3 Apports récents des LLMs et VLMs . . . 69
4.2.4 Positionnement de notre approche . . . 70
4.3 Preliminaires . . . 71
4.3.1 Notations . . . 71
4.4 Méthodologie . . . 71
4.4.1 Présentation Générale . . . 71

4.4.2 Génération de descriptions visuelles (VLM) . . . 72
4.4.3 Extraction des triplets sémantiques (LLM) . . . 73
4.4.4 Ancrage visuel des entités . . . 73
4.4.5 Post-traitement du graphe . . . 75
4.5 Experiments . . . 75
4.5.1 Datasets . . . 75
4.5.2 Protocoles expérimentaux . . . 76
4.5.3 Evaluation du Graphe de Scène Sémantique . . . 77
4.5.4 Evaluation du Graphe ancré visuellement . . . 77
4.5.5 Utilisation des graphes pour le Visual Question Answering (VQA) . 80
4.6 Discussion . . . 82
4.7 Conclusion et Perspectives . . . 83

5 Clustering Spectral et Clustering SNMF 84
5.1 Clustering Spectral . . . 84
5.1.1 Graphes de similarité . . . 84
5.1.2 Laplaciens de graphe et définitions . . . 85
5.1.3 L'algorithme de clustering spectral . . . 86
5.1.4 Origine du clustering spectral . . . 87
5.2 Clustering SNMF . . . 90

6 Localisation et Détection de transformations d'images 93
6.1 Localisation des Transformations d'Images, Détection de Communautés dans des Graphes . . . 93
6.1.1 Introduction . . . 94
6.1.2 Travaux Connexes . . . 97
6.1.3 Procédure globale . . . 97
6.1.4 Hypothèses sur la mesure de similarité . . . 98
6.1.5 Approches de clustering pour la détection d'images falsifiées . . . 99
6.1.6 Jeux de données . . . 100
6.1.7 Application web pour la génération d'images falsifiées . . . 103
6.1.8 Approche expérimentale et méthodologie . . . 103
6.1.9 Du patch-level au niveau pixel-level . . . 104
6.1.10 Métriques . . . 105
6.1.11 Résultats et discussion . . . 106
6.1.12 Conclusion et perspectives . . . 111
6.2 Détection des Transformations d'images . . . 112
6.2.1 ManTraNet : Présentation du modèle . . . 113
6.2.2 Génération de la carte de chaleur avec ManTraNet . . . 113
6.2.3 Extraction de la proportion de l'image suspecte . . . 114
6.2.4 Calcul de la probabilité que l'image soit transformée . . . 114
6.2.5 Recherche des hyperparamètres . . . 116
6.2.6 Expérimentations . . . 117
6.2.7 Résultats et analyse . . . 117
6.2.8 Distribution des features de clustering . . . 117
6.2.9 Discussion et perspectives . . . 120
6.3 Conclusion sur la détection et la localisation . . . 121

III Intégration et Applications : Système de Détection de Fake News Multimodales 122

7 Architecture Globale 124
7.1 Présentation de l'Architecture Complète . . . 124
7.1.1 Motivations et choix de conception : . . . 125
7.2 Intégration de la Prédiction Visuelle . . . 127
7.3 Enrichissement sémantique et par Graphes de Connaissances . . . 128
7.3.1 Construction du graphe de connaissances à partir du texte . . . 128
7.3.2 Construction du graphe de connaissances à partir de l'image . . . 132
7.4 Évaluation de la cohérence sémantique texte-image . . . 134
7.5 Apprentissage sur graphes et prédiction multimodale . . . 135

8 Expérimentations 137
8.1 Entraînement et optimisation du modèle . . . 137
8.1.1 Dataset d'entraînement . . . 137
8.1.2 Embeddings et architectures explorés . . . 138
8.1.3 Stratégie d'entraînement et réglages d'hyperparamètres . . . 140
8.1.4 Étude d'ablation . . . 141
8.1.5 Tests sur d'autres jeux de données . . . 147
8.2 Explicabilité des Résultats . . . 148
8.2.1 Analyses des prédictions sur deux exemples . . . 151
8.2.2 Évaluation de la qualité des graphes extraits . . . 152
8.2.3 Variabilité des graphes générés . . . 152
8.3 Synthèse des expérimentations . . . 158

9 Conclusion, Discussion, Perspectives 160
9.1 Synthèse des contributions . . . 160
9.2 Discussion des limites . . . 162
9.3 Perspectives de recherche . . . 162
9.4 Conclusion . . . 163

Première partie

Contexte et Fondamentaux

1 Introduction

Ce premier chapitre introduit le contexte dans lequel s'inscrit cette thèse. Nous y présentons les enjeux scientifiques liés à la désinformation contemporaine (Section 1.1), la problématique étudiée ainsi que les motivations sous-jacentes (Section 1.2), les objectifs de recherche poursuivis (Section 1.3) et les principales contributions apportées (Section 1.4). Enfin, nous décrivons l'organisation générale du manuscrit (Section 1.5).

1.1 Contexte Général

Dans cette section, nous exposons la nature de la désinformation dans nos sociétés. Nous en analysons les causes (psychologiques, sociologiques...) et ses effets sur le rapport des citoyens au réel. Enfin, nous abordons les différentes formes que peut prendre la désinformation, notamment à travers ses représentations sur les réseaux sociaux.

1.1.1 Les réseaux sociaux, diffuseurs d'une désinformation massive

Dans nos sociétés modernes, l'accès à des quantités massives d'informations s'est largement démocratisé. Lorsque ces informations sont des traductions descriptives du réel, elles révèlent toutes leurs vertus éducatives. Mais au contraire, lorsqu'elles ne relaient que des falsifications et tromperies, elles obscurcissent le jugement des citoyens sur le monde et par conséquent détériorent leur rapport au réel. La désinformation n'est certes pas nouvelle, mais les réseaux sociaux (Twitter, Facebook ...) ont considérablement amplifié et accéléré la diffusion de ce que l'on appelle communément les *fakes news*. Plusieurs raisons expliquent que ces médias sont des vecteurs de propagation des fake news. D'abord, tous les utilisateurs peuvent publier des informations, sans aucune vérification ni compétence journalistique préalable, ce qui engendre la circulation d'informations de moins bonne qualité (voire totalement erronnées) en comparaison avec les médias traditionnels (journaux, télévision, radio). De plus, la structure interconnectée des réseaux sociaux permet la diffusion rapide et virale des fausses informations, créant un effet d'amplification dont les effets sont difficiles à contrôler. Enfin, la personnalisation des algorithmes de recommandation utilisés par ces réseaux peut contribuer à favoriser l'écho des fake news en les mettant en avant auprès d'un public ciblé et vulnérable.

Lors d'évènements politiques de premier plan, tels qu'une guerre ou une élection présidentielle, la propagation de fake news peut avoir des conséquences dévastatrices. Par exemple, elles sont en capacité d'influencer un candidat ou un autre, et de servir d'outil de propagande. C'est pourquoi, l'identification précoce des fake news est un enjeu majeur de notre temps.

Cependant, l'identification des fake news est loin d'être triviale et les différentes acceptions du terme mènent à diverses approches de détection. Outre le soucis de bonne définition, les fake news sont particulièrement difficiles à détecter pour l'être humain, no-

tamment parce que toute information, qu'elle soit ou non authentique, porte en sa qualité une part de vraisemblance.

Dans ce contexte, et face à la massivité du phénomène, s'appuyer uniquement sur le jugement humain ne suffit pas. C'est pourquoi le recours à des processus automatiques répond à l'exigence d'analyse pour la détection des *fake news*. Ce besoin d'outils systématiques constitue l'une des motivations des travaux présentés dans cette thèse.

Cependant, pour comprendre les difficultés intrinsèques à leur détection, il importe de rappeler que les *fake news* sont bien plus anciennes que les réseaux sociaux. En effet, la désinformation est aussi vieille que l'information elle-même, et s'est manifestée à chaque étape de l'évolution des médias. La sous-section suivante revient ainsi sur quelques jalons significatifs de cette histoire, afin de montrer que si les supports ont évolué, les logiques de manipulation de l'opinion restent étonnamment constantes.

Par chance les progrès récents dans le domaine de l'intelligence artificielle ont permis l'émergence d'algorithmes capables d'affronter le fléau des fake news. Nous présentons dans la suite, différentes définitions des fake news communément admises dans la littérature

1.1.2 Fausses nouvelles : Un bref historique

Les fake news, ou "fausses nouvelles" ont toujours existé. Avant la création de l'imprimerie en 1450, leur transmission était essentiellement orale, intégrée au folklore, aux récits des voyageurs, ainsi qu'aux discours de l'Eglise et de la royauté en France. Au XVIIIe siècle, des novellistes (terme idoine pour désigner les journalistes de l'époque) se réunissaient sous un grand marronier de l'allée du Palais-Royal pour y échanger de nombreuses informations fausses, appelées "craques". Donnant ainsi naissance à l'expression "arbre de Cracovie". Au XIXe siècle, la presse française prend une part majeure dans l'information des citoyens, et en 1849, alors que de nombreuses fausses nouvelles sont publiées par ce biais, un *délit de fausses nouvelles* est crée. Bien que le phénomène des fausses informations ne soit donc pas nouveau, le terme *fake news* a connu un grand gain de popularité en 2016 lors des élections présidentielles américaines. Aujourd'hui avec l'explosion des réseaux sociaux et un accès facile à l'information, la propagation des fake news a pris une ampleur sans précédents.

Cet aperçu historique et la Figure 1.1 montrent que la désinformation n'est pas un phénomène nouveau, bien que ses supports et ses canaux de diffusion aient évolué. Pourtant, quant à eux les mécanismes de manipulation ont persisté à travers le temps. Néanmoins, malgré cette continuité, la notion contemporaine de *fake news* ne fait pas pleinement consensus dans la littérature scientifique. Nous apportons donc ci-après des éclaircissements sur les différentes acceptions de ce terme.

1.1.3 Qu'est-ce qu'une fake news ?

Définition du terme : Le concept de *fake news* ne fait pas l'objet d'une définition univoque dans la littérature scientifique. De manière générale, ce terme désigne des informations factuellement fausses qui se présentent comme vraies. Une définition fréquemment adoptée consiste à considérer comme *fake news* toute information intentionnellement et vérifiablement fausse. Selon cette approche, les *fake news* reposent sur deux piliers : *l'intention malicieuse* et *l'inauthenticité*. Cette définition est largement utilisée dans la littérature récente [129][96].

Les différentes définitions du terme varient généralement selon deux axes : le degré d'authenticité du contenu et l'intentionnalité de sa diffusion. Pour qu'un système de détection soit efficace, il est nécessaire de poser une définition opérationnelle et sans ambiguïté



FIGURE 1.1 – A gauche un extrait d'un *occasionnel* (ancêtre de la presse) daté de 1531, représentant un dragon menaçant Paris. Il s'agit naturellement d'une fausse information utilisée dans un contexte de propagande catholique visant à effrayer les protestants [44] https://gallica.bnf.fr. A droite un exemple contemporain d'une fausse information virale, issue du compte Twitter `@Pediavenir`, diffuseur de désinformation à visée parodique. Le tweet combine un texte sensationnaliste à une image manifestement falsifiée de l'acteur américain Brad Pitt. Ces deux cas illustrent une stratégie récurrente de manipulation de l'opinion par le biais d'une mise en scène *multimodale* mêlant texte et image, renforçant l'apparence de véracité du message.

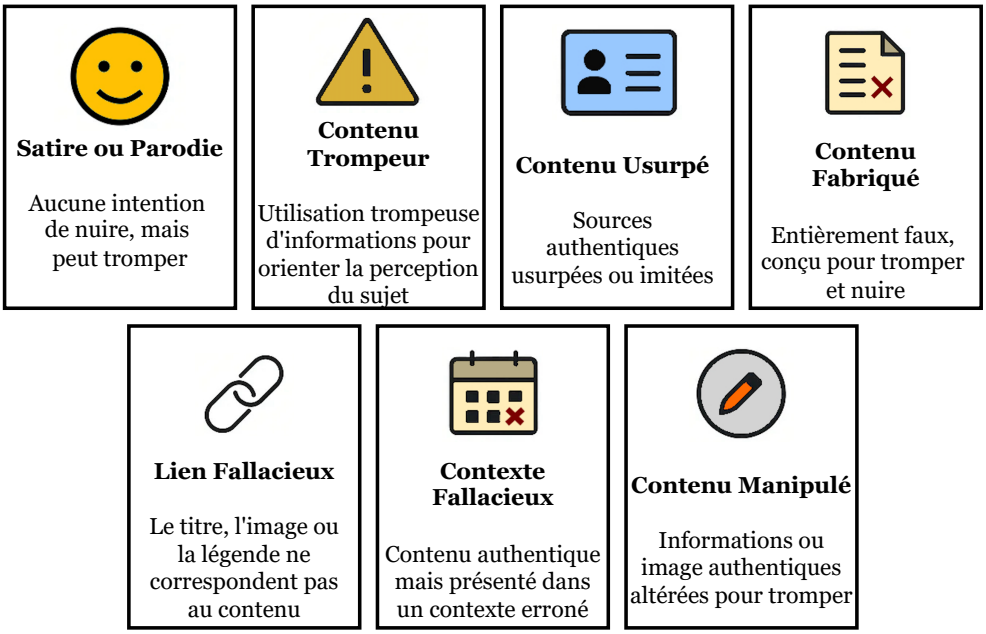


FIGURE 1.2 – Typologie des contenus trompeurs proposée Claire Wardle. Cette classification distingue sept types de mésinformation et désinformation selon l'intention de nuire et la manière dont les faits sont déformés, fabriqués, ou sortis de leur contexte (source : adaptation d'après [158]).

du concept de *fake news*, afin d'identifier clairement les critères mobilisables.

Définition 1.1.1. Une *fake news* est une information intentionnellement et vérifiablement fausse [129].

Cette définition exclut certains contenus trompeurs, comme les parodies dont le contenu peut être mensonger, mais sans visée malveillante ou encore les théories du complot, souvent invérifiables. Elle est donc relativement restrictive. Toutefois, elle peut servir de base à des extensions vers des problématiques connexes. Les caractéristiques exploitées pour identifier les fausses informations selon cette définition restent pertinentes dans des contextes plus larges.

Par ailleurs, Claire Wardle, professeure à Brown et spécialiste de la désinformation, propose une typologie en sept catégories (présentée dans la Figure 1.2), fondée sur l'intention et le type de manipulation [158]. Elle montre que le phénomène des *fake news* ne se limite pas à la désinformation volontaire, mais couvre un spectre plus large de contenus trompeurs.

Ainsi, tout en ayant conscience de la diversité de définitions des fake news, nous en conservons une caractérisation opérationnelle fondée sur l'intentionnalité et la falsification. Au-delà des considérations conceptuelles, nous examinons à présent les effets concrets des fake news.

1.1.4 Effets sur la société

Effets cognitifs et politiques : Les fake news, en questionnant la science, les faits établis et normes sociétales, rédéfinissent les rapports de la population aux faits et à la

vérité [100]. Quantifier les effets des fake news sur la société est une tâche complexe du fait de la diversité des réseaux sociaux et des sujets abordés. La mesure de ces effets peut reposer notamment sur l'analyse de la diffusion des contenus fakes (combien de fois ces contenus ont été partagé, aimés ou commentés), sur des enquêtes d'opinions traduisant le rapport de confiance de la population aux médias, sur l'étude de certains évènements réels (tels que des élections, des manifestations, des émeutes etc).

Amplification par les réseaux sociaux : Par essence, tous les réseaux sociaux ont pour objectif de générer du contenu viral, attirant et retenant le plus grand nombre d'utilisateurs. Cette mécanique permet naturellement aux réseaux d'augmenter leurs profits [93]. Dès lors, cette stratégie commerciale est structurante dans la conception des algorithmes qui les régissent : les contenus proposés aux utilisateurs sont classés automatiquement en prenant compte entre autres de la viralité des contenus. Ce phénomène permet aux fake news d'être largement mises en valeur par les réseaux sociaux, accroissant mécaniquement leur effet négatif sur la société. Par exemple, face à la crise du COVID-19, Tedros Ghebreyesus le directeur général de l'organisation mondiale de la santé (OMS) souligne l'émergence d'une véritable *infodémie* progressant en même temps que la maladie et freinant les politiques sanitaires gouvernementales.

Les fake news constituent donc un enjeu sociétal majeur, dont les impacts dépassent le simple cadre informationnel.

Il importe d'étudier leur structure composite multimodale où textes, images et vidéos interagissent pour renforcer leurs effets persuasifs. C'est l'objet de la sous-section suivante.

1.1.5 Importance du contexte multimodal dans la détection de la désinformation

Constat de la mutlimodalité : Les fake news présentent intrinsèquement un caractère multimodal, combinant généralement des éléments textuels et visuels pour renforcer leur impact. Chaque modalité enrichit le contexte informationnel du message : le texte explicite l'énoncé principal, tandis que l'image ou la vidéo qui l'accompagne vient illustrer, appuyer, ou orienter sa réception. Les contenus trompeurs de la Figure 1.1 illustrent bien ce mécanisme, où les images justifient les assertions mensongères portées par le texte.

Conséquence : Cette structuration multimodale est facilitée et même favorisée par les mécanismes internes des réseaux sociaux. En effet, les contenus incluant à la fois du texte et des médias visuels génèrent davantage d'engagement (clics, partages, commentaires). Dès lors, la multimodalité devient un vecteur stratégique de diffusion et de viralité pour les fake news.

Dans ce cadre, les réseaux sociaux apparaissent comme un écosystème particulièrement propice à la prolifération des fausses informations.

1.1.6 Les fake news sur les réseaux sociaux

Les fake news sur les réseaux sociaux se manifestent de différentes manières :

- Assertions mensongères, souvent sensationnalistes
- Des montages photos ou vidéos
- Usurpation d'identité ou comptes automatisés (bots)
- Rediffusion d'informations fausses, via le partage de contenu entre utilisateurs.

Ainsi les fake news sont des données complexes et hétérogènes. En effet elles peuvent se composer d'objets de nature différentes d'une part (tel que du texte et des images

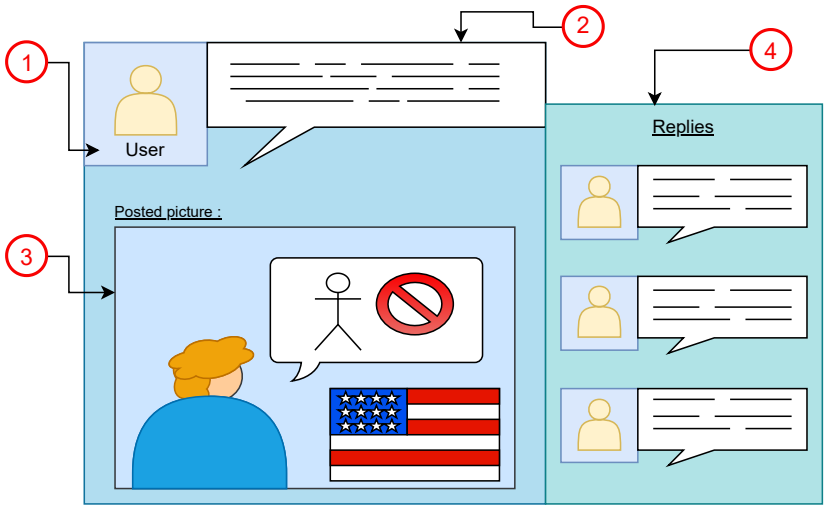


FIGURE 1.3 – Représentation schématique de la structure d'une fake news sur un réseau social. ① : Utilisateur qui poste la fake news, il peut s'agir d'un individu dans l'erreur ou bien malicieux, d'un robot, d'une identité usurpée. ② : Contenu textuel du post. ③ : Contenu visuel du post, il peut s'agir d'une image ou d'une vidéo. ④ : Réponses au post initial, il s'agit du contexte social du post.

par exemple) et d'autre part elles font intervenir de nombreux facteurs contextuels et sociaux (de quel sujet est-il question, à qui est-il destiné, par qui est-il diffusé...). Cette complexité rend leur détection difficile et soulève de nombreux défis scientifiques, que nous présentons dans la section suivante.

1.2 Problématique et Motivations

La diffusion des fake news se caractérise par la complexité des supports qu'elle mobilise, qu'ils soient visuels, textuels ou contextuels. Cependant, les approches automatiques de détection des fake news peinent à exploiter l'ensemble de ces supports, d'une manière à la fois unifiée et explicable.

Limites des approches manuelles : De nombreux travaux journalistiques de *fact-checking* manuel cherchent à désamorcer les *fake news* par une investigation approfondie (voir par exemple les initiatives de l'AFP Factuel [1] ou des Décodeurs du Monde [2]). Ces démarches offrent une détection précise et fortement explicable. Cependant, elles souffrent d'une lenteur importante au regard de la vitesse de diffusion et d'émergence de nouvelles fake news, ce qui limite leur efficacité préventive. Elles mettent néanmoins en évidence l'importance d'une explicabilité approfondie pour rétablir la confiance des citoyens envers l'information.

Limites des approches basées sur le contenu : De nombreuses approches automatiques s'appuient exclusivement sur le contenu des news pour prédire leur inauthenticité. Elles s'appuient notamment sur des méthodes de traitement automatique du langage ou d'analyse d'images, et obtiennent des performances respectables sur certains jeux de données. Toutefois, ce traitement décontextualisé reste insuffisant pour saisir la complexité réelle des fake news. Ainsi, si ces approches constituent un socle utile, elles peuvent être

1. `https://factuel.afp.com`
2. `https://www.lemonde.fr/les-decodeurs/`

enrichies par des connaissances externes, à la manière d'un travail journalistique humain, ouvrant la voie à de meilleures performances et à une éthique renforcée par plus d'explicabilité.

Il apparaît donc prometteur de recourir à une représentation plus unifiée et structurée, capable de relier entre elles les différentes composantes des fake news. Dans cette perspective nous proposons l'usage de graphes de connaissances pour enrichir contextuellement le contenu des news examinées.

Motivations : Face aux limites soulevées, l'un des enjeux consiste à concevoir une représentation des news capable d'intégrer de manière cohérente les différentes modalités impliquées (texte, image) ainsi que des connaissances contextuelles.

Les graphes de connaissances sont des outils formels de structuration de tout type de données. Ils permettent d'intégrer des entités (mots, objets visuels, utilisateurs, sources) et des relations qui les unissent, offrant ainsi une représentation riche et flexible. De plus, ils peuvent être enrichis continuellement de nouvelles connaissances par scrapping ou inférence logique. A cet effet ils offrent un cadre particulièrement adapté pour représenter des données hétérogènes et multimodales comme les fake news.

De plus, les graphes ouvrent la voie à une meilleure explicabilité la détection d'une *fake news* peut être reliée à des relations ou communautés identifiées comme suspectes, ce qui rapproche les méthodes automatiques d'une démarche de vérification humaine.

Ces caractéristiques font des graphes de connaissances une solution structurelle prometteuse pour dépasser les limites des approches actuelles et constituent la base des objectifs poursuivis dans cette thèse.

1.3 Objectifs de la thèse

L'objectif général de cette thèse est de proposer une approche automatique de détection de fake news multimodales qui soit explicable et enrichie par des connaissances externes. La difficulté est de dépasser les limites des approches traditionnelles centrées, soit sur le contenu en laissant de côté l'explicabilité et les connaissances externes, soit uniquement basée sur les connaissances ou le contexte social et laisse tomber le travail sur le contenu. Nous souhaitons donc composer une méthode complète capables d'intégrer simultanément des modalités textuelles, visuelles et contextuelles.

Ainsi, nous poursuivons trois objectifs principaux :

Objectif 1 : Développer une méthode de détection et localisation des transformations visuelles présentes dans des images. Cette objectif adresse le problème de détection automatique et explicable de fake news unimodales visuelles.

Objectif 2 : Concevoir une méthode de structuration des informations d'une news pour chaque modalité sous forme de graphe de connaissances et enrichis par des sources externes. Cela correspond à la représentation unifiée des news multimodales, c'est une étape nécessaire pour tout traitement multimodal explicable.

Objectif 3 : Construire un système *end-to-end* de détection de fake news multimodales, intégrant un module de fusion qui aligne la détection des transformations visuelles et les représentations multimodales enrichies par des connaissances. Nous visons ainsi la mise en œuvre d'un modèle prédictif robuste et explicable.

Ces objectifs constituent le fil directeur des contributions présentées dans ce manuscrit.

1.4 Présentation des contributions

Nous apportons avec cette thèse, trois contributions principales, directement alignées avec les objectifs énoncés précédemment.

Contribution 1 : Détection et localisation des transformations d'images Nous proposons une méthode de détection et de localisation des transformations visuelles dans les images, capable d'identifier une grande variété de falsifications numériques locales explicites. La localisation identifie les zones de l'images altérées, aidant à l'explicabilité. La détection quant à elle, fournit une probabilité de transformation de l'image. Cette approche fournit une base explicable pour la détection des fake news unimodales basée sur le contenu visuel.

Contribution 2 : LLM pour la génération de graphes de connaissances multimodaux Nous proposons plusieurs méthodes qui représentent et structurent les informations issues des fake news sous forme de graphes, en exploitant les capacités des *Large Language Models* (LLM). Nos travaux incluent d'abord la génération de graphes de scène à partir du contenu visuel, puis une méthode de détection visuelle d'entités nommées, et enfin l'extension de ces approches à la construction de graphes de connaissances multimodaux, intégrant simultanément texte et image et enrichis par des bases de données externes.

Cette contribution fournit une représentation explicite, unifiée et contextualisée des contenus multimodaux, nécessaire pour un traitement prédictif en aval.

Contribution 3 : Système de détection de Fake News multimodales Nous avons construit un système *end-to-end* de détection des *fake news* multimodales, intégrant un module de fusion capable d'unifier la détection des transformations visuelles et les représentations multimodales enrichies. Ce système offre une prédiction explicable par les contributions de chacune des modalités, et validée expérimentalement sur plusieurs jeux de données.

Ces contributions forment le socle de la démarche scientifique développée dans ce manuscrit et seront détaillées dans les chapitres suivants.

1.4.1 Publications issues de cette thèse

Les contributions développées dans ce manuscrit ont été reconnues par la communauté scientifique et ont donné lieu aux publications suivantes :

- **Conférence nationale à comité de lecture** : G. Marec, N. Mellouli. *Factorisation de matrices symétriques non négatives appliquée à la détection de communautés dans des graphes et l'analyse forensique d'images*. Conférence Extraction et Gestion de Connaissances (EGC 2024).
- **Journal international à comité de lecture** : G. Marec, N. Mellouli. *Symmetric non negative matrices factorization applied to the detection of communities in graphs and forensic image analysis*. *Data Knowledge and Engineering*, 2025.

Ces publications témoignent de la visibilité et de la diffusion des travaux menés dans cette thèse, ainsi que de leur validation par des instances d'évaluation scientifique externes.

Autres travaux soumis

Une partie des travaux de cette thèse a également été soumise à la conférence internationale ACM Multimedia (ACMMM 2025). Bien que cette contribution n'ait pas été retenue, nous avons pris en compte les retours constructifs des reviewers afin d'enrichir ce manuscrit.

1.5 Organisation du manuscrit

Ce manuscrit est organisé en neuf chapitres.

— Le Chapitre 1 introduit le contexte de la thèse, présente les difficultés liées à la détection des fake news, leurs définitions, leurs effets, ainsi que les problématiques, les objectifs et les contributions de ce travail.
— Le Chapitre 2 propose un état de l'art sur les approches existantes de détection des *fake news*. Nous nous concentrons notamment sur les méthodes basées sur le contenu des news en couvrant à la fois les méthodes unimodales (textuelles ou visuelles), multimodales et fondées sur les connaissances contextuelles.
— Le Chapitre 3 aborde les fondamentaux sur les graphes de connaissances, de leur formalisation à l'apprentissage de représentations vectorielles adaptées pour la détection de fake news.
— Le Chapitre 4 présente une de nos contributions, consacrée à la génération de graphes de scène avec des LLM, pour la structuration de la modalité visuelle.
— Le Chapitre 5 est un chapitre méthodologique sur le clustering de graphes, utile dans la suite pour la localisation des transformations d'images.
— Le Chapitre 6 présente notre approche de détection et localisation des transformations d'images.
— Le Chapitre 7 détaille l'architecture globale de notre système de détection multimodale, ainsi que les choix méthodologiques sous-jacents.
— Le Chapitre 8 présente l'ensemble des expérimentations menées pour entraîner et évaluer notre système de détection de fake news, accompagnées d'analyses quantitatives et qualitatives.
— Enfin, le Chapitre 9 conclut ce manuscrit, en proposant une synthèse des contributions, une discussion des limites et des perspectives de recherche.

Conventions d'écriture

Nous prenons le parti dans ce manuscrit, de conserver en l'état certains termes techniques en anglais issus de la littérature internationale en informatique (par exemple *clustering*, *embedding*, *fine-tuning*), lorsqu'aucune traduction française ne fait consensus ou ne rend compte de la précision du concept. Cette pratique vise à rester cohérent avec l'usage courant dans la communauté scientifique, tout en facilitant la lecture et la comparaison avec les travaux existants.

2 Etat de l'Art et Revue de la Littérature

La désinformation, à travers les *fake news*, se propage aujourd'hui à une vitesse inédite via les réseaux sociaux, menaçant la confiance publique et la qualité du débat démocratique. Face à cet enjeu majeur, la recherche en traitement automatique du langage et en vision par ordinateur a développé de nombreuses méthodes de **détection automatique des fake news** basées sur les **contenus** publiés par les utilisateurs.

Ce chapitre propose un panorama structuré des travaux existants, selon quatre axes principaux :

(i) Approches textuelles : Exploitation de features linguistiques (stylistiques, sémantiques) et recours aux modèles de langage pré-entraînés (BERT, GPT, ...).

(ii) Approches visuelles : Analyses forensiques et sémantiques d'images (CNN, ViT).

(iii) Approches knowledge-based : Intégration de graphes de connaissances et de bases factuelles (Wikidata, ConceptNet, LLMs) pour corroborer ou infirmer des assertions.

(iv) Approches multimodales : Fusion de modalités texte et image, souvent via des mécanismes de co-attention ou d'alignement d'embeddings (SpotFake, MCAN, SDML).

Pour chacun de ces axes, nous présenterons les principes clés, les modèles de référence, ainsi que leurs avantages et limites. Cette analyse permettra de mettre en évidence les défis actuels, au delà aussi des simples performances quantitatives, telles que les difficultés d'explicabilité, ou l'inflation de la valeur des données. Ces difficultés mettent en exergue le manque, et donc la nécessité d'une méthode à la fois **multimodale**, **fortement explicable** et **enrichie par des connaissances externes**, sur laquelle se fonde cette thèse.

2.1 Fake News : Définition et Causes

Dans cette section nous présentons une première définition du problème de détection des fake news, ainsi que la nature des causes les induisant.

2.1.1 Définition

Dans ce travail nous nous intéressons exclusivement à la tâche de détection des fake news c'est à dire à détecter si une information est fausse ou légitime. D'autres tâches, telles que la détection des bots sur les réseaux sociaux bien que connexes, ne seront pas directement étudiées. Shu et al. [129] proposent une définition pour la tâche de détection des fake news sur les réseaux sociaux :

Définition 2.1.1. (Détection des Fake News) Étant donné les interactions sociales $\mathcal{E}$ entre n utilisateurs pour l'information a, la tâche de détection de fausses nouvelles consiste à

prédire si l'article a est une fausse nouvelle ou non, c'est-à-dire, $\mathcal{F} : \mathcal{E} \to \{0, 1\}$, tel que :

$$\mathcal{F}(a) = \begin{cases} 1, & \text{si } a \text{ est une fausse nouvelle,} \\ 0, & \text{sinon.} \end{cases}$$

Ainsi l'on considère le problème de détection des fake news comme un problème de classification binaire des informations, en incluant le contexte social. Cela permet de simplifier le modèle et d'avoir un objectif clair de séparation des informations fausses et vraies. D'un point de vue utilisateur, une classification binaire de l'information facilite l'analyse et la prise de décision quant au traitement des fausses informations. La binarité permet également de simplifier l'évaluation et de faciliter l'annotation des jeux de données d'entraînement.
Cependant il est important de noter qu'il s'agit d'une hypothèse de travail et que traiter le problème des fake news comme un problème de classification multi classes présente aussi de l'intérêt. En effet, du fait de leur complexité, les fake news ne se limitent pas nécessairement à être simplement fausses. Il existe différents niveaux de fausseté, en quantité et en qualité. Certaines données peuvent être ambiguës, contenir à la fois des informations vraies et fausses, vraies mais mal interprétées etc. Certains dataset proposent plusieurs niveaux de vérité. Ainsi le dataset LIAR [153] contient un ensemble de 12.8K déclarations textuelles classifiées en 6 niveaux de vérité : *pants-fire, false, barely-true, half-true, mostly-true, true.*

2.1.2 Compréhension des causes

Une des solutions au problème des fake news est d'identifier et de contrer directement leurs causes. Cette tâche est difficile car les causes sont nombreuses. En effet, si l'on peut penser d'abord à des intentions malicieuses (rentabilité financière, manipulation idéologique [96]), la ténacité des fake news repose grandement sur des biais cognitifs, des erreurs, et des mécanismes propres aux réseaux sociaux [140, 106].

Causes psychologiques

La compréhension des causes psychologiques des fake news permet d'expliquer les mécanismes qui poussent les individus à diffuser de fausses informations. Il est intéressant de noter qu'il y a une différence substantielle entre les croyances des individus et ce qu'ils postent réellement sur les réseaux sociaux. De plus, cette différence s'explique davantage par des erreurs d'inattention que par une volonté de nuire ou de militantisme hyper-partisan [106].
En particulier il existe des *biais cognitifs* qui favorisent largement la propagation des faux contenus [129]. Le *réalisme naïf* est un biais décrivant la propension des utilisateurs à considérer leur propre perception du monde comme la plus proche de la réalité et à dénigrer les perceptions différentes. Le *biais de confirmation* est la préférence des utilisateurs pour des informations qui confirment leurs a prioris. Ainsi certaines métriques sont utiles pour quantifier le rapport des individus à la vérité des informations, parmi lesquelles :

Définition 2.1.2. *Discernement :* Le discernement d'un individu est la différence normalisée entre sa croyance en de vraies informations et sa croyance en de fausses informations. Un discernement de 1 correspond à une croyance totale dans les vraies informations et un rejet total des fausses. Un discernement de -1 correspond à une croyance totale dans les fausses informations et un rejet total des vraies.

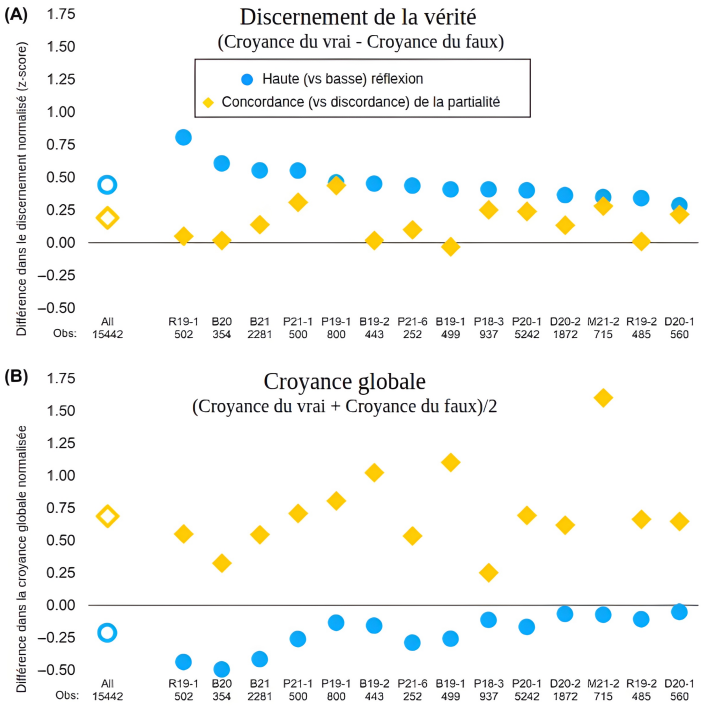


FIGURE 2.1 – Méta-analyse comparative sur la croyance globale et le discernement des individus, en fonction de leur temps de réflexion et de la concordance de leurs opinions avec les informations (source [106]).

Définition 2.1.3. *Croyance globale :* La croyance globale d'un individu correspond à la moyenne entre sa croyance dans les vraies informations et sa croyance dans les fausses.

Définition 2.1.4. Le test de reflexion cognitive [38] (CRT) est une mesure de la propension d'un individu à réfléchir à une question posée et à résister à l'envie de donner la première réponse lui venant à l'esprit.

Les résultats de la méta-analyse présentée dans la figure 2.1[106] montrent que la concordance entre l'opinion soutenue par une informations et l'opinion du lecteur augmente d'une part la croyance globale en l'information (qu'elle soit vraie ou fausse) et d'autre part le discernement de l'individu, mais tout de même moins qu'un CRT 2.1.4 élevé. Ainsi, contrairement à une idée répandue, dans la plupart des cas la recherche de concordance avec ses propres opinions politiques n'est pas le facteur psychologique prépondérant dans la diffusion des fakes news par un individu. En outre un CRT élevé augmente le scepticisme face à toutes les informations.
Une étude de 2021 [107] montre que la véracité des informations n'a que peu d'effet sur leur partage. Pourtant les utilisateurs désirent en majorité diffuser de vraies informations. Ainsi en portant subtilement l'attention d'utilisateurs de Twitter sur la précision des contenus, les auteurs de cette étude parviennent à réduire les partage de fausses informations.
Il est également important de comprendre les ressorts psychologiques en jeux après la détection de fake news. En effet, il est difficile de corriger une mauvaise perception déjà établie. En particulier la correction par fact checking de certaines informations est parfois contre productive et polarise davantage les utilisateurs [106].

Causes Sociologiques

L'étude des causes sociologiques des fake news contribue à la compréhension de la propagation des fausses informations sur les réseaux sociaux et dans la société. Par exemple la modélisation du phénomène des *chambres à échos* (amplification de la valeur d'une idée au sein d'un groupe de personnes aux opinions similaires) [143, 4] permet de comprendre comment se diffusent et s'exacerbent des croyances existantes, rendant les utilisateurs plus susceptibles de prendre pour vraies des informations fausses. De plus, les fausses informations ne se propagent pas de la même manière ou à la même vitesse selon les différents groupes sociaux. De fait, les personnes agées sont plus vulnérables aux fake news [106]. D'autres paramètres tels que le niveau d'éducation, ou de confiance dans les réseaux sociaux [159] ou encore l'orientation politique [106] impactent directement la propagation des fake news.

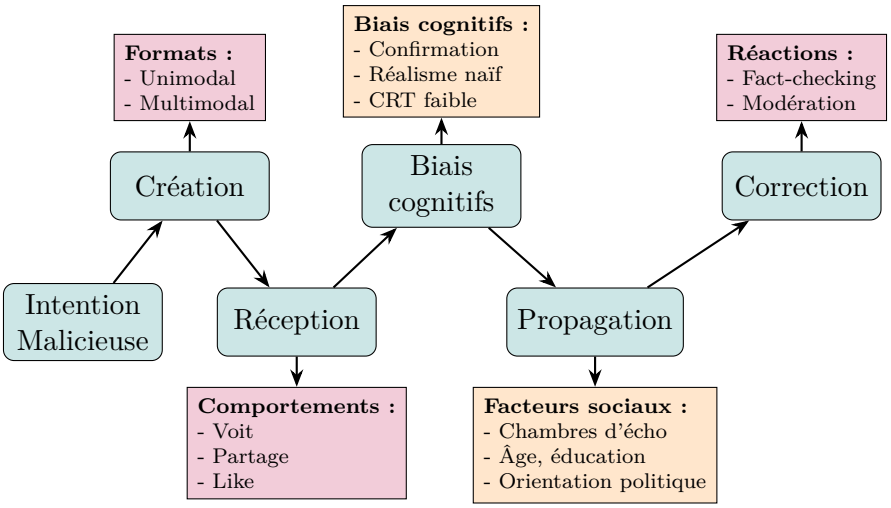


FIGURE 2.2 – Cycle de diffusion d'une fake news sur un réseau social et facteurs influents. Une fake news est postée, constituée d'une ou plusieurs modalités. Les utilisateurs adoptent un comportement, partagent, enrichissent ou ignorent le contenu en fonction de leurs affects, de facteurs sociaux et psychologiques. Le contenu est ensuite potentiellement corrigé, modéré par la plateforme, ou fact-checké .

2.2 Panorama des approches existantes pour la détection

Du fait de la diversité des causes, et la complexité des fake news, les travaux de détection dans la littérature reposent sur plusieurs approches. Ces approches, que nous explicitons dans cette section, interviennent à différentes temporalité du processus de détection des fake news : de l'amont, avec de la récolte de données, à l'aval avec un travail sur la gestion post détection.

2.2.1 Typologie des approches selon le niveau d'intervention

Le figure 2.3 illustre une hiérarchie des approches possibles dans la recherche pour la détection des fake news et la réduction de leur impact, allant de l'analyse des données brutes à l'intégration dans des applications concrètes. Ces approches correspondent chacune à des problématiques de recherches dépendantes les unes des autres.

Approches orientées Données : Les approches orientées données se concentrent sur l'exploitation d'un grand nombre de données provenant de sources diverses pour identifier les motifs récurrents ou anomalies qu'induisent les fake news. Cela inclut la collecte des données, la construction de datasets d'entraînement pour des modèles de détection.

Approches orientées Features : Les approches orientées features se basent sur l'extraction des features pour améliorer la détection. Ces features peuvent être multimodaux ou non, provenir de métadonnées etc. Le choix des bonnes features adaptées aux données et à la tâche de détection est primordial.

Approches orientées Modèle : Les approches orientées modèles se concentrent sur la conception du meilleur modèle pour la détection des fake news. Ces modèles sont essentiellement des algorithmes avancés de machine learning, des architectures complexes de réseaux de neurones profonds.

Approches orientées Applications : Les approches orientées applications consistent à la gestion des fake news après leur détection. Ces approches s'attachent à minimiser l'impact des fake news sur nos sociétés. Cela peut consister en des méthodes de gestion des contenus fallacieux sur le web (suppression, réduction de la visibilité), mise en place de plateformes de fact-checking, mise en place de méthodes de surveillance des réseaux sociaux etc.

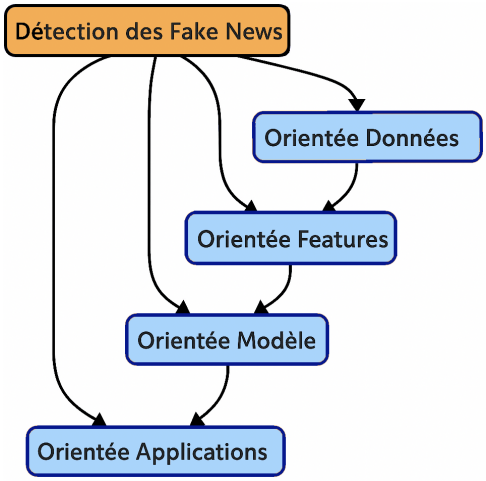


FIGURE 2.3 – Schéma représentant les différentes approches possibles dans la recherche pour identifier et combattre les fake news.

2.2.2 Signal d'entrée pour la détection : contenu, source ou propagation ?

Nous présentons ci-après les approches les plus populaires pour la détection des fake news.

Méthodes basées sur la propagation : L'analyse des schémas de diffusion des informations peut mettre en évidence des fakes news car elles se diffusent différemment des informations authentiques. Il est par exemple possible d'analyser la diffusion des fake news à travers les réseaux sociaux à l'aide de réseaux hiérarchiques de propagation [131]. Ces réseaux de propagation se divisent en deux types, le type macroscopique et le type

microscopique, offrant plusieurs niveaux de granularités sur la compréhension de la diffusion des informations. Pour des informations provenant du réseau Twitter (nouvellement 𝕏), au niveau macroscopique les nœuds du réseau hiérarchique représentent les tweets originaux et leurs retweets (partages des tweets originaux par d'autres utilisateurs) et les arêtes représentent les relations entre les tweets et les posts résultants d'un de leurs retweets. Le réseau complet représente par sa structure la diffusion des informations en cascade. Au niveau microscopique, les nœuds représentent les réponses et commentaires sous les tweet tandis que les arêtes représentent les relations de réponses entre les nœuds. Le réseau au niveau microscopique illustre les discussions et intéractions sociales autour d'un unique tweet original. A partir d'un tel réseau il est possible d'analyser de nombreux paramètres tels que la profondeur des discussions sur un tweet (indiquant l'intérêt des utilisateurs pour le tweet et sa potentielle viralité), le nombre de réponses maximales, la rapidité avec laquelle les tweets sont retweetés etc. Des caractéristiques stylistiques sur les discussions peuvent également être extraites, afin d'estimer le sentiment moyen des réponses par exemple. Les informations fausses correspondent à des discussions au style plus marqué dans lequel l'on peut retrouver de la colère par exemple.

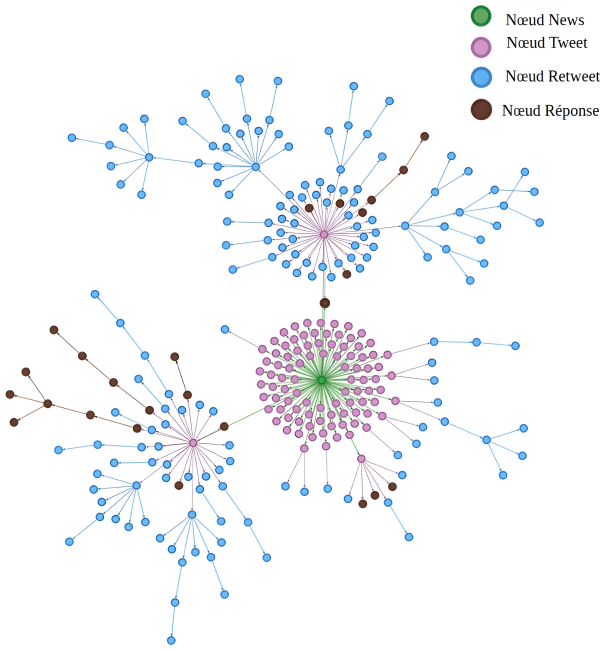


FIGURE 2.4 – SHU et al. [131] représentent les *news* sous forme de graphes, illustrant ici une information issue de *Politifact* (un site de *fact-checking* journalistique américain : `https://www.politifact.com`). Ce graphe de propagation montre la diffusion de l'information à travers les tweets qui y sont liés. Des features statistiques discriminantes sont ensuite extraites pour la détection des *fake news*, telles que la profondeur du graphe, le nombre de nœuds ou encore le temps de réponse entre deux nœuds (source graphe [131]).

Méthodes basées sur l'analyse de la source : Parfois l'analyse du contenu et de la propagation ne suffit pas déterminer si l'information est légitime, notamment parce que les fake news mélangent souvent des informations vraies et fausses [135]. Ainsi des méthodes de détection des fake news basées sur la source utilisent davantage l'analyse de la source des informations plutôt que leur contenu. Par exemple, des travaux de la littérature se concentrent sur les fake news liées à un article de presse en particulier plutôt que de simples déclarations difficilement traçables. En effet, travailler sur des articles de journaux permet d'utiliser des connaissances sur la fiabilité passée de ces journaux. Ainsi certains journaux comme *Le Monde* proposent des informations presque toujours légitimes [91] tandis que

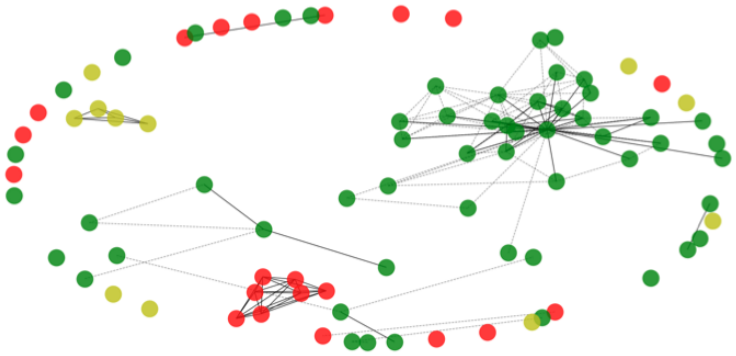


FIGURE 2.5 – SITAULA et al. [135] représentent dans un graphe les liens de co-écriture entre auteurs qui publient seulement des news réelles (en vert), ceux qui publient seulement des fakes (en rouge), et ceux qui publient partiellement des fake (en jaune). Il est remarquable qu'il n'existe pas de chemins du graphe reliant des nœuds d'une couleur à une autre (source graphe[135]).

d'autres tels que *FranceSoir* sont très coutummiers des fausses informations [37]. Ces approches analysent la réputation de la source, d'abord avec sa crédibilité établie par le passé, ensuite en étudiant l'activité de ces sources (par exemple un nombre élevé d'articles publiés en peu de temps peut être indicateur de désinformation). D'autres paramètres sont pris en compte, tels que le nombre d'auteurs d'un article (plus le nombre d'auteurs est élevé, plus la source est fiable), l'historique de publication des auteurs, l'analyse des co-auteurs. Des critères sur la crédibilité du contenu sont également utilisés tel que le nombre de références. Des modèles basés sur la source assez simples et composés de relativement peu de paramètres peuvent obtenir des bons résultats [135].

Méthodes basées sur le contenu : Les approches basées sur le contenu (content-based) se concentrent sur l'interprétation des features extraits des données propres à l'information. C'est à dire le texte qu'elle contient, les images, les vidéos, les fichiers audios qu'elle contient. Majoritairement les informations contiennent des données images et des données textuelles. Progressivement, avec les progrès rapide des deepfakes, les vidéos prendront probablement plus de place dans les fake news. Les vidéos peuvent être interprêtées comme la combinaison d'images et d'audios. Quant aux fichiers audios purs, leur présence est actuellement assez marginale dans les fake news partagées en ligne. Les approches content-based peuvent être séparées en plusieurs catégories. L'approche *style-based* se repose sur les aspects formels du texte de la news. C'est à dire les caractéristiques stylistiques, la détection de sentiments (les émotions présentes dans le texte tel que la colère ou un ton alarmiste par exemple), l'usage des majuscules, de la ponctuation etc. L'approche *knowledge-based* ou fact-checking se concentre sur la détection des incohérences de la news par rapport à la connaissance générale. En cela, selon cette approche, la détection des fake news s'inscrit dans la tâche générale en machine learning de détection d'anomalie. L'approche knowledge-based s'appuie ou bien sur les avis d'expert et donc fait l'objet d'un travail manuel, ou bien s'appuie sur les bases et graphes de connaissances pour automatiser le processus de détection. Cette automatisation de l'approche knowledge-based est dite *computational oriented* [22, 82]

2.3 Jeux de Données pour la Détection des Fake News

Dans cette section, nous présentons les datasets les plus utilisés dans la littérature, pour la détection des fake news, en insistant sur les modalités sur lesquels ils reposent. Les fake news sont difficiles à labelliser, et demandent parfois un véritable travail journalistique de fact-checking. C'est pourquoi les jeux de données de qualité sont relativement peu nombreux. De plus, la complexité des données à l'œuvre dans les fake news (parfois des articles de presse entiers qui mélangent le vrai et le faux ou encore des images deepfakes très vraisemblables), rend l'apprentissage de leurs représentations dépendant d'un grand volume de données. Une autre difficulté très actuelle dans la constitution de datasets avec un grand nombre de données, est l'inflation de la valeur de ces données d'une part, du fait de l'explosion de la demande (notamment liée aux grands modèles de langage et à la gourmandise des architectures de type Transformers) et, d'autre part, en raison des verrous politiques imposés par les grandes entreprises privées qui en détiennent l'accès. Par exemple, l'API de 𝕏 (anciennement Twitter) s'est fortement restreinte en matière de collecte de données, notamment depuis la prise de direction par Elon Musk [13].

2.3.1 Datasets Textuels

Dans un premier temps, nous présentons des jeux de données textuels unimodaux, qui constituent la majorité des corpus utilisés pour la détection des fake news. Cette prédominance s'explique par plusieurs facteurs. D'une part, la collecte de données textuelles est plus simple et moins coûteuse que celle de données multimodales telles que des images ou des vidéos légendées. D'autre part, la labellisation s'avère plus directe : le texte contient souvent des assertions informationnelles explicites, dont la véracité peut être évaluée, idéalement, par un processus de fact-checking manuel.

TABLE 2.1 – Jeux de données textuels pour la détection de fake news

Nom	Taille	Langue	Accès	Thème	Label	Supervision	Date
LIAR	12,8k ex.	en	✔	Politique (USA)	Pants-fire, False, Barely-true, Half-true, Mostly-true, True	Annotation manuelle	2017
Fighting an Infodemic	10,7k ex.	en	✔	Santé publique	Real, Fake	Annotation manuelle	2020
WELFake	72k articles	en	✔	Généraliste	True, Fake	Semi-automatique et manuelle	2021
FEVER	185k ex.	en	✔	Savoirs Encyclopédiques	Supported, Refuted, NotEnoughInfo	Annotation manuelle	2017

LIAR[1] [153] Le dataset LIAR comprend 12800 courtes assertions issues de Politifact [111], un site internet de fact-checking de propositions de politiques américains. Politifact a reçu en 2009 le prix Pulitzer pour la qualité de ses vérifications sur l'élection américaine de 2008. Les données de LIAR ont été labellisées manuellement par des humains, et classifiées selon 6 degrés de falsification, du plus faux au plus vrai : *"pants-fire", "false", "barely-true", "half-true", "mostly-true", "true"*. En outre, LIAR contient des méta-datas telles que l'auteur et l'historique de véracité de ses déclarations. Sa finesse de labellisation fait de LIAR un dataset de référence pour la prédiction multiclasse de fake news basée sur du texte.

Fighting an Infodemic[2][104] Fighting an Infodemic est un dataset unimodal textuel, regroupant des news provenant de sources diverses, des sites de fact checking comme PolitiFact ou Snopes, des posts de réseaux sociaux comme Instagram ou Facebook, des déclarations publiques ou encore des articles de presses. Ces news consistent en de courts textes, tous liés au sujet de la pandémie de Covid-19. La labellisation des données est manuelle et binaire, entre *"Real"* et *"Fake"*. Les deux classes sont équilibrées (52% de Real et 48% de Fake). De plus, il est à noter que les news "Real" sont généralement plus longues que les "Fake".

WELFake[3] WELFake est un dataset textuel composé de 72,000 news, issues de Kaggle, McIntire, Reuters et BuzzFeed. Chacune de ces sources contient des news classifiées de manière binaire *"Real"* et *"Fake"*. WELFake est un dataset composite robuste largement utilisé dans la littérature.

FEVER[4] FEVER est un dataset unimodal textuel de grande ampleur, il contient en effet 185,000 assertions extraites d'informations de Wikipedia à partir de 50,000 pages populaires antérieures à 2017. Chaque annotateur a généré des assertions liées à ces pages ainsi que des mutations de ces assertions basées sur des paraphrases, négations, substitutions d'entités, généralisation ou spécification du propos. Dans un second temps les assertions sont labellisées manuellement suivant trois classes : *Supported, Refuted, NotEnoughInfo* en fonction des preuves textuelles présentes sur la page Wikipedia. FEVER est un dataset de référence pour la vérification et l'extraction de faits, largement cité dans la littérature.

TABLE 2.2 – Exemples d'entrées pour chacun des jeux de données textuels pour la détection de fake news présentés.

Dataset	Extrait de contenu	Label
LIAR	When asked by a reporter whether he's at the center of a criminal scheme [...] Gov. Scott Walker nodded yes.	Pants-fire
Fighting an Infodemic	Chinese converting to Islam after Realising that not muslim was affected by #Coronavirus [...] in the Country.	Fake
FEVER	Claim : Lorelai Gilmore's father is named Robert. → Refuted with Wikipedia evidence.	Refuted
WeLFake	UNBELIEVABLE! OBAMA'S ATTORNEY GENERAL SAYS MOST CHARLOTTE RIOTERS [...] were exercising their constitutional [...] right.	Real

1. https://github.com/tfs4/liar_dataset
2. https://github.com/Omar-Sharif/CONSTRAINT-AAAI2021
3. https://zenodo.org/records/4561253[147]
4. https://fever.ai/dataset/fever.html[142]

2.3.2 Datasets Exploitant le Contexte Social

Plusieurs datasets de la littérature intègrent également le contexte social des fake news sur les réseaux sociaux. De fait la manière dont une information se propage, les caractéristiques des utilisateurs qui la diffusent, ou la viralité d'un contenu, sont autant de signaux pertinent pour évaluer sa véracité. Nous présentons ci-après quelques datasets de références, qui prennent en compte ces dimensions sociales et interactionnelles dans leur construction.

TABLE 2.3 – Jeux de données intégrant le contexte social pour la détection de fake news

Nom	Taille	Langu	Accès	Thème	Label	Nature des données	Date
FakeNews-Net	600k tweets	en	✗	Politique, Divertissement	Real, Fake	Tweets, likes, retweets	2018
PHEME	5,8k fils	en	✓	Rumeurs événementielles	True, False, Unverified	Tweets et réponses	2016
SemEval2017-Task8	4,519 tweets.	en	✓	Rumeurs sur Twitter	Support, Deny, Query, Comment, Real, Fake	Tweets et réponses	2017-2019
CoAID	4,2k news, 296k données contextuelles	en	✓	COVID-19, Santé	Real, Fake	Articles, tweets, likes, retweets	2020

FakeNewsNet [5] [130] FakeNewsNet est un dataset multimodal conçu pour la détection des fake news sur les réseaux sociaux. Toutes les données sont basées sur deux sources fact-checkées : PolitiFact (pour des news politiques) et GossipCop (pour des news de Divertissement). FakeNewsNet est composé de 187,000 tweets fake, et 415,000 tweets réels. Chaque entrée combine à la fois le contenu textuel des informations et le contexte social associé sur Twitter (nouvellement 𝕏), c'est à dire les *likes*, *tweets*, *retweets utilisateurs*. En outre des informations spatiales et temporelles (propagation, dates, localisation) sont également fournies. Ce dataset permet d'étudier conjointement le contenu des news, et leur diffusion, appuyés par une labellisation fiable issue de fact-checking experts. Cependant à notre connaissance, ce jeu de données n'est pour l'heure plus en libre accès. En effet, la récolte des données repose sur l'API de Twitter, dont les accès ont été fortement restreint ces dernières années [13].

PHEME [6][67] PHEME est un dataset de *rumeurs* et *non-rumeurs* sur Twitter qui traitent de 5 évènements majeurs de l'actualité entre 2014 et 2016 : Les attentats de Charlie Hebdo, l'affaire Michael Brown (Ferguson), le crash de l'avion de Germanwings, la fusillade d'Ottawa et la prise d'otage de Sydney. Les données sont hiérarchisées sous

5. https://github.com/KaiDMML/FakeNewsNet
6. https://figshare.com/articles/dataset/PHEME_dataset_of_rumours_and_non-rumours/4010619?file=6453753

forme d'arbres, où chaque rumeur est associée à un tweet et ses réponses associées. Le dataset est bien équilibré, hormi pour les évènements de Charlie Hebdo et Ferguson où les non-rumeurs représentent près de 75% des données. En dépit du nombre de données relativement restreint, PHEME reste un dataset utile pour la détection de rumeurs sur les réseaux sociaux, et reste accessible, malgré les restrictions sur l'API de Twitter.

Semeval2017-Task8 [7] [29] SemEval2017-Task8 est un corpus conçu pour la détection des rumeurs sur les réseaux sociaux et pour l'analyse des réactions des utilisateurs à celles-ci. Le dataset regroupe des tweets sources et des réponses associées, classifiés suivant la posture des utilisateurs : *Support*, *Deny*, *Query*, *Comment* (via crowdsourcing) ainsi que leur véracité *true*, *false* (via annotation experte).

CoAID [8] [25] CoAID est un dataset regroupant des fake news liées à la pandémie de Covid-19, issues de sites internets et de réseaux sociaux, ainsi que l'engagement social (tweets, retweets, réponses) des utilisateurs vis à vis de ces fausses informations. Le dataset regroupe plus de 290,000 engagements sociaux, liés à plus de 5200 news distinctes. Sa spécialisation dans le domaine de la santé en fait un dataset clef pour lutter contre la désinformation médicale.

2.3.3 Datasets Visuels

Les datasets sur les fake news exclusivement visuels sont rares, car la modalité visuelle est davantage employée pour appuyer une information textuelle que pour véhiculer à elle seule une fausse information. En revanche, de nombreux datasets existent pour la détection de transformation d'images ou de vidéos, tâches connexes à la détection des fake news. En effet, la falsification de données visuelles, enlever un élément, en rajouter un autre, modifier les visages présents, sont autant de moyen de falsifier l'information transmise par l'image. Nous présentons ici quelques datasets importants dans le cadre de la détection de transformation visuelle (image ou vidéo). Nous développerons plus en profondeur les problèmes de la détection et de la localisation des transformations d'images dans le Chapitre 4.

TABLE 2.4 – Jeux de données visuels pour la détection d'images ou vidéos falsifiées

Nom	Taille	Modalité	Accès	Falsification	Supervision	Date
FFIW10K	10k vidéos	Vidéo	✔	Face swap	Masques	2021
CASIA v2.0	12,3k images	Image	✔	Copy-move, Splicing	Masques binaires	2013
FaceForensics++	1k vidéos	Vidéo	✔	Deepfakes (Face2Face, FaceSwap, etc.)	Vidéos originales + manipulées, masques	2019
DSO-1	200 images	Image	✔	Splicing réaliste	Masques binaire	2016

Casia v2.0 [9] [108, 31] Casia v2.0 est un dataset sur la transformation locale d'images. Deux transformations sont utilisées, le *copy-move* qui consiste à copier et coller une zone

7. https://alt.qcri.org/semeval2017/task8/
8. https://github.com/cuilimeng/CoAID
9. https://www.kaggle.com/datasets/divg07/casia-20-image-tampering-detection-dataset

de l'image sur une autre zone, et le *splicing* qui consiste à copier et coller une zone d'une image sur une autre image. Les images du dataset sont de tailles (de 320×240 à 800×600 pixels) et formats variables (BMP, TIFF, JPEG). Le dataset est composé d'environ 7200 images authentiques et 5100 transformées dont 3300 ayant subi du copy-move et 1800 du splicing. Les images sont également classées en 9 catégories : *scene, animal, architecture, character, plant, article, nature, indoor, texture*. Les transformations se veulent être le plus réaliste possible, ainsi des observateurs humains ne sont parvenus qu'à une accuracy de 58% en évaluant visuellement le dataset, soit légèrement mieux qu'un classifieur aléatoire, indiquant la bonne capacité des images à tromper même des humains. Casia est un dataset utile pour évaluer la qualité de détecteur de transformation d'images simples, et largement utilisé dans la littérature à cet effet.

FaceForensics++ [10] [120] FaceForensics++ est un dataset de vidéos centrées sur la manipulation de visages. Il contient environ 1000 vidéos falsifiées, accompagnées de leurs versions originales, issues de YouTube, soit près de 2 millions d'images. Quatre méthodes de manipulation différentes sont représentées, couvrant à la fois des approches classiques basées sur des techniques de graphismes (*Face2Face*, *FaceSwap*) et des approches par apprentissage profond (*Deepfakes*, *Neural Textures*). Il s'agit d'un dataset de référence pour la détection de falsfication de visages sur des vidéos.

DSO-1 [11] [15] DSO-1 est un dataset d'images en très haute résolution, composé de 100 images authentiques et 100 images falsifiées, réalisées en intérieur et en extérieur par splicing d'individus. Ce corpus se distingue par le réalisme des manipulations et la qualité visuelle des images, ce qui en fait un dataset pertinent pour évaluer la robustesse des méthodes de détection de falsifications visuelles.

FFIW10K [12] [174] FFIW10K est un dataset de vidéos focalisé sur la manipulation de visages. Il contient 10 000 vidéos falsifiées en haute résolution (supérieure à 480p), avec en moyenne trois visages humains par image. Le corpus est construit à partir de 4 000 vidéos issues de YouTube, segmentées en 12 000 clips de 12 secondes. Les visages de meilleure qualité sont sélectionnés comme source ou cible pour générer les vidéos falsifiées à l'aide de DeepFaceLab, FS-GAN ou FaceSwap. Un module d'évaluation automatique de la qualité des visages permet ensuite de filtrer les vidéos présentant des résultats de mauvaise qualité FFIW10K constitue l'un des plus grands datasets disponibles pour la détection de manipulations faciales en vidéo.

2.3.4 Datasets Multimodaux : Texte et Image

Les jeux de données multimodaux, combinant notamment textes et images, sont particulièrement pertinents pour la détection des fake news, car ils reflètent mieux les cas d'usage réels, notamment sur les réseaux sociaux, où les contenus trompeurs prennent le plus souvent la forme de textes accompagnés d'images. Toutefois, la création de tels datasets est naturellement plus complexe que celle de jeux de données unimodaux, en raison des défis liés à l'annotation croisée et à la collecte de contenus cohérents sur plusieurs modalités. Nous présentons ci-dessous une sélection de jeux de données multimodaux particulièrement pertinents pour la détection des fake news.

10. `https://github.com/ondyari/FaceForensics/blob/master/dataset/README.md`
11. `https://recodbr.wordpress.com/code-n-data/`
12. `https://github.com/tfzhou/FFIW`

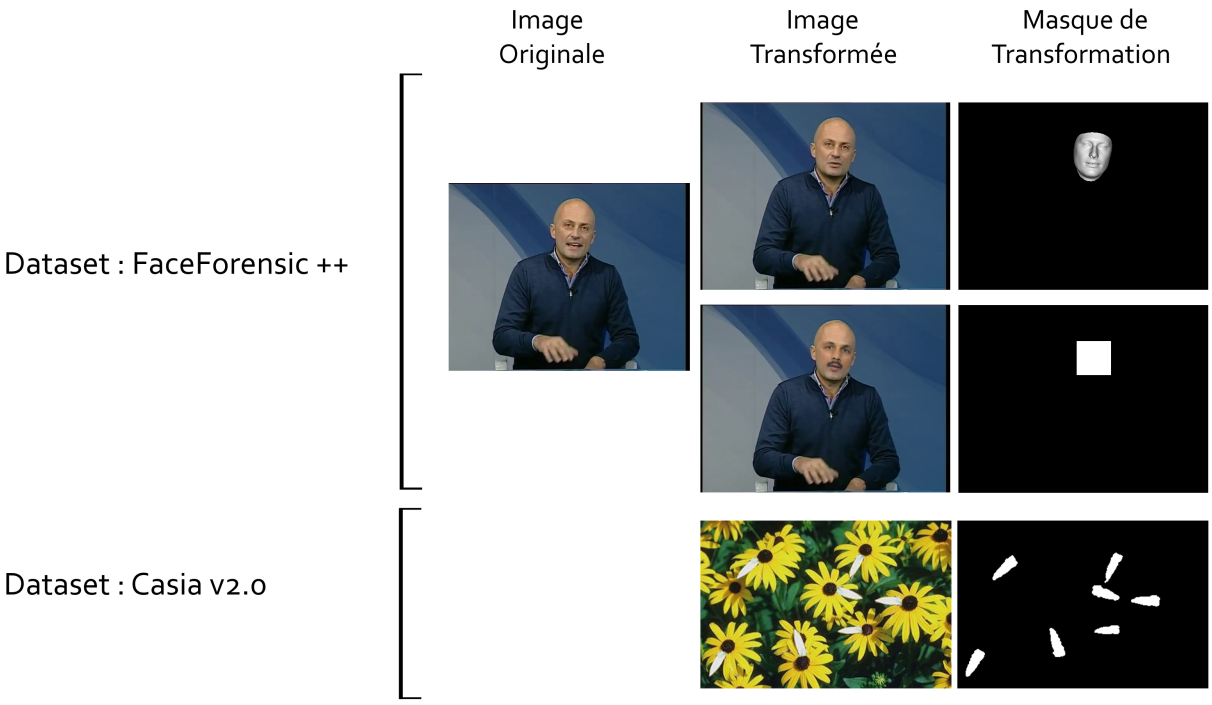


FIGURE 2.6 – Exemples d'images issues des datasets FaceForensics++ [120] et Casia v2.0 [31], ainsi que les masques associés aux transformations. Pour le dataset FaceForensics++, nous présentons une image transformée par *Deepfake* (en bas), et une autre par *NeuralTexture*. Quant à elle, l'image de Casia a subi du *splicing*.

TABLE 2.5 – Jeux de données multimodaux pour la détection de fake news

Nom	Taille	Langue	Accès	Modalités	Supervision	Date
Weibo	∼9,5K posts	zh	✔	Texte, Image, Social	Vrai / Faux (annotation manuelle)	2016
Fakeddit	∼1M posts	en	✔	Texte, Image, Social	2, 3 et 6 classes automatiques	2019
TI-CNN	∼20K articles	en	✔	Texte, Image	Vrai / Faux (provenance)	2018

Weibo [13][60] Weibo est un jeu de données construit à partir du réseau social chinois éponyme, comparable à Twitter dans son fonctionnement. Il comprend plus de 9 500 publications (entre 2012 et 2016), chacune accompagnée de contenu textuel, d'images (jusqu'à deux), et de 16 attributs contextuels liés à l'utilisateur et à l'engagement (nombre de retweets, commentaires, likes, type d'authentification, nombre d'abonnés, etc.). Les rumeurs sont identifiées à l'aide du système de debunking propre à Weibo, fondé sur du crowdsourcing par des utilisateurs qualifiés, tandis que les non-rumeurs proviennent de l'agence officielle *Xinhua News Agency*. Un filtrage est appliqué pour écarter les publications trop bruitées ou redondantes. Ce dataset constitue une ressource précieuse pour l'évaluation de méthodes de détection de fake news multimodales, combinant texte, image et contexte social.

Fakeddit [14] [97] Fakeddit est un jeu de données à très grande échelle construit à partir de 22 subreddits du site Reddit, couvrant plus de 1 million de publications issues de 300 000 utilisateurs distincts, collectées entre 2008 et 2019. Chaque publication est accompagnée de diverses métadonnées, telles que le score, le nom d'utilisateur, le subreddit d'origine, le domaine source, le nombre de commentaires, ainsi que le ratio upvote/downvote. Les labels de véracité sont définis à trois niveaux de granularité : (1) binaire (vrai/faux), (2) ternaire (vrai / faux avec texte véridique / faux avec texte mensonger), et (3) six catégories plus fines (*true*, *satire/parody*, *misleading content*, *imposter content*, *false connection*, *manipulated content*). L'annotation est réalisée automatiquement en fonction du subreddit d'origine, chaque subreddit ayant été manuellement associé à un label global. Fakeddit constitue une ressource précieuse pour l'étude de la désinformation à grande échelle, en particulier dans un cadre multimodal combinant texte, métadonnées et images.

TI-CNN TI-CNN [15][168] est un jeu de données composé de 20 000 articles d'actualité, dont 12 000 étiquetés comme faux et 8 000 comme authentiques. Les fake news proviennent de 240 sites identifiés comme sources peu fiables [16], tandis que les articles authentiques sont issus de médias traditionnels reconnus tels que le *New York Times* ou le *Washington Post*. Chaque article est accompagné d'un ensemble de métadonnées comprenant le titre, le texte, une image, l'auteur et le site d'origine. Ce dataset est conçu pour des approches multimodales mêlant traitement du texte, analyse d'image et métadonnées.

2.4 Exploitation du contenu visuel pour la détection des fake news

Les informations diffusées sur les réseaux sociaux sont composées de manière croissante d'éléments multimédias, notamment d'images accompagnant le texte. Dans cette section, nous nous concentrons sur les méthodes exploitant spécifiquement le contenu visuel dans le cadre de la détection de fake news.

L'ajout d'images dans les publications augmente considérablement leur viralité. Par exemple, sur X (anciennement Twitter), un post contenant une image génère en moyenne 18 % de clics, 89 % de likes et 150 % de retweets supplémentaires par rapport à un post textuel seul [14]. Les images jouent ainsi un rôle essentiel dans la diffusion de l'information, qu'elle soit fiable ou manipulée. D'un point de vue sémantique, les contenus visuels peuvent

13. https://github.com/wangzhuang1911/Weibo-dataset
14. https://github.com/entitize/Fakeddit
15. https://github.com/coderjedi/Data-Mining-Assignment-TICNN?tab=readme-ov-file
16. https://www.kaggle.com/datasets/mrisdal/fake-news

appuyer et illustrer, ou au contraire contredire le message textuel qu'ils accompagnent. Ces incohérences multimodales sont souvent des indicateurs révélateurs de désinformation. L'analyse des images est donc cruciale, d'autant plus que la manipulation de contenu visuel peut intervenir à plusieurs niveaux :

(i) **Manipulation du contenu visuel** : l'image est falsifiée (via splicing, retouche, ou génération artificielle).

(ii) **Image authentique mais détournée** : l'image provient d'un événement réel sans lien avec le contenu textuel (ex. : image d'un ancien conflit utilisée pour illustrer une crise actuelle).

(iii) **Image pertinente mais interprétée de manière trompeuse** : le texte associe à une image réelle une lecture erronée ou biaisée du contexte qu'elle représente.

Dans ce contexte, de nombreuses approches ont été proposées pour analyser les images et en extraire des indicateurs de manipulation ou d'incohérence sémantique. Parmi ces méthodes, les approches fondées sur l'apprentissage profond se sont révélées particulièrement performantes. Elles permettent d'extraire automatiquement des représentations visuelles riches, à différents niveaux de complexité (objets, scènes, etc.), et surpassent largement les méthodes basées uniquemement sur des features manuels ou statistiques dans la plupart des benchmarks récents.

Nous présentons ci-dessous différents types de *features* extraits du contenu visuel, dans le cadre de la détection des fake news.

2.4.1 Les features forensiques

Les features forensiques servent à vérifier l'authenticité d'une image et à capturer les différentes transformations qu'elle aurait pu subir au cours de son historique sur internet. Ces features cherchent à détecter des traces numériques bas niveau (appelées *traces forensiques*) dans l'encodage de l'image.

Les techniques d'extraction de features forensiques cherchent à identifier :

(i) Des transformations spécifiques, telles que le *splicing* (copier-coller d'une région d'une image sur une autre),

(ii) Des traces de recompression, de filtres, ou de résolutions incohérentes,

(iii) Des signes de génération artificielle, notamment via des modèles de type GAN ou diffusion (ex. : Midjourney, DALL· E).

L'objectif est de mettre en évidence une rupture dans l'intégrité de l'image, souvent symptomatique d'une falsification volontaire.

2.4.2 Les features sémantiques

En complément de l'analyse forensique, l'étude du contenu sémantique d'une image permet d'évaluer son impact informationnel et émotionnel. Les images associées à des fake news sont fréquemment choisies pour leur pouvoir de captation émotionnelle : elles suscitent des réactions de colère, d'indignation ou de surprise, renforçant la viralité du message [129, 14].

Les caractéristiques sémantiques (ou *semantic features*) correspondent à des représentations de haut niveau, extraites par des modèles d'analyse d'image. Les réseaux de neurones convolutifs (CNN [70]) ont longtemps constitué l'outil principal pour cette tâche, permettant de détecter objets, scènes et expressions faciales. Plus récemment, les architectures basées sur les transformers (comme ViT [33] ou CLIP [114]) ont démontré une capacité supérieure à capter les relations complexes entre éléments visuels et à produire

des embeddings riches et contextualisés. Contrairement aux approches forensiques, qui s'intéressent à la structure numérique de l'image, les approches sémantiques cherchent à en comprendre le contenu, sa charge symbolique et sa cohérence avec le texte associé.

2.4.3 Les features statistiques

Au-delà des propriétés sémantiques et forensiques, les images liées à un événement peuvent être étudiées sous un angle statistique, en considérant leur distribution au sein d'un corpus. Dans le cas d'un événement réel, on observe en général une diversité d'images provenant de sources multiples, prises sous différents angles et dans des contextes variés. À l'inverse, dans le cas d'une fake news, les contenus visuels sont souvent plus homogènes, parfois répétés ou recyclés. Cette distinction a motivé l'apparition de plusieurs indicateurs statistiques fondés sur la redondance, la cohérence et la diversité visuelle [14, 61] Les scores de *cohérence visuelle* et de *clarté visuelle* permettent de rendre compte de ces différences de distributions. Les travaux sur ces features [14] s'appuient majoritairement sur trois aspects : *Le nombre d'occurrence* des images, leur *popularité* (mesurée par les retweets, commentaires etc) et leur caractéristiques techniques (résolution, largeur, longueur). De plus Jin et al. [61] proposent d'autres features statistiques plus avancés :

Visual Clarity Score (VCS) :

$$\mathrm{VCS} = D_{KL}(p(w|c) \parallel p(w|k)) \tag{2.1}$$

Le VCS mesure la différence de distribution entre deux ensembles d'images : un ensemble d'images d'une news spécifique (event set) et un ensemble d'images couvrant toutes les news (collection set). Cette différence est évaluée en utilisant la divergence de Kullback-Leibler entre des modèles de langages qui représentent le event set et le collection set. $p(w|c)$ correspond à la fréquence d'un mot visuel w dans le collection set, $p(w|k)$ la fréquence dans le event set. Pour définir le modèle de langage approprié il est possible d'utiliser la représentation bag-of-words des images.
Ainsi le VCS permet de mesurer à quel point les images associées à un évènement spécifique se distinguent du reste des images. Si un évènement news contient beaucoup d'images dont la distribution des mots visuels est très différentes de celle des images associées à des évènements réels cela peut indiquer que l'évènement est manipulé. Il est important de sélectionner un bon collection set qui mettrait en évidence une falsification dans l'event set. Par exemple, pour une news sur une attaque militaire, il pourrait être judicieux de constituer un collection set qui comprend des images d'attaque militaires avérées.

Visual Coherence Score (VCoS) :

$$\mathrm{VCoS} = \frac{1}{N(N-1)} \sum_{i,j=1,\ldots,N; i \neq j} \mathrm{sim}(x_i, x_j) \tag{2.2}$$

Le VCoS mesure la cohérence des images dans un événement en particulier en calculant la similarité moyenne entre toutes les paires d'images (x_i, x_j) dans cet ensemble. La similarité peut être ajustée selon les besoins. Nous pouvons utiliser une mesure sémantique, pour mesurer l'homogénéité informationnelle et visuelle des images, ou d'autres similarité telles que la similarité forensique [90] pour mesurer l'homogénéité dans la falsification ou l'authenticité des images.

Visual Similarity Distribution Histogram (VSDH) : Pour une mesure de similarité donnée et un ensemble d'images liées à une news, on construit une matrice de similarité $(m_{i,j})_{i,j}$ où chaque $m_{i,j}$ correspond à la similarité entre les images x_i et x_j. Cette matrice est convertie en un histogramme de distribution en assignant chaque valeur de la matrice à un intervalle spécifique de l'histogramme (bin). Chaque bin représente une plage de valeurs de similarités. Pour un nombre de bin H Le VSDH est définit comme le vecteur de dimension H dont le $i$ème terme contient la fréquence de similarités appartenant au $i$ème bin :

$$\text{VSDH}(h) = \frac{1}{N^2} \left|\{(i,j) \mid i,j \leq N, m_{i,j} \in h - \text{ème bin}\}\right|, \quad h = 1, \ldots, H \tag{2.3}$$

Le VSDH fournit une description de la distribution de la similarité des images. Dans des évènements réels, les images sont assez variées, prises de différents points de vue, cela entraîne une distribution plus large des similarités. Un VDSH qui montre une distribution anormalement concentrée pourrait indiquer qu'un ensemble d'images est artificiellement homogène. Le même effet de concentration serait visible si de nombreuses images similaires sont utilisées pour décrire un évènement, ce qui est courant dans le cas des fake news.

Visual Diversity Score (VDS) :

Le VDS mesure la diversité visuelle dans un ensemble d'images pour un événement donné. Il est calculé en fonction des différences de similarité entre les images, classées par popularité croissantes. Le VDS correspond formellement à la somme des dissimilarités moyennes entre une image et les images plus populaires qu'elle :

$$\text{VDS} = \sum_{i=1}^{N} \frac{1}{i} \sum_{j=1}^{i} (1 - \text{sim}(x_i, x_j)) \tag{2.4}$$

Le VDS prend plus en compte les images les plus populaires, cela réduit l'impact d'images constituant du bruit dans les données. Les vraies news ont tendance à avoir une diversité visuelle plus haute que les fausses [61]. Les fakes news utilisent souvent des images similaires ou recyclées.

Visual Clustering Score (VCS) Le score de clustering visuel évalue la distribution des images en calculant le nombre de clusters formés par toutes les images d'un événement spécifique, en utilisant l'algorithme de clustering hiérarchique agglomératif (HAC). Le HAC fonctionne de la façon suivante : initialement toutes les images forment un cluster (le clustering est réalisé sur les embeddings visuels des images). A chaque itération les clusters les plus proches (selon une distance à définir) sont fusionnés. Jusqu'à un critère d'arrêt (par exemple que tous les clusters sont suffisament éloignés les uns des autres). Des distances comme la distance minimale entre deux éléments de deux clusters, la distance entre les centres de gravité des clusters peuvent être choisies, chacune présentants des avantages et inconvénients. Un VCS élevé signifie que les images sont diverses visuellement et qu'elles se répartissent bien dans différents clusters, cela est typique des évènements authentiques où les images proviennent de sources variées.

2.4.4 Les features contextuels

Les features contextuels exploitent le contexte entourant les images pour identifier les contenus visuels trompeurs. Les fake news recourent fréquemment à des images sorties de leur contexte, utilisées pour illustrer des événements avec lesquels elles n'ont aucun lien

réel. Pour mettre en évidence ces incohérences, les features contextuels peuvent s'appuyer sur les métadonnées ou des sources externes.

Métadonnées Les métadonnées correspondent à des informations techniques associées à l'image, mais indépendantes de son contenu visuel. Il peut s'agir de la date de capture, du type d'appareil utilisé, de la taille du fichier, ou encore de paramètres d'encodage. Ces données peuvent être exploitées dans le cadre d'un processus de fact-checking : par exemple, une discordance entre la date de capture d'une image et la date de l'événement revendiqué peut constituer un indice de manipulation. Cependant, les métadonnées sont souvent altérées ou supprimées lors de la diffusion des images sur les réseaux sociaux. En effet, les plateformes compressent ou reformatent les fichiers, et les utilisateurs peuvent modifier les images (recadrage, filtres, etc.). En pratique, les métadonnées fiables sont donc rarement disponibles en conditions réelles.

Sources externes Une autre approche consiste à rechercher le contexte d'origine de l'image à l'aide de techniques telles que la recherche d'images inversée. Cette démarche permet de tracer l'historique de l'image sur le web, d'identifier ses premières occurrences, et d'en extraire des informations factuelles pertinentes. Plusieurs indicateurs contextuels peuvent alors être définis :

(i) *Timespan* : Le timespan correspond au délai entre la date de publication de la news et la première date de publication connue de l'image associée. Si une image a été publiée longtemps avant l'évènement revendiqué, cela augmente le doute sur l'authenticité de l'information.

(ii) *Inter-claim Similarity* : Cette feature mesure la similarité entre la revendication textuelle accompagnant une image et les autres contenus textuels sur le web dans lesquels cette image apparaît. Une forte dissimilarité augmentant la probabilité de manipulation.

(iii) *Platform Credibility* : Il s'agit de la crédibilité de la plateforme où le contenu visuel a été publié. Des bases de données telles que Media Bias/Fact Check (MBFC), chaque site web où l'image est trouvée via recherche d'images inversée est classé en terme de factualité. Un grand pourcentage de pages provenant de sites peu fiables est un indicateur de fake news potentielle.

2.4.5 Exemple de modèle : MVNN (Qi et al., 2019)

Le modèle MVNN (Multi-domain Visual Neural Network) proposé par Qi et al. [112] constitue une référence majeure dans l'exploitation du contenu visuel seul pour la détection d'images intégrées à des fake news. Il repose sur l'hypothèse que les images associées aux fake news présentent des caractéristiques spécifiques, à la fois au niveau *physique* (compression, artefacts) et au niveau *sémantique* (contenu visuel émotionnel, sensations fortes). MVNN combine ces deux types d'informations par une architecture à trois composants : un sous-réseau dans un domaine de fréquences, un sous-réseau dans le domaines des pixels, et un module de fusion reposant sur un mécanisme d'attention.

Sous-réseau du domaine de fréquences Ce module extrait des features physiques des images en les transformant du domaine des pixels au domaine des fréquences via une transformée en cosinus discrète (DCT), il s'agit d'une technique standard en analyse forensique. En effet, les artefacts physiques laissés par les transformations d'images laissent des motifs périodiques observables dans le domaine des fréquences. Concrètement, pour une image d'entrée, des histogrammes des coefficients de sa DCT sont affinés par une

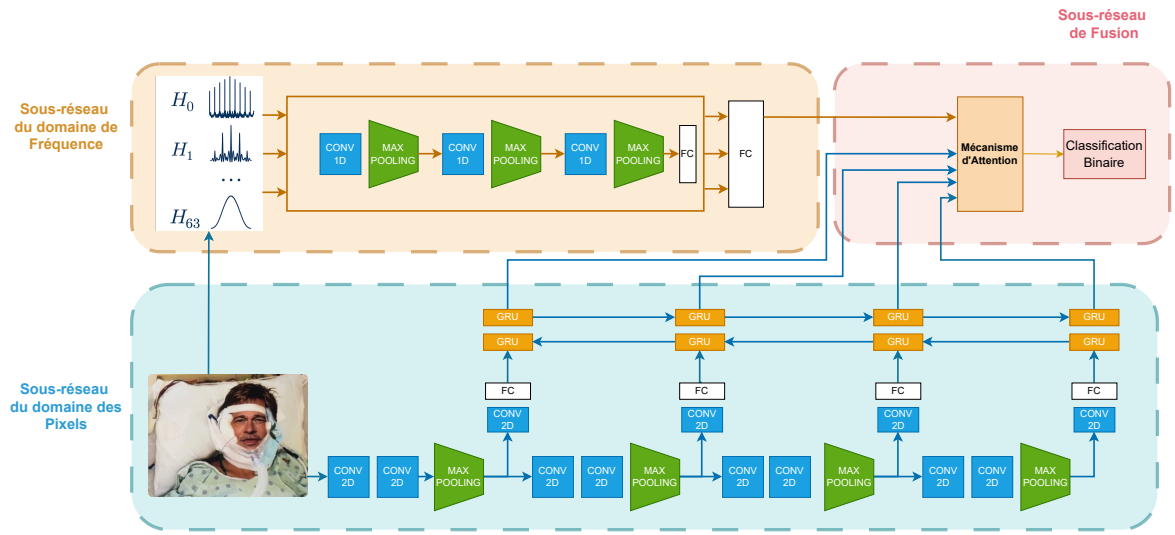


FIGURE 2.7 – Architecture du modèle MVNN proposée par QI et al. [112]. Le modèle repose sur deux sous-réseaux traitant respectivement l'image dans le domaine pixel (gauche) et dans le domaine fréquentiel (droite). Les vecteurs de représentation extraits sont ensuite fusionnés via un module d'attention pour effectuer la classification fake ou real (source : reproduit d'après [112]).

transformée de Fourier, samplés, et traités par un CNN 1D capable de saisir les régularité de motifs dans le domaine fréquentiel. En résulte un vecteur l_0 qui encode ces motifs.

Sous-réseau du domaine des pixels Ce module capture des features sémantiques de haut niveau à partir des pixels bruts. Il repose sur plusieurs couches CNN-based suivies de GRU bidirectionnels. La partie CNN-based contient 4 couches extrayant chacune des features à différents niveaux d'abstraction (des bas vers les hauts niveaux), chacun produisant un vecteur v_t pour $t = 1, \ldots, 4$. Ces vecteurs sont ensuite concaténés via un GRU bidirectionnel, produisant l_1 à l_4. qui modélisent la dépendance entre les différents niveaux d'abstraction. En effet, des features moyen-niveau comme les textures, dépendent à la fois de features bas-niveau comme des lignes, et sont constituant de features haut-niveau comme des objets.

Fusion par attention Enfin, le modèle MVNN fusionne les vecteurs l_0, l_1, l_2, l_3, l_4 à l'aide d'un mécanisme d'attention dynamique qui pondère chaque représentation selon sa pertinence pour la tâche :

$$F(l_i) = v^\top \tanh(W_f l_i + b_f), \quad i \in \{0, 1, 2, 3, 4\} \tag{2.5}$$

$$\alpha_i = \frac{\exp(F(l_i))}{\sum_j \exp(F(l_j))} \tag{2.6}$$

$$u = \sum_i \alpha_i l_i \tag{2.7}$$

Le vecteur u est ensuite projeté dans l'espace des classes (fake/real) par une couche softmax. L'apprentissage est effectué via une fonction de perte d'entropie croisée classique :

$$\mathcal{L} = -\left[y \log p + (1 - y) \log(1 - p)\right] \tag{2.8}$$

2.4.6 Benchmark des approches de détection visuelles

Plusieurs travaux ont évalué des approches reposant uniquement sur le contenu visuel pour la détection de fake news. Le tableau 2.6 présente une sélection de résultats issus de la littérature. On observe que les méthodes basées sur des réseaux convolutifs atteignent des performances solides, avec notamment MVNN [112], ou encore HAMN [36] et Sabir et al. [122] qui se concentrent sur la détection de deepfakes via des architectures CNN ou CNN+RNN. Ces résultats montrent que le contenu visuel seul peut fournir des signaux discriminants, à bas ou haut niveau, bien que l'intégration d'autres modalités reste souvent nécessaire pour une robustesse accrue. Nous approfondirons dans le Chapitre 6 le problème de détection des transformations d'images.

Table 2.6 – Comparaison des approches de détection de fake news basées sur le contenu visuel

Article	**Dataset**	**Accuracy**	**F1**	**Année**
MVNN [112]	Weibo	0.846	0.832	2019
Masciari et al. [86](CNN-based)	PHEME	0.758	0.755	2020
	PolitiFact	0.765	0.692	2020
HAMN [36](CNN-based)	FaceForensic++ (Deepfake)	0.841	-	2020
HAMN [36](CNN-based)	FaceForensic++ (FaceSwap)	0.865	-	2020
Sabir et al. [122](CNN+RNN-based)	FaceForensic++ (Deepfake)	0.860	-	2019
Sabir [122](CNN+RNN-based)	FaceForensic++ (FaceSwap)	0.863	-	2019

2.5 Exploitation du contenu textuel

Le texte constitue la modalité principale dans la plupart des contenus de type fake news, que ce soit dans les articles, les tweets, ou les légendes accompagnant des images. Il porte l'essentiel de l'information et de l'argumentation, et peut contenir des indices stylistiques, cognitifs ou contextuels permettant de discriminer le vrai du faux.

Plusieurs approches ont été proposées pour exploiter le contenu textuel dans le cadre de la détection de fake news. Elles reposent soit sur l'extraction de *features linguistiques explicites* (style d'écriture, complexité, lexique), soit sur des représentations apprises automatiquement par des modèles d'apprentissage profond.

2.5.1 Features linguistiques explicites

Les premières approches de détection de fake news à partir du texte reposent sur l'extraction explicite de *features linguistiques*. Ces features sont des indicateurs mesurables du style d'écriture, de la complexité du texte ou encore de certains biais cognitifs dans la formulation.

Analyse stylistique et reconnaissance d'auteur Brennan et Greenstadt [11] proposent un ensemble de features textuels pour la reconnaissance d'auteurs. Identifier l'auteur à l'origine d'un contenu permet d'évaluer sa fiabilité. En effet, beaucoup d'informations trompeuses sont écrites sous pseudonyme ou par des bots. Cette identification permet en outre de lutter contre des campagnes de désinformation coordonnées adoptant des styles ou des schémas rédactionnels répétitifs. Ces features incluent le nombre de mots différents, la complexité du langage, le Gunning-Fox Index, le nombre de caractères en comptant ou non les espaces, la longueur moyenne des phrases et le Flesch-Kincaid Readability Score. Le Gunning-Fog index est une formule qui estime le nombre d'années d'éducation nécessaires à la compréhension du texte à la première lecture. Le Flesch-Kincaid Readability Score permet d'estimer la difficulté de lecture d'un texte. Un score de 100 correspond à un texte très facile à comprendre, à la portée d'un enfant de 11 ans, tandis qu'un score de 0 est indicateur d'une lecture très difficile, comprise uniquement par les experts du domaine. Respectivement les Gunning-Fog Index (GI) et Flesch-Kincaid Readability Score (FS) sont calculés ainsi :

$$GI = 0.4 \left[\frac{\#\text{words}}{\#\text{sentences}} + 100 \frac{\#\text{difficult words}}{\#\text{words}} \right] \tag{2.9}$$

$$FS = 206.835 - 1.015 \frac{\#\text{words}}{\#\text{sentences}} - 84.6 \frac{\#\text{syllables}}{\#\text{words}} \tag{2.10}$$

Indicateurs lexicaux Reddy et al. [117] présentent également une liste de features utilisés pour la détection du mensonge : Le nombre de syllabes, de mots, de phrases, de phrases courtes, de phrases longues, de syllabes par mot, de longs mots, de mots par phrase, de conjonction, de mots exprimant la certitude, d'auto références, de termes affectifs, de pronoms ainsi que le Flesh-Kincaid score et le taux d'adverbes et adjectifs. Ces features peuvent être calculés explicitement et incorporés après standardisation dans des modèles de classification classiques tels que les SVM, arbres de décision ou régression logistique. Ces features sont facilement interprétables, et ces modèles de classification sont assez efficaces et rapides à entraîner.

Limites et transition vers le deep learning. Malgré leur efficacité initiale, les features explicites atteignent vite leurs limites. Les approches modernes reposent désormais largement sur des représentations vectorielles apprises automatiquement [2]. Les modèles basés sur des réseaux de neurones récurrents (RNN, LSTM, GRU) permettent de capter des relations séquentielles et contextuelles plus fines que les simples motifs lexicaux. Ces RNN sont toutefois souvent surpassés par des modèles convolutifs (CNN) [42], et plus récemment, par des modèles à base de transformeurs comme BERT.

2.5.2 Réseaux de neurones profonds pour l'analyse textuelle

Les approches fondées sur des réseaux neuronaux ont progressivement remplacé les méthodes à base de features explicites, grâce à leur capacité à apprendre automatiquement des représentations sémantiques profondes du texte.

Modèles séquentiels classiques. Des architectures comme les RNN, GRU et LSTM ont été utilisées sur des datasets comme LIAR pour modéliser la séquence des mots. Cependant, leurs performances restent limitées (F1 autour de 21-22%) sur LIAR [42].

Réseaux convolutifs. Des CNN combinés à des embeddings pré-entraînés (Word2Vec [94], GloVe [105]) permettent d'extraire des motifs locaux dans le texte. Par exemple Goldani, Momtazi et Safabakhsh [43] proposent un réseau à capsules [123] (les réseaux à capsules sont des réseaux dérivés des CNN) atteignant 39.5% de précision en classification multi-classes sur LIAR.

Modèles transformer-based Les modèles de type transformer, comme BERT [30], RoBERTa [83], ou XLNet [169], ont montré des performances nettement supérieures sur la détection de fake news. Ils exploitent des représentations contextuelles apprises sur de très grands corpus, avec deux stratégies : Selon l'approche *Feature-based* les embeddings du modèle de langage sont directement utilisés comme entrées pour un autre modèle. Selon l'approche *Fine-tuning* : Les couches finales seules du modèle de langage sont entraînées spécifiquement sur la tâche cible.

Plusieurs études récentes ont montré l'efficacité de ces approches [2]. Par exemple, Jwa et al. [62] utilisent BERT pour comparer l'en-tête d'une news et son corps et obtiennent un score F1 de 0.746 sur le dataset FNC-1 [35]. Quant à eux Gautam, V et Masud [40] utilisent le transformer XLNet [169] pour obtenir une représentation contextuelle des mots et un modèle génératif probabiliste LDA (Latent Dirichlet Allocation) qui encode les différents sujets des news. Ils obtiennent un haut score F1 de 0.96 sur le dataset *Fighting an Infodemic* [104].

Limites et enjeux. Malgré leurs excellents résultats, les *Pre-trained Language Models* (PLM) comme BERT ou XLNET présentent un coût computationnel important (BERT base = 110M paramètres). bien que des variantes comme DistilBERT permettent de réduire ce coût tout en conservant une performance élevée. Cependant, en raison de cette complexité, seul le fine-tuning des modèles est généralement envisageable, plutôt qu'un entraînement complet depuis zéro d'un modèle.

Enfin, plusieurs travaux soulignent que les features textuelles seules peuvent être contournés malicieusement par des contenus habilement rédigés. Cela justifie l'intérêt croissant pour les approches multimodales ou enrichies par des connaissances externes, que nous présenteront dans les sections qui suivent.

2.5.3 Présentation d'un modèle (Wani et al. 2021)

Wani et al. [157] proposent une évaluation systématique de différents modèles neuronaux pour la détection de fake news, en se concentrant exclusivement sur le contenu textuel. L'étude s'appuie sur le dataset *Fighting an Infodemic* [104]. L'approche la plus performante repose sur une version pré-entraînée de `BERT-base cased`, réentraînée sur un large corpus non annoté de tweets COVID, puis fine-tunée pour la tâche binaire de classification (*real* vs *fake*).

Pré-entraînement sur le domaine (LM pretraining) : `BERT-base cased` est réentraîné comme modèle de langage sur un corpus brut de tweets contenant le hashtag `#covid19`, collectés via l'API Twitter. Cette étape vise à adapter les poids du modèle à la distribution lexicale et stylistique propre à la pandémie.

Fine-tuning supervisé : Une couche de classification softmax est ajoutée au-dessus du token spécial `[CLS]` (le token qui encode la représentation globale de chaque phrase, à des fins de classification). L'entraînement est réalisé sur le dataset "Fighting an Infodemic", avec arrêt anticipé basé sur la performance en validation.

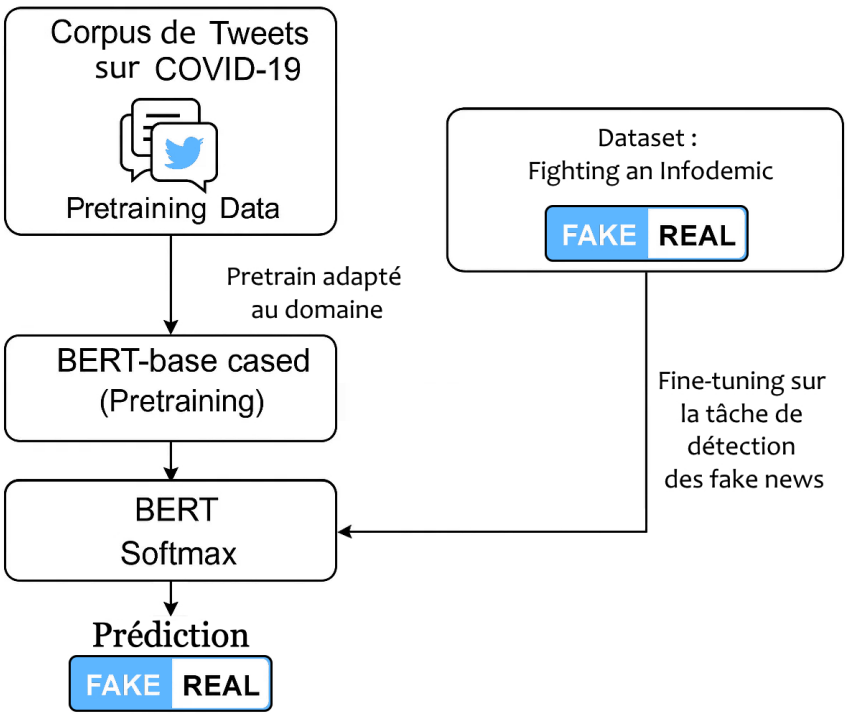


FIGURE 2.8 – Schéma du pipeline textuel pour la détection de fake news proposé par WANI et al. [157]. Le modèle `BERT-base cased` est d'abord adapté au domaine à l'aide d'un pré-entraînement non supervisé sur un corpus de tweets COVID-19, puis fine-tuné sur le dataset *Fighting an Infodemic* [104]. La classification binaire fake-real est réalisée à partir de la représentation du token `[CLS]`.

Résultats : Cette architecture atteint une précision de **98.41%** sur l'ensemble de test, surpassant le baseline SVM (93,32%) de plus de cinq points. Elle surpasse également d'autres modèles séquentiels (CNN, LSTM, BiLSTM), confirmant la supériorité des architecture transformers-based pour cette tâche.

Cet article met en exergue l'efficacité du fine-tuning de modèles de langage sur des corpus spécifiques au domaine cible, avant l'entraînement supervisé. Cette stratégie permet de combiner la puissance des représentations contextuelles apprises sur de grands corpus génériques avec une sensibilité accrue au style propre aux fake news COVID-19. Elle constitue aujourd'hui une référence en matière de détection de désinformation purement textuelle.

2.5.4 Benchmark des approches de détection textuelles

Le tableau 2.7 présente une sélection de modèles textuels représentatifs pour la détection de fake news, couvrant une diversité d'architectures, de périodes et de jeux de données. Les premières approches, comme celle de RUBIN et al. [121], reposent sur des features linguistiques explicites combinées à des classifieurs classiques (SVM), et obtiennent des scores déjà significatifs (F1 = 0.87). Toutefois, les méthodes basées sur du *deep learning*, bien qu'initialement peu performantes sur certains jeux de données complexes (ex. LSTM sur LIAR), montrent une nette amélioration avec l'arrivée des modèles pré-entraînés comme BERT et ses variantes. En particulier, l'adaptation de BERT au domaine spécifique des tweets COVID-19 (WANI et al. [157]) permet d'atteindre des performances remarquables (Précision = 98.4%). Ces résultats confirment à la fois l'importance des modèles contextualisés, et la puissance des transformers pré-entraînés sur de larges corpus pour extraire des représentations sémantiques profondes du texte dans le cadre de la

détection des fake news.

Malgré les bonnes performances de l'état de l'art, les méthodes essentiellement centrées sur le contenu textuel intrinsèque, ne prennent pas en compte le contexte externe la véracité factuelle objective, comme un travail journalistique pourrait le faire par exemple. Or, les auteurs de fake news peuvent produire des textes stylistiquement crédibles, sans pour autant dénoter de faits réels.

Pour pallier cette limite, une autre classe de méthodes cherche à exploiter des sources de connaissance externes (factuelles, structurelles ou sémantiques) pour renforcer la détection des fausses informations. Ces approches, dites *knowledge-based*, intègrent des bases de faits, des graphes de connaissances ou encore des modèles d'inférences logiques, et visent à confronter le contenu d'une news avec des connaissances vérifiables. Nous présentons dans la section suivante les principales approches de ce type et leur contribution dans la lutte contre la désinformation.

TABLE 2.7 – Comparaison de modèles textuels pour la détection de fake news

Article	Dataset	Architecture	Score	Année
RUBIN et al. [121]	Dataset maison [17]	SVM + grammaire, ponctuation, absurdité	F1 = 0.870	2016
GIRGIS, AMER et GADALLAH [42]	LIAR	RNN	F1 ≈ 0.210	2018
GOLDANI, MOMTAZI et SAFABAKHSH [43]	LIAR	Capsule Network	Précision = 0.395	2021
JWA et al. [62]	FNC-1	BERT	F1 = 0.746	2019
GAUTAM, V et MASUD [40]	Fighting an Infodemic	XLNet + LDA	F1 = 0.960	2021
WANI et al. [157]	Fighting an Infodemic	BERT fine-tuned	Précision = 0.984	2021

2.6 Approches Knowledge-based

La détection des fakes news, si elle est conçue uniquement sur le contenu intrinsèque aux informations, peut se révéler inefficace face à des images crédibles et des textes bien rédigés. En effet, certaines formes de désinformation reposent sur des subtilités sémantiques ou des inversions de faits qui échappent aux modèles fondés sur la forme du texte ou sur sa structure linguistique. Pour pallier cette faiblesse, les approches *knowledge-based* s'appuient sur des sources externes de connaissance fiables, qui permettent . Ces connaissances globales, peuvent être contenues dans diverses bases de données, ou des sites d'informations fiables ou encore dans grands modèles de langages. Nous présentons dans cette section les grandes familles de méthodes knowledge-based, illustrées par un modèle représentatif, avant de conclure par un benchmark comparatif.

2.6.1 Méthodes basées sur les graphes de connaissances

Les graphes de connaissances (KGs) constituent une structure de données permettant de représenter des faits sous la forme de triplets `<sujet, prédicat, objet>`. Leur usage dans la détection des fake news repose sur l'idée de comparer les affirmations extraites d'une information suspecte à un ensemble de faits établis, présents dans une base de connaissances vérifiée, comme Wikidata, DBpedia, ou un graphe personnalisé issu de vérificateurs de faits.

Les méthodes exploitant les KGs suivent généralement une des approches suivantes

Extraction d'affirmations factuelles : Le contenu d'un article évalué est transformé en un ensemble de triplets $(s_i, p_i, o_i)_i$, typiquement via des techniques d'extraction comme du parsing sémantique ou de la détection d'entités nommées . L'extraction, peut également se baser sur des ontologies pré-définies. Cette étape reste délicate, notamment sur du texte court (ex. : tweets), où la syntaxe est moins structurée.

Principe de comparaison factuelle : Le contenu d'une information peut être transformé en un ensemble de triplets factuels $K = \{(s_i, p_i, o_i)\}$ via des techniques d'extraction comme du parsing sémantique (tâche d'*open Information Extraction*) ou de la détection d'entités nommées . Ces triplets sont ensuite comparés à un graphe de connaissances G contenant un ensemble de faits établis K_t. Pour chaque triplet extrait k_i, une fonction $f(k_i, G) \in [0, 1]$ estime sa véracité en cherchant une correspondance proche dans G, basée par exemple sur une distance sémantique (ex. BERT-based) entre entités vectorisées (ex. embedding TransE [10] ou ComplEx [144]) . Une forte correspondance permet de valider le fait. A l'inverse, une absence ou contradiction peut indiquer une désinformation potentielle. Par exemple Vedula et Parthasarathy [145] introduisent FACE-KEG, un modèle qui génère un graphe de connaissances pour chaque news et utilise DBpedia comme source de vérité à des fins de classification explicable.

Ces approches permettent de combler certaines lacunes des modèles purement textuels, notamment en identifiant des erreurs factuelles, même dans des textes bien écrits. Leur limite principale réside dans la difficulté d'extraction précise des triplets, ainsi que dans la couverture incomplète ou datée des graphes de connaissances publics. Pan et al. [101] soulignent néanmoins que de bonnes performances sont possibles avec les graphes de connaissances en dépit de leur incomplétude. Notamment en exploitant des embedding sémantiques comme TransE, capables de pallier les manques et imprécisions.

Usage des GNNs : Les méthodes récentes représentent souvent les graphes de connaissances sous une forme apprenable via des Graph Neural Networks (GNNs). Les GNNs sont des réseaux neuronaux conçus pour apprendre sur des données structurées en graphes. Chaque nœud d'un graphe est représenté par un vecteur de features, mis à jour à chaque couche via l'agrégation des informations provenant de ses voisins. Ce mécanisme de propagation permet de capturer les dépendances structurelles entre entités, ce qui les rend particulièrement adaptés à la modélisation des connaissances sous forme de graphes. Dans le contexte de la détection de fake news, les GNNs sont utilisés pour représenter des graphes d'entités extraites (personnes, organisations, lieux) et leurs relations (provenance, liens factuels, co-occurrence), puis pour propager un score de crédibilité entre les nœuds. Des architectures comme le GCN [66] ou le GAT [146] permettent d'exploiter ces graphes pour enrichir les représentations contextuelles et détecter les incohérences structurelles dans les contenus.

Ces architectures permettent de propager l'information entre nœuds reliés, capturant ainsi les dépendances structurelles entre entités. Dans le contexte de la détection de fake news, plusieurs travaux ont montré que les GNN permettent de mieux exploiter les relations sémantiques, syntaxiques ou factuelles entre les éléments d'une information.

Par exemple, Han et al. [49] utilisent des subGNN pour l'embedding de graphes de connaissances associés à des news, classifiées de manière explicable dans un second temps. Cependant, à ce jour, aucun modèle n'intègre simultanément le contenu, le contexte, les relations sémantiques et des connaissances externes dans un même graphe traité par GNN [109].

Par ailleurs, les architectures actuelles de type GCN souffrent de limitations structurelles : elles sont généralement restreintes à deux ou trois couches, du fait des problèmes

de gradients décroissants dans les architectures profondes. La construction de GNN plus flexibles et plus profonds de contexte et de connaissance, représente un défi actuel pour la détection de fake news et plus largement pour les applications NLP basées sur les GNN.

2.6.2 Méthodes basées sur les modèles de langage préentraînés (LLMs)

En parallèle des graphes de connaissances explicites, les modèles de langage préentraînés à grande échelle (LLMs) tels que BERT, GPT ou T5 ont récemment émergé comme des alternatives puissantes pour intégrer des connaissances de manière implicite dans la détection de fake news. Ces modèles, entraînés sur des milliards de tokens issus de multiples sources (livres, encyclopédies, sites web), encapsulent une représentation latente de faits, de relations causales, et de connaissances générales du monde réel. De plus, leur capacité à raisonner et détecter les incohérences en font des outils adaptés à la détection des fake news.

Classification automatisée : Les LLMs sont capables, en formulant l'assertion à vérifier comme une question, de répondre à une requête de classification de fake news à partir de leur savoir intégré. De cette façon, sur le dataset PolitiFact, Li, Zhang et Malthouse [75] obtiennent un F1-score de 0.73 en zero-shot avec le modèle *gpt 3.5 turbo*. Liu et al. [81] soulignent néanmoins les difficultés des VLMs (visual language models), à classifier correctement des fake news multimodales en zero-shot.

Vers une exploitation agentique des LLMs : Plutôt que de demander à un LLM de classer directement une information comme vraie ou fausse via un simple prompt, plusieurs travaux récents proposent d'adopter une approche dite *agentique*, où le modèle suit une procédure structurée d'évaluation, étape par étape.

Par exemple, la méthode présentée dans [75] propose un pipeline en plusieurs modules : analyse de la source (URL), détection de langage sensationnaliste ou biaisé, recherche de contradictions, évaluation du bon sens, etc. Chaque module est exécuté indépendamment à l'aide de LLMs, puis leurs résultats sont synthétisés dans une décision finale. Cette décomposition permet une meilleure explicabilité du raisonnement, une performance accrue par rapport au prompting direct (zero-shot), et une adaptation plus fine à des cas complexes ou ambigus. Sur Politifact ils parviennent à obtenir un score F1 de 0.88. Ce type d'approche transforme le rôle du LLM de simple classificateur à un agent cognitif capable d'effectuer une analyse structurée. Ainsi les méthodes LLM-based présentent des avantages conséquents, elles ne nécessitent pas de graphe structuré explicite, sont facilement adaptables à de nouveaux domaines, et peuvent manipuler des faits récents si les données d'entraînement sont régulièrement mises à jour. Néanmoins, elles restent sensibles aux hallucinations des modèles, aux biais culturels et politiques des corpus d'entraînement, et aux coût computationnels, même en simple inférence. La combinaison de LLMs au sein de méthodes hybrides avec des outils classiques est également prometteuse pour renforcer la robustesse des systèmes de détection de fake news [81].

2.6.3 Exemple de modèle : DEAP-FAKED (Mayank et al., 2021)

Mayank, Sharma et Sharma [87] présentent en 2021 une méthode de classification des fake news se reposant uniquement sur le titre de celles-ci et sur des connaissances externes, issues de graphes de connaissances. L'intérêt d'utiliser les titres est multiple,

d'abord ils concentrent souvent l'essentiel de l'information de l'article, ensuite, ils incarnent généralement le style de l'article tout entier (clickbait, style journalistique formel etc.). En outre, se concentrer uniquement sur les titres permet d'en fournir une représentation vectorielle efficace avec moins de données et moins de bruit, qu'en traitant tout le contenu de l'article. Plus précisément : DEAP-FAKED est un framework constitué de trois composantes :

(1) Un encodeur de news : Réalise un encodage vectoriel du titre des news évaluées
(2) Un encodeur d'entités : Identifie les entités nommées dans le titre et encode chacune des entités à partir d'un graphe de connaissance
(3) Une couche de classification : Renforce l'apprentissage de l'encodage des news et des entités, à une fin de classification des fake news

Encodeur de news : L'encodage de données séquentielles est un problème très classique en NLP, résolu généralement par l'emploi de réseaux récurrents tels que les LSTM. Empiriquement les auteurs ont choisi un biLSTM à 2 couches.

Encodeur d'entités : Les entités nommées présentes dans le titre ("Joe Biden", "France" etc.) sont extraites par *Named entity Recognition* (NER) via la bibliothèque python SpaCy et le modèle RoBERTa. Ces entités sont dans un second temps associées à l'entité la plus proche contenue dans le graphe de connaissance à grande échelle de Wikidata [154]. L'embedding *ComplEx* [144] basé sur la factorisation tensorielle dans un espace complexe, est utilisé pour encoder l'entité. Enfin les représentations d'entités sont agrégées par moyenne pour produire une unique représentation vectorielle du contexte de connaissance associé au titre.

Couche de classification : Les représentations des news et des entités précédemment calculées sont concaténées pour créer une représentation globale des news et des entités en jeu. Les vecteurs issus des deux encodeurs sont concaténés, puis passés dans une couche entièrement connectée pour la prédiction finale. Le modèle atteint un F1-score de 0.78 sur le dataset CoAID [25], démontrant l'utilité de l'intégration de connaissances externes pour améliorer la classification à partir de très peu de texte.

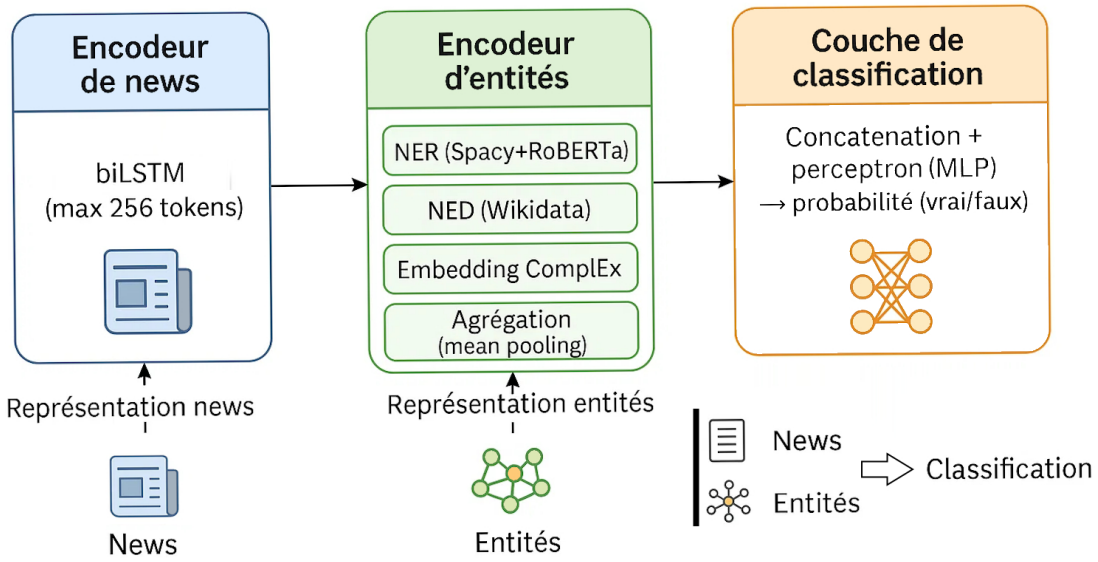


FIGURE 2.9 – Framework du modèle DEAP-FAKED [87]. Les titres des news sont encodés par un LSTM, puis les entités nommées sont match à un graphe de connaissance externe et encodées par ComplEx. Enfin les représentations sont concaténées et servent d'entrée à un perceptron multicouche pour la classification binaire.

2.6.4 Benchmark des approches Knowledge-based

TABLE 2.8 – Benchmark des approches de détection de fake news basées sur des connaissances

Article	Dataset	Architecture	Score	Année
Li et al. [75]	Politifact	GPT 3.5 turbo zero-shot	F1 : 0.73	2024
Li et al. [75]	Politifact	LLM agentique	F1 : 0.88	2024
DEAP-FAKED [87]	CoAID	BiLSTM, Wikidata, ComplEx	F1 : 0.78	2021
HAN et al. [49]	PolitiFact	sub-GNN	F1 : 0.919	2021
FACE-KEG [145]	FEVER	Graph transformer, DBpedia	F1 : 0.712	2021
Boissonneault et al. [8]	LIAR	ChatGPT zero-shot	F1 : 0.865	2024

Les résultats présentés dans le tableau 2.9 illustrent la diversité des approches knowledge-based pour la détection de fake news, allant des modèles fondés sur des graphes de connaissances explicites (comme DEAP-FAKED ou FACE-KEG) aux méthodes plus récentes utilisant des LLMs dans des cadres de fine-tuning ou zero-shot. Des approches relativement simples comme DEAP-FAKED, qui combinent un encodeur séquentiel et des embeddings issus d'un graphe de connaissances (Wikidata + ComplEx), atteignent des scores compétitifs (F1 : 0.78 sur CoAID), tout en exploitant peu de contenu pour une seule news (uniquement le titre). À l'inverse, des architectures plus complexes comme sub-GNN [49], qui exploitent des sous-graphes construits autour des entités extraites, atteignent des scores élevés (F1 : 0.919), mais au prix d'une complexité computationnelle accrue.

Depuis à peine une demi décennie, l'intégration des LLMs dans les pipeline de détection des fake news émerge également comme une alternative prometteuse. BOISSONNEAULT et HENSEN [8] parviennent à un haut score F1 de 0.865, avec ChatGPT (version non renseignée) en zero-shot sur le dataset LIAR, tandis que les travaux récents de Li et al. [75] montrent qu'un agent LLM enrichi par une base de connaissances surpasse nettement une simple utilisation zero-shot(F1 : 0.88 vs. 0.73).

En résumé, les approches knowledge-based apportent une valeur ajoutée certaine, notamment en contexte de faible ressource textuelle. L'intégration explicite de graphes (KGs) reste performante, mais les méthodes récentes tendent à les intégrer de manière implicite dans des architectures LLMs hybrides, plus flexibles mais également plus coûteuses à entraîner ou à inférer.

2.7 Exploitation de Contenu Multimodal : textuel et visuel

Le contenu textuel d'une publication fournit des indicateurs linguistiques, tandis que son contenu visuel peut renforcer ou au contraire contredire le message véhiculé. L'exploitation combinée de ces modalités permet donc de détecter des incohérences caractéristiques des fake news. La plupart des travaux de détection et classification des fake news qui utilisent le contenu multimodal, texte et image, se basent sur la même idée, l'image et le texte sont encodés vectoriellement séparément puis agrégés, avant d'être transmis à un classifieur binaire. Les modèles diffèrent sur les backbones visuels et textuels utilisés, ainsi que sur les méthodes d'agrégation. En 2017, Jin et al. [60] encodent le texte et le contexte social (#, etc.) de manière jointe avec un RNN et l'image avec un backbone VGG-19 accompagnés mécanismes d'attention qui permettent de prendre en compte l'encodage du texte et du contexte dans celui de l'image. En 2019, Singhal et al. [134] présentent SpotFake qui encode séparément l'image avec VGG-19 et le texte avec BERT. Des perceptrons multicouches sont ajoutés pour apprendre des représentations en faible dimension et pour la classification. En 2021, Wu et al. [161] présentent MCAN, qui encode à la fois le texte et l'image au niveau des pixels, mais aussi au niveau du domaine des fréquences (comme dans le modèle 2.7 présenté précédemment). Les représentations sont ensuite agrégées avec des mécanismes de co-attention. MCAN obtient de meilleurs résultats que SpotFake.

2.7.1 Présentation d'un modèle : MVAE (Khattar et al. 2019)

En 2019 Khattar et al. [64] proposent un autoencodeur multimodal variationnel 2.10 pour la détection des fake news à partir du texte et d'un contenu visuel associé. De la même manière que les autres modèles multimodaux, MVAE encode à la fois le texte et l'image avant d'en construire une représentation unifié. Pour ce faire MVAE se base sur une architecture d'autoencodeur, c'est à dire l'apprentissage d'un encodeur et d'un décodeur de l'entrée multimodale, desquels sont extraits une représentation adaptée à la détection des fake news. Ainsi MVAE est composé de trois parties, un encodeur, un décodeur et un module de classification binaire des fake news.

L'Encodeur : L'encodeur est chargé de transformer les informations textuelles et visuelles en une représentation vectorielle dans un seul espace latent $\mathbb{R}^n$. Il est divisé en deux parties, un encodeur visuel et un encodeur textuel. L'encodeur textuel est un RNN composé de LSTM bi-directionnels qui capturent les dépendances contextuelles des mots. La représentation finale du texte est obtenue après un passage de la dernière sortie LSTM à travers une couche entièrement connectée. L'encodeur visuel repose sur un CNN pré entraîné (VGG-19) pour extraire les caractéristiques visuelles de l'image. La sortie du VGG-19 est passée à travers plusieurs couches entièrement connectée pour obtenir le vecteur de caractéristiques visuelles. Les deux vecteurs de représentation sont concaténés, leur moyenne μ et variance σ sont calculés et le vecteur latent est tiré d'une distribution normale $\mathcal{N}(\mu, \sigma)$.

Le Décodeur : Le décodeur reconstruit les données textuelles et visuelles originales à partir de la représentation latente obtenue en sortie de l'encodeur. Il est également décomposé en deux parties. L'une reconstruit les probabilités des mots à partir de couches entièrement connectées et de bi-LSTM, l'autre reconstruit les features visuels de l'image avec des couches entièrement connectées.

Le classifieur de fake news : Le classifieur de fake news utilise la représentation multimodale apprise pour classifier si une news est fake ou non. Le vecteur latent en sortie du décodeur est passé à travers deux couches entièrement connectées puis une fonction d'activation sigmoïde pour obtenir la probabilité finale.

Le modèle est entraîné en utilisant une fonction de perte qui combine trois éléments : Une perte de reconstruction qui mesure la qualité de l'autoencodage en comparant les données d'entrées et les données reconstruites. Une perte de divergence KL pour ajuster les paramètres μ et σ de manière à se rapprocher d'une distribution normale. Une perte d'entropie croisée pour la classification. Le modèle MVAE permet d'apprendre une représentation multimodale, compacte et significative des données. La représentation des données textuelles et visuelles au sein du même espace latent permet de capturer des corrélations et dépendances entre les deux types de données, ce qui permet de détecter des incohérences si l'image et le texte ne sont pas alignés (ce qui est généralement le cas des fake news).

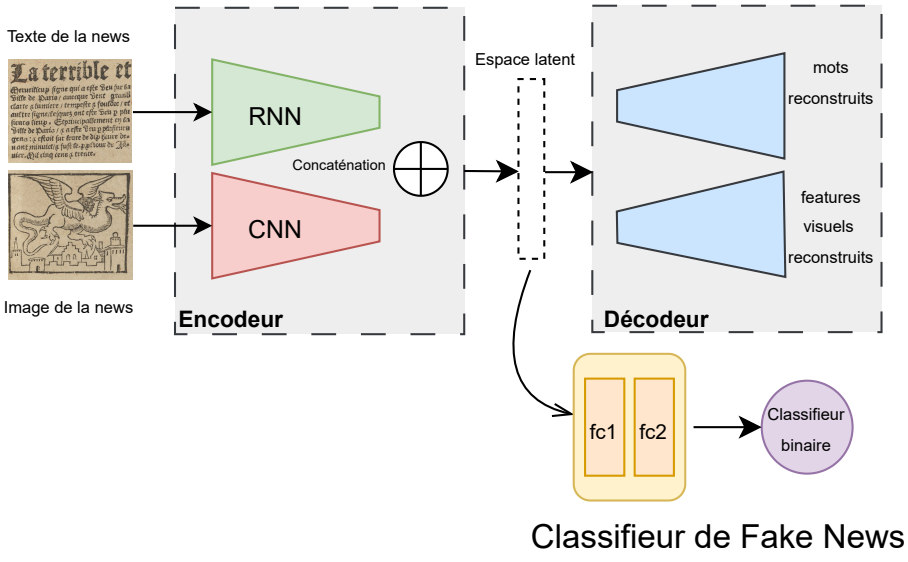


FIGURE 2.10 – Architecture schématique du MVAE [64]. Le texte et l'image sont encodés respectivement par un biLSTM et VGG-19, concaténés puis reconstruits à partir de leur représentation dans un espace latent. La représentation latente est utilisée en entrée d'un perceptron à deux couches pour la classification binaire.

2.7.2 Benchmark des approches de détection multimodales

Les résultats du tableau 2.9 montrent une progression claire des performances des approches multimodales au fil des années, en parallèle des avancées en vision par ordinateur et en traitement du langage naturel. Les premiers modèles (comme celui de JIN et al. [60]) utilisaient des architectures simples de type LSTM + CNN (VGG-19), avec des fusions

TABLE 2.9 – Benchmark des approches de détection de fake news basées sur le contenu multimodal

Article	Dataset	Architecture	Score	Année
Jin et al. [60]	Weibo	LSTM, att-RNN, VGG-19	Acc : 0.78	2017
MVAE [64]	Weibo	Autoencodeur, VGG-19, biLSTM	Acc : 0.82	2019
SpotFake [134]	Weibo	VGG-19, BERT	Acc : 0.89	2019
MMIFND [41]	FakeNewsNet	VGG-16, LSTM, BERT	F1 : 0.76	2020
MCAN [161]	Weibo	VGG-19, CNN (freq), BERT	Acc : 0.90	2021
SDML [178]	Weibo	ViT-B/16, BERT-16	Acc : 0.93	2025

explicites et parfois des mécanismes d'attention. Avec l'introduction dans les pipelines de détection de modèles pré-entraînés plus puissants comme BERT pour le texte et VGG-19 (ou ViT plus récemment) pour l'image, les performances se sont nettement améliorées. Ainsi, dés 2019, l'approche multimodale SpotFake surpasse l'approche unimodale visuelle MVNN 2.7 sur le dataset weibo de 5% d'accuracy, démontrant l'intérêt des approches multimodales pour la détection des fake news.

Dans le même esprit, MCAN [161] innove en combinant l'information du domaine des pixels et des fréquences via un mécanisme de co-attention, ce qui lui permet d'obtenir un score de 89.9%.

Enfin, les modèles récents comme SDML [178], exploitant des architectures transformer pour les deux modalités (ViT pour l'image, BERT pour le texte), franchissent une nouvelle étape avec une accuracy supérieure à 92%.

Cependant, cette montée en performance s'accompagne d'une complexité accrue en termes de calcul et de supervision. Par ailleurs, l'interprétabilité de ces modèle demeure limitée : les mécanismes d'attention permettent parfois de visualiser l'importance des modalités, mais les raisons profondes d'une prédiction restent souvent opaques.

En résumé, l'exploitation du contenu multimodal améliore significativement la détection des fake news. Les travaux récents tendent à s'orienter vers des fusions plus complexes (espace latent partagé, co-attention) qui ouvrent des perspectives prometteuses, mais posent également de nouveaux défis en termes de coût, de généralisation et d'explicabilité.

2.8 Conclusion de l'état de l'art

Synthèse critique de la littérature : L'analyse critique de la littérature révèle que, bien que la détection automatique des fake news ait bénéficié de progrès considérables, notamment récemment via les modèles de langage pré-entraînés (BERT, LLMs, ...) (et plus généralement grâce aux transformers) ou via les graphes de connaissances ou encore via des architectures multimodales avancées, aucune approche n'a jusqu'à présent, à notre connaissance, intégré les trois contraintes suivantes :

(i) Multimodalité complète : texte et image

(ii) Explicabilité forte : offrant une traçabilité fine des décisions du classifieur

(iii) Intégration de connaissances externes actualisées.

Quelques travaux offrent néanmoins des pistes prometteuses : FACE-KEG exploite des graphes de connaissances à des fins d'explicabilité pour justifier chaque triplet factuel analysé, mais se cantonne au contenu textuel, des pipelines agentiques basés sur les LLMs décomposent un raisonnement de fact-checking en étapes explicites, mais sans fu-

sion des modalités visuelle et textuelle, DEAP-FAKED s'appuie sur le texte et intègre des connaissances externes via graphe de connaissances.

Cette thèse s'inscrit dans la suivance de ces approches et vise à combler les lacunes identifiées et à proposer une détection plus interprétable des fake news multimodales.

2.8.1 Limites des comparaisons quantitatives

Il convient de noter que nous ne comparons pas quantitativement, dans ce manuscrit, toutes les approches présentées dans cet état de l'art. Et ce pour plusieurs raisons : Les problématiques ciblées peuvent différer (détection de rumeurs, manipulation d'images, événements ciblés, etc.), ce qui limite la comparabilité directe. De plus, de nombreux modèles ne sont pas disponibles en ligne (code, poids ou configuration), rendant impossible leur reproduction expérimentale. Enfin, les modalités exploitées (texte seul, multimodalité texteimage, métadonnées sociales, etc.) ainsi que les jeux de données et métriques varient considérablement d'une étude à l'autre.

2.8.2 Tableau de synthèse qualitative

Pour clore cet état de l'art, nous proposons une Table 2.10 qualitative qui synthétise les principales approches de détection et qui positionne scientifiquement ce travail de thèse.

TABLE 2.10 – Synthèse qualitative des principales approches de détection de fake news

Méthode	Deep Learning	Visuel	Textuel	Knowledge	Explicabilité	Date
MVAE [64]	✓	✓	✓	✗	✗	2019
MVNN [112]	✓	✓	✗	✗	✓	2019
MMIFND [41]	✓	✓	✓	✗	✗	2020
GAUTAM, V et MASUD [40]	✓	✗	✓	✓	✗	2021
FACE-KEG [145]	✓	✗	✓	✓	✓	2021
DEAP-FAKED [87]	✓	✗	✓	✓	✓	2021
SDML [178]	✓	✓	✓	✗	✗	2025
Notre approche	✓	✓	✓	✓	✓	2025

Ce tableau récapitulatif met en évidence l'absence d'approches qui satisfait simultanément les contraintes de multimodalité, d'explicabilité et d'intégration de connaissances externes. Nous positionnons nos recherches dans ce manque, en visant à proposer une méthode unifiant toutes ces dimensions.

Deuxième partie

Fondements Formels et Applicatifs pour le Modèle de Détection

L'objet de cette partie est de définir à la fois les fondements méthodologiques (cf. Chapitre 3 pour les Graphes de Connaissances et Chapitre 5 pour les méthodes de clustering de graphes) qui serviront dans notre architecture finale de détection de fake news multimodales (cf. Chapitre 7). Nous présentons également leur application au sein de contributions intermédiaires (cf. Chapitre 6 pour la Détection et la Localisation de transformations d'images, et Chapitre 4 pour la Génération de Graphes de Scène).

Ces fondements constituent une partie essentielle du socle de l'approche proposée dans cette thèse.

Interlude : Vue d'ensemble de l'architecture proposée

Avant de détailler les différentes composantes méthodologiques, nous présentons ici une vue d'ensemble de l'architecture finale de détection que nous proposons (Figure 2.11). Cette illustration a pour objectif de donner au lecteur une première vision synthétique du pipeline complet. Les différents modules seront expliqués en détail dans le Chapitre 7.

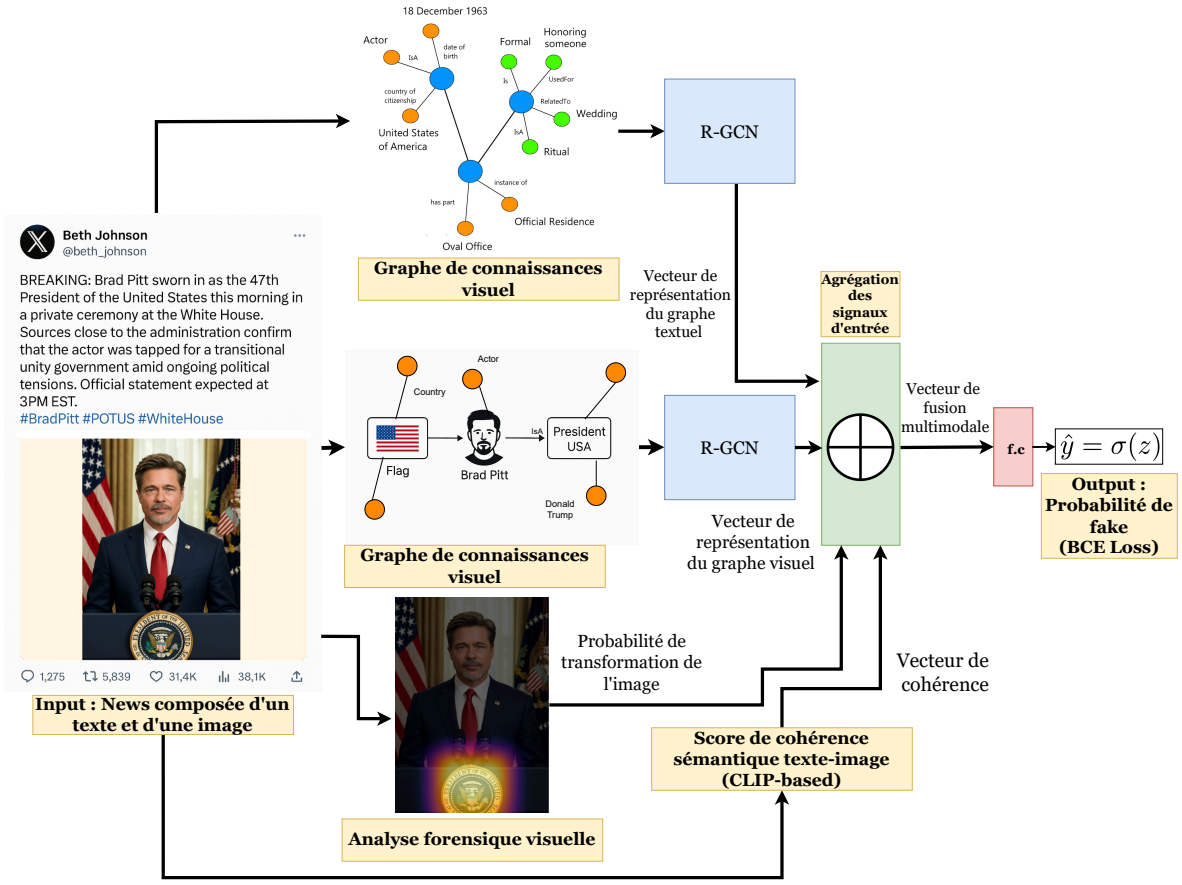


FIGURE 2.11 – Aperçu global de l'architecture proposée pour la détection multimodale de fake news.

3 Fondamentaux des Graphes de Connaissances

Dans notre état de l'art, nous avons mis en évidence l'intérêt d'architectures qui utilisent des connaissances externes pour améliorer la détection des fake news (cf. Section 2.6). Notamment, des travaux tels que ceux de Mayank, Sharma et Sharma [87] ou Pan et al. [101] démontrent l'intérêt des graphes de connaissances pour le fact-checking d'actualités, à des fins d'explicabilité des prédictions ou de traitement de données multimodales.
C'est pourquoi, dans ce chapitre et le suivant, nous nous attardons à définir les graphes de connaissances au sens large, leur formalisation, leurs déclinaisons (comme les graphes de scènes), leur exploitation (en présentant des embeddings tels que transE ou ComplEx), et leurs applications (dans des tâches de Visual Question Answering, ou à travers des architectures comme R-GCN).

3.1 Introduction

3.1.1 Définition

Motivation Les graphes de connaissances (ou *knowledge graphs* (KG)) sont apparus comme une réponse à un besoin croissant : représenter de manière structurée, et exploitable par machine des ensembles complexes de connaissances. Face à la masse croissante de données hétérogènes et multimodales, il devient nécessaire de disposer d'outils formels qui articulent les entités du monde réel et les relations qui les lient, tout en permettant une automatisation du raisonnement sur celles-ci.

Formalisation Classiquement, un graphe de connaissances est un graphe orienté et labellisé dont les nœuds représentent des entités du monde réel (par exemple un personnage historique, un concept scientifique, une ville) et dont les arêtes représentent des relations sémantiques explicites entre ces entités (par exemple, *est né à*, *capitale de*, *a inventé*). Cette structure permet non seulement de modéliser des *faits* sous forme de triplets `<sujet,relation,objet>`, mais aussi de relier des données hétérogènes, d'inférer de nouvelles relations implicites, et d'alimenter des modèles d'intelligence artificielle. Il existe des variations sur la définition des graphes de connaissances. En particulier, il est possible d'ajouter à chaque entité des attributs ou des propriétés. Ainsi une entité *voiture* peut être spécifiée de couleur *verte* et de marque *Citroën*. Puisque ces attributs peuvent se ramener à des faits canoniques tels que `<voiture, est, verte>`, nous considérons dans la suite la définition suivante :

Définition 3.1.1. Soient $\mathcal{E}$ un ensemble d'entités et $\mathcal{R}$ un ensemble de relations.
Un *graphe de connaissances* est défini comme un ensemble de triplets

$$\mathcal{G} = \{(s, r, o) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$$

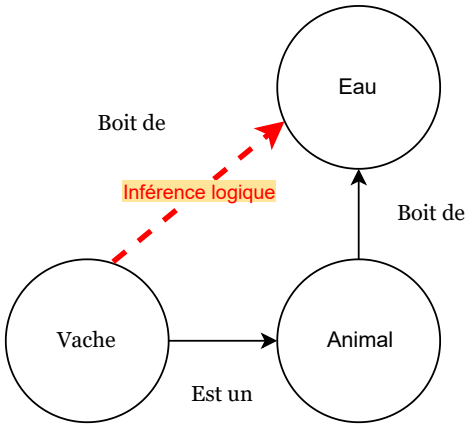


FIGURE 3.1 – Exemple de graphe de connaissances élémentaire, composé des faits `<Une Vache, est un, Animal>` et `<Animal, boit de, Eau>`. Le fait `<Une Vache, boit de, Eau>` peut s'inférer par syllogisme.

où s (le sujet) et o (l'objet) sont des entités, et r est une relation dirigée du sujet vers l'objet.

Cette définition correspond à la représentation des connaissances sous la forme de triplets répondant au standard dit RDF (Resource Description Framework).

Inférence de nouvelles connaissances Des langages tels que OWL (Web Ontology Language) permettent de décrire des triplets RDF, d'exprimer des contraintes logiques et d'inférer de nouvelles connaissances au sein des graphes. D'autres approches, tels que des embeddings, ou des réseaux de neurones profonds permettent d'apprendre des représentations quantitatives des graphes de connaissances et d'en extraire de nouveaux faits, ou d'en exploiter la sémantique.

Graphes de connaissances publics et privés De nombreux graphes de connaissances publics, liés à des entités du monde réel ou bien à des concepts généraux, sont mis à disposition pour les chercheurs où le grand public. D'autres, regroupant des connaissances sensibles ou spécifiques à un domaine, sont plus restreintes. Ainsi, on distingue deux familles de knowledge graphs : les *open knowledge graphs*, accessibles publiquement (ex. DBpedia, Wikidata, ConceptNet [154]), et les *enterprise knowledge graphs*, utilisés en entreprise pour des cas d'usage tels que la recherche sémantique, la recommandation ou l'analyse métier [51]. Un des graphes de connaissances les plus complets et utilisé est Wikidata. Il s'agit d'un graphe collaboratif en libre accès qui fournit des données à la fois pour Wikipédia et pour les utilisateurs. En 2021, Wikidata contenait plus de 90 millions d'entités distinctes et plus d'un milliard de relations entre ces entités CHAUDHRI et al. [18].

Incomplétude et enrichissement continu Malgré les efforts de la communauté scientifique, un graphe de connaissances, aussi grand soit-il, demeure structurellement incomplet. Cependant, comme souligné par MAYANK, SHARMA et SHARMA [87], l'incomplétude des graphes de connaissances n'est pas toujours un frein à leurs capacités prédictive (en particulier dans le domaine des fake news), mais plutôt le reflet d'une marge de progres-

sion. Ainsi, les graphes de connaissances sont sujets à un enrichissement continuels des données qui les composent (ou bien générés *from scratch*), pour une tâche applicative donnée. Ainsi VEDULA et PARTHASARATHY [145] génère des graphes de connaissances dans un processus de fact-checking automatique, pour une meilleure explicabilité. L'enrichissement des graphes, peut se faire de manière dynamique, ou automatiquement à partir d'un modèle, d'une base de données, de textes etc. Cette facilité d'extension traduit la grande flexibilité des knowledge graphs pour représenter de la connaissance.

Requêtes L'exploration et les requêtes sur un graphe de connaissances sont identiques à la navigation sur celui-ci. Ainsi, pour extraire de l'information d'un graphe, les algorithmes de parcours de graphes sont une bonne approche en premier lieu. Cependant, pour l'exploration de toutes les inférences intéressantes HOGAN et al. [51] il est également possible d'utiliser des langages de requêtes spécifiques aux graphes tel que SPARQL. De la même manière que le SQL sur les bases de données relationnelles, le SPARQL permet d'exécuter des requêtes sur un graphe, avec des filtres, des jointures, des requêtes par motifs etc.

Listing 3.1 – Exemple de requête SPARQL retournant les musées situés à Paris

```
PREFIX ex: <http://example.org/>

SELECT ?museum WHERE {
  ?museum a ex:Museum .
  ?museum ex:locatedIn ex:Paris .
}
```

3.1.2 Distinction entre ontologies et graphes de connaissances

Dans la littérature, il n'est pas toujours fait de distinction claire entre graphes de connaissances, ontologies, et bases de connaissances. Ainsi l'ontologie YAGO (Yet Another Great Ontology) développée par l'institut Max-Planck (*Sarrebuck*, Allemagne) est appelée ontologie tout en faisant référence à un graphe de connaissance ainsi qu'à une base de connaissance [34]. Parfois la distinction entre graphe de connaissances et ontologie peut se faire en terme de quantité, c'est à dire qu'un graphe de connaissances correspondrait simplement à une ontologie à grande échelle. Parfois, un graphe de connaissances est vu comme l'addition d'une *base de connaissances/ontologie* et d'un *raisonneur*. EHRLINGER et WÖSS [34] proposent la définition suivante, sur laquelle nous nous appuyons :

Définition 3.1.2. Un graphe de connaissance acquiert et intègre les informations contenues dans une ontologie et applique un raisonneur afin d'inférer de nouvelles connaissances

3.2 Les graphes de connaissances pour le traitement des fake news

Beaucoup d'approches *content-based* pour la détection des fake news s'attachent d'avantage à la forme des informations qu'à leur fond. C'est le cas par exemple des approches *style-based* qui se concentrent sur le style d'écriture des articles. Au contraire, les approches de type *knowledge-based* (i.e *fact-checking*) sont prometteuses car elles se concentrent exclusivement sur la valeur informationnelle de la nouvelle. Les méthodes de détection de

fake news basées sur les graphes de connaissances sont donc content-based *et* knowledge-based. Les graphes de connaissance contiennent des connaissances générales sur l'etat du monde, qui sont à même de fact-checker des informations complexes. Bien que les graphes de connaissances soient généralement utilisés pour agréger ce type de connaissances globales, leur structure sous forme de triplets RDF `<sujet, prédicat, objet>` est exploitable pour décrire des concepts (graphes de concepts), ou des images (graphes de scènes). Ainsi les graphes de connaissances sont des objets structurellement suffisamment riches pour saisir la complexité d'informations précises sur des sujets divers et parfois multimodales. C'est pourquoi, dans le cadre de cette thèse, nous faisons le choix de nous appuyer sur les graphes de connaissances comme socle formel pour détecter les *fake news*, en exploitant leur richesse structurelle et leur potentiel de généralisation.

3.2.1 Exploration des stratégies d'intégration des graphes de connaissances

L'intégration de graphes de connaissances dans un pipeline de détection des *fake news* soulève plusieurs problématiques méthodologiques. Leur résolution dépend des choix de conception, du type d'information à vérifier, et des ressources disponibles. Nous décrivons ici différentes stratégies envisageables pour articuler efficacement l'usage des graphes de connaissances avec le contenu informatif à analyser.

Sélection d'un graphe pertinent au sujet : Chaque information est généralement rattachée à un sujet plus large (par exemple, une déclaration politique liée au contexte d'une élection présidentielle). Il est donc crucial de sélectionner un graphe de connaissances pertinent pour ce domaine. Une première possibilité consiste à s'appuyer sur des graphes thématiques préexistants, ou à extraire dynamiquement un sous-graphe depuis une base généraliste telle que DBpedia ou Wikidata, en filtrant les faits liés au sujet global.

Comparaison vérité / désinformation : Une autre approche consiste à travailler avec deux graphes de connaissances construits ou annotés a posteriori : l'un contenant des faits vérifiés, l'autre des assertions issues de fake news. Ces deux graphes peuvent alors être comparés pour chaque instance, à l'aide d'une mesure de similarité sémantique adaptée (ex. similarité cosinus sur des embeddings d'entités), afin d'estimer le degré de divergence entre les représentations factuelles et suspectes [101].

Incomplétude et complexité d'exploitation : Un des défis liés à l'usage des graphes de connaissances réside dans leur incomplétude, c'est-à-dire l'absence de certains faits pourtant vrais mais absents du graphe. Ce phénomène est particulièrement problématique dans le cadre des fake news, qui mobilisent souvent des relations précises, rares ou atypiques. À l'inverse, les grands graphes généralistes comme Wikidata présentent une distribution fortement déséquilibrée, centrée sur les relations les plus fréquentes. Pour atténuer ce biais, plusieurs stratégies sont envisageables : recourir à des scores de similarité sémantique (par exemple, basés sur des modèles de type BERT) pour tolérer des variations relationnelles proches, enrichir le graphe via des techniques de complétion automatique, ou encore intégrer des correspondances synonymiques à partir de dictionnaires prédéfinis.

Un second enjeu tient à la complexité même de l'exploitation de ces graphes. Leur taille, leur densité variable et leur richesse sémantique soulèvent la question de la granularité optimale à adopter : doit-on extraire un sous-graphe minimal centré sur le sujet de la news, ou au contraire intégrer un voisinage étendu ? Une extraction trop restreinte risque de négliger des faits pertinents, tandis qu'un graphe trop large peut introduire

du bruit et diluer les signaux utiles. Trouver un compromis entre richesse informative et bruit contextuel constitue ainsi un aspect crucial de toute méthode basée sur un graphe de connaissances externe comme Wikidata.

Évaluation de la plausibilité des faits : Certaines méthodes évaluent directement la plausibilité d'un triplet de type `<sujet, prédicat, objet>` au sein d'un unique graphe de connaissance contenant des informations dont la véracité est examinée. Par exemple, Pan et al. [101] s'appuient sur la connectivité du graphe. Cette idée tient de l'intuition qu'un fait est d'autant plus crédible que le graphe contient un chemin sémantiquement plausible reliant le sujet et l'objet, via une ou plusieurs relations intermédiaires.

Les approches présentées ci-dessus ne sont pas mutuellement exclusives et peuvent être combinées selon le niveau d'ambition et de ressources ainsi que la modalité traitée (texte, image, multimodal). Elles illustrent la richesse méthodologique offerte par les graphes de connaissances, tout en soulignant les compromis à opérer entre complexité, couverture et capacité de généralisation.

3.3 Embedding de graphes de connaissances

Bien que les graphes de connaissances permettent de rassembler de très nombreuses données dans un seul objet, ils demeurent en l'état trop complexes pour être exploités efficacement par des algorithmes de machine learning. Les *embeddings* de graphes sont des méthodes de conversion des graphes en vecteurs de petites dimensions exploitables algorithmiquement. Il s'agit d'un problème difficile car d'une part il s'agit d'une compression importante des données dans un espace latent et d'autres part il est souhaitable de conserver la structure de graphe sous jacente dans le vecteur associé au graphe [152].

La plupart des techniques d'embedding actuelles considèrent les graphes de connaissances comme un ensemble de triplets appelés *faits* qui sont composés d'une entité *sujet* d'une entité *objet*, et d'une *relation* correspondant à l'arête du graphe les reliant. Ces techniques reposent également souvent sur une fonction de score à définir qui mesure la plausibilité de chaque fait. Les faits contenus dans le graphe étant plus plausibles que les faits ne l'étant pas. Un embedding est alors obtenu en maximisant la plausibilité totale des faits observés. Cette manière d'obtenir un embedding ne prend en compte que les faits. Récemment, des méthodes d'embedding utilisent d'autres informations du graphe comme le type des entités ou des chemins de relations entre entités. Dans cette sous-section nous parlerons des modèles utilisant uniquement les faits.

Dans cette section, on note $\mathbb{D}^+ = \{(h, r, t),\ h \in \mathbf{E},\ t \in \mathbf{E},\ r \in \mathbf{R}\}$ l'ensemble des faits observés dans le graphe de connaissance, avec $\mathbf{E}$ l'ensemble des entités et $\mathbf{R}$ l'ensemble des relations. Typiquement un embedding de graphes de connaissance se déroule en 3 étapes : *(i)* La représentation des entités et des relations. *(ii)* La définition d'une fonction de score. *(iii)* Apprendre les représentations des entités et relations. La première étape consiste en le choix de la forme des embeddings. En général les entités sont souvent représentées par des vecteurs et les relations par des opérations sur ces vecteurs, c'est à dire des vecteurs, matrices, tenseurs, distributions gaussiennes etc. La deuxième étape consiste à définir une fonction score définie pour chaque fait (h, r, t) et qui mesure leur plausibilité. Un fait présent dans le graphe est plus susceptible d'avoir un score élevé. La troisième étape est l'apprentissage des représentations des entités et relations en maximisant la plausibilité totale des faits observés. Selon que l'on utilise une fonction de score *distance-based* ou

semantic-based on parle de modèle à distance translationnelle ou de modèle à matching sémantique.

3.3.1 Modèles à distance translationnelle

EmbeddingTransE

L'embedding TransE [10] est un des plus simples et des plus utilisés dans la littérature pour l'embedding de graphes de connaissances. L'idée derrière TransE est de plonger les entités et les relations dans le même espace de vecteurs $\mathbb{R}^d$ et d'interpréter les relations comme des vecteurs de translations entre la tête h et la queue t du fait. De sorte qu'un fait $(h, r, t) \in \mathbb{D}^+$ est traduit par des vecteurs $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$ tels que $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$. Intuitivement cette interprétation a du sens car la représentation de mots en vecteurs est capable de capturer des régularités linguistiques, ainsi l'on comprend naturellement la pseudo-égalité $Paris + EstUneVilleDe \approx France$. La fonction de score peut alors se définir comme :

$$f_r(h, r, t) = -||h + r - t||_2$$

Ainsi le score est maximal (et proche de 0) lorsque $(h, r, t) \in \mathbb{D}^+$. TransE est simple et efficace mais souffre de faiblesses de caractérisation lors des relations 1 vers plusieurs, plusieurs vers 1 ou plusieurs vers plusieurs. En effet, en reprenant l'exemple précédent, si $Paris + EstUneVilleDe \approx France$ et $Paris + EstUneVilleDe \approx Europe$ alors TransE apprendra des représentations très proches des entités *France* et *Europe* qui sont pourtant bien différentes. Une des solutions à ce problème est d'autoriser les entités a posséder une représentation distincte pour chaque relation dans lesquelles elles sont impliquées. Dès lors chaque représentation devient spécifique à une relation. Les méthodes TransH, TransR et TransD qui sont des variations de TransE tentent toutes de répondre à ce problème, au détriment parfois de la complexité temporelle.
Dans l'article originel TransE est appris avec une descente de gradient stochastique.

Embedding gaussiens

Certains modèles [50] considèrent que les entités doivent être représentées avec une certaine incertitude et donc représentent les composantes des faits comme des vecteurs issus de distributions gaussiennes multivariée :

$$\begin{cases} \mathbf{h} \sim \mathcal{N}(\mu_h, \Sigma_h) \\ \mathbf{t} \sim \mathcal{N}(\mu_t, \Sigma_t) \\ \mathbf{r} \sim \mathcal{N}(\mu_r, \Sigma_r) \end{cases}$$

Puis, de la même manière que pour TransE, l'idée est de comparer les variables $\mathbf{t}-\mathbf{h}$ et $\mathbf{r}$. Une façon de le faire est de prendre pour score $f_r(h, t)$ la divergence de Kullback-Leibler entre les distributions de $\mathcal{N}(\mu_t - \mu_h, \Sigma_t + \Sigma_h)$ et $\mathcal{N}(\mu_r, \Sigma_r)$. De sorte que :

$$f_r(h, t) \propto -\text{tr}(\Sigma_r^{-1}(\Sigma_h + \Sigma_t)) - \mu^T \Sigma_r^{-1} \mu - \ln \frac{\det(\Sigma_r)}{\det(\Sigma_h + \Sigma_t)} \tag{3.1}$$

Il existe d'autres mesures de distance entre distributions et il existe également d'autres manières d'exploiter l'idée de représentations bruitées des entités et relations (par exemple transG).

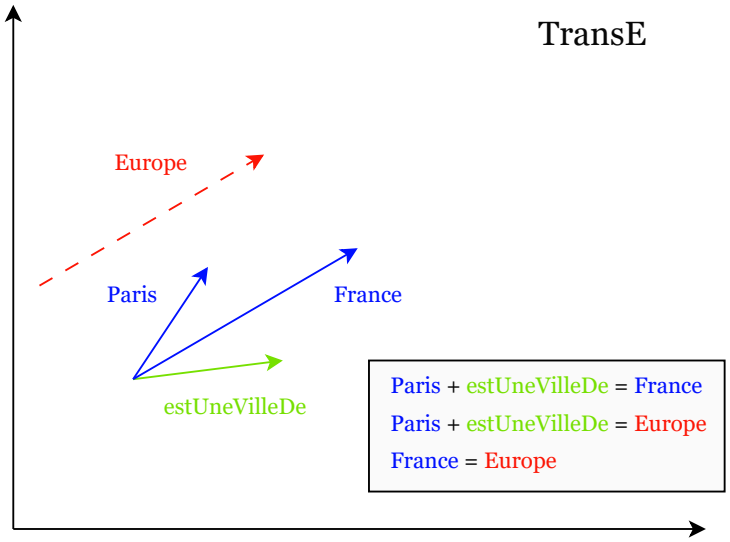


FIGURE 3.2 – Représentation vectorielle d'un graphe de connaissances selon le modèle TransE. Chaque entité (ex. *Paris*, *France*) est projetée dans un espace vectoriel. Les relations sont modélisées comme des translations : une relation `estUneVilleDe` est représentée par un vecteur tel que $\textit{Paris} + \texttt{estUneVilleDe} \approx \textit{France}$. Ce principe impose des approximations sémantiques, comme illustré ici par l'égalité des représentations des entités *France* et *Europe*.

Méthode	**Emb. entités**	**Emb. relations**	**Fonction de score $f_r(h,t)$**	**Constraintes/Régularisations**
TransE [10]	$h,t \in \mathbb{R}^d$	$r \in \mathbb{R}^d$	$-\|h+r-t\|_{1/2}$	$\|h\|_2 = 1,\ \|t\|_2 = 1$
TransH [156]	$h,t \in \mathbb{R}^d$	$r, w_r \in \mathbb{R}^d$	$-\left\|(h - w_r w_r^\top h) + r - (t - w_r w_r^\top t)\right\|_2^2$	$\|h\|_2 < 1,\ \|t\|_2 < 1,\ \|w_r^\top r\|/\|r\|_2 \leq \epsilon,\ \|w_r\|_2 = 1$
TransR [76]	$h,t \in \mathbb{R}^d$	$r \in \mathbb{R}^k,\ M_r \in \mathbb{R}^{k\times d}$	$-\|M_r h + r - M_r t\|_2^2$	$\|h\|_2,\ \|t\|_2,\ \|r\|_2 \leq 1,\ \|M_r h\|_2,\ \|M_r t\|_2 \leq 1$
TransD [57]	$h, w_h \in \mathbb{R}^d$, $t, w_t \in \mathbb{R}^d$	$r, w_r \in \mathbb{R}^k$	$-\|(w_r w_h^\top + I)h + r - (w_r w_t^\top + I)t\|_2^2$	$\|h\|_2,\ \|t\|_2,\ \|r\|_2 \leq 1,$ $\|(w_r w_h^\top + I)h\|_2,\ \|(w_r w_t^\top + I)t\|_2 \leq 1$
TranSparse [56]	$h,t \in \mathbb{R}^d$	$r \in \mathbb{R}^k,\ M_r(\theta_r) \in \mathbb{R}^{k\times d},$ $M_r^1(\theta_r^1),\ M_r^2(\theta_r^2) \in \mathbb{R}^{k\times d}$	$-\|M_r(\theta_r)h + -M_r(\theta_r)t\|_{1/2},$ $-\|M_r^1(\theta_r^1)h + r - M_r^2(\theta_r^2)t\|_{1/2}$	$\|h\|_2,\ \|t\|_2,\ \|r\|_2 \leq 1,$ $\|M_r(\theta_r)h\|_2,\ \|M_r(\theta_r)t\|_2 \leq 1,$ $\|M_r^1(\theta_r^1)h\|_2,\ \|M_r^2(\theta_r^2)t\|_2 \leq 1$
ManifoldE [162]	$h,t \in \mathbb{R}^d$	$r \in \mathbb{R}^d$	$-\left(\|h+r-t\|_2^2 - \theta_r^2\right)^2$	$\|h\|_2,\ \|t\|_2,\ \|r\|_2 \leq 1$
KG2E [50]	$h \sim \mathcal{N}(\mu_h, \Sigma_h),$ $t \sim \mathcal{N}(\mu_t, \Sigma_t)$	$r \sim \mathcal{N}(\mu_r, \Sigma_r),$ $\mu_h, \mu_t, \mu_r \in \mathbb{R}^d,$ $\Sigma_h, \Sigma_t, \Sigma_r \in \mathbb{R}^{d\times d}$	$-\mathrm{tr}(\Sigma_r^{-1}(\Sigma_h + \Sigma_t)) - \mu^\top \Sigma_r^{-1}\mu,$ $-\ln \frac{\det(\Sigma_r)}{\det(\Sigma_h+\Sigma_t)} - \mu^\top \Sigma^{-1}\mu - \ln\det(\Sigma),$ $\mu = \mu_h + \mu_r - \mu_t,\ \Sigma = \Sigma_h + \Sigma_r + \Sigma_t$	$\|\mu_h\|_2,\ \|\mu_t\|_2,\ \|\mu_r\|_2 \leq 1,$ $c_{\min} I \preceq \Sigma_h \preceq c_{\max} I,$ $c_{\min} I \preceq \Sigma_t \preceq c_{\max} I,$ $c_{\min} I \preceq \Sigma_r \preceq c_{\max} I$

TABLE 3.1 – Comparaison des méthodes d'embedding à distance translationnelle et des fonctions scores associées. Table issue de [152]

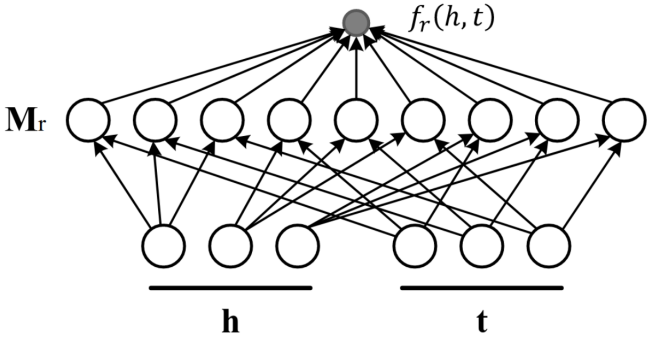


FIGURE 3.3 – Représentation schématique de la fonction de score du modèle RESCAL [99]. Figure issue de WANG et al. [152]

Embedding ComplEx

Le modèle d'embedding ComplEx [144] peut être vu comme une extension de TransE dans laquelle les entités et relations sont plongées dans un espace vectoriel complexe. Il repose sur une fonction de score de type translationnel, qui exploite le produit hermitien afin de modéliser à la fois les relations symétriques et antisymétriques, ce qui nest pas possible avec TransE. Plus précisément, pour un triplet (h, r, t), le score est défini par

$$f_r(h,t) = \Re\Big(\langle \mathbf{w}_r, \mathbf{e}_h, \overline{\mathbf{e}_t} \rangle\Big) \tag{3.2}$$

où $\mathbf{e}_h, \mathbf{e}_t \in \mathbb{C}^K$ sont les vecteurs complexes des entités et $\mathbf{w}_r \in \mathbb{C}^K$ celui de la relation. Cette formulation conserve la simplicité et l'efficacité des modèles translationnels tout en capturant une plus large variété de relations, notamment les relations antisymétriques.

3.3.2 Modèles à matching sémantique

Les modèles à matching sémantiques utilisent des mesures de similarités pour quantifier la plausibilité des faits en associant les caractéristiques sémantiques latentes contenues dans les vecteurs représentant les entités et les relations.

RESCAL

RESCAL [99] associe chaque entité à un vecteur afin de capturer sa sémantique latente. Les relations quant à elles sont représentées par des matrices qui modélisent les interactions deux à deux des sémantiques latentes de la tête et la queue. Le score est défini par la forme bilinéaire suivante :

$$f_r(h,t) = h^T M_r t = \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} [M_r]_{ij} \dot{[h]}_i \dot{[t]}_j \tag{3.3}$$

avec h,t$\in \mathbb{R}^d$ des vecteurs représentant les entités et M_r une matrice associée à la relation. Ce score capture (grâce à la matrice M_r) toutes les interactions possibles entre les composantes des entités tête et queue.

Le désavantage de RESCAL est l'introduction de $\mathcal{O}(d^2)$ paramètres par relation. C'est pourquoi plusieurs méthodes de matching sont des extensions de RESCAL, améliorant la complexité mais concédant sur l'adaptabilité de la méthode.

Méthode	**Emb. entités**	**Emb. relations**	**Score** $f_r(h,t)$	**Contraintes / Régularisations**
RESCAL [99]	$h,t \in \mathbb{R}^d$	$M_r \in \mathbb{R}^{d\times d}$	$h^\top M_r\, t$	$\|h\|_2 \le 1,\ \|t\|_2 \le 1,\ \|M_r\|_F \le 1$ $M_r = \sum_i \pi_r^i\, u_i\, v_i^\top$
TATEC [39]	$h,t \in \mathbb{R}^d$	$r \in \mathbb{R}^d,\ M_r \in \mathbb{R}^{d\times d}$	$h^\top M_r\, t + h^\top r + t^\top r + h^\top D t$	$\|h\|_2 \le 1,\ \|t\|_2 \le 1,\ \|r\|_2 \le 1,\ \|M_r\|_F \le 1$
DistMult [166]	$h,t \in \mathbb{R}^d$	$r \in \mathbb{R}^d$	$h^\top \mathrm{diag}(r)\, t$	$\|h\|_2 = 1,\ \|t\|_2 = 1,\ \|r\|_2 \le 1$
HoLE [98]	$h,t \in \mathbb{R}^d$	$r \in \mathbb{R}^d$	$r^\top (h \star t)$	$\|h\|_2 \le 1,\ \|t\|_2 \le 1,\ \|r\|_2 \le 1$
ComplEx [144]	$h,t \in \mathbb{C}^d$	$r \in \mathbb{C}^d$	$\Re\left(h^\top \mathrm{diag}(r)\, \bar{t}\right)$	$\|h\|_2 \le 1,\ \|t\|_2 \le 1,\ \|r\|_2 \le 1$
Analogy [78]	$h,t \in \mathbb{R}^d$	$M_r \in \mathbb{R}^{d\times d}$	$h^\top M_r\, t$	$\|h\|_2 \le 1,\ \|t\|_2 \le 1,\ \|M_r\|_F \le 1$ $M_r M_r^\top = M_r^\top M_r,\ M_r M_{r'} = M_{r'} M_r$
SLM [136]	$h,t \in \mathbb{R}^d$	$r \in \mathbb{R}^k,\ M_r^1, M_r^2 \in \mathbb{R}^{k\times d}$	$r^\top \tanh(M_r^1 h + M_r^2 t)$	$\|h\|_2,\ \|t\|_2, \|r\|_2 \le 1$ $\|M_r^1\|_F, \|M_r^2\|_F \le 1$
MLP [32]	$h,t \in \mathbb{R}^d$	$r \in \mathbb{R}^d$	$w^\top \tanh(M^1 h + M^2 r + M^3 t)$	$\|h\|_2 \le 1,\ \|t\|_2 \le 1,\ \|r\|_2 \le 1$

TABLE 3.2 – Comparaison des méthodes d'embedding à matching sémantiques et des fonctions scores associées. Table issue de [152].

3.4 Apprentissage des représentations

Une fois la fonction de score définie, il reste à apprendre les embeddings des entités h et t. Il existe deux hypothèses de travail sur le graphe de connaissance pour l'apprentissage, *l'hypothèse de monde ouvert* et *l'hypothèse de monde fermé*.

3.4.1 Hypothèse de monde ouvert

Sous l'hypothèse de monde ouvert l'ensemble $\mathbb{D}^+$ des faits (h, r, t) observés dans le graphe de connaissance n'est composé que de faits avérés. Et les faits non observés dans le graphe sont faux ou bien vrais mais manquants. Ces faits non observés constituent l'ensemble $\mathbb{D}^-$. Pour l'entraînement, des éléments de $\mathbb{D}^-$ peuvent être construits à partir d'éléments de $\mathbb{D}^+$ de manière heuristique. Deux exemples importants de loss pour l'apprentissage des représentations sont les suivants :

$$\min_{\Theta} \sum_{(h,r,t)\in(\mathbb{D}^+\cup\mathbb{D}^-)} \log(1+\exp(-(\mathbb{1}_{h,r,t\in\mathbb{D}^+} - \mathbb{1}_{h,r,t\in\mathbb{D}^-}) f_r(h,t)) \tag{3.4}$$

$$\min_{\Theta} \sum_{(h,r,t)\in\mathbb{D}^+} \sum_{(h',r',t')\in\mathbb{D}^-} \max(0, \gamma - f_r(h,t) + f_{r'}(h',t')) \tag{3.5}$$

La loss logistique 3.4 performe mieux avec les matching sémantiques, tandis que la loss de classement par paire 3.5 performe mieux sur les modèles à distance translationnelle. De plus la minimisation de cette loss force l'inégalité $f_r(h,t) \geq \gamma + f_{r'}(h',t')$ et donc force le score des faits observés à être plus grand que celui des faits non observés.

Processus d'apprentissage

Pour optimiser les loss, la descente de gradient stochastique avec des minibatchs obtient de bonnes performances. Il est également possible d'utiliser un learning rate adaptatif avec des méthodes de type AdaGrad. L'initialisation de l'embedding peut se faire à partir d'une loi uniforme, gaussienne, d'un modèle d'embedding pré entraîné simple (comme transE) ou encore l'emploi de modèle word2vec classique sur les entités. Les exemples pour l'entraînement provenant de $\mathbb{D}^+$ sont directement tirés de manière aléatoire des faits contenus dans le graphe. L'ensemble $\mathbb{D}^-$ peut être construit de la manière suivante en corrompant aléatoirement des faits appartenant à $\mathbb{D}^+$: $\mathbb{D}^- = \{(h', r, t), h \in E, h' \neq h, (h, t, r) \in \mathbb{D}^+\} \cup \{(h, r, t'), t \in E, t' \neq t, (h, t, r) \in \mathbb{D}^+\} \cup \{(h, r', t), r \in R, r \neq r', (h, t, r) \in \mathbb{D}^+\}$. Malheureusement cette construction ne permet pas d'assurer la disjonction totale des ensembles $\mathbb{D}^+$ et $\mathbb{D}^-$. C'est pourquoi d'autres méthodes plus fines qu'une corruption aléatoire permettent d'amoindrir ce risque.

3.4.2 Hypothèse de monde fermé

Sous l'hypothèse de monde fermé, tous les faits non observés dans le graphe sont considérés faux. Une loss possible pour l'apprentissage des représentations est l'erreur quadratique suivante :

$$\min_{\Theta} \sum_{(h,r,t)\in E\times R\times E} (\mathbb{1}_{(h,r,t)\in\mathbb{D}^+} - f_r(h,t))^2 \tag{3.6}$$

Cependant l'hypothèse de monde fermé est hautement irréaliste, dans la mesure ou la plupart des graphes de connaissances de grande échelle sont incomplets. Les expériences

montrent également de moins bons résultats sous l'hypothèse de monde fermé, par rapport à l'hypothèse de monde ouvert [152].

3.5 Embedding de graphes, au delà des faits

Les graphes de connaissances contiennent bien plus d'informations exploitables que les simples faits (h, r, t) parmi elles on trouve le type des entités, les chemins de relations, les descriptions textuelles, les règles logiques. Nous présentons quelques méthodes pour ajouter ces informations à l'embedding de graphes.

3.5.1 Type des entités

Les types des entités correspondent à des catégories sémantiques liées aux entités d'un graphe. Ils contiennent des informations importantes sur la nature des entités et sont largement utilisés dans les graphes de connaissances. Une manière directe de traiter le type des entités, est de les faire correspondre naturellement à un fait (h, r, t). Par exemple l'entité *Paris* de type *Ville* peut être traitée comme le fait `<Paris, estUn/Une, Ville>`. D'autres méthodes plus élaborées telle que SSE (Semantically smooth embedding) [46] utilisent les types comme critère de proximité entre les données : deux données du même type ont des embeddings plus proches l'un de l'autre que des embeddings d'autres données d'un type différent. Le type des entités peut également être utile en tant que contrainte sur les relations. Par exemple la relation $EstUneCapitaleDe$ impose une entité sujet de type $Ville$ et une entité objet de type $Pays$.

3.5.2 Chemin de relations

Les chemins entre entités contiennent également de nombreuses informations. Ils sont généralement définis comme une suite de relations $r_1 \to \cdots \to r_\ell$ liant une entité sujet h à une entité objet t. Notamment pour la prédiction de relation entre deux entités, il est utile de trouver des chemins les reliant au sein du graphe. La majorité des approches existantes pour traiter les chemins de relations est de les considérer comme des compositions des relations les constituant. Une idée est de considérer un nouveau triplet (h, p, t) où p désigne le chemin de relations $r_1 \to \cdots \to r_\ell$ et où l'on emploie le score $f_p(h, t) = -||h + (r_1 + \ldots r_\ell) - t||$ par analogie au modèle TransE.

L'incorporation des chemins au modèle augmente ses performance, cependant elle est couteuse en temps computationnel dans la mesure où un très grand nombre de chemins constituent le graphe. C'est pourquoi certaines méthodes s'enrichissent uniquement de certains chemins bien choisis.

3.5.3 Description textuelle

Les graphes de connaissances actuels comprennent souvent des descriptions textuelles de leurs entités, riches en information sémantique. Une approche performante [155] est de considérer un modèle joint qui plonge à la fois les entités, les relations et les descriptions textuelles dans le même espace de vecteurs. Ce modèle joint consiste en la fusion de trois modèles : (i) un modèle knowledge, dérivé de TransE et qui traite les entités et relations (ii) un modèle texte dérivé de SkipGram (iii) un modèle d'alignement qui permet de lier dans le même espace les embeddings des entités et des mots. Un des mécanismes pour le modèle d'alignement utilise le lien existant entre le KG Wikidata et Wikipédia. Chaque page wikipédia est associé à une entité du graphe wikidata. Or, chaque lien (anchors)

renvoie à une page wikipédia. Dès lors il est possible d'associer la phrase contenant le lien à l'entité de wikidata, faisant ainsi le lien entre texte et entités. La loss finale du modèle joint prend la forme d'une loss combinée : $\ell = \ell_K + \ell_T + \ell_A$

3.6 Réseaux de neurones profonds pour l'embedding des graphes de connaissances

Les modèles d'embedding de graphes de connaissances classiques, tels que TransE [10] ou DistMult [166], représentent les entités et relations dans un espace vectoriel, en se basant uniquement sur la structure des triplets (h, r, t). Toutefois, ces approches ignorent souvent la structure globale du graphe, les interactions de voisinage, et la diversité des types de relations. Pour dépasser ces limitations, des architectures basées sur les réseaux de neurones à graphes (*Graph Neural Networks*, GNNs) ont été proposées. Nous présentons dans cette section, les architectures GCN [66] et R-GCN [125] efficaces pour affiner les représentations vectorielles des graphes et notamment pour des tâches prédictives de classification des fake news [109].

3.6.1 Des CNNs aux GCNs

Présentation : Les Graph Convolutional Networks (GCNs) [66] sont des réseaux de neurones à plusieurs couches qui propagent l'information d'un nœud du graphe vers ses voisins par un mécanisme de convolution. Cette propagation améliore la représentation contextuelle de l'embedding, car ainsi, les embeddings des nœuds sont interdépendants. Un GCN peut enrichir un embedding préalable d'un graphe (adapter un TransE pour une tâche donnée par exemple), ou bien être la seule méthode d'embedding des nœuds d'un graphe.

Motivation : Les CNNs excellent sur les données à grille régulière, c'est à dire sur des données de type uniforme et localement structurées, typiquement les pixels d'une image. En particulier ils permettent d'extraire de ces données des features multi-échelle bas et haut niveau. Cependant des données comme les fake news issues de réseaux sociaux sont hautement hétérogènes, et présentent souvent une géométrie non euclidienne. Ces données irrégulières et l'efficacité des CNNs ont motivé Defferrard, Bresson et Vandergheynst [26] à extrapoler les capacités des CNNs à des graphes, à travers les GCNs. En effet les graphes ont démontré leur capacité à encoder des données à la géométrie complexe et peuvent être étudiés avec des outils mathématiques puissants comme la théorie spectrale. Ainsi Defferrard, Bresson et Vandergheynst [26] définissent une convolution spectrale sur des graphes, en s'appuyant notamment sur le théorème de convolution [85] qui définit les convolutions comme des opérateurs linéaires qui diagonalisent dans la base de Fourier (representée par les vecteurs propres de l'opérateur laplacien). Cette définition a ouvert la voie au GCN classique de Kipf [66].

Propriétés attendues d'une convolution sur graphe : Généraliser les CNNs à des graphes implique trois étapes fondamentales pour conserver l'analogie : *(i) Conception de filtres localisés* : définir des opérations de convolution qui restent concentrées autour d e chaque nœud (localisation spatiale ou spectrale) *(ii) Coarsening du graphe* : regroupement des sommets similaires en vue d'extraire un graphe réduit qui approxime bien le premier *(iii) Pooling sur graphe* : extraction d'un graphe réduit, composé de moins de nœuds et qui approxime le premier afin d'augmenter la réceptivité des filtres.

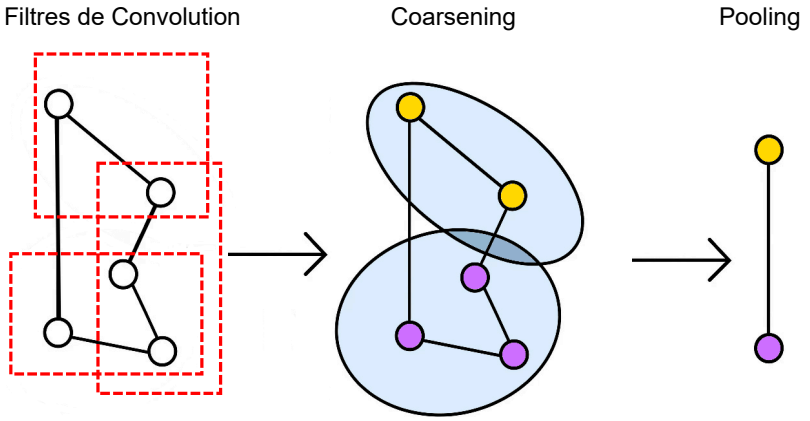


FIGURE 3.4 – Représentation schématique des trois étapes nécessaires à l'établissement d'une architecture GCN, analogue à l'architecture CNN.

Ces trois phases correspondent aux opérations de convolution, coarsening et pooling dans les architectures CNNs classiques. Il est à noter que certains travaux [20] ne font pas de distinction entre coarsening et pooling.

Fondements spectraux

Deux approches sont possibles pour définir des filtres de convolutions sur les graphes, *spatiales* ou *spectrales*. DEFFERRARD, BRESSON et VANDERGHEYNST [26] proposent l'approche spectrale dans la mesure où elle permet de définir sans ambiguïté une translation, et une convolution [85].

Laplacien et décomposition dans une base de Fourier : Soit $G = (V, E, A)$ un graphe non orienté à n nœuds, A la matrice d'adjacence et D celle des degrés ($D_{ii} = \sum_j A_{ij}$). Le Laplacien normalisé symétrique se diagonalise dans une base de Fourier :

$$L = I - D^{-1/2} A D^{-1/2} = U \Lambda U^T \tag{3.7}$$

avec U (vecteurs propres) et $\Lambda = \mathrm{diag}(\lambda_0, \ldots, \lambda_{n-1})$ (valeurs propres). Et où la transformée de Fourier d'un signal $x \in \mathbb{R}^n$ est définie par

$$\hat{x} = U^T x, \quad x = U \hat{x} \tag{3.8}$$

Translation dans le domaine spectral : Ils définissent en outre une translation spectrale via la convolution avec une impulsion de Dirac : pour un nœud pivot i, la translation de x en i vers j s'écrit :

$$(T_i x)(j) = (\delta_i \star_G x)(j) = \sum_{l=0}^{n-1} u_l(i) \hat{x}(\lambda_l) u_l(j) \tag{3.9}$$

où δ_i est le vecteur de Dirac en i et $u_l(k)$ la composante k-ème du l-ième mode de Fourier (coordonnées (l, k) de U).

Filtrage spectral et polynômes de Tchebychev : Pour le graphe G, la convolution spectrale de deux signaux x, y s'écrit

$$x \star_G y = U\Big((U^T x) \odot (U^T y)\Big) \tag{3.10}$$

où $\odot$ désigne le produit de Hadamard. Pour un filtre paramétrique g_θ, on a $y = g_\theta(L)x = U\, g_\theta(\Lambda)\, U^T x$, avec y le signal x filtré par g_θ. Pour s'affranchir de la complexité algorithmique de la diagonalisation du laplacien, on approxime efficacement $g_\theta(\Lambda)$ par un polynôme de Tchebychev d'ordre $K-1$:

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k \tag{3.11}$$

ce qui entraîne

$$y = \sum_{k=0}^{K-1} \theta_k L^k x \tag{3.12}$$

avec $L^k_{ij} = 0$ si la distance $d_G(i,j) > K-1$ (voisinage d'ordre K-1). Ceci permet donc de calculer le signal filtré sans connaissance explicite de U ou Λ.

3.6.2 Le GCN classique

Kipf [66] choisissent $K = 2$, normalisent le laplacien $\tilde{L} = \frac{2}{\lambda_{max}} L - I$ et approximent $\lambda_{max} \approx 2$ de sorte que :

$$g_\theta(\tilde{L}) \star x \approx \theta_0 x + \theta_1 (L - I)x = \theta_0 - \theta_1 D^{-1/2} A D^{-1/2} x \tag{3.13}$$

Et en restreignant le nombre de paramètres avec $\theta_0 = -\theta_1 = \theta$ on obtient :

$$g_\theta(\tilde{L}) \star x \approx \theta(I + D^{-1/2} A D^{-1/2})x = \theta \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} x \tag{3.14}$$

Avec, $\tilde{A} = A + I$.

Ce qui peut se généraliser à un signal $X \in \mathbb{R}^{N \times C}$ avec un vecteur de features de dimension C pour chaque nœud et F filtres :

$$Z = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta, \tag{3.15}$$

D'où l'équation de récurrence suivante, pour le signal en sortie de la $(l+1)$-ème couche de graph-conv :

$$H^{(l+1)} = \sigma(\hat{A}\, H^{(l)} W^{(l)}), \tag{3.16}$$

avec $\hat{A} = \tilde{D}^{-1/2}(A + I)\, \tilde{D}^{-1/2}$, $H^{(0)} = X$, la matrice d'adjacence, $W^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$ est le jeu de poids appris, $\tilde{D}$ la matrice des degrés de $\tilde{A} = A + I$. La fonction d'activation σ (ReLU par défaut), introduit la non-linéarité entre deux couches graph-conv comme dans un CNN.

Interprétation spatiale et apprentissage : Chaque nœud agrège ses propres caractéristiques et seulement celles de ses voisins immédiats (car $K = 2$), multiplie par $W^{(l)}$, puis applique une activation non linéaire. Le modèle typique utilise deux ou trois couches et une fonction loss adaptée à la tâche prédictive.

Limitations des GCN : Le GCN classique est sensible au bruit et reste peu profond. De plus il est invariant aux différents types de relations et a une compréhension contextuelle limitée puisqu'il ne prend en compte dans la représentation de chaque nœud, que les voisins immédiats. Des variantes (GAT, GraphSAGE, R-GCN) visent à renforcer l'expression et la profondeur des GCN.

Conclusion : Le GCN de KIPF [66] allie simplicité et efficacité, et reste une base essentielle pour le deep learning sur graphes. Cependant, bien que les GCN soient efficaces pour des graphes simples, leur expressivité reste limitée dans le cadre de graphes hétérogènes avec des relations typées, comme c'est le cas des graphes de connaissances.

3.6.3 Relational Graph Convolutional Networks (R-GCN)

Les *Relational Graph Convolutional Networks* (R-GCN) [125] sont une extension des GCNS classiques aux graphes hétérogènes, dans lesquels les arêtes sont liées à plusieurs types de relations. Cette propriété est importante dans le cadre des graphes de connaissances, où chaque relation entre entités possède une sémantique propre (par exemple `est`, `vitÀ`, `aFondé`).

La mise à jour des représentations de nœuds à la couche $l+1$ s'effectue comme suit :

$$h_u^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{v\in\mathcal{N}_r(u)}\frac{1}{c_{u,r}}W_r^{(l)}\,h_v^{(l)} + W_0^{(l)}\,h_u^{(l)}\right) \tag{3.17}$$

où $h_u^{(l)}$ est l'embedding du nœud u à la couche l, $\mathcal{R}$ est l'ensemble des types de relations du graphe, $\mathcal{N}_r(u)$ désigne les voisins de u connectés via la relation r, $W_r^{(l)}$ est la matrice de poids associée à la relation r, $W_0^{(l)}$ est un poids de régularisation pour l'état propre de u (self-loop), $c_{u,r}$ est un facteur de normalisation (souvent proportionnel au degré de v pour la relation r), σ est une fonction d'activation non linéaire (ex. ReLU).

Cette architecture permet de capturer les différentes natures de relations entre entités, ce qui est essentiel pour l'encodage sémantique dans les graphes de connaissances. Les R-GCN ont notamment été utilisés avec succès dans des tâches telles que la classification de nœuds, la complétion de triplets, et la détection de faits plausibles dans des environnements multirelationnels.

3.6.4 Pooling de graphes

Une fois les représentations vectorielles des nœuds calculées par un GCN ou un R-GCN ou tout autre modèle d'embedding de graphes (ex. TransE), il est souvent nécessaire d'agréger l'information au niveau du graphe entier (par exemple pour une classification de graphe ou une vérification de faits impliquant plusieurs entités). Cette opération s'effectue à l'aide d'un mécanisme de *pooling* (souvent élémentaire), qui permet de synthétiser quantitativement l'ensemble des embeddings de nœuds en une seule représentation globale.

Pooling simple (moyenne/max) : Les méthodes de pooling les plus simples consistent à agréger les embeddings des nœuds par une opération élémentaire telle que le maximum sur chaque dimension :

$$(h_G)_i = \max_{u\in V}(h_u)_i, \quad \text{pour } i = 1,\ldots,d \tag{3.18}$$

ou la moyenne :

$$(h_G)_i = \frac{1}{|V|}\sum_{u\in V}(h_u)_i, \quad \text{pour } i = 1,\ldots,d. \tag{3.19}$$

Le maximum permet de selectionner les informations saillantes dans le graphe, c'est à dire que peu importe où se trouve l'information elle est sélectionnée si et seulement si elle est maximum, cela permet de détecter des motifs rares mais significatifs. La moyenne permet

de lisser les contributions de chacun des nœuds et donne une vue d'ensemble du graphe, en contrepartie les motifs rares risquent d'être noyés dans du bruit.

Ces méthodes sont rapides et efficaces, et supposent que tous les nœuds contribuent également à la représentation globale. Cependant elles constituent une forte compression du graphe et la perte d'information induite peut être significative. Elles sont largement utilisées dans la littérature [164][48]

Pooling attentionnel : Pour affiner l'agrégation des embeddings des nœuds, on peut utiliser un mécanisme de *graph attention pooling*, qui pondère chaque nœud en fonction de sa contribution :

$$h_G = \sum_{v \in V} \alpha_v \, h_v, \qquad \alpha_v = \frac{\exp(\text{score}(h_v))}{\sum_{u \in V} \exp(\text{score}(h_u))}. \tag{3.20}$$

La fonction score$(\cdot)$ peut être un simple perceptron ou une projection linéaire. Ce mécanisme permet de mettre en valeur les nœuds les plus informatifs pour la tâche cible [19].

Autres Poolings : De nombreux autres poolings plus sophistiqués sont détaillés dans la littérature, nécessitant parfois plusieurs itérations avant de passer d'un graphe à un vecteur interprêtable [77]. Entre chaque itération une méthode de coarsening via message passing est nécessaire (*cf.* Fig 3.4)

Ces différentes approches peuvent être choisies selon le niveau de contrôle et de précision souhaité dans l'agrégation de l'information apprise sur les graphes.

4 Application, LLMs pour la Génération de Graphes de Scènes

Motivation : Dans le cadre de cette thèse consacrée à la détection des fake news, nous avons montré l'importance de représenter explicitement la sémantique des contenus, qu'ils soient textuels ou visuels. Or, les images jouent souvent un rôle crucial dans la désinformation : elles peuvent être manipulées, sorties de leur contexte, ou simplement interprétées de manière ambiguë. Il devient donc essentiel de disposer d'une représentation structurée du contenu visuel, capable de capturer non seulement les objets présents, mais aussi les relations sémantiques qui les unissent. Les graphes de scènes répondent précisément à ce besoin : ils permettent de transformer une image en une structure exploitable pour le raisonnement automatique, facilitant l'analyse du sens véhiculé visuellement, et donc la détection d'incohérences ou d'anomalies caractéristiques des fake news.

Résumé de la contribution : Dans cette contribution, nous introduisons une nouvelle méthode *end-to-end* de génération de graphes de scènes (Scene Graph Generation, SGG) s'appuyant sur les capacités de raisonnement combinées des modèles vision-langage (VLMs), des grands modèles de langage (LLMs) et de techniques d'ancrage visuel. Notre approche consiste tout d'abord à produire, à partir d'une image, une description textuelle riche et contextualisée à l'aide d'un VLM. Cette description est ensuite analysée par un LLM afin d'en extraire des triplets sémantiques de la forme `<sujet, relation, objet>`, qui sont ensuite ancrés visuellement à l'aide de Grounding DINO [80], un modèle de localisation d'objets *open-set* (i.e sur un vocabulaire ouvert, non limitée par des classes prédéfinies) fondé sur des transformers.

Cette pipeline permet de construire automatiquement des graphes de scène structurés, ancrés visuellement et adaptables à différents niveaux de granularité, selon les besoins applicatifs. Nous évaluons notre méthode sur trois jeux de données complémentaires Visual Genome [69], OpenImages [72] et VOC3000 [124] selon deux axes : la qualité de l'ancrage visuel, et la richesse sémantique des graphes générés. Grâce à la richesse lexicale et contextuelle des LLMs, notre méthode permet de générer des relations fines, rares ou implicites, difficilement accessibles aux approches classiques.

Enfin, nous démontrons l'utilité de ces graphes dans des tâches en aval telles que le Visual Question Answering (VQA), illustrant ainsi le potentiel des LLMs comme raisonneurs sur des contenus multimodaux, capables d'en développer des représentations structurées pour une meilleure compréhension et interprétabilité des scènes visuelles.

4.1 Introduction

Un graphe de scène est une représentation structurée d'une image qui encode sous forme de triplets RDF `<sujet, relation, objet>` les objets, leurs attributs et les relations qui les unissent. Il s'agit d'un cas particulier de graphe de connaissances (cf. Chapitre 3) adapté à un contenu visuel spécifique, là où les graphes de connaissances contiennent

des informations globales. Le concept de graphe de scène répond au besoin d'une représentation sémantique riche pour la compréhension des scènes visuelles, en particulier pour accomplir des tâches complexes telles que le Visual Question Answering (VQA) ou l'image captioning.
La génération automatique de graphes de scènes (SGG) est une tâche difficile qui repose généralement sur deux approches : *bottom-up*, où les objets et leur bounding boxes sont prédits avant les relations ; et *top-down*, où les objets et leurs relations sont prédits conjointement [177, 17]. Aujourd'hui, la détection des objets est correctement maitrisée grâce aux architectures CNN-based [63]CNNsurvey. Néanmoins, la prédiction des relations reste un problème difficile, du fait des relations rares (problème de longue-queue de distribution) ou sémantiquement complexes.
Au cours des dernières années, de nombreuses approches ont été proposées pour renforcer la contextualisation des relations : propagation itérative du contexte avec des RNNs et GNNs, intégration de connaissances externes avec des graphes de connaissances ou des connaissances statistiques ou linguistiques. Plus récemment, l'usage des transformers a permis d'affiner la modélisation contextuelle des relations, cependant au détriment de temps de calculs significatifs. Ces approches demeurent aujourd'hui imparfaites et peinent à inférer des relations implicites ou sémantiquement riches. En effet, fournir une véritable compréhension holistique de la scène, impose d'en dépasser les représentations fragmentées, comme simple juxtaposition d'objets reliés entre eux. Cela peut être accompli en renforçant dans les modèles de SGG, l'intégration de connaissances externe contextuelles, conceptuelles ou factuelles. Dans ce contexte, les récents modèles de langage de grande taille (LLMs) apparaissent prometteurs pour mener à bien cette mission. En effet, leur extension multimodale, textuelle et visuelle, les modèles vision-language (VLMs), permet la combinaison du traitement d'images et raisonnement linguistique, ouvrant la voie à de nouvelles méthodes de SGG, capables d'inférer des relations subtiles à partir de descriptions visuelles.

Dans cette contribution nous explorons les capacités des LLMs et VLMs à générer des graphes de scènes. En intégrant leur richesse sémantique et contextuelle nous ambitionnons d'améliorer la compréhension de scènes complexes. Nous proposons une nouvelle méthode de génération de graphes de scène, end-to-end, légère, à la granularité paramétrable et entièrement basée sur les LLMs. Notre méthode s'appuie sur une pipeline en trois étapes : (1) génération d'une description textuelle avec un VLM ; (2) extraction de triplets `<sujet, relation, objet>` avec un LLM ; (3) Ancrage visuel des entités. Afin de répondre à ces ambitions, nos contributions principales sont les suivantes :

(i) Proposition d'une nouvelle pipeline end-to-end de génération de graphes de scènes, simple et modulable, sans fine-tuning
(ii) Une meilleure couverture des relations rares et complexes grâce aux LLMs.
(iii) Une évaluation comparative sur trois datasets (Visual Genome, OpenImages, et VOC3000), ainsi qu'une application à la tâche de VQA sur près de 4000 questions.

4.2 Travaux Connexes

Cette section présente un aperçu des principales approches pour la génération de graphe de scène. Nous discutons les approches classiques fondées sur des architectures CNN, RNN ou GNN, puis les tentatives d'intégration de connaissances a priori dans ces pipelines. Enfin nous discutons d'approches plus récentes tirant parti des larges modèles de langage.

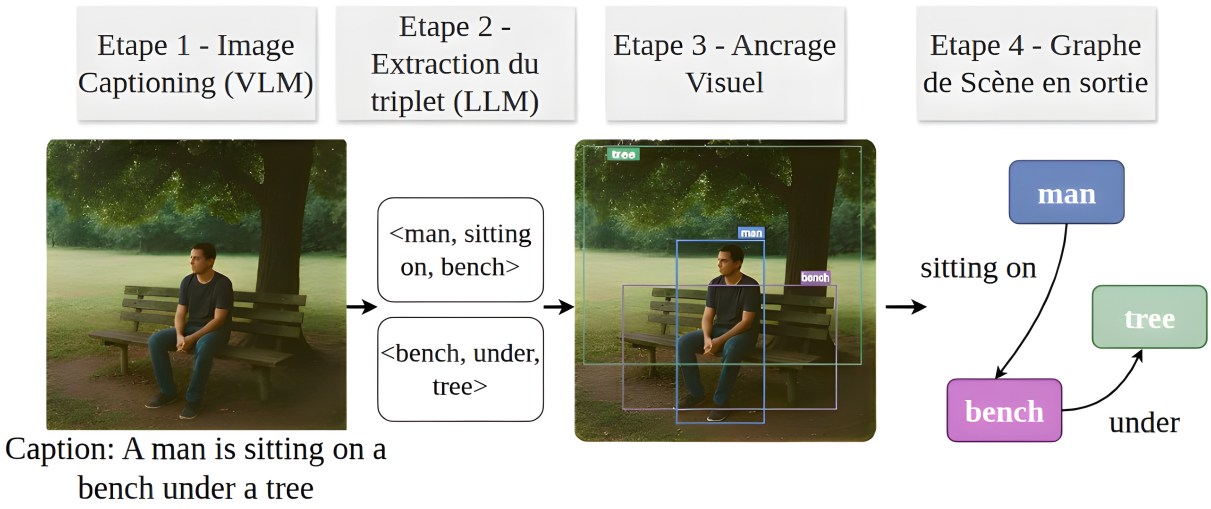


FIGURE 4.1 – Présentation de notre pipeline modulaire basé sur LLM pour la génération de graphes de scènes. Une image est d'abord décrite par un modèle vision-langage (VLM), puis un LLM extrait des triplets `<sujet, relation, objet>`. Ceux-ci sont ensuite, standardisés et ancrés visuellement dans l'image, ce qui produit un graphe de scènes structuré et interprétable.

4.2.1 Génération historique des graphes de scènes

Un graphe de scène représente une image sous forme de triplets `<sujet, predicat, objet >`. Les méthodes de génération de graphe de scènes combinent donc compréhension visuelle (détection et classification des objets) et compréhension sémantique (prédiction des relations). Si la plupart des pipelines de SGG emploient des backbones CNN-based, les approches de prédiction des relations ne font pas consensus, et pour causes, elles présentent chacune des faiblesse. En effet, les méthodes CNN-based sont limitées pour les relations complexes (manque de compréhension contextuelle), les méthodes RNN-based peinent à capturer les longues dépendances contextuelles et sont coûteuses en calculs et quant à elles, les méthodes GNN-based sont sensibles au bruit (propagation du bruit à travers le graphe) et sont également coûteuses en calculs [163].

4.2.2 Intégration de connaissances a priori

Pour améliorer la qualité de ces relations, des approches intégrant des connaissances a priori ont été développées, entre autres des graphes de connaissances [21, 173], graphes conceptuels ou encore des approches probabilistes comme l'usage de conditional random fields (CRF) [139, 23]. Ces approches enrichissent le contexte sémantique, par exemple, un éléphant étant gris, il est plus probable qu'un triplet décrivant sa couleur sur une image soit `<éléphant, est, gris>` plutôt que `<éléphant, est, rose>`.
Plus récemment, l'émergence des modèles de langages de grande taille (LLMs) et modèles de langage visuel (VLMs) ont ouvert de nouvelles perspectives pour la SGG [176]llavas-gleveragingscenegraphs

4.2.3 Apports récents des LLMs et VLMs

Ces dernières années, les LLMs et VLMs entraînés sur d'immenses corpus textuels et multimodaux, et composés de milliards de paramètres, ont démontré leur capacité à raisonner sur du contenu visuel et textuel sémantiquement riche, et à inférer des relations

complexes ou implicites.
Certains travaux s'appuient sur des légendes textuelles générées à partir d'images comme support sémantique pour la SGG. Ainsi Kim et al. [65] proposent une méthode faiblement supervisée qui génère un graphe de scène à partir d'une légende exhaustive.
D'autres approches adoptent une démarche inverse. Plutôt que de générer un graphe de scène à l'aide de LLMs, LLaVA-SG [150], propose d'améliorer les performances de compréhension (par exemple en VQA) d'un VLM (LlaVA) à partir d'un Graphe de scène. Enfin [176] améliore la prédiction de relations avec des LLMs dans le cadre de la tâche de la Panoptic Scene Graph Generation (PSG), voisine de la SGG, dans laquelle les objets sont segmentés en même temps que les relations sont prédites.

4.2.4 Positionnement de notre approche

Notre approche se distingue des travaux précédents par sa simplicité, sa modularité, et son indépendance vis-à-vis de tout entraînement supervisé. Elle repose sur une pipeline en trois étapes : (i) une riche description de l'image est générée par un modèle vision-langage (VLM), (ii) De cette légende, un LLM extrait des triplets `<sujet, relation, objet>`, (iii) Enfin un module de grounding visuel aligne les entités textuelles aux régions de l'image afin d'ancrer visuellement les objets prédits.

Contrairement à LLM4SGG [65], qui exploite une légende fournie pour générer des pseudo-labels servant à entraîner un modèle de SGG classique, notre méthode ne nécessite aucun fine-tuning ni supervision supplémentaire. Elle fonctionne entièrement en inférence, avec des composants pré-entraînés, et génère directement un graphe structuré depuis l'image seule.

Par ailleurs, là où LLaVA-SG [150] adopte une approche inverse injectant un graphe de scène dans un VLM pour renforcer ses performances, nous construisons le graphe comme une sortie interprétable et indépendante, utilisable pour diverses applications en aval (ex. VQA, raisonnement visuel).

Enfin, à la différence de VLPrompt [176], qui repose sur une segmentation panoptique supervisée et un prompting complexe dédié à la tâche PSG, notre pipeline est adaptable à différents niveaux de granularité, et s'appuie sur des modèles open-source (ex. LLaVA, Llama 3, GroundingDino) tirant profit des pré-entraînements des LLMs, et facilitant un usage easy-to-use.

À notre connaissance, il s'agit de la première méthode end-to-end à combiner de manière explicite un VLM, un LLM, et un module de grounding open-set pour générer un graphe de scène complet, structuré, visuellement ancré, et interprétable sans supervision ni réentraînement.

Méthode	**Caption**	**LLM**	**Grounding**	**Training**	**End-to-End**
LLM4SGG [65]	Fourni	✔	Faible	✔	✘
LLaVA-SG [150]	VLM	✘	Implicite	✔	✘
VLPrompt (PSG) [176]	✘	✔	✔	✔	✘
Notre approche	VLM	✔	✔	✘	✔

TABLE 4.1 – Comparaison avec des méthodes récentes LLM/VLM-based de SGG. Notre méthode est la seule entièrement end-to-end de l'image jusqu'au graphe sans entraînement supplémentaire.

4.3 Preliminaires

Dans cette section, nous formalisons brièvement la tâche de génération de graphe de scènes. Un graphe de scène est défini par un ensemble de triplets enrichis de leurs ancrages visuels

$$G(I) = \{(s, r, o, b_s, b_o) \in E \times R \times E \times B^2\} \tag{4.1}$$

où E, R et B désignent respectivement les ensembles d'entités, de relations, et de bounding boxes associées à l'image x, et où b_s et b_o sont les bounding boxes correspondant au sujet s et à l'objet o.
Dans notre approche top-down, la prédiction d'un graphe G à partir d'une image x s'écrit :

$$P(G|x) = P(O, R|x) \cdot P(B|x, O) \tag{4.2}$$

Ainsi, nous prédisons conjointement objets et relations puis ancrons visuellement les entités détectées, avec une étape de grounding.

4.3.1 Notations

- $\hat{G}(x)$: graphe de scène prédit par notre pipeline à partir d'une image x.
- $G^*(x)$: graphe de vérité terrain
- $G_{\text{raw}}(x)$: graphe sémantique brut produit par le LLM à partir de la description d'image.
- $c(x)$: description textuelle générée à partir de l'image x via un VLM.

4.4 Méthodologie

Face aux limites des approches classiques de génération de graphes de scène (SGG), nous proposons une pipeline exploitant les capacités des grands modèles de langages, visuels et textuels. L'objectif est de construire un graphe de scène sans supervision directe à partir de LLMs pré-entraînés. Notre méthode se décompose en quatre étapes : (i) Génération d'une description visuelle de l'image à partir d'un VLM (ii) Extraction de triplets uniquement sémantiques, depuis la description de l'image, avec un LLM (iii) Ancrage visuel des triplets sémantiques (iv) Post-traitement, dépendant de la tâche applicative

Cette architecture est entièrement paramétrable : à chaque étape, le choix du modèle, du prompt, et du niveau de granularité peut-être ajusté afin d'adapter la précision et la cohérence du graphe, ou son niveau de détails selon l'application visée.

4.4.1 Présentation Générale

La génération d'un graphe de scène $\hat{G}(x)$ à partir d'une image x peut se formaliser comme une composition de fonctions, chacune paramétrable en fonction du modèle utilisé, du prompt associé et du niveau de granularité attendu. La forme générale du pipeline est donnée par :

$$\hat{G}(x) = g_{\theta_3} \circ f_{\gamma_p} \circ LLM_{\theta_2} \circ VLM_{\theta_1}(x) \tag{4.3}$$

Chaque composant est défini comme suit :

- $\theta_1 = (M_{VLM}, p_v, \gamma_v)$: paramètre du modèle de description visuelle (VLM), où M_{VLM} désigne le modèle utilisé, p_v le prompt et γ_v la granularité visuelle.

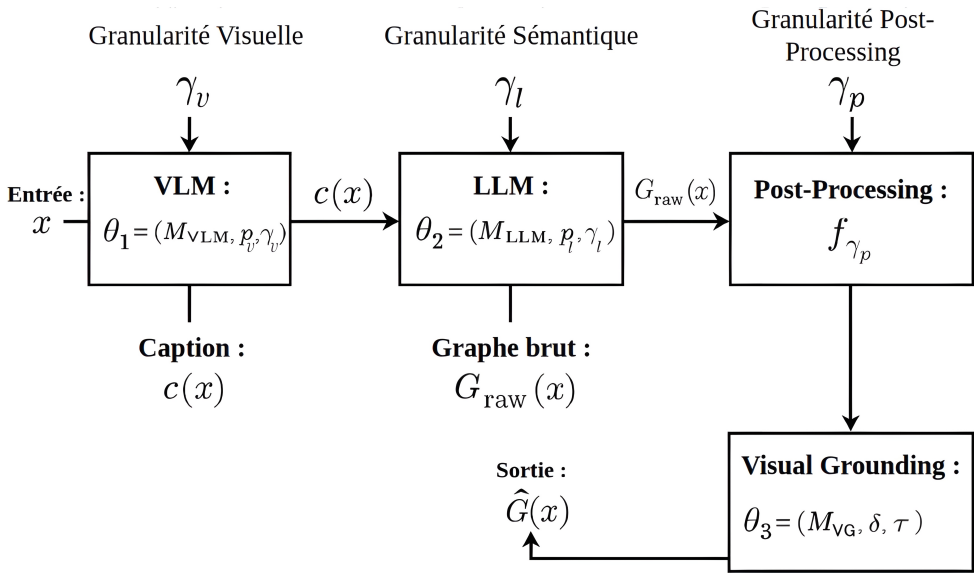


FIGURE 4.2 – Architecture modulaire de notre pipeline de génération de graphe de scène basée sur un LLM. Chaque composante est paramétrée indépendamment. Le modèle vision-langage (VLM) génère une légende $c(x)$ à partir d'une image d'entrée x, en utilisant les paramètres $\theta_1 = (M_{VLM}, p_v, \gamma_v)$, dont γ_v contrôle la granularité visuelle. Le grand modèle de langage (LLM) extrait un graphe de scène brut $G_{raw}(x)$ à partir de cette légende, avec une granularité sémantique γ_l contrôlée par $\theta_2 = (M_{LLM}, p_l, \gamma_l)$. Une fonction de post-traitement f_{γ_p} transforme le graphe brut selon une granularité γ_p, avant une étape finale d'ancrage visuel réalisée par le modèle M_{VG} avec les paramètres δ et τ, produisant le graphe final $\hat{G}(x)$.

- $\theta_2 = (M_{LLM}, p_l, \gamma_l)$: paramètre du modèle d'extraction des triplets (LLM), où M_{LLM} désigne le modèle utilisé, p_l le prompt et γ_l la granularité sémantique.
- f_{γ_p} désigne la fonction de post-traitement du graphe où γ_p est le paramètre de granularité pour le post-traitement, contrôlant notamment le niveau de normalisation des relations appliqué aux triplets
- g_{θ_3} désigne le modèle de grounding visuel des entités où $\theta_3 = (M_{VG}, \delta, \tau)$ est le paramètre du module de grounding visuel (Visual Grounding), et M_{VG} est le modèle utilisé, δ le type de stratégie de matching, et τ un ou plusieurs seuils de confiance.

Le pipeline commence par la génération d'une description textuelle $c(x)$ de l'image, produite par un VLM. Cette légende est ensuite parsée par un LLM pour en extraire un ensemble de triplets $\langle s, r, o \rangle$, formant un graphe sémantique brut $G_{\text{raw}}(x)$. En fonction des besoins applicatifs, ce graphe est ensuite filtré et standardisé par une fonction f_{γ_p} de post-traitement. Enfin, une étape d'ancrage visuel g_{θ_3} est appliquée pour associer les entités à des régions spécifiques de l'image.

Cette architecture permet un contrôle fin de la génération, et rend possible des compromis adaptables entre richesse sémantique, précision visuelle et qualité structurelle du graphe final.

4.4.2 Génération de descriptions visuelles (VLM)

Dans notre approche, la première étape consiste à générer une description textuelle de l'image, décrivant les entités présentes et les interactions entre elles, à l'aide d'un VLM. L'usage d'un VLM permet de représenter le contenu visuel sous forme de phrases bien structurée, facilitant l'extraction des triplets par le LLM dans la suite.
Les VLM s'appuient en général sur des architectures bimodales, qui combinent un en-

codeur d'images (basé sur des Vision Transformers) et un décodeur textuel (souvent un LLM) [73]. Ces modèles sont entraînés en alignement intermodal image et texte, sur de larges corpus multimodaux. Un des défauts de ces modèles est le problème d'hallucinations, le modèle peut générer dans la description textuelle des éléments non présents dans l'image [58]. Ce phénomène est accentué dans les scènes complexes comprenant de nombreuses entités ou bien des entités rares. L'hallucination peut être atténuée en ajustant la granularité des description exigée par le prompt. Une granularité fine enrichit le contenu du caption mais augmente le bruit, tandis qu'à l'inverse une granularité trop grossière peut omettre des éléments signifiants.

4.4.3 Extraction des triplets sémantiques (LLM)

À partir de la description textuelle $c(x)$ produite par le VLM, nous extrayons un ensemble de triplets $\langle s, r, o \rangle$ décrivant les entités et leurs interactions dans l'image. Cette tâche est appélée *parsing sémantique* et consiste à transformer des phrases en représentations formelles, ici, des ensembles de triplets. Les méthodes classiques [5], sont efficaces dans des contextes bien cadrés, avec des textes propres à la structure linguistique stable. Elles se montrent peu adaptées aux descriptions naturelles générées par des modèles VLM, qui sont souvent longues, bruitées, redondantes ou partiellement désorganisées.
C'est pourquoi nous adoptons ici un LLM qui agit d'avantage comme un *raisonneur sémantique* que comme un simple parseur. Grâce au *prompt engineering*, nous contraignons le modèle à produire une sortie au format `JSON`, structurée sous forme d'une liste de triplets $< sujet, relation, objet >$. Ce choix permet une extraction souple, faiblement supervisée, et directement exploitable pour construire un graphe de scène.
Nous utilisons le framework Ollama [141] pour tester localement différents LLMs avec un nombre de paramètres inférieur à 10B, tels que *Llama3*, *Mistral* ou *Deepseek*. Nous ajustons chacun avec différentes granularités d'extraction contrôlées par le prompt. Le choix du modèle influence fortement la précision des triplets extraits (*cf.* 4.2). Afin de traiter des descriptions textuelles longues, nous appliquons un découpage en fragments avec chevauchement [133], en définissant une taille de fragment de 80 caractères et un chevauchement de 40 caractères. Cette approche permet d'éviter la perte de relations se situant à la frontière entre deux phrases.
Cette méthode permet de produire un graphe brut $G_{\text{raw}}(x)$, qui sera ensuite traité pour ancrage visuel et/ou simplification selon la tâche ciblée.

4.4.4 Ancrage visuel des entités

Une fois les triplets $\langle s, r, o \rangle$ extraits, nous procédons à leur ancrage visuel, c'est-à-dire à l'association des entités s et o à des régions spécifiques de l'image x, délimitées par des bounding box. Cette étape permet d'aligner la structure sémantique du graphe avec les régions visuelles pertinentes, ce qui est essentiel pour certaines tâches applicatives telles que le VQA spatial ou le raisonnement visuel.

Pour réaliser l'ancrage visuel des entités, nous utilisons le modèle GroundingDINO [80], conçu pour détecter des régions d'image pertinentes à partir d'une requête textuelle. Pour chaque entité d'un triplet (sujet ou objet), nous utilisons sa forme textuelle comme requête, et le modèle prédit un ensemble de boîtes candidates, chacune associée à un score de correspondance appris de manière multimodale. L'avantage d'un modèle comme GroundingDINO est sa capacité à ancrer visuellement des entités en vocabulaire ouvert, contrairement à des modèles tels que YOLO [118] par exemple. Ainsi, GroundingDINO

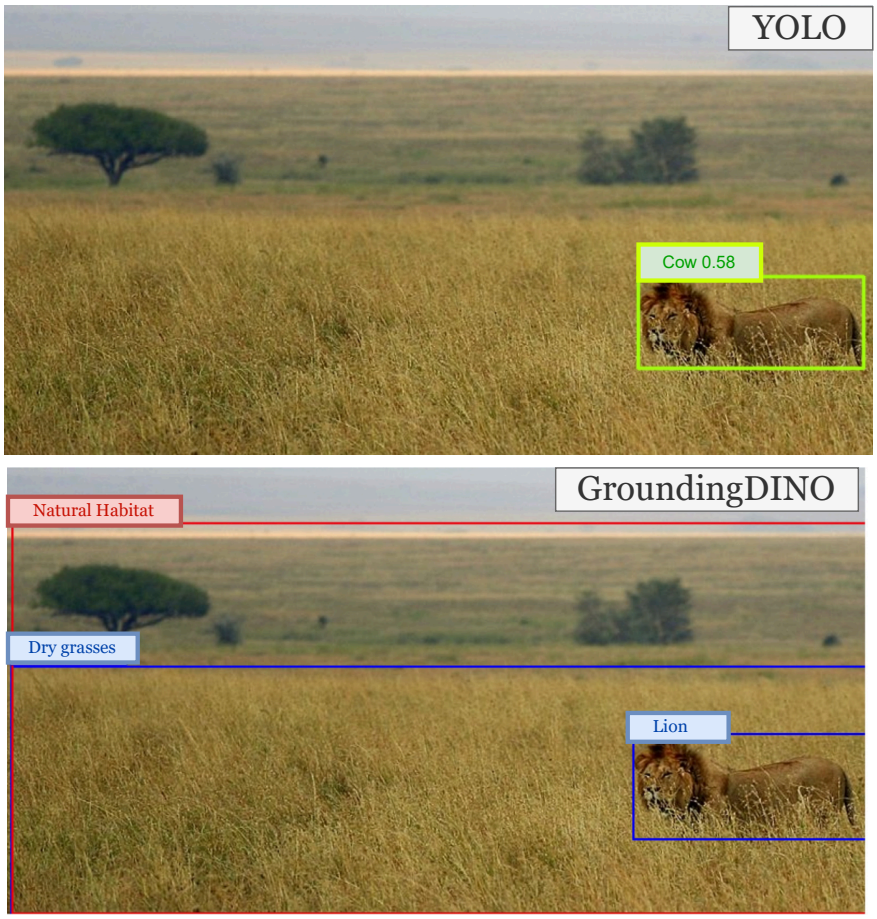


FIGURE 4.3 – Exemple de détection performée par YOLO et GroundingDINO sur une image du dataset OpenImages [72]. YOLO détecte correctement la box du lion, mais échoue sur l'étiquette, imposant un matching sémantique après la détection. Au contraire, GroundingDINO permet de partir des entités extraites par le LLM et de les détecter directement dans l'image, améliorant la richesse et les performances de la détection

peut tirer pleinement parti du vocabulaire étendu des LLM, chacune des entités prédites par le LLM peut être ancrée. En effet, un modèle tel que YOLO impose de détecter d'abord les bounding box des entités, puis dans un second temps de faire un matching sémantique entre les entités extraites par le LLM et les étiquettes des entités détectées par YOLO, ce qui est imprécis et coûteux.

Nous conservons uniquement la boîte ayant le meilleur score :
(stratégie $\delta = \underset{bbox}{\text{argmax}}(\text{scores})$), si celui-ci dépasse un seuil de confiance τ. Les coordonnées de la box obtenue sont ensuite ajoutées au triplet initial, produisant un graphe enrichi $G_{\text{grounded}}(x) = g_{\theta_3}(G_{\text{raw}}(x))$. Cette stratégie d'ancrage permet de relier les entités extraites automatiquement à leurs occurrences visuelles dans l'image, sans supervision explicite, tout en garantissant une cohérence avec l'espace sémantique des entités textuelles.

Notons que bien que l'ancrage visuel soit toujours effectué dans notre pipeline, il n'est pas nécessairement exploité dans toutes les tâches : pour certaines applications telles que la génération textuelle à partir du graphe, l'information de position peut être ignorée sans perte majeure.

4.4.5 Post-traitement du graphe

La dernière étape de notre pipeline consiste à filtrer, nettoyer et, si besoin, normaliser les triplets extraits afin d'obtenir un graphe de scène exploitable. Cette phase est contrôlée par un paramètre de granularité noté γ_p, qui détermine le niveau de transformation appliqué au graphe brut. Nous appliquons d'abord un nettoyage minimal : suppression des triplets incomplets et fusion des entités textuellement proches à l'aide d'une mesure de similarité sémantique (nous utilisons BERTScore).

La granularité γ_p intervient notamment dans la gestion des relations. Par défaut, les LLMs ont tendance à produire des relations verbales riches, telles que `< Tasse, est posée sur le coin gauche, Table>`. Ce niveau de détail peut être problématique dans certaines tâches, par exemple dans le VQA basé sur des systèmes à choix fermés, où une relation plus simple comme `'ON'` peut faciliter l'alignement avec les réponses attendues. Dans ce cas, nous proposons une standardisation des relations via un alignement sémantique basé sur BERTScore avec un ensemble de relations de référence, telles que celles du dataset Visual Genome (`'ON'`, `'IS'` etc.).

Cependant, dans notre propre tâche de VQA, l'évaluation étant elle-même effectuée à l'aide d'un LLM, nous faisons le choix de conserver les triplets dans leur forme complète. Le LLM évaluateur étant capable de raisonner sur des structures sémantiques complexes, cette décision permet de préserver la richesse expressive du graphe sans perte de performance.

Le graphe final $\hat{G}(x)$, produit à l'issue de cette étape, est ainsi directement exploitable pour des tâches aval telles que le raisonnement, la génération de réponse ou l'évaluation par comparaison à un graphe de référence $G^*(x)$.

4.5 Experiments

Dans cette section, nous évaluons notre approche de génération de graphes de scène basée sur les LLMs, selon différents scénarios. L'objectif est double : d'une part, mesurer la qualité sémantique des triplets extraits ; d'autre part, évaluer l'utilité de ces graphes dans des tâches de détection visuelle et de question-réponse visuelle (VQA). Nous comparons plusieurs variantes de notre pipeline, en faisant varier les modèles (LLM, VLM), la granularité des descriptions générées, ainsi que les modalités d'ancrage spatial. Nos expériences sont conçues pour explorer la robustesse, la généricité et les limites de notre méthode dans des contextes visuels variés.

4.5.1 Datasets

Nous menons nos expériences sur trois jeux de données visuelles aux profils complémentaires :

Visual Genome [69] : Un dataset de référence pour la génération de graphes de scène, contenant plus de 100k images annotées avec objets, relations et descriptions. Nous utilisons un sous-ensemble de 100 images.

OpenImages V6 [72] : Un dataset large et hétérogène, avec annotations d'objets et de relations. Pour garantir une densité relationnelle suffisante, nous sélectionnons aléatoirement 100 images du split "train" dont les annotations ground truth contiennent au moins 5 triplets. La sélection est effectuée de manière uniforme selon le nombre de triplets, afin de couvrir différents niveaux de complexité.



FIGURE 4.4 – Image issue de Visual Genome [69]. Dans notre graphe de scène généré le triplet `<man, uses, phone>` est prédit. En effet l'homme habillé en gris fait un geste de la main trompeur, interprétable comme un appel téléphonique même pour un humain. Ce genre de relation ambiguë et subtile est difficile à prédire.

VOC3000 [124] : Un dataset basé sur PASCAL VOC, annoté pour les objets et leurs relations, centré sur des *visual phrases* comme *person riding horse*. Il permet d'évaluer la capacité des modèles à capturer des interactions visuelles typiques mais compactes, où les objets sont fortement dépendants de leur contexte relationnel. Nous sélectionnons aléatoirement 100 images qui contiennent au moins 3 triplets.

Chacun de ces jeux est utilisé dans différentes expériences, soit pour évaluer la qualité sémantique des triplets générés, soit pour mesurer les performances d'ancrage visuel via la tâche SGDet.

4.5.2 Protocoles expérimentaux

Nous évaluons notre approche selon trois expériences distinctes. Tout d'abord, nous mesurons la qualité des graphes sémantiques bruts G_{raw} (sans ancrage visuel) sur le dataset Visual Genome. Pour cela, nous faisons varier à la fois le modèle (LLM, VLM) et les niveaux de granularité γ_v, γ_l, afin d'analyser leur impact sur la qualité des triplets extraits. Ensuite, nous évaluons les graphes complets $\hat{G}(x)$, incluant l'ancrage visuel des entités, notamment à l'aide de la métrique `sgdet` (scene graph detection) qui mesure le nombre de triplets de la vérité terrain, correctement prédits, visuellement et sémantiquement. Cette évaluation est menée sur trois datasets : Visual Genome, OpenImages [72], et VOC3000 [124].

Enfin, nous testons l'utilisation des graphes dans un contexte de Visual Question Answering (VQA). Nous nous appuyons sur un ensemble de 4000 questions réparties sur 50 images issues de Visual Genome, et comparons les réponses générées par un LLM à partir des graphes extraits, selon différentes variantes de notre pipeline.

Toutes les expériences sont réalisées en local sur une machine équipée d'un GPU nvidia RTX4060.

4.5.3 Evaluation du Graphe de Scène Sémantique

Dans cette première expérience, nous évaluons la capacité de notre pipeline à produire des graphes sémantiques cohérents à partir de simples descriptions textuelles d'images, sans ancrage visuel. L'objectif est d'analyser l'impact combiné du VLM (image captioning) et du LLM (extraction de triplets) sur la qualité finale du graphe G_{raw}.

Setup expérimental : Chaque image est d'abord décrite à l'aide d'un VLM selon deux niveaux de granularité $\gamma_v \in \{\text{low}, \text{high}\}$, contrôlés par le prompt : le niveau *high* encourage une description exhaustive (spatialité, objets, relations), tandis que le niveau *low* se concentre sur les éléments essentiels. Deux VLMs sont testés : LLaVA 7B [79] et Llama3.2-Vision 11B [1]. LLaVA s'avère globalement plus précis et informatif dans nos tests.

Les descriptions sont ensuite analysées par un LLM pour extraire des triplets de la forme `<sujet, relation, objet>`. Nous évaluons trois modèles : Qwen2.5 (0.5B) [165], Zephyr (7B) [55], et Llama3 (8B) [92]. Deux niveaux de granularité $\gamma_l \in \{\text{low}, \text{high}\}$ sont testés côté LLM : le niveau *high* favorise une extraction exhaustive et détaillée (au risque d'hallucination), tandis que le niveau *low* produit des triplets plus synthétiques.

Évaluation : La qualité des triplets extraits est évaluée par appariement avec les triplets de la vérité terrain. Pour$x \in X$ l'ensemble des 100 images de Visual Genome testées, on définit une précision et un recall moyen.

$$\text{Precision} = \frac{1}{|X|} \sum_{x \in X} \frac{|\{t \in G_{\text{raw}}(x) \mid \exists\, t' \in G^*(x),\, \text{sim}(t, t') > \tau_p\}|}{|G_{\text{raw}}(x)|} \tag{4.4}$$

$$\text{Recall} = \frac{1}{|X|} \sum_{x \in X} \frac{|\{t' \in G^*(x) \mid \exists\, t \in G_{\text{raw}}(x),\, \text{sim}(t, t') > \tau_r\}|}{|G^*(x)|} \tag{4.5}$$

où $\text{sim}(\cdot, \cdot)$ est une similarité BERTScore moyenne (sur les trois composants du triplet), avec $\tau_p = 0.8$ et $\tau_r = 0.75$.

Résultats : Les résultats sont présentés dans le tableau 4.2. On observe une amélioration marquée des scores avec l'augmentation de la taille du LLM. Le modèle Qwen2.5 (0.5B) échoue régulièrement à structurer ses réponses et produit parfois des sorties erronées (langue incorrecte, absence de format JSON).

Le meilleur score est obtenu pour la configuration LLaVA (high), Llama3 (low), avec un F1-score de 0.78. Cette combinaison maximise l'informativité de la description tout en limitant le bruit à l'extraction. À l'inverse, un excès de granularité dans les deux composantes ($\gamma_v = \text{high}$, $\gamma_l = \text{high}$) tend à introduire des redondances et relations secondaires, nuisant au F1-score.

4.5.4 Evaluation du Graphe ancré visuellement

Dans cette expérience, nous évaluons la qualité des graphes complets $\hat{G}(x)$, dans lesquels chaque entité est associée à une boîte englobante détectée automatiquement. L'objectif est d'estimer dans quelle mesure notre pipeline est capable de générer des triplets corrects à la fois sémantiquement (entités + relation) et spatialement (localisation des objets).

Granularité		Qwen2.5 0.5B			Zephyr 7B			Llama3 8B		
VLM	LLM	P	R	F1	P	R	F1	P	R	F1
Low	Low	0.14	0.10	0.10	0.50	0.64	0.54	**0.63**	**0.67**	**0.63**
Low	High	0.17	0.08	0.09	0.40	0.61	0.46	**0.49**	**0.65**	**0.53**
High	High	0.42	0.36	0.33	0.44	0.84	0.56	**0.55**	**0.85**	**0.65**
High	Low	0.32	0.38	0.30	0.57	0.85	0.67	**0.72**	**0.87**	**0.78**

TABLE 4.2 – Evaluation sur Visual Genome pour différentes granularités des LLM et modèles, avec LLaVA comme VLM. Métriques : Precision (P), Recall (R), et F1-score (similarité BERT-based ; thresholds : 0.8 pour P, 0.75 pour R). Meilleurs scores en gras.

Protocole d'évaluation : Nous utilisons la métrique SGDet (Scene Graph Detection), qui évalue la proportion de triplets $\langle subject, relation, object \rangle$ correctement prédits, en considérant qu'un triplet est correctement prédit si :

- les labels sémantiques de $\langle subject, relation, object \rangle$ sont similaires à ceux du triplet de référence (similarité sémantique BERTScore). Où les relations prédites ont été classifiées parmi un ensemble restreint de relations de références, avec un matching sémantique.
- les boîtes prédictes pour *subject* et *object* ont une IoU ≥ 0.3 avec celles du triplet groundtruth.

Nous évaluons cette métrique sur trois jeux de données : Visual Genome, OpenImages et VOC3000.

Choix du modèle d'extraction : Nous avons également testé un LLM alternatif, `deepseek-r1-8B`, orienté raisonnement, pour extraire les triplets. Malgré un léger gain sur certaines métriques internes, il ne dépasse pas Llama3 en SGDet et augmente considérablement le temps de traitement par image (jusqu'à $\times 5$ selon les scènes). Pour des raisons d'efficacité, nous ne retenons donc pas ce modèle dans la suite. La combinaison LlaVA pour le VLM, et Llama pour le LLM étant celle qui fonctionne le mieux.

Résultats : Les résultats sont présentés dans le tableau 4.3. Sur Visual Genome, nous atteignons un score SGDet de 9.4%, ce qui reste modeste au regard des méthodes supervisées traditionnelles (souvent entre 15 et 30% selon les configurations). Ce score reste néanmoins intéressant, dans la mesure où notre approche ne repose ni sur de l'apprentissage, ni sur un vocabulaire restreint, et que la détection repose uniquement sur une similarité de concepts (et non sur des classes pré-apprises).
Sur VOC3000, les résultats sont meilleurs, avec un SGDet de 15.6%. Cela peut s'expliquer par la nature du dataset, centré sur des scènes plus simples (moins de 7 triplets sur les données groundtruth), avec moins d'objets simultanés, mais des relations bien définies.

En revanche, les résultats sur OpenImages sont nettement plus faibles. Cette baisse s'explique principalement par deux facteurs : Les images sémantiquement riches du dataset contiennent souvent des scènes de foule, avec de nombreux objets similaires (par ex. plusieurs personnes, véhicules, etc.), rendant l'ancrage visuel très difficile. De plus, certaines images sont mal annotées, par exemple des images très riches sémantiquement (example, un port de pêche) ne contiennent dans le groundtruth que 2 ou 3 triplets ce qui diminue la cohérence entre les triplets prédits et les annotations de référence. Il serait intéressant d'examiner plus en détails les annotations du dataset.

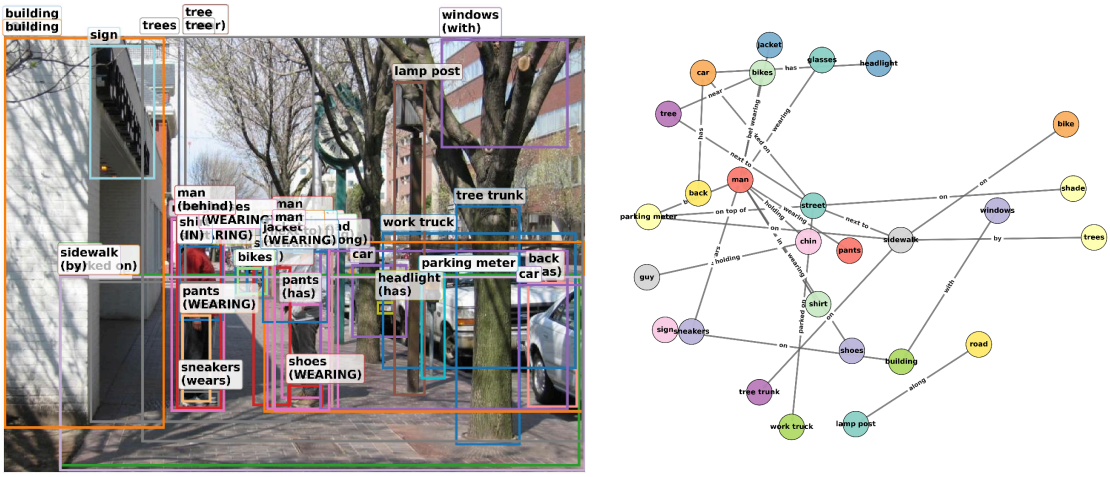


FIGURE 4.5 – Exemple de graphe de scène généré sur une image de Visual Genome. A gauche l'image, les bounding boxes et relations associées, à droite une représentation sous forme de graphe.

Comparaison avec l'état de l'art : Des travaux récents comme ceux de YU et al. [170] ou de KIM et al. [65] avec LLM4SGG ont obtenu des performances intéressantes sur la tâche SGDet en utilisant des approches faiblement supervisées. LLM4SGG attteint par exemple des scores de $R@50 = 18.4\%$ et $mR@100 = 6.26\%$ sur le dataset Visual Genome. Quant à euxYU et al. [170] obtiennent également un $mR@100 = 6.2\%$ sur Visual Genome. Le Recall@50 (R@50) mesure la proportion de triplets correctement prédits parmi les 50 plus confiants générés par le modèle, en comparant aux triplets de vérité terrain. Il favorise les relations fréquentes. À l'inverse, le mean Recall@100 (mR@100) calcule le rappel moyen par classe de relation, ce qui donne un poids égal aux classes rares et fréquentes, et permet d'évaluer la capacité du modèle à généraliser sur l'ensemble des relations, y compris les moins représentées.

Même si la comparaison numérique directe avec notre méthode n'est pas possible, ces résultats montrent que notre méthode permet de générer des graphes riches et pertinents sans supervision et ouvre des perspectives solides pour la SGG zéro-shot.

Dataset	**Model**	**SGDet**
VOC3000	LLaVA + LLaMA3	0.130
	LLaVA + DeepSeek	**0.156**
	LLaMA3.2-Vision + LLaMA3	0.153
OpenImages	LLaVA + LLaMA3	0.020
	LLaVA + DeepSeek	0.007
	LLaMA3.2-Vision + LLaMA3	**0.024**
Visual Genome	LLaVA + LLaMA3	**0.094**

TABLE 4.3 – Scene Graph Detection (SGDet) scores across datasets. A predicted triplet is considered correct if both its spatial localization and semantic similarity (subject, predicate, object) match a ground truth triplet. Best results per dataset are in bold.

4.5.5 Utilisation des graphes pour le Visual Question Answering (VQA)

Cette expérience vise à évaluer si les graphes générés par notre méthode contiennent une information sémantique suffisamment riche et précise pour pouvoir répondre correctement à des questions en langage naturel. Plutôt que d'utiliser un classificateur dédié au VQA, nous faisons l'hypothèse que les LLMs, en tant que raisonneurs puissants, sont capables d'exploiter ces graphes comme contexte structuré pour produire des réponses pertinentes. L'objectif est donc de mesurer l'utilité effective du graphe comme support de raisonnement, à travers les performances obtenues sur une tâche de question-réponse visuelle.

Protocole : Nous utilisons un ensemble de 4000 paires `(question, réponse)` issues de Visual Genome, réparties sur 50 images distinctes. Pour chaque image x, nous disposons d'un graphe prédit $\hat{G}(x)$, et ancré visuellement. Contrairement à l'évaluation SGDet, les relations du graphe ne sont pas normalisées ni appariées à un vocabulaire de référence : nous conservons des relations riches et verboses, telles qu'extraites par le LLM.

Nous comparons différentes configurations de génération de graphe (VLM, LLM, granularités) en observant leur capacité à supporter une tâche de question-réponse. Pour chaque question associée à une image, nous construisons une prédiction de réponse en interrogeant le graphe via un prompt conçu pour un LLM.

Ce prompt contient : (i) une consigne indiquant que le modèle doit répondre à la question en s'appuyant uniquement sur le contenu du graphe ; (ii) une incitation explicite à répondre "I don't know" si l'information n'est pas disponible ; (iii) un contexte extrait du graphe, sous forme textuelle structurée. Ce contexte correspond à un sous-graphe sélectionné dynamiquement : nous extrayons d'abord les entités clés de la question, puis nous collectons les triplets de $\hat{G}(x)$ contenant des entités similaires. Par exemple, pour la question "What is the color of the clock ?", les entités extraites seraient *color* et *clock*, et les triplets du contexte incluraient des éléments comme `(tshirt, is, red)`, `(clock, displays, 10pm)` ou `(clock, is, grey)`. Si aucun triplet n'est jugé pertinent, l'ensemble du graphe est fourni.

Ce mécanisme vise à réduire la charge cognitive du LLM en lui fournissant un contexte ciblé, facilitant ainsi l'accès à l'information pertinente lorsque celle-ci est présente dans le graphe. Le modèle de langage utilisé pour produire la réponse est LLaMA3.

Évaluation : Nous comparons la réponse prédite au texte de la réponse de référence en mesurant la similarité de type SBERT (cosine similarity entre embeddings). Trois métriques sont reportées :

- **Accuracy@0.5** : proportion de réponses avec une similarité > 0.5.
- **Score moyen** : moyenne des similarités SBERT sur l'ensemble des questions.
- **% de réponses "I don't know"** : indicateur de non-réponse ou d'échec du raisonnement.

Résultats : La figure 4.6 illustre les performances de l'évaluation VQA en fonction du modèle de VLM choisi et des granularité du LLM et VLM. En fait, le paramètre qui a le plus d'incidence est la granularité du LLM. Une granularité fine induit près d'un quart de bonne réponses (23% pour l'accuracy@0.5). A l'inverse, la granularité grossière du LLM induit une forte augmentation de l'incapacité du LLM à trouver à la réponse aux questions (62% de 'I don't know').

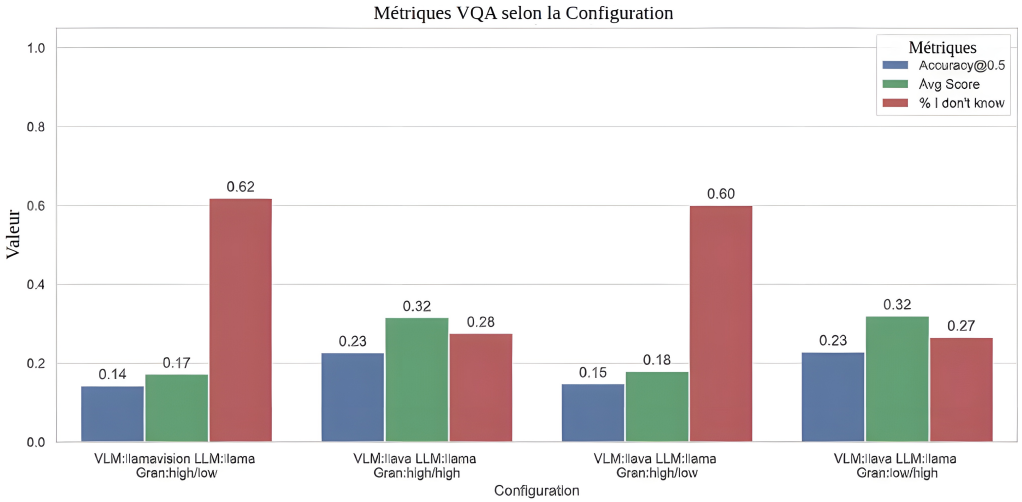


FIGURE 4.6 – Résultats VQA pour différentes configurations de génération de graphes. Lorsque la granularité du LLM est faible, le pourcentage d'incertitude augmente significativement. La granularité du VLM a peu d'effet sur la tâche de VQA.

Distribution des scores. La figure 4.7 illustre la distribution du score de similarité entre la réponse prédite et la réponse attendue. Les meilleurs résultats sont obtenus lorsque la granularité du LLM extracteur de triplet est fine, et la granularité du VLM n'a que peu d'incidence sur cette distribution.

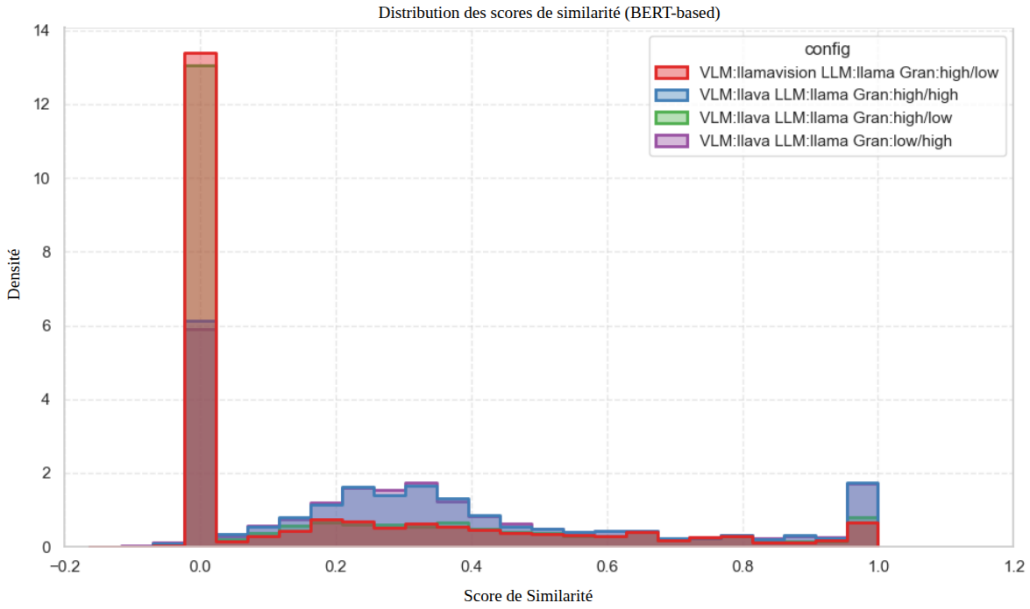


FIGURE 4.7 – Distribution des scores de similarité SBERT sur les réponses VQA. Les scores faibles ont une plus forte occurence lorsque la granularité du LLM est faible.

Ces résultats indiquent que les graphes générés par notre pipeline véhiculent une information sémantique suffisamment riche pour permettre de répondre correctement à une part significative (~25%) des questions visuelles de type *commonsense*. Il est important de souligner que certaines de ces questions sont ambiguës et peuvent admettre plusieurs réponses valides. Par exemple, à la question *"What wears the man ?"*, des réponses comme *"beard"*, *"shoes"* ou *"glasses"* peuvent toutes être considérées comme acceptables.

Plusieurs pistes peuvent être envisagées pour améliorer ces performances. Cela inclut l'enrichissement des graphes afin d'en accroître l'exhaustivité, une meilleure sélection des

sous-graphes contextuels, ainsi que l'exploration d'autres modèles de raisonnement. Par ailleurs, les résultats obtenus suggèrent que le LLM utilisé comme raisonneur nécessite un niveau de sémantique contextuelle suffisamment élevé pour inférer des réponses précises.

4.6 Discussion

Nos expériences montrent qu'une pipeline de génération de graphe de scène (SGG) de bout en bout fondée sur des LLM est non seulement faisable, mais également prometteuse. La flexibilité du *prompting*, la capacité à raisonner sur des relations complexes, ainsi que l'injection de connaissances a priori via la maitrise du langage, en font une alternative crédible aux approches SGG classiques centrées vision.

Cette approche présente toutefois des limites :

(i) Ancrage visuel. L'ancrage spatial précis des entités demeure difficile, en particulier lorsque l'image contient *plusieurs objets visuellement similaires*. Sans détection explicite, la localisation des triplets peut devenir ambiguë et conduire à des erreurs d'assignation région-entité. Dans le cadre de la thèse, cela impacte directement la vérification image-texte : une entité mal ancrée fragilise toute vérification de cohérence avec l'énoncé textuel.

(ii) Sélection des relations pertinentes. La sélection de relations utiles pour des tâches aval (ex. VQA) n'est pas triviale. Les LLM extraient un large éventail d'interactions, mais toutes ne sont pas informatives. Pour la détection de fake news, il faut prioriser les relations *factuelles* (qui, quoi, où, quand, interactions causales/spatio-temporelles) et réduire le bruit des relations, par exemple en restreignant leur nombre afin d'alimenter efficacement un graphe hétérogène de connaissances et des modules éventuels de contrôle de cohérence.

(iii) Coût computationnel. Le coût croît avec la taille du LLM. Les modèles plus puissants produisent des graphes plus structurés et cohérents, au prix d'une inférence plus lente et d'exigences matérielles plus élevées. Sur une *NVIDIA RTX 4060*, le temps moyen d'inférence pour générer un graphe complet à partir d'une image est d'environ 7-8 minutes, ce qui reste prohibitif pour le temps réel ou le traitement massif.

(iv) Hallucination. Les LLM ont tendance à sur-généraliser ou à déduire des relations absentes de l'image. Ce risque augmente avec des descriptions fines ou des prompts à forte granularité, et peut conduire à des triplets plausibles mais erronés, notamment dans des scènes ambiguës. Dans notre objectif de détection de fake news, ces hallucinations constituent une source de faux positifs et doivent être quantifiées et régulées autant que faire se peut.

Lien avec l'objectif de détection de fake news multimodales. La SGG produit une représentation sémantique structurée (objets, attributs, relations) qui sert de couche d'interface entre l'image et la news, et les connaissances externes (événements, lieux, calendriers). Cette structuration facilite (1) le *claim-image checking* (i.e la vérification que l'information textuelle est bien soutenue par l'image qui l'accompagne), (2) la détection d'usage hors contexte (mismatch lieu/date/identité), et (3) l'enrichissement d'un graphe hétérogène global. En outre, cette représentation de l'image sous forme de graphe est complémentaire des signaux visuels bas niveau (ex. traces forensiques, artefacts visuels) déjà montrés utiles pour la détection visuelle de fake news et bénéfiques en fusion multimodale.

En résumé, notre méthode LLM-based offre un fort pouvoir de généralisation et de richesse sémantique, mais doit gagner en efficacité, en robustesse dans les scènes complexes, et en *task-alignment* pour servir de brique fiable à la détection de fake news multimodales.

4.7 Conclusion et Perspectives

Nous avons présenté une pipeline SGG entièrement modulaire, de bout en bout, reposant sur des LLM : un modèle vision-langage pour le captioning, suivi d'un LLM d'extraction structurée en triplets. Nous montrons qu'il est possible de générer des graphes de scène riches et cohérents sans recourir à des détecteurs classiques.

Nos résultats valident la viabilité de cette approche et mettent en lumière plusieurs chantiers d'amélioration : choix du VLM/LLM, design de prompts, contrôle de l'hallucination, et compromis performance-coût. Bien que la solution soit flexible et facilement déployable, des modèles *fine-tuned* et/ou distillés pourraient améliorer la précision (au prix de l'accessibilité).

5 Clustering Spectral et Clustering SNMF

Ce chapitre présente deux types de clustering de graphe, que nous appliquerons dans la suite sur des *graphes de similarités forensiques* afin de localiser des transformations locales d'images (copy-move, splicing, flou partiel, etc.).

5.1 Clustering Spectral

Nous tentons dans cette section d'expliciter les ressorts théoriques du clustering spectral, et d'en fournir une explication intuitive, en nous appuyant sur les travaux de Luxburg [84]. Le clustering spectral est une méthode de partitionnement de données, simple à implémenter et qui montre souvent de meilleures performances que d'autres algorithmes de clustering classiques (k-means, EM, ...). Il s'appuie sur l'analyse spectrale d'un graphe $G(V, E, W)$ représentant les données et leurs similarités. Et, précisément, dans le cadre de cette thèse, nous disposerons d'une représentation des images sous forme de graphes de similarités permettant de comparer, au regard des transformations subies, des zones de l'image entre elles.

5.1.1 Graphes de similarité

Soit $(x_i)_{i=1}^n$ un ensemble de données. On cherche à partitionner ces données à dessein, pour une application choisie. Pour ce faire on dispose d'un critère de *proximité* entre les données, quantifié par une mesure de similarité $w(x_i, x_j) = w(x_j, x_i) \in [0, 1]$. On représente alors les données par un graphe non orienté et pondéré :

$$G = (V, E, W), \quad |V| = n,$$

où $V = \{v_i\}$, $E \subset V \times V$, et $W = (w_{ij}) \in \mathbb{R}_+^{n \times n}$ est la matrice de poids/similarités (symétrique, $w_{ij} = w_{ji} \geq 0$, et $w_{ii} = 0$). On note le degré $d_i = \sum_j w_{ij}$, $D = \mathrm{diag}(d_1, \ldots, d_n)$.

A partir de cette nouvelle représentation des données, le problème du clustering spectral se formule ainsi : on cherche une partition du graphe en communautés (une communauté est un sous-graphe de G) telles que celles-ci sont faiblement reliées entre elles et dont les sommets sont fortement connectés entre eux au sein d'une même communauté.

Constructions usuelles de E

Plusieurs constructions distinctes de l'ensemble des arêtes E sont possibles, nous présentons les plus usuelles.

Graphe ε-voisinage : Pour $\varepsilon > 0$ On connecte deux sommets si et seulement si ils sont à une distance inférieure à ε. Tous les points reliés sont donc à la même échelle ε. Ainsi on considère souvent qu'ajouter un poids n'ajouterait pas d'informations, c'est pourquoi

on décide généralement de ne pas pondérer ces graphes. Dans la pratique, il est parfois difficile de trouver un bon ε car les données multi-échelles sont fréquentes. Par exemple deux gaussiennes d'écart-types et moyennes très différents.

graphes des k plus proches voisins : On connecte v_i à v_j si et seulement si v_j est parmi les k plus proches voisins de v_i. Cette définition implique que le graphe est orienté, la relation de plus proche voisin n'étant pas symétrique. Dans certains contextes, on peut ignorer l'orientation des arêtes, pour rendre volontairement le graphe non orienté. En effet, le graphe est considéré non orienté dès lors qu'il existe une arête permettant de relier deux sommets différents. On parle alors de graphe des *k-plus-proches-voisins mutuels*. Dans les deux cas on pondère les arêtes avec la mesure de similarité.

Une propriété utile du graphes des k-plus-proches-voisins est sa capacité à relier des points se situant à différentes échelles, par opposition au graphe des k-plus-proches-voisins-mutuels (*cf.* Figure 5.1). En effet ce dernier a tendance à connecter les points se situant à la même échelle. Comme pour le choix de ε, il n'existe pas vraiment de résultat théorique permettant de choisir *a priori* le k optimal.

Le graphe entièrement connecté : On connecte tous les sommets entre eux et on pondère les arêtes par la mesure de similarité. La construction du graphe de similarité ayant pour but de mettre en exergue la *proximité* entre les données, le graphe entièrement connecté est pertinent seulement si la mesure de similarité est assez riche pour contenir entièrement la quantification de cette proximité. En effet, dans un tel graphe, toute l'information de proximité entre les données est contenue dans la mesure de similarité. Au contraire des $\varepsilon-$graphes et des graphes des k-plus-proches-voisins qui ne relient que des données déjà *proches* (respectivement au sens de la distance ε, ou au sens du voisinage).

5.1.2 Laplaciens de graphe et définitions

On considère un graphe non orienté et pondéré

$$G = (V, E, W), \quad |V| = n,$$

On définit les quantités suivantes :

- degré d'un sommet : $d_i = \sum_{j=1}^{n} w_{i,j}$
- La matrice des degrés est la matrice diagonale D qui contient les degrés $d_1, \cdots, d_n$ sur la diagonale.
- On définit les vecteurs $\mathbb{1}_A := (f_1, \cdots, f_n)$ avec $f_i = 1$ si $v_i \in A \subset V$, 0 sinon. Ces vecteurs encodent l'information sur les sommets appartenant ou non au sous-ensemble A.
- $W(A, B) := \sum_{i,j \in A,B} w_{i,j}$, pour A et B deux sous-ensembles de sommets.
- On utilise les deux quantités suivantes pour mesurer la taille d'un sous-ensemble de sommets : $|A|$:= nombre de sommets, $\mathrm{vol}(A) := \sum_{i \in A} d_i$, le volume.
- On appelle laplacien du graphe la matrice $L := D - W$

Propriétés Pour tout $u \in \mathbb{R}^n$,

$$u^\top L\, u = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}\, (u_i - u_j)^2 \;\geq\; 0,$$

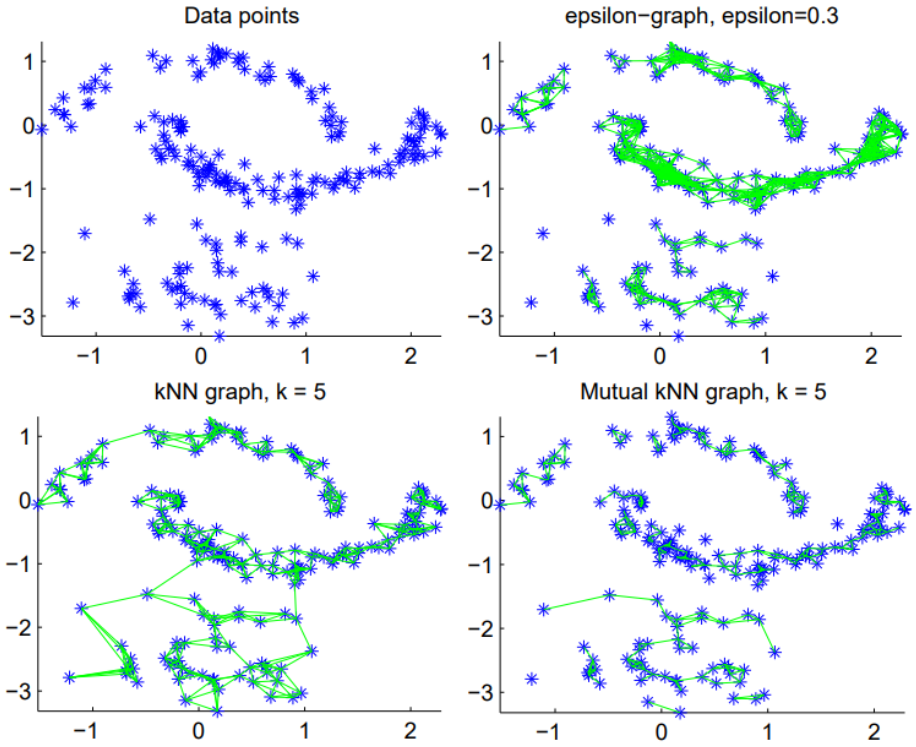


FIGURE 5.1 – Exemple issu de [84]. On observe ci dessus l'effet des différentes constructions de graphes de similarités susnommés. On comprend la difficulté de choisir un ε de qualité lorsque les données appartiennent à plusieurs échelles. De plus, on voit la capacité du kNN graphe à relier des données appartenant à des échelles distinctes et au contraire la tendance du kNN mutuel graphe à relier des données appartenant à des échelles identiques.

avec égalité ssi u est constant sur chaque composante connexe de G. En particulier, L est **symétrique définie positive** et son spectre est *non négatif* :

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

La multiplicité de la valeur propre 0 est égale au nombre k de composantes connexes $(A_1, \ldots, A_k)$; l'espace propre associé est engendré par les indicateurs $\{\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}\}$.

Remarque Dans les graphes de similarité forensiques, les degrés d_i peuvent être très hétérogènes (multi-échelles, densités locales différentes). Des versions normalisées du laplacien sont alors généralement plus stables ;

5.1.3 L'algorithme de clustering spectral

Il existe plusieurs variantes de l'algorithme de clustering spectral, nous présentons l'une d'entre elles ici.

Algorithme 1 : Clustering spectral non normalisé

Entrées : $W \in \mathcal{M}_{n,n}(\mathbb{R})$: matrice de similarité, k : nombre de clusters
Output : $A_1, \ldots, A_k$: clusters

1. Construire le graphe de similarité à partir de W.
2. Calculer le Laplacien L du graphe.
3. Calculer les k premiers vecteurs propres $u_1, \ldots, u_k$ de L.
4. Former $U \in \mathcal{M}_{n,k}(\mathbb{R})$ en plaçant $u_1, \ldots, u_k$ en colonnes.
5. Pour chaque $i \in \{1, \ldots, n\}$, définir $y_i \in \mathbb{R}^k$ comme la i-ème ligne de U.
6. Appliquer k-means aux n points y_i pour obtenir les clusters $C_1, \ldots, C_k$.
7. Définir $A_i = \{j \mid y_j \in C_i\}$ pour $i = 1, \ldots, k$.

Bien que de prime abord le fonctionnement du clustering spectral puisse sembler obscur, nous détaillons ci-dessous chacune des étapes, et dans la section suivante ses origines et les idées qui le sous-tendent :

Étape 1 : Construction du graphe de similarité On modélise les données comme un graphe $G = (V, E)$ dont les sommets représentent les points et les arêtes sont pondérées par les similarités données dans W. Cette représentation permet d'utiliser les outils de la théorie spectrale des graphes.

Étape 2 : Calcul du Laplacien Le Laplacien non normalisé est défini par $L = D - W$, où D est la matrice diagonale des degrés. Cette matrice encode la structure globale du graphe et est fondamentale pour la suite.

Étape 3 : Extraction des vecteurs propres On calcule les k vecteurs propres associés aux plus petites valeurs propres de L. Ces vecteurs forment une base de directions qui reflète la structure globale du graphe.

Étape 4 : Construction de la matrice U On place ces k vecteurs propres en colonnes pour former la matrice $U \in \mathbb{R}^{n \times k}$.

Étape 5 : Représentation des données La i-ème ligne de U, notée $y_i \in \mathbb{R}^k$, est la nouvelle représentation du point x_i dans un espace où la structure de graphe est encodée de manière à rendre les groupes plus séparables. Ces représentations y_i serviront directement de base pour l'étape de clustering.

Étape 6 : Clustering par k-means On applique un clustering basique tel que k-means sur ces nouvelles représentations y_i. Dans cet espace spectral, les groupes sont plus séparables qu'avec les coordonnées initiales.

Étape 7 : Attribution finale Chaque cluster obtenu est traduit en un ensemble d'indices A_i correspondant aux points du cluster d'origine.

5.1.4 Origine du clustering spectral

Dans cette sous-section, nous expliquons l'intuition qui sous-tend le clustering spectral. L'objectif d'une méthode de clustering est de regrouper en paquets des données qui sont similaires. A travers le prisme des graphes de similarité, construire ces clusters revient

à déterminer des communautés au sein d'un graphe. Une approche intuitive de partitionnement d'un graphe en communautés est l'approche dite *mincut*. Pour un nombre de communautés k, l'approche mincut [84] consiste à choisir k sous ensembles de sommets $A_1, \cdots A_k$ qui minimisent la quantité :

$$\text{cut}(A_1, \cdots A_k) = \frac{1}{2} \sum_{1 \leq k \leq n} W(A_i, \bar{A}_i) \tag{5.1}$$

Intuitivement, cette minimisation isole les parties du graphes qui sont faiblement reliées au reste du graphe. Cependant, ce problème conduit souvent à isoler simplement des sommets uniques du graphe, tandis que le clustering consiste précisément à rassembler les données similaires. Pour contrer ce problème il est possible de pénaliser le cut par la taille des partitions A_i, ceci conduisant à deux nouvelles approches :

$$\text{RatioCut}(A_1, \cdots A_k) = \frac{1}{2} \sum_{1 \leq k \leq n} \frac{W(A_i, \bar{A}_i)}{|A_i|} = \sum_{1 \leq k \leq n} \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|} \tag{5.2}$$

$$\text{NCut}(A_1, \cdots A_k) = \frac{1}{2} \sum_{1 \leq k \leq n} \frac{W(A_i, \bar{A}_i)}{\text{vol}(A_i)} = \sum_{1 \leq k \leq n} \frac{\text{cut}(A_i, \bar{A}_i)}{\text{vol}(A_i)} \tag{5.3}$$

Cependant ces deux problèmes sont NP-Difficile (c'est à dire qu'il est vérifiable en temps polynomial qu'une solution candidate est bien solution). Le clustering spectral est un moyen de les relaxer. Dans la suite on s'intéressera uniquement au problème RatioCut mais des techniques similaires peuvent être employées pour relaxer NCut.

Approximation du problème RatioCut pour $k = 2$

On s'intéresse d'abord à l'approximation de RatioCut dans le cas $k = 2$. Soit $A \subset V$, en définissant le vecteur $f = (f_1, \cdots, f_n) \in \mathbb{R}^n$ avec

$$f_i = \begin{cases} \sqrt{\frac{|\bar{A}|}{|A|}} & \text{si } v_i \in A \\ -\sqrt{\frac{|A|}{|\bar{A}|}} & \text{si } v_i \in \bar{A} \end{cases}$$

on obtient

$$f^T L f = |V| \cdot \text{RatioCut}(A, \bar{A})$$

On a donc ramené le problème de minimisation RatioCut à un problème de minimisation impliquant le laplacien du graphe de similarité. De plus, on a $\sum_{1 \leq i \leq n} f_i = 0$. C'est à dire que f est orthogonal à $\mathbb{1}$. En outre $||f|| = \sqrt{n}$. Cela mène à réécrire le problème :

$$\min_{A \subset V} f^T L f \text{ sous contrainte que } f \perp \mathbb{1} \text{ et } ||f|| = \sqrt{n}$$

C'est un problème d'optimisation discrète car les f_i ne peuvent prendre que deux valeurs. On peut relaxer ce problème en autorisant f à prendre des valeurs réelles, cela menant au problème suivant :

$$\min_{f \in \mathbb{R}^n} f^T L f \text{ sous contrainte que } f \perp \mathbb{1} \text{ et } ||f|| = \sqrt{n}$$

La solution de ce problème est le vecteur propre correspondant à la deuxième plus petite valeur propre de de L. Afin de partitionner le graphe nous avons tout de même besoin d'avoir une solution discrète

$$\begin{cases} v_i \in A \text{ si } f_i \geq 0 \\ v_i \in \bar{A} \text{ si } f_i < 0 \end{cases}$$

Cette heuristique est assez simpliste et non extensible pour $k > 2$. C'est pourquoi on peut également employer un $k-$means en considérant les f_i comme des points de $\mathbb{R}$ que l'on affecte à deux clusters C et $\bar{C}$:

$$\begin{cases} v_i \in A \text{ si } f_i \in C \\ v_i \in \bar{A} \text{ si } f_i \in \bar{C} \end{cases}$$

Approximation du problème RatioCut pour un k arbitraire

De la même manière que précédemment, pour une partition $V = \bigsqcup_{1 \leq j \leq k} A_i$, on définit k vecteurs $h_j = (h_{1,j}, \cdots, h_{n,j})^T$ par

$$h_{i,j} = \begin{cases} 1/\sqrt{|A_j|} \text{ si } v_i \in A_j \\ 0 \text{ sinon} \end{cases} \qquad \forall i \in \{1, \cdots, n\}, \forall j \in \{1, \cdots, k\}$$

On définit $H \in \mathcal{M}_{n,k}(\mathbb{R})$ qui contient ces k vecteurs comme colonnes (La matrice H correspond à la matrice U de l'algorithme de clustering spectral présenté en 1)). Chaque colonne est de norme 1 et est orthogonale aux autres, donc la matrice H est orthogonale. On peut vérifier que

$$h_i^T L h_i = \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|} = (H^T L H)_{i,i}$$

D'où

$$\text{RatioCut}(A_1, \cdots, A_k) = \sum_{i=1}^{k} (H^T L H)_{i,i} = \text{Tr}(H^T L H)$$

On a donc transformé la minimisation de $\text{RatioCut}(A_1, \cdots, A_k)$ en une minimisation de trace de matrice faisant intervenir le laplacien du graphe de similarité. Comme lorsque $k = 2$ on s'intéresse au problème relaxé naturel associé :

$$\min_{H \in \mathcal{M}_{n,k}(\mathbb{R})} \text{Tr}(H^T L H) \text{ sous contrainte que } H^T H = I$$

D'après Luxburg [84], la solution à ce problème de minimisation est donnée par la matrice H contenant les k premiers vecteurs propres de L comme colonnes. On conclut en appliquant un clustering (par exemple k-mean) sur les n lignes de cette matrice. Chacune d'entre elles correspondant à une des n données.

Formulation basée sur la norme de Frobenius On peut relier la formulation du problème de clustering spectral fondée sur la minimisation de la trace à un problème de minimisation d'une norme de Frobenius. En effet, en partant de la définition du Laplacien non normalisé $L = D - W$, on a :

$$\text{Tr}(H^T L H) = \text{Tr}(H^T (D - W) H) = \text{Tr}(H^T D H) - \text{Tr}(H^T W H). \tag{5.4}$$

Le premier terme $\text{Tr}(H^T D H)$ ne dépend pas de la structure des arêtes dans W de manière discriminante pour le clustering, et peut être vu comme constant si D est fixé.

Ainsi, minimiser $\mathrm{Tr}(H^T L H)$ revient à maximiser $\mathrm{Tr}(H^T W H)$ sous la contrainte $H^T H = I$:

$$\min_{H^T H = I} \mathrm{Tr}(H^T L H) \quad \Longleftrightarrow \quad \max_{H^T H = I} \mathrm{Tr}(H^T W H). \tag{5.5}$$

Or, on a l'identité suivante :

$$\|W - HH^T\|_F^2 = \|W\|_F^2 - 2\,\mathrm{Tr}(H^T W H) + \|HH^T\|_F^2. \tag{5.6}$$

Sous la contrainte $H^T H = I$, on a $\|HH^T\|_F^2 = k$ (constant). Minimiser $\|W - HH^T\|_F^2$ revient donc à maximiser $\mathrm{Tr}(H^T W H)$, et donc, par l'équivalence précédente, à minimiser $\mathrm{Tr}(H^T L H)$.

On établit ainsi que :

$$\min_{H^T H = I} \mathrm{Tr}(H^T L H) \quad \Longleftrightarrow \quad \min_{H^T H = I} \|W - HH^T\|_F^2, \tag{5.7}$$

les deux formulations étant équivalentes à un terme constant près.

Convergence de l'algorithme

Il a été démontré que l'algorithme de clustering spectral non normalisé peut ne pas converger (quand le nombre de données $n \to \infty$) , ou bien converger vers des solutions triviales, par exemple un cluster ne contenant qu'une seule donnée. Mathématiquement, on peut prouver que $\frac{1}{n} L$ converge vers un opérateur limite T lorsque $n \to \infty$. Cependant, les propriétés spectrales de T peuvent être telles qu'elles empêchent la convergence. Ce phénomène peut survenir même pour de petites tailles d'échantillon et produire des résultats peu fiables. Pour éviter ces problèmes, il est nécessaire que les valeurs propres de L correspondant aux vecteurs propres utilisés soient nettement inférieures au degré minimal du graphe, soit $\lambda_i \ll \min_j d_j$ pour $i = 1, \ldots, k$. Dans le cas contraire, les vecteurs propres obtenus peuvent s'apparenter à des fonctions de Dirac, ne séparant qu'un unique sommet du reste du graphe, ce qui mène à des partitions non pertinentes.

5.2 Clustering SNMF

Les matrices de similarité sont symétriques, et des méthodes comme le SNMF (Semi-Nonnegative Matrix Factorization) exploitent cette propriété en imposant que la factorisation préserve cette symétrie intrinsèque des données. Le clustering SNMF consiste à compresser la matrice de similarité dans un espace latent, naturellement interprété comme un espace de clusters. Soit une matrice de similarité $W \in \mathcal{S}_{n,n}(\mathbb{R})$, où $\mathcal{S}_{n,n}(\mathbb{R})$ désigne l'ensemble des matrices réelles symétriques de taille n, sa factorisation symétrique non négative peut s'écrire comme la solution du problème suivant :

$$\min_{H \geq 0} \|W - HH^T\|_F^2 \tag{5.8}$$

où $\|\cdot\|_F$ désigne la norme de Frobenius, H est une matrice non négative de dimension $n \times k$, et k est le nombre de clusters désiré.

Le problème d'optimisation de l'équation (5.8) sera désigné dans la suite par *problème SNMF*, puisqu'il repose sur l'approche de factorisation semi-non négative de matrices.

Lien avec la détection de communautés. Plutôt que d'utiliser la matrice de similarité brute, on utilise une version normalisée :

$$D^{-1/2}WD^{-1/2}.$$

Dans la perspective de la détection de communautés dans le graphe de similarité, cette normalisation aligne le problème SNMF avec la maximisation du *normalized cut* du graphe [71].

Maximiser le *normalized cut* revient à maximiser la dissimilarité entre clusters et la similarité à l'intérieur des clusters [128], ce qui correspond précisément aux caractéristiques attendues des communautés dans un graphe.

De plus, le problème SNMF se formule d'une manière proche du *problème spectral* [84], également fondé sur la détection de communautés dans les graphes. Le clustering SNMF est donc intuitivement relié à la détection de communautés.

Paramétrage et interprétation. En général, on choisit $k \ll n$ de sorte que cette factorisation corresponde à une compression des données. Le paramètre k est fixé par l'utilisateur selon le contexte applicatif, de manière à refléter la structure des données contenues dans la matrice de similarité W.

En effet, une solution H satisfait $W \approx HH^T$ et correspond donc à la compression de n^2 valeurs en $n \times k$ valeurs. En écrivant $W = [a_1, \ldots, a_n]$, où $(a_i)_{1 \leq i \leq n}$ désigne les vecteurs colonnes de W, on a :

$$a_i \approx H \cdot [h_{i,1}, \ldots, h_{i,k}]^T$$

avec $h_{i,j}$ les coefficients de la matrice H.

Le vecteur

$$[h_{i,1}, \ldots, h_{i,k}]^T$$

est alors interprété comme les coordonnées de la donnée a_i dans un espace latent de dimension k. La matrice H joue donc le rôle de matrice de changement de base entre l'espace des données et cet espace latent.

La non-négativité de H permet une interprétation naturelle de cet espace latent comme un espace de clusters : une donnée a_i appartient au cluster C_ℓ correspondant à la plus grande coordonnée $h_{i,\ell}$ de son vecteur de représentation.

Résolution numérique. La solution H du problème SNMF est obtenue à l'aide d'un algorithme de minimisation. Nous avons retenu la méthode du gradient projeté, qui offre un bon compromis entre qualité de minimisation et temps de calcul [71].

Soit W la matrice de similarité de taille $n \times n$. Nous cherchons à résoudre le problème SNMF. Compte tenu de l'existence de la bijection

$$\phi : H = [h_1, \ldots, h_k] \in \mathcal{M}_{n,k}(\mathbb{R}) \;\longmapsto\; x = [h_1^T, \ldots, h_k^T] \in \mathbb{R}^{n \times k},$$

on peut indifféremment considérer x ou H.

On optimise alors la fonction de perte

$$f : x \mapsto \|W - HH^T\|_F^2$$

de gradient

$$\nabla f : x \mapsto \phi^{-1}\Big(4(HH^T - W) \cdot H\Big).$$

Les itérations successives sont données par :

$$(x_{t+1})_{i,j} = \max\left((x_t)_{i,j} - \alpha\,(\nabla f(x_t))_{i,j},\ 0\right), \quad \forall i, j \in \{1, \ldots, n\} \times \{1, \ldots, k\}$$

où $\alpha > 0$ est le pas de gradient.

L'optimisation est arrêtée après un nombre maximal d'itérations fixé empiriquement.

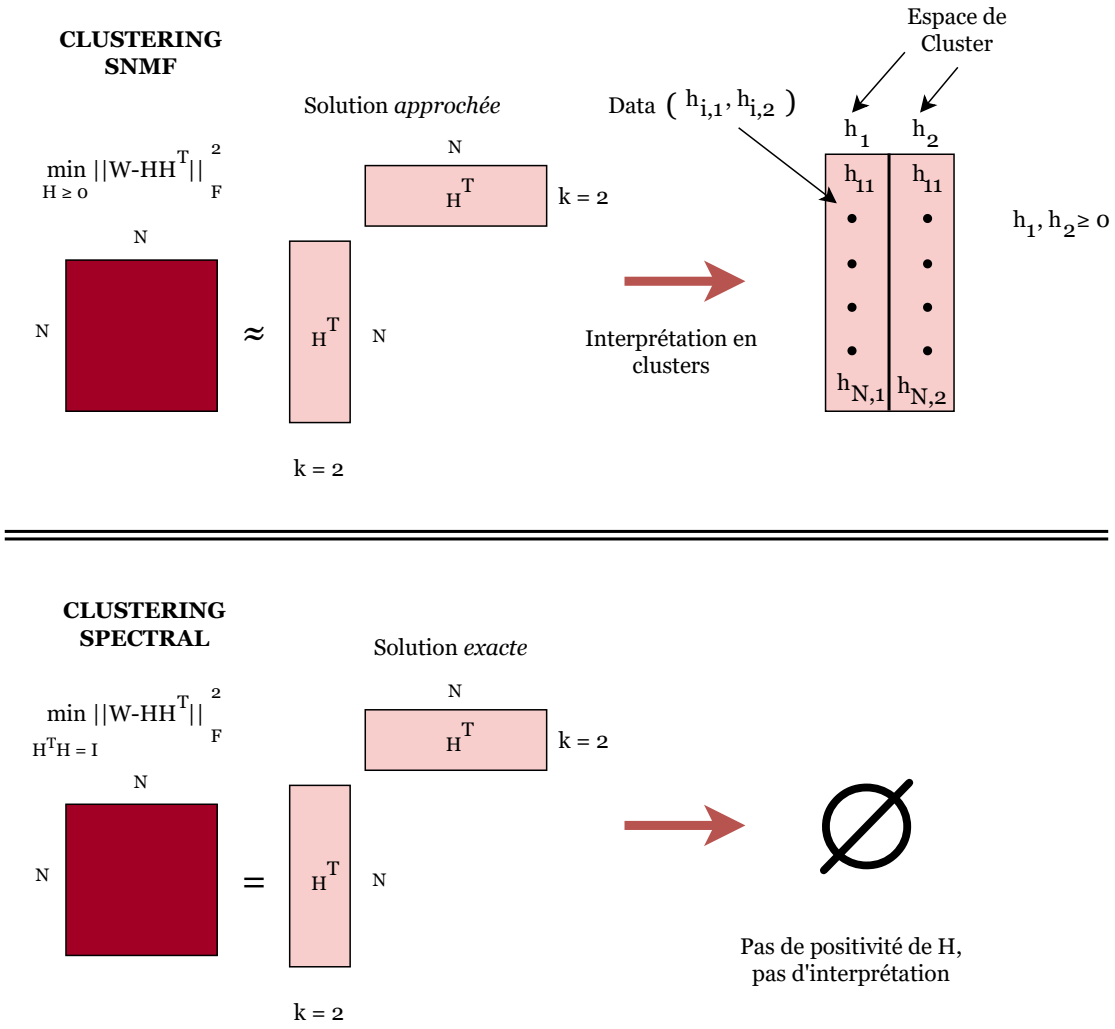


FIGURE 5.2 – Comparaison des clustering SNMF et Spectral. Le clustering spectral repose sur un clustering auxiliaire comme k-means là où le SNMF est auto-suffisant grâce à la positivité de H.

Comparaisons Clusterings SNMF et Spectral Le problème spectral présente l'avantage de posséder une solution globale et optimale, obtenue à partir des vecteurs propres de la matrice W [84]. Toutefois, en perdant la contrainte de non-négativité sur H, la structure en clusters n'est plus directement déductible, la contrainte d'orthogonalité étant plus forte que celle de positivité. Il est alors nécessaire d'appliquer une méthode de clustering auxiliaire sur les lignes de H afin d'extraire la structure des clusters. Ainsi, le clustering spectral se révèle moins intuitif que la méthode SNMF pour regrouper les données à partir de la matrice de similarité. La Figure 5.2 illustre les différences entre SNMF et clustering spectral en termes d'interprétabilité de leurs prédictions, ainsi que sur le caractère approximatif ou exact de la solution au problème d'optimisation associé. En outre KUANG, DING et PARK [71] montrent que le clustering SNMF performe généralement mieux que le clustering spectral.

6 Localisation et Détection de transformations d'images

Dans le chapitre précédent, nous avons présenté en détail les méthodes de clustering Spectral et SNMF. Dans ce chapitre, nous illustrons leur application à un problème concret : la **détection** et la **localisation** de transformations locales dans les images. Ces transformations constituent un enjeu majeur dans la constitution de *fake news* multimodales, car elles permettent de manipuler le contenu visuel afin d'influencer la perception des lecteurs.

Nous abordons les transformations d'images selon deux problématiques complémentaires :

Détection des transformations : Déterminer si une image a subi une modification (splicing, copy-move, etc.) ou si elle est authentique. La détection produit une caractérisation probabiliste de la présence de transformations sur l'image, ce qui constitue une feature précieuse pour la détection globale de *fake news* multimodales.

Localisation des transformations : Identifier précisément les zones altérées et les zones authentiques de l'image, dans une optique d'explicabilité. La localisation fournit une information visuelle interprétable, essentielle pour comprendre et expliquer la manipulation.

6.1 Localisation des Transformations d'Images, Détection de Communautés dans des Graphes

Résumé : Dans le contexte de la détection des fake news, un défi majeur réside dans la détection, la caractérisation, et la localisation des manipulations d'images, où des altérations sont réalisées afin de tromper les observateurs. En lien avec la problématique de détection d'anomalies, des méthodes récentes abordent la localisation des transformations d'images comme un problème de détection de communautés au sein de graphes associés aux images.

Dans cette contribution, nous proposons d'utiliser une méthode de *clustering* de communautés basée sur la factorisation de matrices symétriques non négatives. Par le jeu de plusieurs expériences de détection d'altérations dans des images manipulées, nous évaluons la robustesse de la méthode et discutons des améliorations potentielles. Nous présentons également un processus permettant de générer automatiquement des images falsifiées visuellement et sémantiquement cohérentes. Enfin, nous fournissons une application web afin d'illustrer ce processus.

6.1.1 Introduction

Les informations diffusées sur les réseaux sociaux contiennent souvent des images, et notamment, les fausses informations sont fréquemment accompagnées d'images manipulées. Par fausse information, nous entendons un contenu intentionnellement trompeur. En effet, une image véhicule souvent un contenu informationnel qui peut être altéré par des transformations telles que le *splicing* (copier une ou plusieurs régions/objets d'une image et les coller dans *une autre* image), ou le *copy-move* (copier une ou plusieurs régions/objets d'une image et les coller dans la *même* image).

Par exemple, l'inclusion d'un chef d'état dans la photographie d'une réunion diplomatique pourrait induire en erreur les consommateurs quant aux relations internationales entretenues par les pays impliqués.

De plus, l'émergence récente et la démocratisation des modèles génératifs ont facilité la production d'images falsifiées de haute qualité, connues sous le nom de *deepfakes* [172], ainsi appelées en raison de leur génération par des réseaux neuronaux profonds. Cependant, au-delà du consommateur moyen, certains domaines ont un besoin critique de détecter les images falsifiées, notamment les journalistes et les enquêteurs en criminalistique.

Par conséquent, le défi de la détection et de la localisation des transformations d'images a gagné en popularité ces dernières années, et actuellement, les stratégies de détection de falsifications d'images se divisent en deux catégories : *actives* et *passives*. L'approche active consiste à intégrer des signatures ou des filigranes dans les images, tandis que l'approche passive repose uniquement sur les caractéristiques intrinsèques de l'image elle-même. L'approche passive est la seule méthode efficace lorsqu'il s'agit d'images circulant librement, sans contrôle, et ayant été partagées de nombreuses fois, comme c'est le cas sur les plateformes de réseaux sociaux [172]. C'est pourquoi, dans cette contribution, nous nous concentrons exclusivement sur une approche passive.

Compte tenu de la diversité des transformations d'images, les travaux antérieurs se sont concentrés sur la détection de transformations spécifiques comme le *copy-move* [148], la compression JPEG [54], ou le *splicing*, en utilisant des méthodes traditionnelles du domaine de la vision par ordinateur [6][45]. Cependant, certaines transformations peuvent être difficiles à détecter et à définir mathématiquement. Ainsi, les approches d'apprentissage profond se sont révélées pertinentes pour relever ce défi au cours de la dernière décennie.

Bref état de l'art sur la localisation des transformations d'images : Les méthodes à l'état de l'art pour la détection de falsifications d'images reposent principalement sur des architectures d'apprentissage profond telles que les réseaux de neurones convolutifs (CNN) et les réseaux de neurones récurrents (LSTM) pour identifier efficacement les falsifications [172][116]. Ces méthodes excellent dans la capture des dépendances spatiales et temporelles au sein des données visuelles, ce qui les rend particulièrement efficaces pour l'analyse des motifs de falsification. Cependant, les avancées récentes des modèles basés sur l'attention ont établi un nouveau paradigme dans les tâches de vision par ordinateur, y compris dans la détection et la localisation des transformations d'images. En effet, ces modèles, grâce à leur capacité à se concentrer sélectivement sur les parties les plus pertinentes d'une image, ont démontré des capacités remarquables tant pour la détection que pour la localisation de régions falsifiées.

Par exemple, Rao et al. [116] intègrent un mécanisme d'attention basé sur un champ aléatoire conditionnel (CRF) dans un cadre CNN, ce qui entraîne des améliorations substantielles des performances en capturant efficacement les relations contextuelles au sein

de l'image. Cette intégration illustre le potentiel des approches basées sur l'attention pour affiner et améliorer le processus de détection, en particulier dans les cas complexes ou subtils de falsification.

Les méthodes basées sur l'attention représentent ainsi une piste prometteuse pour faire progresser l'analyse *forensique* des images, car elles combinent l'interprétabilité des mécanismes d'attention avec les solides capacités d'extraction de features des modèles d'apprentissage profond. Bien que notre travail ne se concentre pas sur les architectures basées sur l'attention, ces modèles offrent une perspective complémentaire à notre approche. Explorer leur intégration pourrait constituer une piste intéressante pour de futurs travaux, notamment dans le développement de modèles hybrides exploitant les forces des deux approches.

Transformations d'images et détection d'anomalies : Certaines recherches considèrent la détection de transformations locales dans une image comme un exemple de détection d'anomalies, un problème récurrent en apprentissage automatique [16]. La détection d'anomalies consiste à trouver un moyen automatique d'identifier les données anormales au sein d'un jeu de données [103]. Ici, il s'agit de détecter un ensemble de pixels anormaux dans l'image, correspondant aux zones transformées. Plusieurs approches se sont révélées efficaces, notamment l'utilisation d'une mesure de similarité qui quantifie la proximité des données entre elles. Les données qui sont éloignées des autres selon cette mesure sont alors considérées comme anormales.

Localisation et Graphes de Similarité Forensique : Un article récent [89] réinterprète la détection et la localisation de falsifications d'images comme un problème de détection de communautés au sein d'un graphe. Cette méthode introduit une représentation des images sous forme de graphes, appelés *Forensic Similarity Graphs* (FSG). Ces graphes contiennent des informations de bas niveau sur les transformations subies par les images associées. Chaque nœud du FSG représente une petite région carrée de l'image appelée *patch*. Ces nœuds sont reliés par des arêtes pondérées par un score de similarité caractérisant la proximité des patches en termes de transformations subies. De plus, chaque FSG est identifiable par sa matrice d'adjacence, appelée matrice de similarité. La détection de transformations sur une image correspond ainsi à une partition des nœuds du FSG représentant l'image [89], *i.e.* à la détection de communautés dans les graphes. Cette approche a montré de meilleures performances que plusieurs méthodes à l'état de l'art pour la localisation de transformations [89].

Cependant, deux outils principaux sont nécessaires : (i) une mesure de similarité entre images, (ii) une méthode de *clustering*. La mesure de similarité doit être capable de comparer et d'extraire les artefacts de transformation liés à deux images. Il s'agit d'une tâche complexe, particulièrement adaptée à des modèles puissants tels que les réseaux de neurones. Dans ce travail, nous utilisons une mesure de similarité pré-entraînée [90], basée sur un réseau de neurones convolutif (CNN).

La méthode de *clustering* choisie permet d'exploiter l'information encodée par la mesure de similarité. Il est donc crucial d'utiliser une méthode de *clustering* adaptée au problème étudié.

Le *clustering spectral* [84] (*cf.* Chapitre 5) est une approximation d'une solution au problème du *min-cut* (un problème classique de détection de communautés dans un graphe) et donne de bons résultats [89] pour la détection de transformations sur une image. Dans ce travail, nous utilisons une autre méthode de *clustering*, le *SNMF clustering* [71] (factorisation en matrices symétriques non négatives (*cf.* Chapitre 5)).

La figure 6.1 illustre notre approche proposée. À partir d'une image falsifiée et d'une mesure de similarité capturant l'information forensique (les artefacts de transformation) contenue dans l'image, nous construisons un graphe de similarité forensique et sa matrice de similarité correspondante, qui encode la similarité par paires entre les patches de l'image. L'étape suivante consiste à appliquer le *SNMF clustering* à la matrice de similarité (équivalent à la détection de communautés dans le graphe de similarité forensique) pour prédire les zones falsifiées dans l'image.

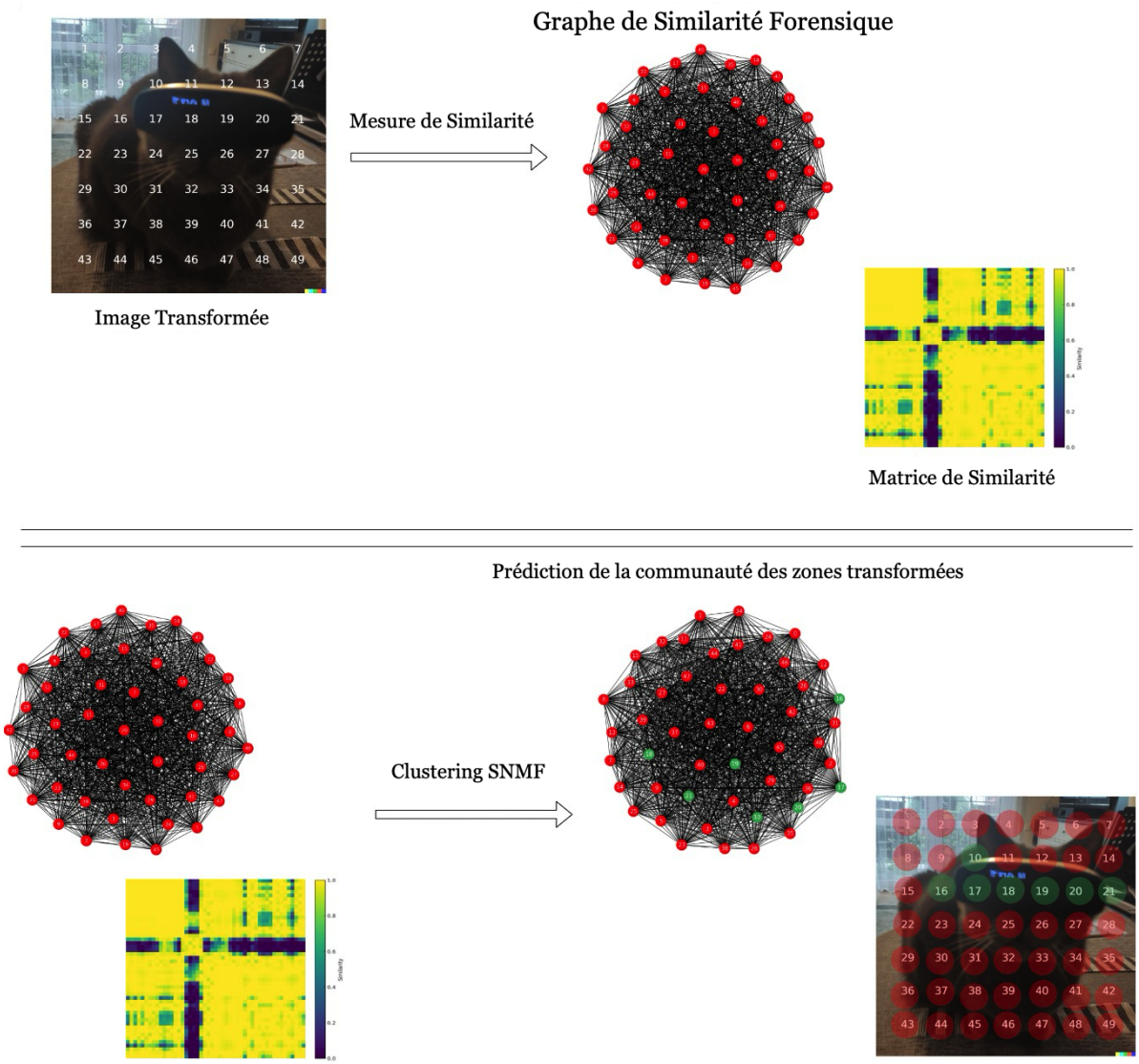


FIGURE 6.1 – Synthèse de l'approche de notre contribution. Nous exploitons la mesure de similarité de MAYER et STAMM [90] à l'aide du clustering SNMF, pour distinguer, à l'aide d'une détection de communauté sur un graphe, les zones transformées des zones authentiques de l'image.

Notre contribution s'organise de la manière suivante :

Dans la sous-section 2, nous passons en revue les travaux liés à la détection de falsifications d'images, en mettant l'accent sur les approches basées sur la détection de communautés. La sous-section 3 décrit la procédure générale appliquée à chaque image pour localiser les transformations. La sous-section 5 présente les avantages des différentes méthodes de clustering adaptées à la tâche de détection de communautés dans le contexte de la localisation de transformations d'images. La sous-section 6 introduit les jeux de données utilisés dans nos expériences, comprenant à la fois des ensembles de données

couramment employés dans la littérature sur la détection de transformations d'images et un nouveau jeu de données spécialement conçu pour cette étude. Dans la sous-section 7, nous présentons notre application web développée pour illustrer une méthode automatique de génération d'images falsifiées tout en préservant leur cohérence sémantique. La sous-section 8 détaille la méthodologie expérimentale, la configuration et les métriques d'évaluation utilisées. Enfin, les dernières sous-sections décrivent les expériences menées, les résultats obtenus et l'analyse associée.

6.1.2 Travaux Connexes

Mayer et Stamm [89] présentent une procédure simple d'utilisation pour détecter et localiser les transformations d'images. Cette méthode repose sur une mesure de similarité $S : (X_1, X_2) \mapsto S(X_1, X_2) \in [0, 1]$ entre deux images, qui capture les traces forensiques (informations de bas niveau) laissées dans les images par les transformations. Cette mesure est calculée par un CNN entraîné sur des millions d'images [90].

Ils proposent de représenter chaque image sous la forme d'un graphe de similarité forensique (*Forensic Similarity Graph*, FSG) dont les nœuds sont des *patches* de l'image et dont les arêtes sont pondérées par la mesure symétrique S entre ces *patches*. Cette représentation en graphe confère à la méthode de détection une certaine robustesse face aux perturbations locales et au bruit. La taille des *patches* recouvrant l'image peut être choisie en fonction de la granularité des transformations. De plus, la représentation FSG de l'image simplifie la détection des transformations au niveau du pixel en les détectant d'abord au niveau des *patches*.

6.1.3 Procédure globale

Pour illustrer la procédure proposée par Mayer et Stamm [89], une représentation en pseudocode (*cf.* Algorithme 2) est fournie. Ce pseudocode décrit les étapes générales de leur approche pour localiser les transformations d'images à l'aide de graphes de similarité forensique. Chaque cluster obtenu correspond à un ensemble de *patches* d'image ayant subi les mêmes types de transformation.

Algorithme 2 : Procédure pour la localisation des transformations d'images

Entrées : Paramètre de clustering θ, méthode de clustering C_θ, mesure de similarité S, image X, nombre de patchs n
Output : Clusters $Y = (Y_1, \ldots, Y_n)$
Séparer X en patchs $X_1, \ldots, X_n$;
Calculer les $\frac{n^2-n}{2}$ similarités between entre chaque paire de patchs ;
Construire la matrice de similarité $M(S, X)$;
Déterminer les clusters $Y \leftarrow C_\theta(M)$;

La procédure (*cf.* Algorithme 2) peut être résumée comme suit :

- étape 1 (Ligne 1) : L'image X est divisée en n *patches*, $X_1, \ldots, X_n$, à l'aide d'une grille régulière. Chaque *patch* chevauche ses voisins de 50%, ce qui améliore la précision de la détection au niveau du pixel.
- étape 2 (Ligne 2) : La mesure de similarité forensique S est calculée pour chaque paire de *patches*, ce qui donne $\frac{n^2-n}{2}$ scores de similarité.
- étape 3 (Ligne 3) : Ces scores alimentent la matrice de similarité $M(S, X)$, qui capture les relations entre les *patches*.
- étape 4 (Ligne 4) : La méthode de *clustering* C_θ est appliquée à $M(S, X)$, produisant des clusters $Y = (Y_1, \ldots, Y_n) \in \mathbb{R}^{n \times k}$, avec k le nombre de clusters, où chaque cluster correspond à un groupe de *patches* affectés par la même transformation.

6.1.4 Hypothèses sur la mesure de similarité

Le choix de la mesure de similarité est crucial, car il définit ce qui est ou n'est pas désigné comme une transformation. Dans ce travail, nous utilisons une mesure de similarité entre images conçue et entraînée par Mayer et Stamm [90]. Cette mesure est calculée par un réseau de neurones composé d'un extracteur de features basé sur un CNN, adapté du réseau MISLnet [3], et d'un réseau de neurones à trois couches appelé *similarity network*. La mesure de similarité forensique introduite dans [90] prend en compte le modèle de caméra, le type de transformation, ainsi que les paramètres de ces transformations, et s'adapte également à des modèles de caméras et à des transformations inconnus [90].

Le module extracteur de features, basé sur des CNN, a été entraîné sur 2×10^6 *patches* d'images de tailles 128×128 et 256×256 pixels, étiquetés avec leurs traces forensiques. Ces traces incluent des transformations spécifiques telles que le flou gaussien, le redimensionnement bilinéaire, la compression JPEG, ainsi qu'un ensemble de 50 modèles de caméras, dont par exemple l'iPhone 6 et le Samsung Galaxy Note 4 [90]. Ces caractéristiques forensiques représentent les traces forensiques des images, c'est-à-dire les informations contenues dans l'image qui caractérisent les transformations qu'elle a subies ainsi que sa source.

Cependant, bien que les types de traces forensiques utilisés lors de l'entraînement soient finis, Mayer et Stamm [90] démontrent que leur mesure de similarité se généralise efficacement à des traces forensiques non rencontrées durant l'entraînement. Concernant les transformations, dans les cas où les paires d'images comportent une transformation connue et une transformation inconnue, 19 comparaisons sur 24 sont correctement classées avec une précision supérieure à 90%. De plus, pour les paires d'images où les deux transformations sont inconnues, 5 comparaisons sur 6 sont correctement classées avec une précision similaire. Ils montrent également, en prenant l'exemple des transformations de redimensionnement, que la mesure de similarité est capable de détecter si les paramètres de transformation (par exemple, le facteur d'échelle dans le cas du redimensionnement) appliqués à deux images sont identiques.

Plus précisément, l'extraction de caractéristiques forensiques conserve un haut niveau de performance lorsqu'elle est appliquée à des transformations inédites, car les représentations profondes capturent des motifs forensiques essentiels qui ne sont pas spécifiques à une tâche. Par exemple, [88] a montré que les caractéristiques apprises pour l'identification de modèles de caméras étaient efficaces pour les tâches de détection de manipulations. Notamment, la précision de détection de manipulation s'est améliorée en utilisant des caractéristiques transférées depuis un réseau de modèles de caméras, suggérant que certaines caractéristiques encodent une information forensique fondamentale applicable à plusieurs tâches. En outre, Mayer et Stamm ont démontré que leur mesure de similarité forensique pouvait identifier si deux images proviennent du même modèle de caméra, même lorsque ce modèle n'était pas inclus dans l'ensemble d'entraînement [90].

Cette capacité de généralisation peut être attribuée à la nature hiérarchique des caractéristiques apprises par le CNN. En effet, les caractéristiques de bas niveau extraites par les couches les moins profondes sont indépendantes de la tâche et peuvent être réutilisées pour divers objectifs forensiques, tandis que les caractéristiques de haut niveau issues des couches plus profondes ont tendance à être plus spécifiques à la tâche. Cette mesure a donc l'avantage de ne pas être spécifique à un seul type de transformation ou de caméra.

étant donné que les tailles de *patches* sont fixées à 128×128 ou 256×256 pixels, on note que les dimensions d'une image peuvent ne pas toujours correspondre à la taille des *patches*. Cela se produit lorsque le reste de la division euclidienne de la largeur ou de la hauteur de l'image par la taille des *patches* n'est pas proportionnel à $(1 - \text{overlap}) \times$ patchsize. Dans ce cas, certaines zones en bordure de l'image restent non couvertes par

les *patches*. Pour traiter ce problème, il n'est pas possible d'ajouter du *padding*, car cela introduirait un nouveau type de transformation dans l'image, ce qui ajouterait du bruit dans la matrice de similarité et affaiblirait le processus de *clustering*. Cependant, Mayer et Stamm [89] ont montré que la procédure est robuste à la compression JPEG. Par conséquent, lorsque les images ont des dimensions inadaptées, il est possible soit de les traiter telles quelles en acceptant la perte d'information induite, soit de les compresser pour ajuster leurs dimensions de manière appropriée.

Le *similarity network* évalue la similarité de deux images en fonction de la proximité des traces forensiques qu'elles contiennent, en associant un score à chaque paire de vecteurs de caractéristiques. Il contient deux couches entièrement connectées ainsi qu'une couche dont le rôle est d'agréger les vecteurs de caractéristiques des deux images. Enfin, il se termine par un unique neurone de similarité avec une fonction d'activation sigmoïde, calculant un score de similarité compris entre 0 et 1 (voir Figure 6.2).

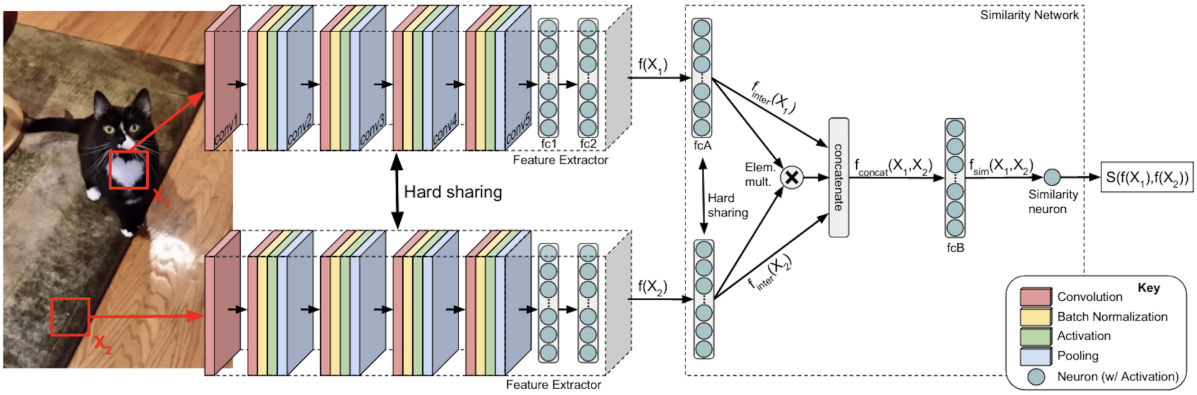


FIGURE 6.2 – Schéma de l'architecture du réseau de neurones qui calcule la mesure de similarité entre deux images [90]. Le réseau se compose d'une première partie, constituée de deux CNN extracteurs de features, qui convertit les images en vecteurs de features de faible dimension. Ces vecteurs sont ensuite fournis en entrée au *similarity network*, qui calcule un score compris entre 0 et 1, indiquant si les images contiennent les mêmes traces forensiques.

6.1.5 Approches de clustering pour la détection d'images falsifiées

Pour exploiter efficacement l'information contenue dans la mesure de similarité, il est essentiel de concevoir une méthode de *clustering* robuste et adaptée aux matrices de similarité. L'efficacité, dans ce contexte, recouvre trois aspects essentiels : des performances de calcul rapides pour traiter de grands ensembles de données, une simplicité de mise en œuvre avec une dépendance minimale à un grand nombre d'hyperparamètres, et une forte capacité à localiser avec précision les transformations dans les données.

Le *clustering* spectral répond bien à ces critères. Il exploite les vecteurs propres de la matrice de similarité pour identifier les clusters, permettant un regroupement précis et flexible. Cependant, malgré ses avantages, le *clustering* spectral ne capture pas intrinsèquement la structure sous-jacente des clusters dans les données [71].

Cette limitation rend nécessaire l'utilisation d'une méthode de *clustering* auxiliaire pour partitionner les représentations de faible dimension obtenues à partir de l'embedding spectral.

Bien que le choix de cette méthode auxiliaire ne soit pas figé, l'algorithme des *k-means* est un choix courant en raison de sa simplicité, de son efficacité computationnelle et de

son usage très courant [84]. Néanmoins, la dépendance aux *k-means* introduit des étapes supplémentaires ainsi qu'un réglage potentiel d'hyperparamètres, ce qui peut compliquer le flux de travail.

En revanche, la méthode SNMF [71] constitue une alternative intéressante en permettant de regrouper directement les données sans recourir à une méthode auxiliaire (*cf.* 5) telle que *k-means*. En décomposant la matrice de similarité en facteurs non négatifs, le SNMF assure à la fois un calcul efficace et la capacité de délimiter les clusters, offrant ainsi une approche simplifiée et interprétable pour les tâches de *clustering*.

On s'attend donc à obtenir de meilleurs résultats pour la localisation avec le clustering SNMF plutôt que le clustering spectral, d'autant que les travaux de Kuang, Ding et Park [71] appuient en théorie cette supériorité.

6.1.6 Jeux de données

Dans cette contribution, nous utilisons quatre jeux de données : Columbia [52], DSO-1 [15], ainsi que deux jeux de données construits pour cette contribution, que nous appellerons respectivement *DALLE2* et *Img2Text2Img*. Columbia et DSO-1 sont des jeux de données utilisés dans d'autres travaux [89][90] et présentent l'avantage d'avoir un historique de transformations connu. Ils servent donc de référence pour vérifier que la méthode fonctionne correctement.

Le jeu de données DALLE2 est utilisé pour tester la méthode sur des transformations plus complexes. DALLE2 peut être considéré comme un jeu de données de test difficile à évaluer quantitativement. En effet, l'incertitude concernant l'historique des transformations entraîne un manque de précision au niveau du *ground truth*. Comme nous n'avons aucun contrôle sur la provenance des images authentiques de ce jeu de données, nous ne pouvons pas affirmer qu'elles n'ont pas subi d'autres transformations en plus de celles réalisées avec DALL·E 2. (Dans la suite, nous distinguons DALL·E 2, qui désigne l'algorithme de génération d'images, et DALLE2, qui désigne le jeu de données d'images créées à l'aide de DALL·E 2).

Le quatrième jeu de données, Img2Text2Img, est un jeu de données que nous avons entièrement construit à partir d'un processus automatique illustré par la Figure 6.7, et dont la description complète est disponible sur GitHub [1]. Ce processus permet de tester la détection de transformations plus complexes, telles que le *splicing* avec *stable diffusion* [2].

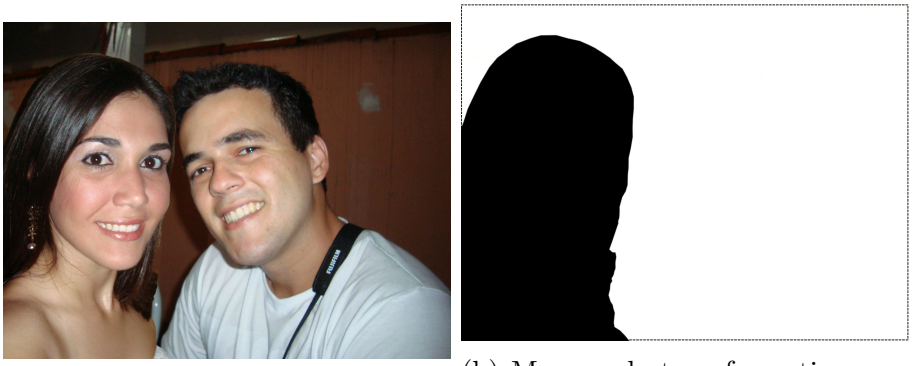
Columbia et DSO-1 : Columbia contient 180 images transformées, ayant subi un *splicing* manuel. Comme expliqué précédemment (*cf.* définition 6.1.1), la transformation par *splicing* consiste à copier une ou plusieurs régions ou objets d'une image et à les coller dans une autre image. Les images du jeu de données Columbia représentent principalement des scènes d'intérieur. Elles sont relativement petites (moins d'un million de pixels) et les transformations sont grossières. Par conséquent, le temps de calcul pour chaque image est assez court (de l'ordre d'une minute) et les résultats sont meilleurs que sur les trois autres jeux de données. La Figure 6.3 présente trois exemples d'images authentiques et falsifiées issues du jeu de données Columbia.

DSO-1 (*cf.* 2) contient 200 images d'une résolution de 1536×2048, dont 100 authentiques et 100 transformées. Les images transformées présentent un individu ajouté à une scène contenant déjà au moins une autre personne. En raison de la cohérence visuelle et sémantique de la transformation, les images transformées de DSO-1 peuvent facilement

1. GitHub : Img2Text2Img Dataset
2. GitHub : stable diffusion model



FIGURE 6.3 – Exemples d'images du jeu de données Columbia. Images authentiques en haut, images issues d'un *splicing* en bas [52].



(a) Image falsifiée

(b) Masque de transformation correspondant

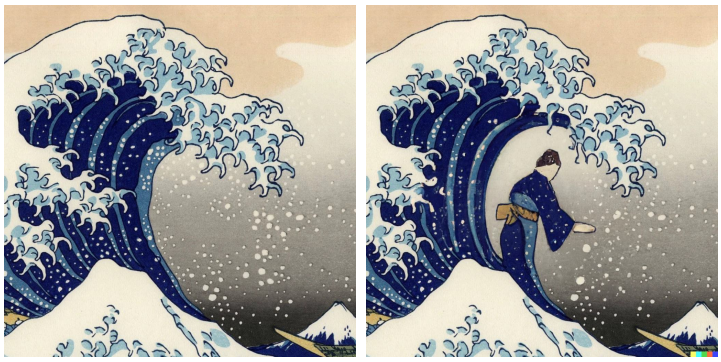
FIGURE 6.4 – Exemple d'image issue du jeu de données DSO-1, avec l'image falsifiée à gauche et le masque de transformation correspondant à droite [15].

être prises pour authentiques par un observateur humain. Compte tenu de leur résolution plus élevée par rapport aux images de Columbia, les temps de calcul pour une seule image sont plus longs (de l'ordre de cinq minutes). Bien que les transformations soient de même nature dans DSO-1 et Columbia, DSO-1 constitue un jeu de données plus difficile que Columbia pour la localisation des transformations. La Figure 6.4 montre un exemple d'image falsifiée issue de DSO-1 ainsi que son masque de transformation associé.

Les expériences menées sur Columbia et DSO-1 servent à calibrer les hyperparamètres et à vérifier le bon fonctionnement de la méthode.

DALLE2 : DALLE2 est un jeu de données construit pour ce travail. Il contient 60 images authentiques et 60 images transformées, de taille 1024×1024. Chaque image est générée par l'algorithme DALL·E 2 d'OpenAI [115] de la manière suivante : une image accompagnée d'une description textuelle est fournie à DALL·E 2, puis une partie de l'image définie par l'utilisateur est remplacée par DALL·E 2 conformément à la description. La Figure 6.5 illustre un exemple du jeu de données DALLE2. Les images construites de cette manière proviennent d'une base de données en libre accès contenant des images générées par DALL·E 2 [24]. Nous ne disposons pas d'informations sur l'historique de transformation des images authentiques ; elles ont donc pu subir d'autres transformations en plus de celles réalisées avec DALL·E 2. Cela peut poser un problème pour l'évaluation du modèle, car notre méthode peut détecter plusieurs transformations différentes, et nous n'avons accès qu'à la transformation effectuée par DALL·E 2 comme vérité terrain.

Cependant, ce type d'images reflète une situation réaliste, car elles présentent une forte cohérence visuelle et leur description de génération est saisie manuellement. L'architecture de DALL·E 2 est basée sur un modèle de diffusion, un modèle génératif profond efficace



(a) Image authentique (b) Image falsifiée

FIGURE 6.5 – Exemple issu du jeu de données DALLE2, l'image (b) a été transformée à partir de l'image (a) selon la description suivante : *"Surfing The Great Wave off Kanagawa | A Japanese surfer on a wave | woodblock print by the Japanese ukiyo-e artist Hokusai, 1831"*.

dans de nombreux domaines, notamment la génération d'images et de vidéos [167]. Les images générées par DALL·E 2 atteignent parfois un tel niveau de réalisme que la détection de falsification devient souvent impossible pour un humain. Le jeu de données DALLE2 apparaît donc plus difficile à traiter que Columbia et DSO-1. En effet, d'une part, la transformation subie est plus subtile qu'un simple déplacement de pixels, et d'autre part, on peut s'interroger sur l'efficacité de la mesure de similarité employée face à des images générées par réseaux de neurones.



(a) Image falsifiée (b) Masque correspondant

FIGURE 6.6 – Exemple issu du jeu de données Img2Text2Img, avec l'image falsifiée à gauche et le masque de transformation correspondant à droite.

Img2Text2Img : Pour construire le jeu de données Img2Text2Img, nous avons mis en place une stratégie de construction d'images entièrement automatisée visant à reproduire un type spécifique de transformation : la modification localisée d'une image à l'aide d'un modèle génératif. Ce jeu de données se distingue de DALLE2 car nous disposons d'informations complètes sur l'historique de transformation des images. Il peut donc être utilisé pour évaluer quantitativement les *clusterings*.

Le processus débute par une photographie originale que nous avons prise. Nous sélectionnons ensuite une région carrée de cette image et la décrivons textuellement à l'aide du

modèle GIT (*Generative Image-to-Text Transformer*) [149]. Cette description est ensuite utilisée comme entrée pour générer une nouvelle image via le modèle *Stable Diffusion* [110], un modèle génératif profond de type texte-vers-image. Après génération, nous supprimons l'arrière-plan (à l'aide de l'outil rembg [3], qui repose sur un modèle U^2-Net [113] pour la détection d'arrière-plan) de cette nouvelle image, que nous intégrons ensuite dans l'image originale.

Le fait de partir d'une description textuelle de la zone authentique garantit une cohérence sémantique entre l'image originale et l'image modifiée, tandis que la suppression de l'arrière-plan dans la zone modifiée assure la cohérence visuelle. La Figure 6.6 montre un exemple issu du jeu de données Img2Text2Img. Bien que l'image soit sémantiquement et visuellement cohérente, la zone falsifiée peut être détectée par un œil averti, par exemple à cause du manque de précision du texte sur le panneau.

Les modèles GIT et Stable Diffusion sont utilisés via les API HuggingFace [4] [5]. Notre jeu de données Img2Text2Img se compose de 100 images authentiques et 100 images falsifiées. Nous avons initialement capturé 100 photographies, chacune d'une résolution de 1024×1024 pixels, afin d'avoir un contrôle complet sur leur historique de transformation.

6.1.7 Application web pour la génération d'images falsifiées

Nous avons également développé une application web destinée à illustrer le processus automatique de création d'images utilisé pour constituer le jeu de données Img2Text2Img [6]. Le site permet aux utilisateurs de soumettre des images de moins de 16Mo et d'appliquer une transformation locale basée sur un *prompt* dérivé d'une zone de l'image, comme décrit dans la section précédente. Le *prompt*, le masque de transformation et l'image transformée sont affichés sur le site et enregistrés dans le cloud à des fins exclusivement de recherche. Selon la taille de l'image, le temps de traitement varie, mais il est généralement de l'ordre d'une minute. Lorsque le site reste inactif pendant un certain temps, le serveur passe en mode veille, ce qui peut entraîner un délai supplémentaire lors de son réactivation. Le site est encore en cours de développement.

6.1.8 Approche expérimentale et méthodologie

D'une part, nous nous intéressons à comparer l'efficacité des méthodes de *clustering* spectral et SNMF, et d'autre part, nous voulons tester la capacité de la méthode à détecter des transformations locales de type *deepfake* sur des images. Le nombre de clusters k est fixé à 2 afin de distinguer uniquement les zones authentiques et transformées dans les images. Il est possible d'utiliser plus de deux classes tout en conservant les mêmes méthodes de *clustering*, afin de distinguer différents types de transformations au sein d'une même image. Cependant, nous ne disposons pas actuellement de jeux de données et de vérités terrain adaptés pour ce faire.

Ces expériences sont en partie menées sur des jeux de données couramment utilisés dans la littérature [52][15]. À chaque itération, le pas de gradient α dans le *clustering* SNMF diminue géométriquement (avec un facteur $\beta = 0.1$) à partir d'une valeur initiale $\alpha_0 = 1$. Intuitivement, à chaque itération, α est choisi pour garantir que la fonction de perte se rapproche d'un minimum, tout en restant suffisamment grand pour maintenir une convergence efficace.

3. https://pypi.org/project/rembg/
4. https://huggingface.co/microsoft/git-large-r-coco
5. https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0
6. web-app

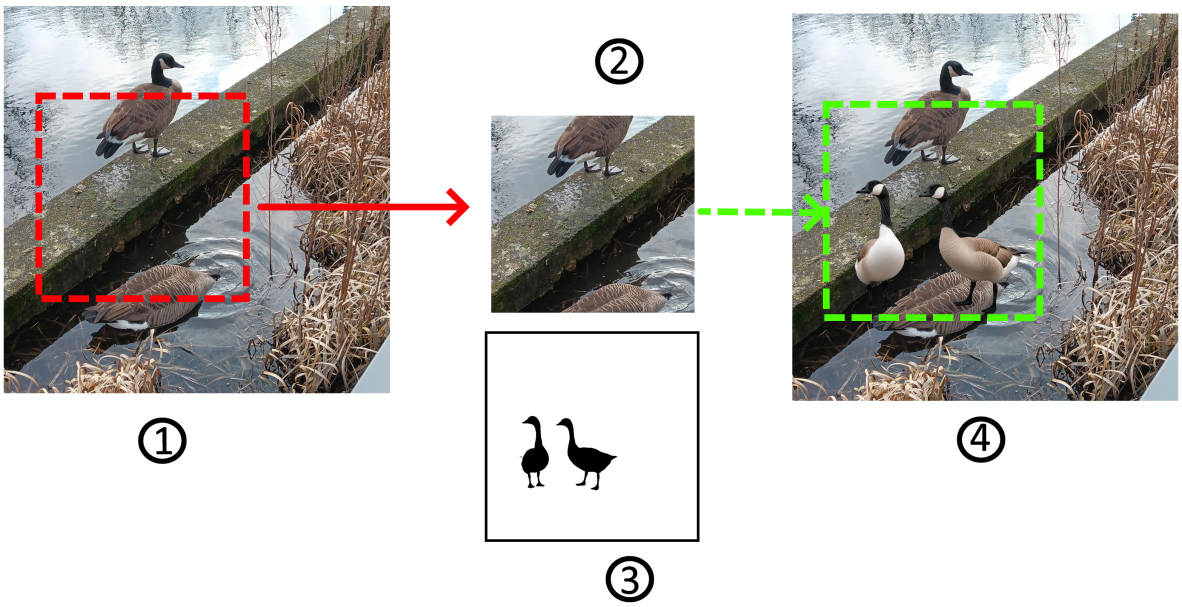


FIGURE 6.7 – Procédure de construction du jeu de données Img2Text2Img : ① - Sélection d'une zone carrée (pointillée rouge) dans l'image authentique. ② - La zone sélectionnée est extraite et convertie en texte : *"two geese standing on a ledge next to a body of water."* ③ - Masque de l'image complète associé à la transformation. ④ - La zone carrée, transformée à partir du texte, est ajoutée à l'image authentique après suppression de l'arrière-plan.

Les tailles de *patches* choisies pour chaque jeu de données sont résumées dans la Figure 6.1.10. Le calcul des métriques est effectué au niveau du pixel (la conversion des classes de *patches* au niveau pixel suit la procédure décrite dans [89], avec un seuil fixé à 0.5).

6.1.9 Du patch-level au niveau pixel-level

Pour un nombre fixé de clusters $k \geq 2$, la prédiction est affinée au niveau du pixel. Pour un pixel donné p de coordonnées (x, y), et l'ensemble $\mathcal{R}(p)$ des *patches* $R(p)$ contenant p, son vecteur de clusters $v_k(p) \in \mathbb{R}^k$ est défini comme la somme pondérée des fonctions indicatrices de cluster des *patches*. Les poids sont déterminés par l'inverse de la distance entre le pixel et le centre de chaque *patch*, normalisée par la taille du *patch*. Ainsi, un pixel situé plus près du centre d'un *patch* est davantage influencé par celui-ci que lorsqu'il se trouve plus près de sa bordure.

$$v_k(p) = \sum_{R \in \mathcal{R}(p)} \frac{\text{cluster}(R) \cdot \text{patchsize}}{\sqrt{2} \cdot \|p - \text{center}(R)\|_2} \tag{6.1}$$

où $\text{cluster}(R) \in \{0, 1\}^k$ est défini comme un vecteur *one-hot* tel que $\text{cluster}(R)_i = 1$ si le *patch* R appartient au cluster i, et 0 sinon.

Le cluster d'un pixel p est alors défini comme suit :

$$\text{cluster}(p) = \underset{i}{\text{argmax}}\ v_k(p(x, y)) \tag{6.2}$$

Cela implique que le cluster associé à un pixel p est déterminé par le cluster dominant (celui ayant le poids le plus élevé) parmi les clusters présents dans les *patches* contenant p.

6.1.10 Métriques

Nous utilisons plusieurs métriques pour mesurer la qualité de nos prédictions binaires au niveau du pixel. Pour l'analyse au niveau pixel, à partir de la prédiction pour chaque *patch*, nous examinons la proportion de *patches* contenant le pixel qui sont classés comme transformés. Dans nos expériences, étant donné un recouvrement de 50% entre les *patches*, chaque pixel se voit attribuer une valeur dans l'ensemble $\{0, 0.25, 0.5, 0.75, 1\}$. Cette valeur est ensuite lissée en appliquant un flou gaussien sur l'ensemble de ces valeurs, produisant une valeur finale comprise entre 0 et 1 pour chaque pixel. Cette valeur finale représente la probabilité que le pixel associé soit transformé.

Pour la détection finale, nous comparons ces valeurs à un seuil, les meilleurs résultats étant obtenus avec un seuil de 0.5. Dans la suite, TP, TN, FP, FN désignent respectivement le nombre de pixels correctement prédits comme transformés/authentiques et incorrectement prédits comme transformés/authentiques :

(i) *MCC (Matthews correlation coefficient) :*

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{6.3}$$

Le MCC mesure la qualité des prédictions en tenant compte des quatre issues possibles d'une classification binaire. Une valeur de -1 correspond à une prédiction complètement incorrecte, 0 à une prédiction aléatoire, et $+1$ à une prédiction parfaite. Le MCC est particulièrement utilisé lorsque les classes sont fortement déséquilibrées, ce qui n'est pas souvent le cas dans nos données, les zones transformées représentant généralement des surfaces significatives de l'image. Une valeur supérieure à 0.5 est généralement considérée comme indicative d'une bonne méthode de prédiction.

(ii) *F1-score :* Le F1-score est couramment utilisé pour mesurer la qualité d'une classification binaire. Il est calculé à partir de deux métriques importantes : la précision et le rappel. La précision est la proportion de prédictions positives correctes parmi toutes les prédictions positives. Le rappel est la proportion de données positives correctement prédites parmi toutes les données positives. La précision mesure la capacité du modèle à prédire correctement les données positives, tandis que le rappel mesure sa capacité à détecter les données positives. Le F1-score est la moyenne harmonique de la précision et du rappel, ce qui permet d'agréger ces deux mesures et d'évaluer la performance globale du modèle.

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{6.4}$$

Un F1-score proche de 1 indique une performance idéale ; proche de 0, il est faible.

(iii) *Courbe ROC :* La courbe ROC (*Receiver Operating Characteristic*) est un outil graphique qui mesure la capacité du modèle de classification à distinguer deux classes. Son principal intérêt réside dans sa capacité à prendre en compte à la fois le taux de vrais positifs (sensibilité) et le taux de faux positifs (spécificité), pour différents seuils de décision. Pour définir la courbe ROC, nous utilisons les probabilités de transformation pour chaque pixel telles que précédemment calculées, avant comparaison au seuil.

(iv) *Score AUC :* L'AUC (*Area Under the ROC Curve*) mesure l'aire sous la courbe ROC et fournit une mesure agrégée des performances pour tous les seuils de classification possibles. Une AUC de 1 indique une performance parfaite, tandis qu'une AUC de 0.5 correspond à une performance équivalente au hasard.

(v) *Indice de silhouette :* L'indice de silhouette mesure la qualité du *clustering* en fonction de la cohésion et de la séparation des clusters. Pour chaque donnée, sa cohésion est définie comme sa distance moyenne aux autres données de son cluster. Sa séparation est définie comme sa distance moyenne au cluster le plus proche. L'indice de silhouette correspond à la moyenne, sur l'ensemble des points, de la différence normalisée entre la séparation et la cohésion. Ainsi, l'indice de silhouette varie entre 1 et -1, une valeur proche de 1 indiquant une bonne séparation et cohésion, et à l'inverse, une valeur proche de -1 indiquant une mauvaise séparation et cohésion.

6.1.11 Résultats et discussion

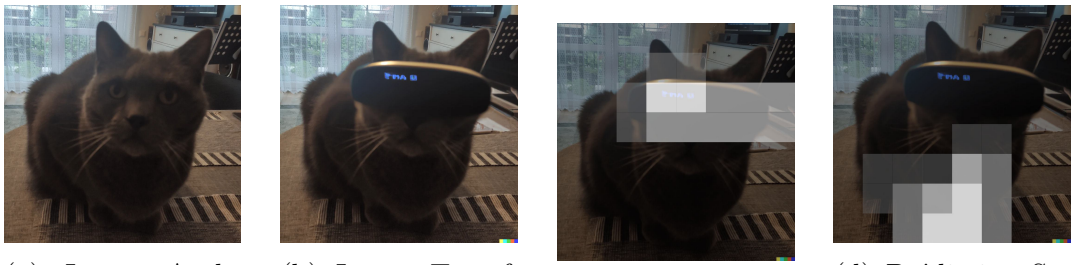
Dans cette section, nous détaillons les résultats expérimentaux obtenus dans le cadre de notre étude. Les résultats sont d'abord présentés individuellement pour chaque jeu de données, en mettant en évidence les observations spécifiques à chacun. Nous proposons ensuite une synthèse des tendances communes à tous les jeux de données, comprenant les métriques globales de performance, des analyses quantitatives, les temps de calcul, ainsi que des exemples illustratifs tels que des matrices de similarité.

Jeu de données Columbia : Le jeu de données Columbia a été utilisé dans nos expériences principalement pour calibrer la méthode proposée, plutôt que comme un banc d'évaluation complet. Cette décision découle des caractéristiques du jeu de données : les transformations par *splicing* qu'il contient sont grossières et facilement identifiables, ce qui le rend moins difficile pour des techniques forensiques avancées. Nous avons donc sélectionné un sous-ensemble de 30 images de Columbia pour les tests. Les images testées proviennent d'un appareil photo Canon G3, dans lesquelles des parties d'images prises avec un Nikon D70 ont été ajoutées. Compte tenu des petites dimensions des images Columbia, nous avons choisi d'utiliser des *patches* de 128×128 pixels. En effet, l'utilisation de *patches* de 256×256 pixels entraîne une granularité trop faible, ce qui conduit à ignorer certains détails dans les images.

Comme prévu, les meilleurs résultats parmi les jeux de données testés ont été obtenus sur Columbia, en termes de métriques utilisées (*cf.* Tableau 6.1). De plus, sur Columbia, nous avons obtenu de meilleures performances pour toutes les métriques avec le *clustering* SNMF par rapport au *clustering* spectral, avec une augmentation de 1 à 5%. Nous observons également que SNMF et *Spectral clustering* coïncident pour certaines images. Le MCC est nettement supérieur à 0.5 (*cf.* Tableau 6.1) pour les deux méthodes, confirmant que les deux sont efficaces.

Jeu de données DSO-1 : Les résultats obtenus sur DSO-1 sont assez similaires à ceux de Columbia, bien que légèrement inférieurs. Nous avons testé l'ensemble des 100 images de DSO-1 avec des *patches* de 256×256 pixels. Pour les deux méthodes, le MCC est supérieur à 0.5, tandis que le score AUC dépasse 0.9 (*cf.* Tableau 6.1 et Figure 6.10), ce qui indique une bonne classification pour SNMF comme pour le *clustering* spectral. Cependant, sur ce jeu de données, c'est le *clustering* spectral qui obtient des résultats légèrement supérieurs à SNMF. Le fait que les performances soient bonnes sur Columbia et DSO-1 montre que les expériences sont bien calibrées (même si tous les hyperparamètres ne sont pas forcément optimaux).

Jeu de données DALLE2 : Comme attendu, les résultats sur DALLE2 sont plus contrastés. Que ce soit avec le *clustering* spectral ou le *clustering* SNMF, le MCC est



(a) Image Authentique (b) Image Transformée (c) Prédiction SNMF (d) Prédiction Spectrale

FIGURE 6.8 – La 18e image du jeu de données généré avec DALL·E 2. Plus les zones ombrées sont claires, plus la transformation est fortement détectée.

proche de 0 (*cf.* Tableau 6.1), ce qui indique que les prédictions sont proches d'un comportement aléatoire. De même, le fait que le score AUC soit proche de 0.5 (*cf.* Tableau 6.1 et Figure 6.10) confirme que le modèle se comporte comme un classificateur de pixels aléatoire. En raison de l'absence de vérité terrain fiable, les métriques sont difficiles à interpréter en l'état.

Cependant, en examinant chaque prédiction individuellement, nous pouvons faire les observations suivantes : pour le *clustering* spectral, aucune des prédictions n'est correcte, et la méthode échoue à interpréter les matrices de similarité. En revanche, pour le *clustering* SNMF, les prédictions semblent pertinentes pour 19 images sur les 60 testées (images nos 3, 5, 11, 18, 20, 21, 26, 27, 28, 31, 32, 36, 37, 38, 41, 46, 48, 52, 59). Par exemple, la Figure 6.8 illustre une comparaison entre les prédictions de SNMF et celles de *Spectral clustering* sur la 18e image. La transformation appliquée correspond à l'ajout d'un casque de réalité virtuelle sur un chat. Cette transformation semble correctement détectée par SNMF, mais pas du tout par la méthode spectrale. Environ un tiers des images du jeu DALLE2 présentent ce type de résultats.

Il arrive que la méthode identifie des transformations supplémentaires sur des images dites authentiques, comme illustré dans la Figure 6.9. Dans cette image générée par DALL·E 2 (59e image), notre méthode détecte à la fois le cameraman et l'enfant comme zones transformées. La partie haute de l'image, clairement altérée, n'est pas détectée, mais le cameraman l'est. La détection de l'enfant est difficile à interpréter : il peut s'agir d'une erreur de détection ou, au contraire, d'une véritable transformation préalable de l'image dite authentique.



(a) Image Authentique (b) Image Transformée (c) Prédiction SNMF

FIGURE 6.9 – La 59e image du jeu de données généré avec DALL·E 2. Bien que la transformation concerne principalement l'homme en train de photographier, l'enfant et l'homme sont tous deux détectés comme des zones falsifiées.

Ces 19 images avec des prédictions SNMF pertinentes ne semblent partager aucune caractéristique commune ; certaines sont des dessins, d'autres sont photoréalistes, et elles ne

traitent pas des mêmes thèmes. Les explications possibles sont multiples : la transformation peut être trop complexe pour être capturée par la mesure de similarité employée, ou bien le manque de connaissance sur l'historique des transformations constitue un obstacle majeur à l'évaluation. Les expériences sur le jeu Img2Text2Img permettent de répondre à cette question, puisque d'une part l'historique des transformations est connu et maîtrisé, et d'autre part les transformations sont d'une complexité comparable à celles du jeu DALLE2.

Jeu de données Img2Text2Img : Sur le jeu de données Img2Text2Img, nous avons testé les 100 images transformées avec des *patches* de 128×128 pixels et de 256×256 pixels. Les résultats sont meilleurs avec les plus petits *patches*, car certaines transformations sont de petite taille et leur détection est facilitée par une granularité plus fine. Avec les *patches* de 128×128 pixels, nous avons obtenu un MCC de 0.54 avec SNMF et de 0.53 avec le *clustering* spectral (*cf.* Tableau 6.1). Ces valeurs étant supérieures au seuil généralement accepté de 0.5, cela valide la pertinence de la méthode. Les autres métriques confirment également l'efficacité de la classification. Ces résultats montrent que la méthode de détection fonctionne correctement malgré la complexité des transformations induites par *Stable Diffusion*.

Ces observations confirment l'hypothèse formulée à partir des expériences sur DALLE2 : les mauvais résultats sur ce dernier sont davantage liés au manque de connaissance sur les transformations subies par les images qu'à la complexité des transformations générées par DALL·E 2. Par ailleurs, sur Img2Text2Img, SNMF surpasse légèrement le *clustering* spectral.

Métriques	Méthode de Clustering	Patchsize	Columbia	DSO-1	DALLE2	Img2Text2Img
F1 score	Clustering Spectral	128	0.88		0.54	**0.94**
		256		**0.95**	**0.58**	0.92
	SNMF	128	**0.90**		0.45	**0.94**
		256		0.94	0.57	0.91
MCC	Clustering Spectral	128	0.62		0.00	0.53
		256		**0.72**	**0.1**	0.43
	SNMF	128	**0.69**		-0.1	**0.54**
		256		0.66	0.01	0.39
IoU	Clustering Spectral	128	0.78		0.37	**0.88**
		256		**0.91**	**0.41**	0.86
	SNMF	128	**0.83**		0.29	**0.88**
		256		0.89	0.40	0.84
AUC	Clustering Spectral	128	0.89		0.49	0.94
		256		**0.94**	**0.54**	0.89
	SNMF	128	**0.94**		0.44	**0.95**
		256		0.91	0.51	0.86

TABLE 6.1 – Métriques de performance (F1-score, MCC, IoU et AUC) pour les méthodes de *clustering* spectral et de *clustering* SNMF sur différents jeux de données (Columbia, DSO-1, DALLE2, Img2Text2Img) et différentes tailles de *patches* (128 et 256 pixels)

Résumé et analyse des résultats globaux

Cette sous-section propose une analyse complète des résultats globaux obtenus par l'application des méthodes de *clustering* SNMF et spectral pour la détection de falsifications d'images sur les quatre jeux de données : Columbia, DSO-1, DALLE2 et

Img2Text2Img. L'évaluation repose sur un ensemble de métriques de performance (par exemple, F1-score, MCC, IoU, AUC), ainsi que sur des métriques d'efficacité computationnelle, des observations de convergence, et des inspections visuelles telles que les matrices de similarité et les courbes ROC. Ces analyses visent à mettre en évidence les forces et les limites de chaque approche de *clustering*, en fournissant à la fois des perspectives quantitatives et qualitatives sur les défis expérimentaux propres à chaque jeu de données.

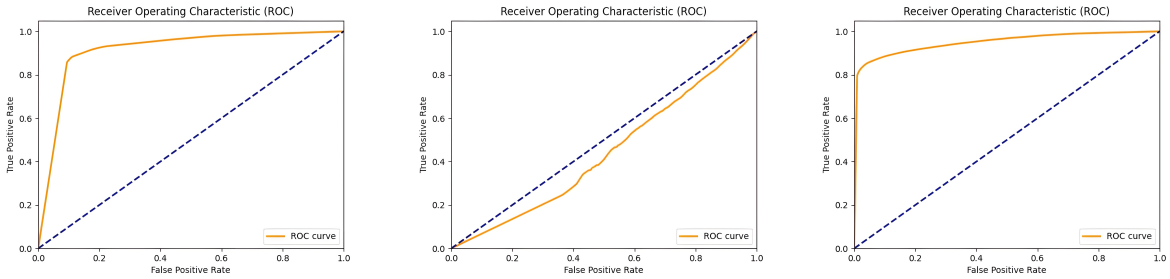
	Columbia	DSO-1	DALLE2	Img2Text2Img	Complexité Algorithmique
Nombre de patchs par image	$n = 70$	$n = 165$	$n = 49$	$n = 225$	
Résolution de l'Image	568×757	1536×2048	1024×1024	1024×1024	
Résolution du Patch	128×128	256×256	256×256	128×128	
Calcul de la Similarité Calcul de la matrice	$\approx$ 20s	$\approx$ 4min	$\approx$ 20s	$\approx$ 3min	$O(n^2 \times \textit{calcul similarit})$
Localisation Pixel-level Temps de calcul	$\approx$ 0.1s	$<$ 1s	$<$ 1s	$<$ 1s	
Clustering Spectral Temps de calcul par image	$\approx$ 0.01s	$\approx$ 0.1s	$\approx$ 0.001s	$\approx$ 0.2s	$O(n^3)$
Clustering SNMF Temps de calcul par image	$\approx$ 1s	$\approx$ 3s	$\approx$ 0.5s	$\approx$ 0.5s	$O(n^2 \times k)$ *par itration*
Ecart-type du pourcentage de pixels transformés	7.05%	6.22%	26.61%	6.00%	
Pourcentage moyen de pixels transformés	20.80%	10.55%	40.95%	7.20%	

TABLE 6.2 – Quelques métriques sur les différents jeux de données et temps de calcul moyens pour une seule image, à différentes étapes. Expériences réalisées sur une machine équipée d'un processeur Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz et d'un GPU GeForce GTX 1080Ti.

Les métriques de performance (F1-score, MCC, IoU et AUC) pour les méthodes de *clustering* spectral et SNMF sont calculées pour différents jeux de données (Columbia, DSO-1, DALLE2, Img2Text2Img) et tailles de *patches* (128 et 256 pixels). Le Tableau 6.1 permet une comparaison quantitative des performances, tandis que le Tableau 6.2 compare les temps de calcul sur l'ensemble des jeux de données. Ces mesures fournissent un éclairage sur la robustesse des méthodes de *clustering* et sur leur capacité à traiter des transformations d'images variées : *splicing* simple pour Columbia et DSO-1, et *deepfake splicing* pour DALLE2 et Img2Text2Img.

Le Tableau 6.2 présente également des données quantitatives sur les caractéristiques des jeux de données, telles que la résolution des images et le nombre de *patches* par image, ainsi que sur les paramètres expérimentaux, y compris les temps de calcul pour les méthodes SNMF et spectrale, et le pourcentage moyen de pixels transformés par image. Ces résultats montrent que certaines caractéristiques des jeux de données influencent fortement les performances. Par exemple, la résolution élevée des images du jeu DSO-1 entraîne des temps de calcul sensiblement plus longs, la construction de la matrice de similarité étant environ 10 fois plus lente que pour les jeux DALLE2 et Columbia. En outre, DALLE2 se distingue par sa forte hétérogénéité : il présente à la fois le pourcentage moyen le plus élevé de pixels transformés et la plus forte variation de ce pourcentage, ce qui rend la détection plus difficile.

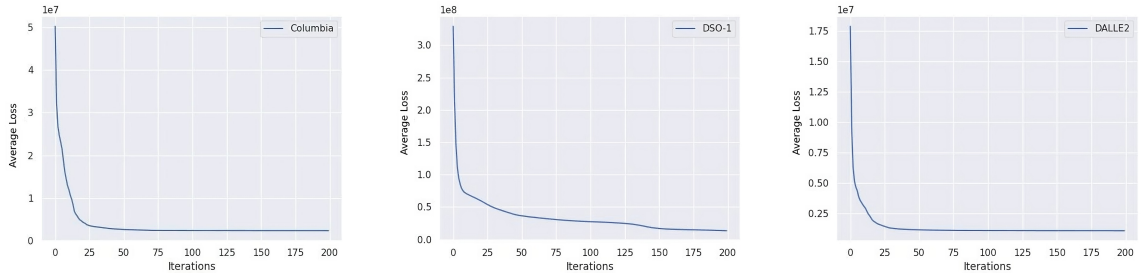
La Figure 6.10 montre les courbes ROC pour le *clustering* SNMF sur trois jeux de données : DSO-1, DALLE2 et Img2Text2Img. Ces courbes illustrent le compromis entre le taux de vrais positifs (TPR) et le taux de faux positifs (FPR) et sont directement liées aux valeurs AUC du Tableau 6.1. Pour DALLE2, les performances sont proches de celles



(a) DSO-1 (b) DALLE2 (c) Img2Text2Img

FIGURE 6.10 – Courbes ROC pour le clustering SNMF sur trois datasets, DSO-1, DALLE2 et Img2Text2Img.

d'un classificateur aléatoire, alors que pour DSO-1 et Img2Text2Img, les résultats sont nettement meilleurs.



(a) Columbia (b) DSO-1 (c) DALLE2

FIGURE 6.11 – Représentation de la perte moyenne $||A - HH^T||_F^2$ (*cf* Equation 5.8 Chapitre 5) à chaque itération de la descente de gradient utilisée pour le clustering SNFM, pour chacun des datasets.

La Figure 6.11 illustre le comportement de convergence de l'algorithme SNMF sur trois jeux de données (Columbia, DSO-1 et DALLE2). Les courbes montrent l'évolution de la fonction de perte $||A-HH^T||_F^2$ au fil des itérations de la descente de gradient. Toutes les expériences convergent en moins de 200 itérations, ce qui témoigne de la robustesse du processus d'optimisation.

Enfin, la Figure 6.12 montre des exemples de matrices de similarité pour trois images (issues respectivement de Columbia, DSO-1 et DALLE2). Comme attendu, la diagonale est maximale (chaque *patch* étant identique à lui-même), tandis que les zones sombres indiquent de fortes dissimilarités, révélant des transformations. Dans un scénario simple avec une seule transformation, la matrice serait quasiment binaire, mais en conditions réelles, la diversité des transformations complique fortement l'interprétation visuelle.

Comparaison des résultats : Le tableau 6.3 compare les résultats que nous obtenons sur le jeu de données Columbia, très standard dans la littérature sur les métriques F1, IoU et AuC. Notre approche, supplante très légèrement celle de Mayer et Stamm qui emploie le clustering spectral pour la détection des communautés.

Synthèse des résultats : Les résultats mettent en évidence à la fois les forces et les limites des méthodes SNMF et spectrale pour la détection de falsifications d'images. Sur les jeux de données simples (Columbia, DSO-1), les deux méthodes donnent de bons résultats, avec un léger avantage pour SNMF sur Columbia et pour le *clustering* spectral sur DSO-1.

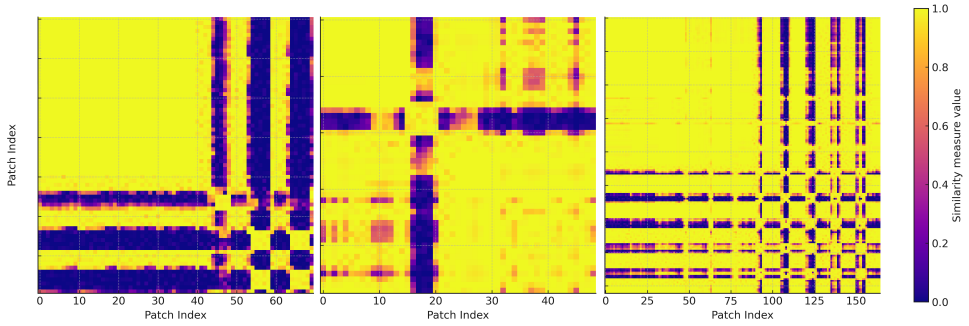


FIGURE 6.12 – Trois matrices de similarité, chacune correspondant à une image de Columbia, DSO-1 et DALLE2

TABLE 6.3 – Comparaison des métriques de localisation des transformations sur le jeu de données Columbia.

Méthode	**F1 (%)**	**IoU (%)**	**AUC (%)**
SPAN [53]	81.4	—	—
ManTraNet [171]	—	—	82.4
ObjectFormer [151]	—	—	**95.5**
PIM [68]	68.0	60.0	—
Spectral (Mayer & Stamm) [89]	89.0	—	95.0
Nous (Spectral)	88.0	78.0	89.0
Nous (SNMF)	**90.0**	**83.0**	94.0

En revanche, DALLE2 présente de grandes difficultés liées à la complexité des transformations et au manque de vérité terrain fiable, ce qui explique les performances faibles, même si SNMF fournit des prédictions pertinentes pour environ un tiers des images. Sur Img2Text2Img, les deux méthodes obtiennent des résultats acceptables, SNMF ayant un léger avantage.

Ces observations confirment que les performances dépendent fortement des caractéristiques du jeu de données (résolution, taille et complexité des transformations). Aucune méthode ne peut être considérée comme strictement supérieure, mais les deux montrent leur capacité à détecter des transformations locales de nature variée.

6.1.12 Conclusion et perspectives

Nous sommes confiants quant à la validité de la procédure générale décrite dans l'Algorithme 2 (*cf* sous-section6.1.3) pour localiser les transformations au sein d'une image. Cette approche est intuitive, puisqu'elle formule la détection de transformations comme un problème de détection d'anomalies, et son efficacité empirique a été démontrée dans cette étude. Nos résultats confirment la validité et la robustesse de la méthode sur des jeux de données variés et pour différents types de transformations.

Nos expériences ne permettent pas d'affirmer la supériorité du *clustering* SNMF par rapport au *clustering* spectral, mais les résultats sont encourageants, en particulier sur DALLE2 où SNMF parvient à extraire des informations qualitatives sur certaines images, contrairement au *clustering* spectral. Nous proposons également un outil de génération d'images modifiées par *deepfake splicing* avec cohérence sémantique, offrant ainsi une ressource utile pour la recherche et l'expérimentation.

Parmi les améliorations possibles, il serait pertinent, dans un contexte d'investigation

forensique, de travailler avec plus de deux classes. Cela permettrait de distinguer différents types de transformations, et non seulement de séparer zones transformées et authentiques (ce qui serait utile pour des images DALLE2 dont l'historique des transformations est inconnu). Algorithmiquement, utiliser plus de deux classes ne change pas le *clustering* des *patches*. Nous pourrions également évaluer le modèle sur davantage de données, par exemple en construisant un jeu d'images transformées par un modèle de diffusion récent comme DALL·E 2 ou par d'autres algorithmes de *deepfake*, tout en veillant à ce que chaque image subisse une seule transformation.

Enfin, une piste d'amélioration consisterait à travailler sur le *clustering* pour mieux interpréter les similarités. étant donné que les méthodes spectrale et SNMF donnent toutes deux de bons résultats, mais parfois différents, une approche de type *ensemble clustering* [138][59] pourrait être envisagée afin d'agréger plusieurs méthodes et d'obtenir un consensus de *clustering* supérieur à chaque méthode prise individuellement.

6.2 Détection des Transformations d'images

Nous disposons à présent d'une méthode de localisation des transformations d'images, très utile à des fins d'explicabilité notamment, lorsque qu'une fake news est constituée d'une image falsifiée. Cependant, avant toute localisation, nous aimerions en plus, obtenir un score de probabilité de transformation. Pour ce faire nous avons tenté dans un premier temps d'extraire des matrices de similarité de Mayer et Stamm [90] des features quantitatives discriminantes pour la détection des images falsifiées. Malheureusement nous démontrons l'échec d'une telle approche. Finalement, pour extraire une probabilité vraiment informative de la falsification d'une image, nous nous appuyons sur ManTraNet [171], une méthode de localisation des transformations d'images moins efficace que la notre, mais dont nous parvenons à extraire une probabilité de falsification pertinente.

Ainsi, dans cette section nous présentons d'abord notre méthode concrète de prédiction des transformations appliquées aux images, basée sur le réseau pré-entraîné de Yue Wu et Natarajan [171]. ManTraNet génère au niveau du pixel, une carte de chaleur (heatmap) prédictive qui quantifie la probabilité de transformation. De cette heatmap, nous extrayons deux *features* principales :

1. **La proportion de l'image suspecte** : c'est la fraction de pixels dont la valeur de heatmap dépasse un seuil fixé (i.e proportion de pixels *chauds*). Interprétable comme la quantité de transformation subie par l'image, répondant à la question "*à quel point l'image a-t-elle été transformée ?*".
2. **La probabilité que l'image soit transformée** : cette probabilité est calculée à partir de la heatmap seuillée, redimensionnée en 512×512, que l'on découpe en patchs de taille patchsize $\times$ patchsize avec un recouvrement de 50%. Si un patch contient plus de 75% de pixels chauds, alors cela augmente fortement la probabilité de transformation. Un seul patch chaud suffit pour obtenir une probabilité élevée, et cette probabilité croît de façon progressive avec le nombre de patchs suffisamment chauds. Cette probabilité permet de répondre à la question "*Est ce que l'image a été transformée ?*".

Nous détaillons dans la suite chacune de ces étapes ainsi que des expérimentations qui quantifient les performances de notre approche et explicitent le choix de cette architecture.

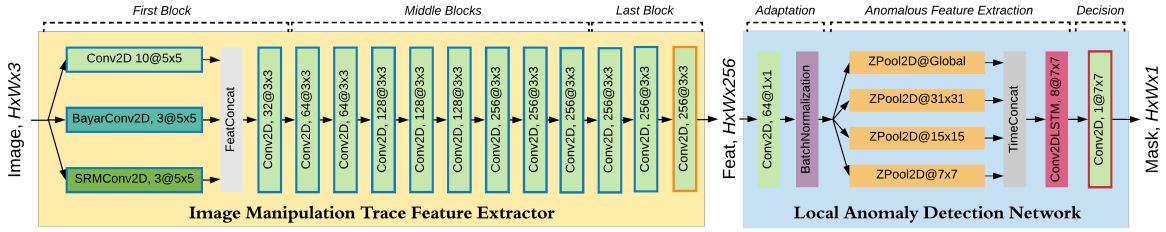


FIGURE 6.13 – Architecture du réseau ManTraNet, composé d'un extracteur de features forensiques, et d'un détecteur d'anomalies locales (schéma tiré de [171]).

6.2.1 ManTraNet : Présentation du modèle

Présentation générale

ManTraNet [171] est un algorithme de détection et localisation de transformation d'images. Il prend en entrée une image et calcule en sortie une heatmap correspondant à une prédiction probabiliste de la zone transformée, au niveau du pixel. ManTraNet est pré-entraîné sur des données synthétiques associées à 385 classes de transformation distinctes. Dans un second temps, le modèle est entraîné sur quatre types de transformations distincts, correspondant aux transformations les plus utilisées : *copy-move*, *splicing*, *removal*, *enhancement*.

Architecture

D'un point de vue architecture, ManTraNet est composé de deux sous-réseaux de neurones :

Extracteur de features CNN-based : (avec un backbone VGG [132]) qui capture de l'information bas niveau (features forensiques) sur la manipulation des images, dans un vecteur de taille fixée.

Réseau détecteur d'anomalie : , qui compare les features forensiques locaux aux features forensiques dominant d'une région locale. Son activation dépend de la différence entre le feature local évalué, et le feature dominant, ainsi plus une zone est transformée, plus l'activation est grande.

6.2.2 Génération de la carte de chaleur avec ManTraNet

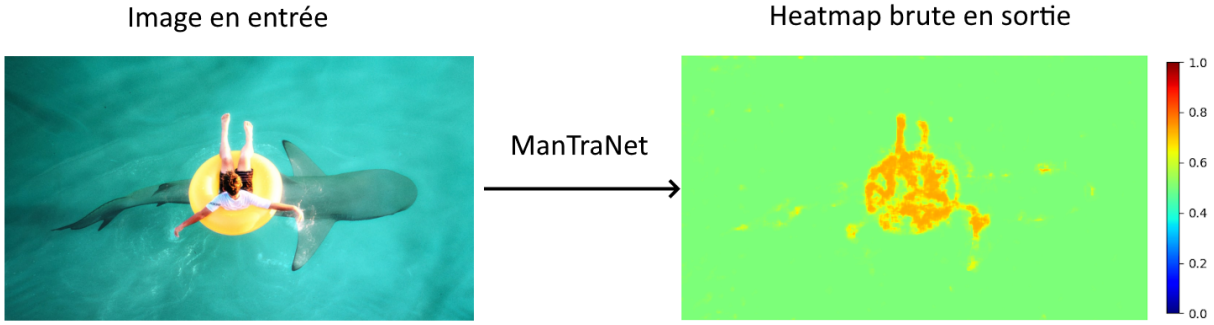


FIGURE 6.14 – Exemple d'entrée et sortie calculée par ManTraNet.

ManTraNet prend en entrée une image de taille arbitraire et produit en sortie une *heatmap* H de même résolution, où chaque pixel $H(i,j)$ encode un score de probabilité (compris entre 0 et 1) que la région locale autour du pixel ait subi une manipulation.

On note l'image d'entrée par $I \in \mathbb{R}^{M\times N\times 3}$ (trois canaux couleur). Lorsque cette image comporte plus de 6×10^5 (taille déterminée empiriquement), nous en effectuons une redimension par interpolation bilinéaire afin de réduire le nombre de pixels sous ce seuil. Cette redimension est d'une part nécessaire du fait de la contrainte VRAM disponible sur le GPU local (Nvidia RTX4060) utilisé dans nos expériences et d'autre part possible car les auteurs de ManTraNet soulignent la robustesse de leur méthode à la compression. Quant à elle, nous notons la heatmap brute produite par ManTraNet :

$$H_{\text{brut}} = \Big(H_{\text{brut}}(i,j)\Big)_{1\le i\le M,\ 1\le j\le N}, \quad H_{\text{brut}}(i,j) \in [0,1].$$

Dans la suite, nous redimensionnons cette heatmap brute pour obtenir une carte H de taille fixe 512×512 afin de calibrer notre processus de détection.

$$H = \Big(H(i,j)\Big)_{1\le i,j\le 512}, \quad 0 \le H(i,j) \le 1.$$

Le but de cette première étape est donc d'obtenir une carte de chaleur de taille standardisée H qui nous servira de base pour calculer la probabilité de transformation.

6.2.3 Extraction de la proportion de l'image suspecte

Nous seuillons la heatmap (seuil $\tau \in [0,1]$) pour la normaliser et mettre en exergue les zones transformées. Empiriquement le seuil de 0.6 permet de bien discriminer les zones authentiques des zones transformées. Soit

$$B(i,j) \;=\; \begin{cases} 1, & \text{si } H(i,j) \;>\; \tau, \\ 0, & \text{sinon.} \end{cases}$$

Le résultat $B \in \{0,1\}^{512\times 512}$ est une carte binaire où chaque pixel $B(i,j) = 1$ est qualifié de *pixel chaud* (suspect).

Feature 1 : proportion de l'image suspecte. On note la proportion de pixels chauds par

$$S_\tau \;=\; \frac{1}{512 \times 512} \sum_{i=1}^{512} \sum_{j=1}^{512} B(i,j)\,. \tag{6.5}$$

Ainsi, S_τ correspond à la fraction de l'image considérée comme suspecte.

6.2.4 Calcul de la probabilité que l'image soit transformée

Si la simple proportion S_τ informe sur la quantité globale de pixels suspects, elle ne renseigne pas pleinement sur la répartition spatiale de ces pixels. Or, la transformation d'une image se traduit généralement par l'altération de zones connexes de l'image. En particulier, quelques pixels chauds répartis aléatoirement peuvent entraîner une même valeur de S_τ qu'un petit bloc concentré de pixels chauds, alors que ce dernier signe davantage une manipulation localisée. Nous utilisons cette idée pour calculer un estimateur de la probabilité de transformation de l'image.
D'abord nous découpons la heatmap seuillée en patchs carrés et analysons la densité de pixels chauds dans chaque patch, dans le but de passer d'une information pixel-level, à une information image-level sur la transformation de l'image.

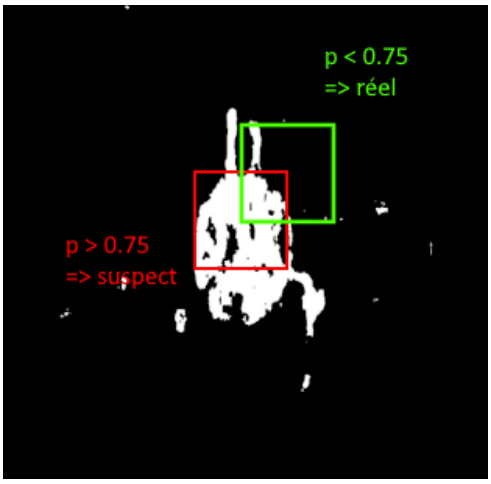


FIGURE 6.15 – Exemple d'un patch détecté comme suspect (en rouge) et d'un patch détecté comme réel (en vert) sur une heatmap calculée par ManTraNet

Construction des patchs avec recouvrement

Nous disposons de la carte binaire $B \in \{0,1\}^{512\times512}$. On souhaite la découper en patchs (sous-régions) de taille patchsize × patchsize avec un recouvrement de 50% en chaque dimension. Cela revient à fixer un pas (stride) de 32 pixels. Des patchs trop petits sont trop sensibles au bruit et augmentent par conséquent le nombre de faux-positifs. A l'inverse, des patchs trop grands sont trop insensibles aux transformations locales et augmentent le nombre de faux négatifs.

Détection des patchs suspects

Soit n_h, n_v les nombres de patchs horizontaux et verticaux, pour chaque patch $P_{k,\ell}(1 \leq k \leq n_v, 1 \leq \ell \leq n_h)$, on calcule la proportion de pixels chauds (pixels suspects) à l'intérieur du patch :

$$p_{k,\ell} = \frac{1}{\text{patchsize} \times \text{patchsize}} \sum_{(i,j)\in P_{k,\ell}} B(i,j). \tag{6.6}$$

On qualifie alors un patch de suspect si

$$p_{k,\ell} > 0{,}75,$$

c'est-à-dire si plus de 75% des pixels du patch sont chauds. On définit alors l'indicateur binaire

$$C_{k,\ell} = \begin{cases} 1, & \text{si } p_{k,\ell} > 0{,}75, \\ 0, & \text{sinon.} \end{cases}$$

On note alors

$$N = \sum_{k=1}^{n_v} \sum_{\ell=1}^{n_h} C_{k,\ell} \tag{6.7}$$

le nombre total de patchs de l'image considérés comme suspects.

Définition de la probabilité P

Pour estimer la probabilité de transformation P, l'idée est que la présence d'au moins un patch très chaud (i.e. $N \geq 1$) constitue un signe fort d'une altération généralisée de l'image. De plus, plus le nombre N de patchs chauds est élevé, plus la probabilité d'une

transformation est grande. On propose donc de calculer une probabilité P sous la forme suivante :

$$P \;=\; 1 - \exp\Big(-\alpha\, N\Big), \tag{6.8}$$

où $\alpha > 0$ est un paramètre de sensibilité. Cette forme exponentielle garantit que dès $N = 1$, la probabilité P se rapproche rapidement de 1 si α est grand, et qu'elle augmente de façon progressive lorsque N croît.

Une variante possible est de tenir compte non seulement du nombre de patchs très chauds, mais aussi du niveau moyen de chaleur dans les patchs chauds. Pour cela, on peut introduire la quantité

$$\overline{p}_{\text{chaud}} \;=\; \frac{1}{N} \sum_{k,\ell\,:\,C_{k,\ell}=1} p_{k,\ell} \tag{6.9}$$

lorsque $N \geq 1$. Si $N = 0$, on pose $\overline{p}_{\text{chaud}} = 0$ par convention. On peut alors définir par exemple :

$$P \;=\; 1 - \exp\Big(-\big(\alpha\, N + \beta\, \overline{p}_{\text{chaud}}\big)\Big), \tag{6.10}$$

où $\alpha, \beta > 0$ sont deux paramètres à ajuster. Cette seconde formule fait croître P plus vite si les patchs chauds correspondent à une chaleur moyenne $(p_{k,\ell})$ élevée.

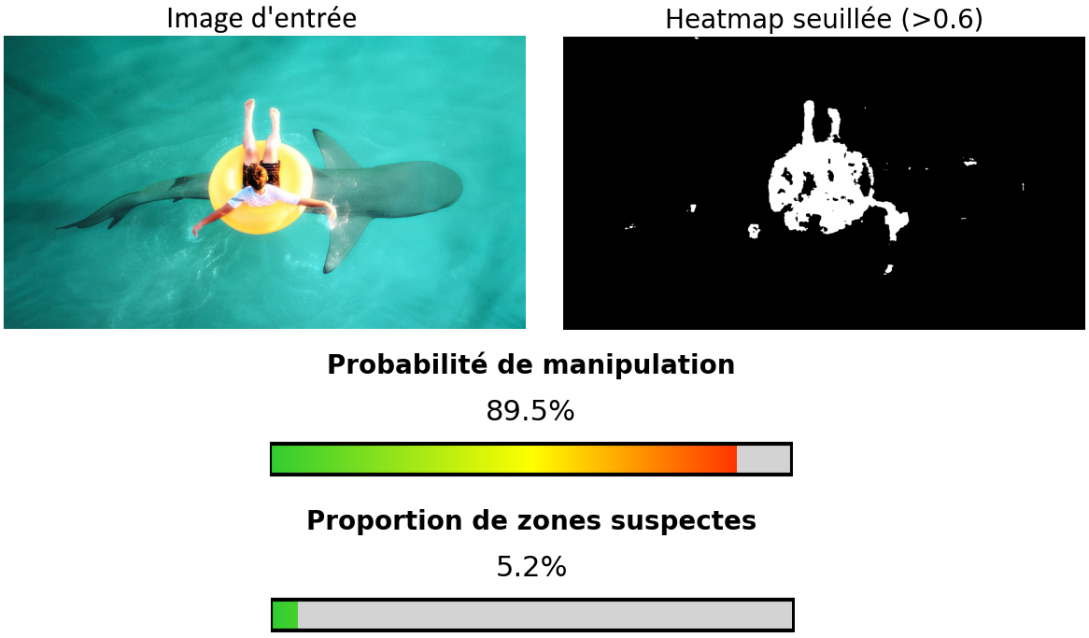


FIGURE 6.16

FIGURE 6.17 – Illustration d'une image d'entrée (a) et de la heatmap brute (b) générée par ManTraNet.

6.2.5 Recherche des hyperparamètres

Pour déterminer les valeurs optimales de α, β et patchsize, nous adoptons une stratégie de Grid-search :

Grid search : On explore d'abord l'espace via une grille :

$$(\alpha,\ \beta,\ \text{patchsize}) \;\in \{0.1, \ldots, 1\}^2 \times \{16, 32, \ldots, 96\}$$

soit 500 combinaisons. Pour chaque triplet, $(\alpha, \beta, \text{patchsize})$, on mesure l'accuracy, le F1-score et l'AUC sur le même dataset (Columbia). On conserve finalement les paramètres : $\alpha = 0.5, \beta = 0.6$ et patchsize $= 64$.

Puisque la détection s'effectue sur des heatmaps standardisées, fixer ces hyperparamètres est pertinent, peu importe le dataset. Cependant, la Grid-Search pourrait être affinée, par exemple en augmentant la précision des hyperparamètres autour d'un maximum local des métriques.

6.2.6 Expérimentations

Dans cette sous-section, nous présentons nos expériences de détection d'images falsifiées ou transformées sur trois jeux de données, en comparant la méthode MantraNet et deux variantes basées sur les algorithmes de localisation par clustering spectral/SNMF (*cf.* section 5.2). Nous complétons ces expériences par l'essai d'une approche de machine learning classique de classification de type SVM basée sur des descripteurs extraits des matrices de similarité (*cf.* section 5.2 *fig.* 6.12).

Evaluation de ManTraNet pour la détection

Dans cette expérience nous évaluons trois méthodes distinctes de détection des transformation d'images. Chacune des trois correspond à la procédure décrite dans la sous-section 6.2.4 pour le calcul de la probabilité P de transformation. Cependant elles diffèrent par la méthode de localisation sur laquelle se base la heatmap. Ainsi nous testons des méthodes basées sur (i) ManTraNet, (ii) Clustering Spectral, (iii) Clustering SNMF.

Jeux de données : Nous testons nos méthodes sur trois datasets distincts, Columbia et DSO-1, déjà présentés dans ce travail (*cf.* Chapitre 2 et Section 6.1) et DGM^4 un dataset de fake news multimodales que nous présenterons en détail dans le chapitre dédié à notre architecture sur la détection des fake news.

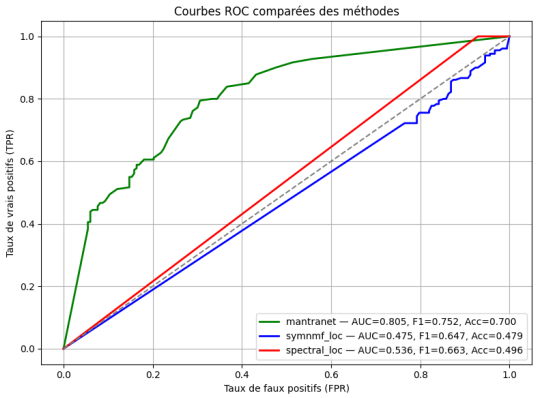
6.2.7 Résultats et analyse

Courbes ROC sur la détection fake/authentique : Sur Columbia et DSO (Fig. 6.18a et c), la méthode ManTraNet atteint des AUC de l'ordre de 0.80-0.70. Les approches basées sur la localisation par clustering spectral ou SymmFNMF échouent systématiquement (AUC $\approx$ 0,50). L'échec des méthodes de localisation basées sur la mesure de similarité forensique était attendu, en effet dans le cas des images authentiques, des zones de l'images sont souvent séparées en deux clusters, malgré leur similarité. Ainsi les méthodes de clustering SNMF et Spectral pour la localisation ne se comportent pas vraiment comme des heatmap de probabilité de transformation comme ManTraNet (*cf.* Figure 6.19).

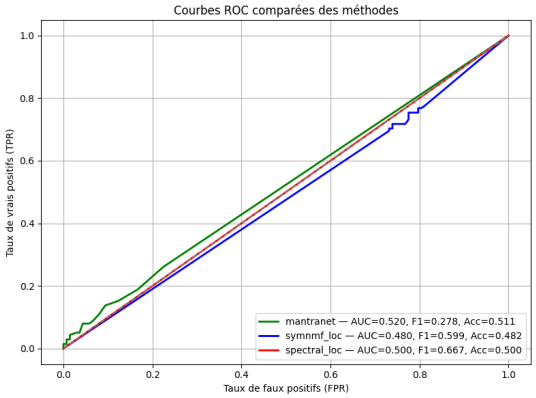
En revanche, sur DGM (Fig. 6.18b), aucun des algorithmes ne dépasse un AUC de 0.52, soulignant la difficulté de notre approche à capturer un signal discriminant pour la détection des images transformées sur des datasets plus complexes.

6.2.8 Distribution des features de clustering

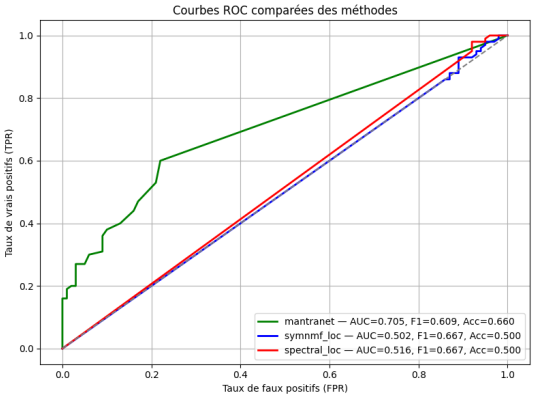
Nous testons également une méthode de détection des images transformées, à partir de features quantitatifs extraits des matrices de similarité forensique, suivant une idée analogue à Mayer et Stamm [90]. Nous choisissons des features/métriques potentiellement capables de discriminer les images fakes des images authentiques. Ainsi nous calculons d'abord les métriques suivantes :



(a) Columbia



(b) DGM



(c) DSO

FIGURE 6.18 – Comparaison des performances (ROC) des trois variantes sur chaque jeu de données. La courbe verte correspond à ManTraNet, les courbes rouge et bleue aux localisations Spectrale et SNMF respectivement.

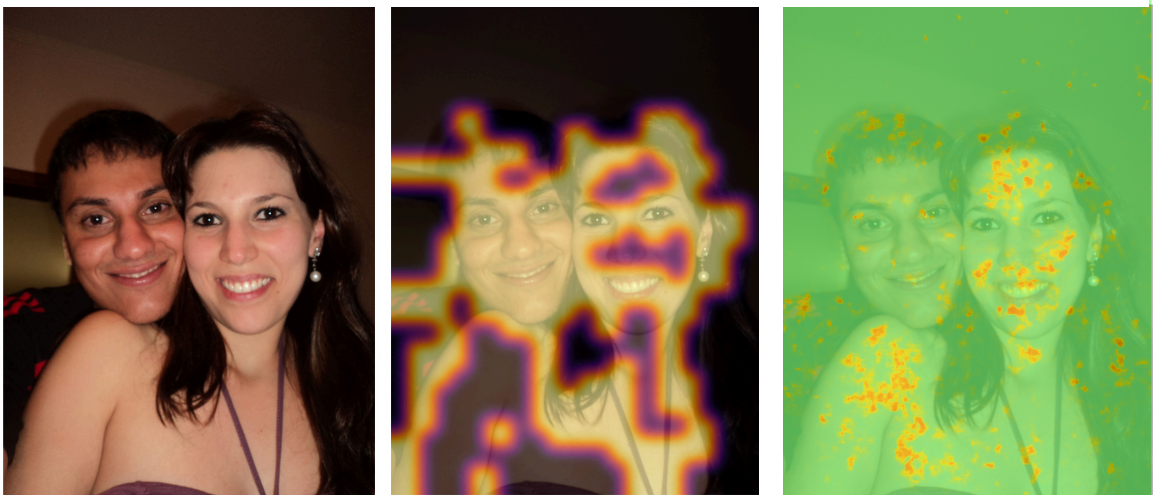


Image authentique | Masque de prédiction Clustering Spectral | Masque de prédiction ManTraNet

FIGURE 6.19 – Exemple de localisation basée sur le Clustering Spectral et sur ManTraNet sur une image authentique. On observe que malgré l'authenticité, le clustering spectral force la séparation en deux clusters significatifs, rendant difficile le calcul d'un score de probabilité pour la détection. Ce n'est pas le cas de ManTraNet, qui affiche une heatmap relativement uniforme sur toute l'image authentique.

λ_2 **(Valeur propre de Fiedler) :** Cette métrique correspond à la deuxième plus petite valeur propre du Laplacien normalisé du graphe construit à partir de la matrice de similarité. Elle mesure la connectivité globale du graphe : une valeur élevée indique une structure fortement connectée, tandis qu'une valeur faible reflète une séparation plus marquée entre sous-graphes.

Gap spectral : Le *gap spectral* est défini comme la différence $\lambda_3 - \lambda_2$ entre la troisième et la deuxième plus petite valeur propre du Laplacien. Un gap important suggère une séparation claire entre deux clusters principaux et le reste de la structure, renforçant la stabilité de la partition.

Coefficient de silhouette : Le coefficient de silhouette évalue la qualité du partitionnement spectral obtenu (2 clusters ici). Il compare, pour chaque élément, la distance moyenne intra-cluster à la distance moyenne vers le cluster le plus proche. La valeur varie entre -1 et 1, une valeur proche de 1 indiquant des clusters bien séparés et compacts.

Normalized Cut (NCut) : La métrique NCut calcule la somme des coupes normalisées (*cf.* équation 5.3) entre deux partitions du graphe. Une valeur faible indique une séparation efficace avec peu de connexions inter-clusters.

Ratio de tailles de clusters : Le ratio est défini comme le rapport $\frac{\min(|C_1|,|C_2|)}{|C_1|+|C_2|}$ entre la taille du plus petit cluster et le nombre total de nœuds. Il mesure l'équilibre entre les partitions : une valeur proche de 0.5 correspond à deux clusters de tailles similaires, tandis qu'une valeur faible signale un fort déséquilibre.

Modularité : La modularité Q est une mesure de la qualité d'un partitionnement des nœuds d'un graphe en communautés. Elle compare la proportion d'arêtes intra-communauté observée à celle attendue si les arêtes étaient distribuées aléatoirement, tout en conservant le degré de chaque nœud. Sa valeur est comprise entre -0.5 et 1, et un score supérieur à 0.3 est généralement considéré comme révélateur d'une structure communautaire significative. La modularité se définit comme :

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

où A_{ij} est le poids de l'arête entre i et j, k_i et k_j sont les degrés des nœuds i et j, m est le nombre total d'arêtes pondérées du graphe, c_i est la communauté du nœud i, et $\delta(c_i, c_j) = 1$ si i et j appartiennent à la même communauté, 0 sinon.

SVM sur features statistiques : classifieur linéaire entraîné sur six mesures graphiques (λ_2, gap, silhouette, ncut, ratio, modularité) issues des matrices de similarité.

Histogramme des descripteurs : Nous estimons les distributions propres à chaque classe (image authentique/fake) pour chacune des métriques. Des distributions significativement différentes indiquent une potentielle discrimination interclasse exploitable pour la détection. Cependant, les distributions des six métriques (Fig. 6.20) montrent un fort recouvrement entre les deux classes, suggérant une faible capacité discriminative de ces seules features.

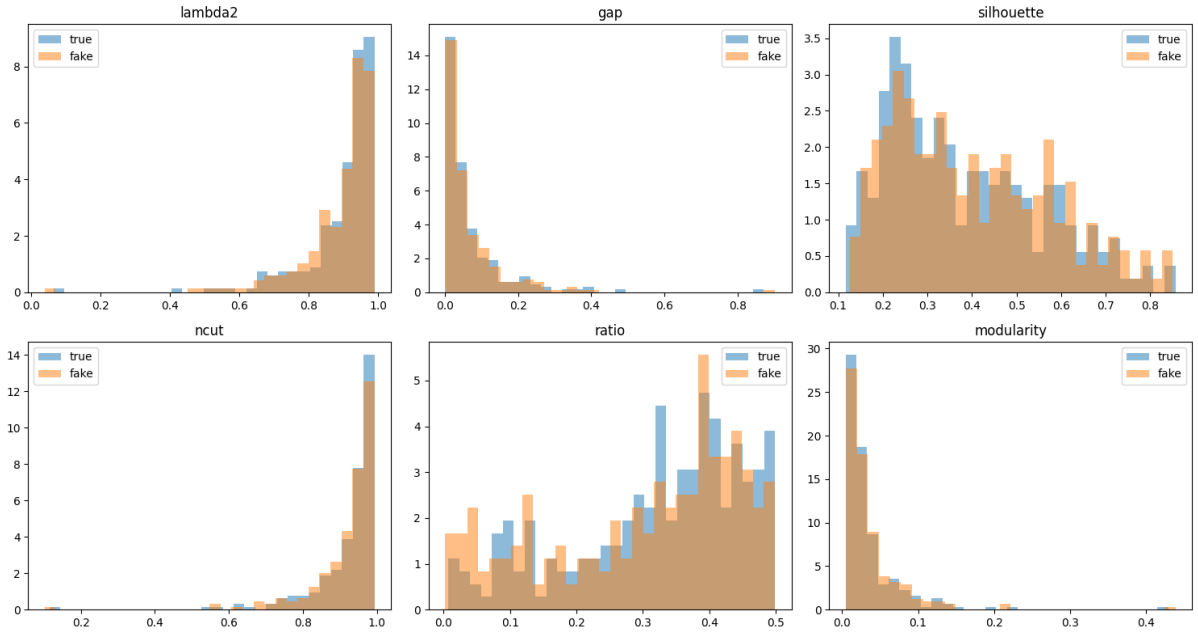


FIGURE 6.20 – Histogrammes des descripteurs (λ_2, gap, silhouette, ncut, ratio, modularité) pour images authentiques (*blue*) et falsifiées (*orange*) du jeu DSO.

Classification SVM sur les features : Afin de confirmer cette intuition, nous entraînons un SVM sur ces six descripteurs, sur le jeu de données DSO-1 pour la détection. Ainsi, l'approche SVM conduit à une AUC de seulement 0.535 sur DSO-1 (Fig. 6.21), confirmant que ces features ne suffisent pas à séparer les images authentiques des falsifiées.

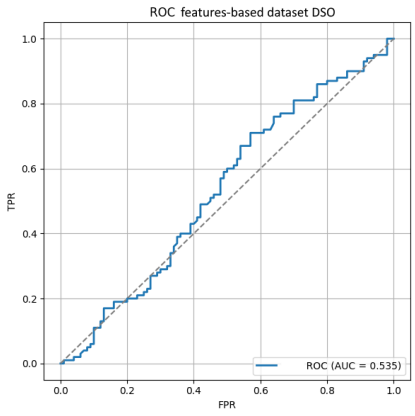


FIGURE 6.21 – ROC de la classification SVM entraînée sur les six descripteurs pour le jeu DSO (AUC = 0.535).

6.2.9 Discussion et perspectives

Les résultats obtenus mettent évidence la capacité de notre approche basée sur ManTraNet à détecter les transformations d'images. Cependant, face à des jeux de données complexes, avec des transformations subtiles (ex. jeu de données DGM[4]), la détection est plus difficile. En outre, nous avons montré qu'il est difficile de construire une méthode de détection efficace basée sur des features statistiques liés aux matrices de similarité forensiques. Plusieurs pistes peuvent être étudiées pour améliorer la détection des transformations d'images : Nous pouvons affiner notre méthode basée sur ManTraNet,

mieux classifier les heatmaps qu'en utilisant l'hypothèse de connexité sur les transformations locales d'images. Nous pouvons également exploiter les mêmes idées développées ici, sur d'autres heatmaps de localisation des transformations d'images, pré entraînées sur d'autres architectures que ManTraNet. Nous n'avons pas trouvé de telles architectures publiques et fonctionnelles. Nous avons notamment effectué des tests avec le modèle HiFi-IFDL [47] qui propose une classification hiérarchique des transformations d'images. Cependant, malgré un code public, nous n'avons pas réussi à obtenir de meilleures performances que le hasard. L'autre approche possible est de reprendre depuis zéro une méthode de détection des transformations d'images. Il s'agit d'un travail colossal et fort intéressant, que nous n'aborderons pas dans cette thèse.

6.3 Conclusion sur la détection et la localisation

Dans ce chapitre nous avons défini deux méthodes, une pour la localisation des transformations d'images, et une pour la détection. La localisation des transformations d'images sert notre pipeline globale de détection des fake news à des fins d'explicabilité, une image suspecte peut ainsi être complétée d'un masque de localisation qui présente à l'utilisateur les raisons de la suspiction. Quant à la détection, nous avons présenté un protocole complet pour passer d'une image brute à deux features scalaires permettant de prédire ' si l'image a subi une transformation ou non. Ces features peuvent également être utiles pour l'explicabilité ou servir à l'entraînement d'un classifieur supervisé dans un système de détection plus large. C'est précisément ce que nous faisons dans la suite. Nous utilisons la probabilité de transformation d'image comme signal d'entrée dans notre pipeline de détection de fake news.

Troisième partie

Intégration et Applications : Système de Détection de Fake News Multimodales

Dans cette partie nous présentons la contribution finale de cette thèse : un modèle de détection de fake news multimodales s'appuyant sur des graphes de connaissances et la détection et localisation des transformations d'images. Nous détaillons dans le Chapitre 7 l'architecture globale de notre approche, dont la structure s'inscrit dans la continuité des modèles exhibés dans notre *Etat de l'Art* (*cf.* Chapitre 2), et dont le fonctionnement est commandé par les outils théoriques et applicatifs développés au cours de notre Partie II. Dans le Chapitre 8 nous procédons à plusieurs expérimentations de notre modèle et nous les discutons. Certaines sont structurantes (étude d'ablation), d'autres servent à l'appréciation qualitative et quantitative du modèle. Quant à lui, le Chapitre 9 synthétise les conclusions, discussions et perspectives de recherche, liées à ce travail de thèse.

7 Architecture Globale

Dans ce chapitre nous présentons notre architecture multimodale ainsi que chacun des modules la composant. Afin de prédire l'authenticité ou non d'une information constituée d'un texte et d'une image, notre objectif est d'intégrer la probabilité de transformation d'une image (*cf.* 6) ainsi que des connaissances externes pour enrichir le contexte de l'information. Nous ajoutons en outre, une mesure de cohérence sémantique texte image, à même de quantifier la similarité entre les informations textuelles et visuelles.

En premier lieu, nous présentons l'architecture complète de notre modèle dans la Section 7.1. Nous détaillons ensuite chacun des modules : l'intégration visuelle dans la Section 7.2, le module *knowledge-based* utilisant des graphes de connaissances dans la Section 7.3, et le score de cohérence visuelle et textuelle dans la Section 7.4. Enfin nous présentons le module d'apprentissage des représentations des graphes, constitués de deux R-GCNs et la fusion multimodale finale pour la prédiction dans la Section 7.5.

7.1 Présentation de l'Architecture Complète

La Figure 7.1 illustre l'architecture complète de notre pipeline multimodale, knowledge-based et explicable pour la détection des fake news. Une information en entrée est constituée d'un **texte** et d'une **image**, à partir desquels sont extraits quatre signaux distincts :

1. **Probabilité de transformation de l'image** : issue du module de détection de falsification présenté au Chapitre 6, elle reflète la vraisemblance qu'une image ait subi une modification locale (flou, splicing, copy-move etc.).
2. **Score de cohérence texte-image** : mesure de similarité sémantique permettant d'évaluer l'alignement entre les informations véhiculées par la modalité textuelle et la modalité visuelle.
3. **Graphe de connaissances basé sur le texte** : représentation structurée des entités et relations extraites du contenu textuel, permettant de raisonner sur des faits et des connaissances externes.
4. **Graphe de connaissances basé sur l'image** : représentation des entités et relations issues de l'analyse visuelle (via génération de graphe de scène *cf.* Chapitre 4), offrant une description sémantique de la scène représentée et des connaissances sur les entités en jeu.

Ces quatre signaux sont ensuite intégrés dans un module de fusion multimodale réalisée par une concaténation pondérée des différentes représentations. Les coefficients associés à chaque modalité sont appris conjointement avec le modèle, ce qui permet d'ajuster automatiquement la contribution relative de chaque signal. Il ne s'agit pas d'une forme d'attention au sens strict, mais d'un mécanisme de pondération paramétrique, plus simple à mettre en œuvre mais visant la même fin. Les représentations ainsi pondérées sont ensuite enrichies au moyen de deux couches entièrement connectées, qui assurent la fusion multimodale.

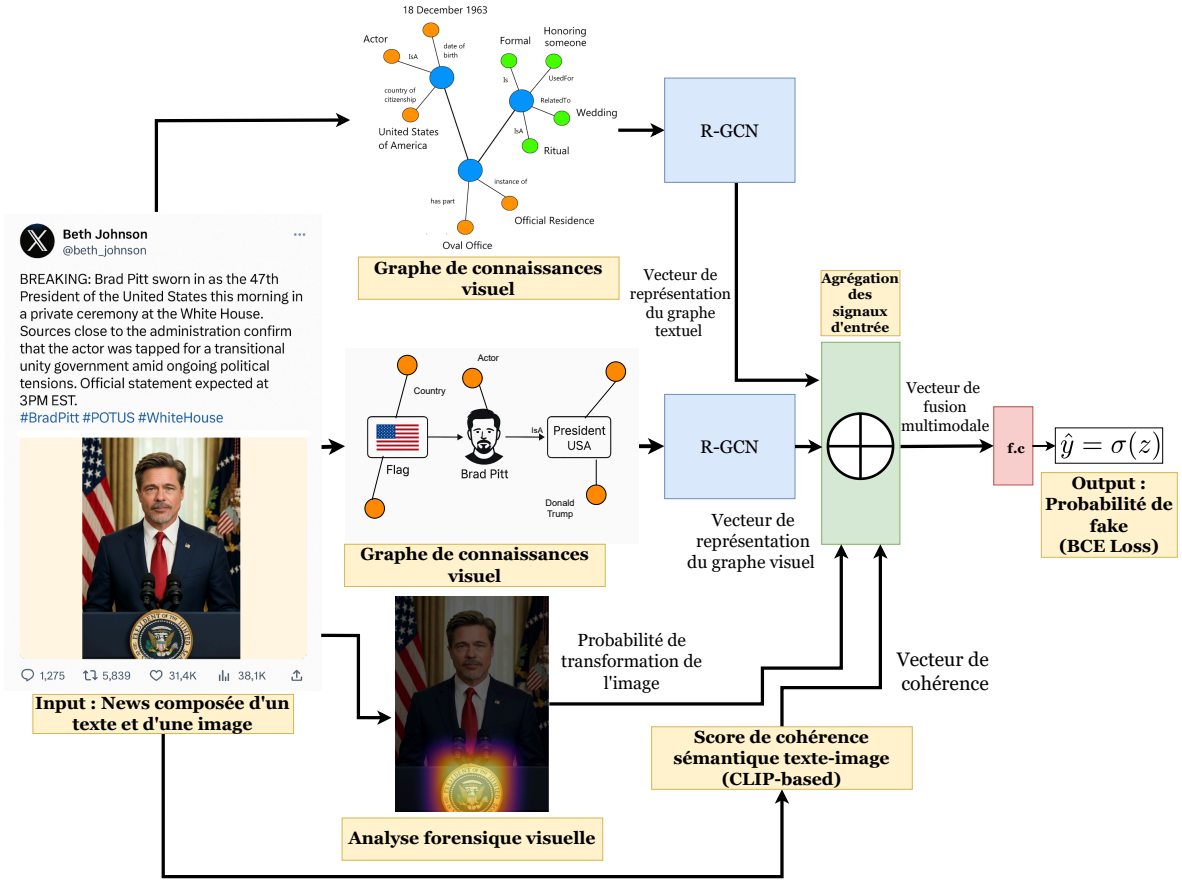


FIGURE 7.1 – Architecture complète de notre modèle de détection des fake news. Une information en entrée est composée d'une image et d'un texte desquels sont extraits 4 signaux d'entrées : La probabilité de transformation de l'image, un score de cohérence texte-image, un graphe de connaissances basés sur le texte et un basé sur l'image. Ces signaux sont agrégés à l'aide d'un module de pondération. Puis leur représentation est enrichie par deux couches entièrement connectées. Une fonction sigmoïde est appliquée en sortie pour la prédiction binaire finale.

Enfin, une fonction sigmoïde est appliquée sur la sortie du réseau pour produire une probabilité de falsification de l'information. Une valeur proche de 1 indique une information considérée comme fake, tandis qu'une valeur proche de 0 correspond à une probabilité élevée d'authenticité. L'organisation modulaire du pipeline présente deux avantages majeurs : (i) chaque signal peut être analysé indépendamment, offrant ainsi une capacité d'explication fine des prédictions (ii) la structure est extensible et permet l'ajout ou le retrait de modalités ou de modules selon les besoins de l'application.

7.1.1 Motivations et choix de conception :

La conception de notre architecture multimodale pour la détection des fake news repose sur plusieurs idées directrices.

Représentation de l'information par des graphes de connaissances : Dans un premier temps, nous avons relevé que des modèles comme DEAP-FAKED [87] (Chapitre 2) sont performants même à partir d'une quantité de texte restreinte, enrichie par des bases de connaissances externes. Ces travaux nous ont convaincu de l'intérêt d'un processus d'extraction d'informations depuis le texte suivi d'un enrichissement de connaissances

(à calibrer) pour une meilleure compréhension *contextuelle* de la sémantique du texte. De même, pour l'explicabilité, des travaux tels que FACE-KEG [145] renforcent notre conviction quant au potentiel des graphes de connaissances pour fournir des justifications structurées et interprétables des prédictions. Enfin, comme discuté au Chapitre 4, la génération de graphes de scènes nous conduit à représenter également la modalité visuelle sous forme de graphe. Cette représentation permet de capturer efficacement l'information sémantique contenue dans l'image, et de l'exploiter conjointement avec les connaissances issues du texte.

Deux modalités, deux graphes : Une fois l'image et le texte convertis en graphes, plusieurs choix architecturaux sont envisageables. Par exemple, il serait possible d'agréger le graphe de scène et le graphe de connaissances textuel au sein d'un *unique* graphe combinant l'information textuelle et visuelle. Nous avons toutefois choisi de conserver deux graphes de connaissances distincts, et d'apprendre séparément leur représentation vectorielle au moyen de deux GNNs indépendants.

Ce choix se justifie pour plusieurs raisons. Tout d'abord, l'unification des graphes complexifie l'apprentissage : un graphe unique contient davantage de nœuds, rendant l'optimisation d'un GNN plus difficile et plus coûteuse en ressources, tant en phase d'entraînement qu'en inférence. En particulier, cette configuration nécessite davantage de données d'apprentissage pour capturer correctement la structure et les interactions entre modalités.

Ensuite, maintenir l'information sur la modalité d'origine de chaque nœud dans un graphe unifié impose d'ajouter des attributs spécifiques aux nœuds, ce qui alourdit la structure, ou de représenter cette modalité comme une relation, ce qui introduit de nouveaux nœuds et modifie la structure initiale du graphe.

Enfin, notre décision est également guidée par l'efficacité démontrée, dans l'état de l'art, des architectures multimodales qui traitent séparément les modalités textuelle et visuelle, sans alignement préalable, et qui réalisent la fusion uniquement à un stade ultérieur du traitement.

Usage de la probabilité de transformation locale : Comme détaillé tout au long de ce manuscrit, les fake news exploitent fréquemment des images ayant subi des altérations locales, qu'il s'agisse de montages, de retouches ou de suppressions ou intégration d'éléments. C'est pourquoi nous intégrons dans notre pipeline le score de probabilité de transformation locale issu du module basé sur ManTraNet [171] et présenté au Chapitre 6, afin de quantifier le degré de suspicion d'une falsification visuelle de l'image. Ce score fournit un signal complémentaire aux indicateurs sémantiques du modèle (cohérence texte-image, graphes de connaissances). Contrairement aux informations contenues dans les graphes de connaissances, ce score reflète une compréhension bas niveau de l'image, dans laquelle se manifeste les traces forensiques laissées par une potentielle édition de l'image. En outre, il renforce l'explicabilité du système en permettant la visualisation des zones suspectes identifiées, offrant ainsi à l'utilisateur une justification visuelle des prédictions.

Score de cohérence textuelle-visuelle : Parfois, les fake news contiennent des images réelles mais sorties de leur contexte, ou encore associées à des descriptions textuelles trompeuses. Afin de détecter ce type d'incohérence entre les deux modalités, nous intégrons un score de cohérence sémantique texte-image. Plusieurs travaux ont démontré qu'une mesure de similarité adéquate est capable de discriminer les fake news des authentiques. Ce score agit ainsi comme un signal complémentaire aux indices visuels et contextuels, et vise à contribuer également à l'explicabilité des prédictions.

Utilisation des GNNs pour la représentation vectorielle des graphes de connaissances : Pour exploiter pleinement la structure des graphes de connaissances dans un modèle d'apprentissage, il est nécessaire de les convertir en représentations vectorielles préservant à la fois leur structure topologique et leur sémantique. Les *Graph Neural Networks* (GNNs) constituent un paradigme particulièrement adapté à cette tâche, car ils propagent et agrègent l'information entre nœuds en fonction de la structure du graphe, permettant ainsi de capturer les dépendances locales et globales entre entités. Notamment, nous avons montré dans le Chapitre 3 que les GCNs sont simples et efficaces pour traiter des graphes en tirant partie de leur structure. En particulier les variantes relationnelles telles que les R-GCNs, sont efficaces pour encoder des graphes hétérogènes et multi-relationnels comme nos graphes de connaissances. Dans notre architecture, l'utilisation de R-GCNs permet non seulement de générer des embeddings riches pour chaque graphe, mais aussi d'intégrer de manière explicite les relations entre entités, renforçant ainsi la capacité du modèle à raisonner sur des informations complexes.

7.2 Intégration de la Prédiction Visuelle

Du Chapitre 6 nous avons tiré trois indicateurs qui peuvent être utiles à la détection des fake news : (i) La probabilité de transformation de l'image (ii) La proportion de pixels suspects (iii) Une carte de localisation des zones transformées

Proportion de pixels suspects : Dans notre architecture nous n'utilisons pas la proportion de pixels suspects pour la prédiction. En effet, nous considérons qu'il ne s'agit pas là d'une métrique particulièrement informative. L'information transformée ne dépend pas de manière évidente de la surface altérée. On pense par exemple à un visage soumis à du splicing ou du deepfake, selon le cadrage de l'image la surface visée est très variable et l'information altérée est constante. En outre cette métrique est assez bruitée, en effet, elle ne prend pas en compte l'hypothèse de connexité des zones transformées. Ainsi, une heatmap produite par MantraNet sur une image (authentique ou non) produit souvent de nombreuses zones éparses de pixels suspects, qui ne contribuent pas à la probabilité de falsification de l'image mais significativement à la proportion de pixels suspects. Cependant à des fins d'explicabilité sur une image effectivement transformée, nous pouvons utiliser cette métrique pour quantifier a posteriori la falsification de l'information.

Carte de localisation des zones transformées : Nous disposons également d'une carte de localisation des zones transformées, que nous n'employons pas non plus pour la détection. De fait, nous avons mis en évidence dans le Chapitre 6 la difficulté de notre méthode de localisation basée sur une mesure de similarité forensique [90] et le clustering SNMF [71] à être étendue dans le cadre de la simple détection. C'est pourquoi nous utilisons cet indicateur uniquement après la détection, pour améliorer l'explicabilité de notre méthode.

Probabilité de transformation de l'image : Nous utilisons directement la probabilité de transformation de l'image calculée dans le Chapitre 6 comme variable d'entrée $h_{\text{fake-img}} \in [0, 1]$ dans le module d'agrégation des signaux car il constitue un indicateur direct de falsification visuelle et factuelle.

Synthèse Parmi les trois indicateurs issus du module de détection et localisation de transformations d'images, seule la *probabilité de transformation* est utilisée comme signal

direct dans notre pipeline multimodal. Les deux autres, à savoir la proportion de pixels suspects et la carte de localisation, ne contribuent pas directement à la classification, mais jouent un rôle important dans l'explicabilité du modèle en fournissant une interprétation visuelle des prédictions. Ainsi, notre approche combine un signal synthétique exploitable en détection avec des indicateurs visuels plus détaillés qui renforcent la transparence et l'interprétabilité des résultats.

7.3 Enrichissement sémantique et par Graphes de Connaissances

Notre architecture multimodale intègre un module *knowledge-based* qui représente, sous forme de graphes de connaissances, l'information extraite du texte et de l'image associée à une publication. Ces graphes constituent une structure explicite et interprétable, adaptée à la vérification d'information, car ils permettent de raisonner non seulement sur les entités présentes, mais aussi sur les relations qui les relient.

Dans cette section nous explicitons l'intégration au modèle de ces deux graphes qui repose sur les étapes suivantes :

1. **Extraction automatique de faits** à partir du contenu textuel et visuel, sous la forme de triplets `<sujet,relation,objet>`.
2. **Construction des graphes de connaissances**, organisés séparément pour les modalités textuelle et visuelle, afin de produire deux représentations pertinentes pour la vérification d'information.
3. **Encodage initial des graphes** au moyen d'embeddings translationnels (par exemple, TransE, cf. Section 3.3), qui projettent les entités et relations dans un espace vectoriel continu.
4. **Propagation contextuelle des représentations** grâce à un réseau R-GCN (cf. Section 3.6.3), permettant de capturer les dépendances locales et globales entre entités.
5. **Pooling global** du graphe afin de produire un vecteur unique pour chaque graphe, exploité par la suite dans le module d'agrégation des signaux.

Cette modélisation *knowledge-based* structure l'information dans un espace où la plausibilité d'une information peut être évaluée en fonction de faits connus et de relations établies. Elle constitue une composante essentielle de l'explicabilité de notre pipeline, en rendant possible la justification des prédictions au travers de structures interprétables et participent aussi largement à la détection, d'un point de vue quantitatif.

7.3.1 Construction du graphe de connaissances à partir du texte

Nous présentons étape par étape la procédure suivie et décrite dans le pseudo-code 3 pour construire le graphe de connaissances à partir de la modalité textuelle de la news.

Algorithme 3 : ConstruireGrapheTextuel

Entrées : $\mathcal{A}_{text}$: texte brut ; $\mathcal{LLM}$: extracteur de triplets ; `spaCy` : tokenizer/parseur ; $\mathcal{E}$: encodeur sémantique (SBERT `all-MiniLM-L6-v2`) ; $\mathcal{R}_{can}$: ensemble de relations canoniques ; τ_{rel} : seuil de mapping relation ; τ_{dup} : seuil de fusion sémantique

Output : T : graphe de triplets normalisés et enrichis ; Π : métadonnées de provenance

1. $P \leftarrow \text{SegmenterEnPhrases}(\mathcal{A}_{text})$; `// ex : spaCy`

2. $T \leftarrow \emptyset$;

3. pour chaque $p \in P$ **faire**

  $T_{\text{raw}} \leftarrow \mathcal{LLM}(p)$; `// JSON : liste de ⟨sujet, relation, objet⟩`

  $T \leftarrow T \cup T_{\text{raw}}$;

4. pour chaque $(s, r, o) \in T$ **faire**

  $(s, o) \leftarrow \text{SimplifierEntiteCommune}(s, o, \texttt{spaCy})$;

  $r \leftarrow \text{MapperRelation}(r, \mathcal{R}_{can}, \mathcal{E}, \tau_{rel})$; `// projection vers` $\mathcal{R}_{can}$

5. $T \leftarrow \text{FusionParInclusion}(T)$;

6. $T \leftarrow \text{FusionSemantique}(T, \mathcal{E}, \tau_{dup})$;

7. $(T, \mathcal{C}_E) \leftarrow \text{ClassifierEntites}(T)$;

8. $(T, \mathcal{C}_{EN}) \leftarrow \text{ClassifierEntitesNommees}(T)$;

9. $(T, \Pi) \leftarrow \text{EnrichirAvecKnowledge}(T, \mathcal{C}_{EN}, \mathcal{C}_E)$;

10. $T \leftarrow \text{Nettoyer}(T)$;

11. retourner (T, Π) ;

Pour chaque actualité textuelle $\mathcal{A}_{\text{text}}$, nous produisons un graphe orienté enrichi par des connaissances externes, Cette production se structure en plusieurs étapes. Chacune participe à la robustesse et à la couverture sémantique du graphe final.

Extraction de triplets sémantiques via LLM (LLaMA3) : En premier lieu, nous exploitons un modèle de type LLM (ici LLaMA3) pour segmenter le texte en triplets $<$ `sujet`, `relation`, `objet` $>$ (de manière analogue au Chapitre 4). Cette étape correspond aux lignes 1 à 3 de l'Algorithme 3. Pour ce faire nous incluons une fenêtre contextuelle de deux phrases, avec chevauchement d'une phrase, afin d'augmenter les capacités du LLM à saisir le contexte sémantique entre les différentes entités en jeu.

L'utilisation d'un LLM présente plusieurs avantages par rapport aux approches classiques de parsing textuel. Du fait de leur capacité de raisonnement, elle permet en effet de capter des relations complexes, parfois implicites ou métaphoriques, et d'offrir une grande flexibilité via le prompt sans nécessiter d'entraînement spécialisé. Toutefois, ces modèles demeurent fondamentalement stochastiques, difficiles à calibrer et susceptibles de produire des hallucinations. Malgré ces limites, l'objectif de cette étape est d'obtenir une représentation structurée de la sémantique contenue dans le texte, qui servira de base aux étapes ultérieures de construction et d'enrichissement des graphes de connaissances.

Normalisation des entités : Nous normalisons les entités et relations afin d'uniformiser le graphe, de réduire le bruit et d'en améliorer la cohérence. Cela correspond aux lignes 4 à 6 de l'Algorithme 3 Les entités extraites sont d'abord simplifiées en utilisant la bibliothèque Python `spaCy` afin de ne conserver que la dernière tête nominale, ce qui permet de réduire les variantes lexicales complexes à leur forme canonique. Il s'agit d'une

hypothèse de travail dans la mesure où cette simplification induit nécessairement une perte d'information. Cela est donc un compromis entre exhaustivité et simplicité. Ensuite, une fusion par inclusion de chaînes de caractères est appliquée pour regrouper les entités dont l'une est contenue dans l'autre. Enfin, une fusion sémantique est réalisée en comparant les représentations vectorielles issues de l'encodeur SBERT [119] (`all-MiniLM-L6-v2`).

Fusion Sémantique et choix de l'encodeur Deux entités sont fusionnées si la similarité cosinus entre leurs embedding SBERT dépasse le seuil τ_{dup}. Le choix d'encodeur SBERT est motivé par son entraînement et son évaluation sur un large éventail de tâches de classification et de similarité de phrases, incluant notamment de la classification de sentiments [102], de questions [74] et d'opinion [160]. Ces benchmarks couvrent des cas variés allant de la détection de paraphrases à la classification de sentiment, démontrant la robustesse de SBERT dans la capture de similarités sémantiques entre phrases courtes. Cela en fait un choix pertinent pour nos tâches de normalisation d'entités et de relations, où la qualité de la mesure de similarité textuelle est essentielle.

Néanmoins, nous soulignons que SBERT n'est pas optimisé spécifiquement pour la représentation de mots isolés, contrairement à des modèles lexicaux comme Word2Vec ou fastText. Dans certains cas, la similarité captée par SBERT peut refléter davantage des associations issues des phrases d'entraînement que des proximités purement lexicales. Cependant, nous pensons que dans notre cadre, cette limite est compensée par la capacité de SBERT à mieux capturer des nuances sémantiques pertinentes.

Normalisation des relations. L'ensemble des relations extraites (directement en sortie du LLM, ou de Wikidata et ConceptNet) est hétérogène. Pour harmoniser la structure on projette chaque relation vers un ensemble de relations canoniques restreint (ligne 4 de l'Algorithme 3) :

$$\mathcal{R}_{can} = \{\texttt{IsA}, \texttt{PartOf}, \texttt{RelatedTo}, \texttt{Causes}, \texttt{LocatedIn}, \texttt{MemberOf}, \texttt{UsedFor}\} \quad (7.1)$$

Chaque relation r est encodée par un transformeur léger pré-entraîné (`all-MiniLM-L6-v2`) et rapprochée, par similarité cosine, de sa plus proche dans $\mathcal{R}_{\text{can}}$. Le mapping n'est retenu que si la similarité dépasse le seuil τ_{rel}, garantissant une correspondance fiable avec les relations canoniques. Les relations non appariées reçoivent le label générique `RelatedTo`. Cette homogénéisation réduit le bruit syntaxique et facilite l'exploitation ultérieure des arêtes du graphe par un R-GCN.

Entités nommées et entités communes Une des difficultés de l'extraction des faits depuis le texte, est de séparer les entités *nommées*, (c'est à dire les noms propre tel que des des personnes célèbres, des lieux, des organisations...) des entités *communes* (qui correspondent à des noms communs, des objets, des concepts etc). Cette étape correspond aux lignes 7 et 8 de l'Algorithme 3. Les entités nommées sont fortement porteuses de l'information contenue dans une news et peuvent être enrichies largement à partir de base de données spécifiques comme WikiData ou DBpedia. Quant à elles, les entités communes, sont en général moins informatives, mais peuvent porter une thématique globale et être enrichie par des informations conceptuelles précieuses avec des bases de connaissances comme ConceptNet [137] ou WordNet [95]. Nous illustrons la différence entre entités nommées et communes sur la Figure 7.2.

La distinction entité commune/nommée est opérée automatiquement à l'aide de Wikidata. Pour chaque entité extraite, une requête SPARQL permet d'identifier son identifiant unique (QID) et son type à partir de la propriété `P31` (*instance of*). Les classes

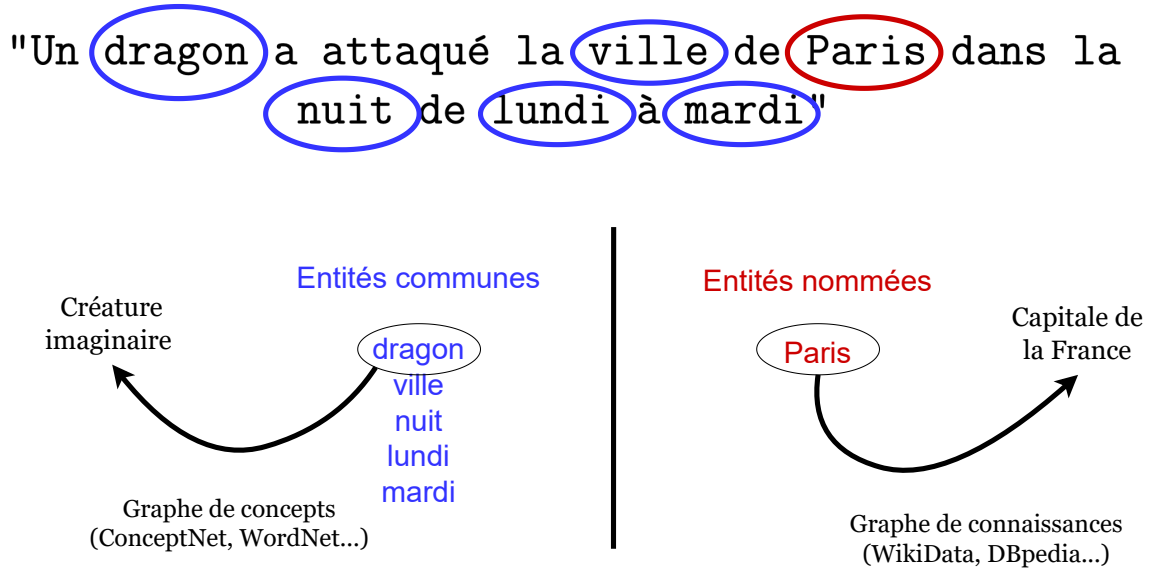


FIGURE 7.2 – Exemple de classification des entités sur une assertion. Les entités nommées et communes sont enrichies d'après des bases de connaissances distinctes.

obtenues sont ensuite mappées vers un ensemble restreint de catégories prioritaires (par exemple *Person*, *Organization*, *Location*, *Event*). Lorsqu'une entité correspond à l'une de ces classes, elle est considérée comme une **entité nommée** et enrichie par ses relations directes issues de Wikidata. Dans les autres cas (classe *Other* ou absence de QID), l'entité est traitée comme une entité commune et son enrichissement repose alors sur ConceptNet.

Enrichissement des entités nommées via Wikidata Les entités *nommées* sont enrichies à l'aide de bases de connaissances externes, telles que Wikidata ou DBpedia, qui offrent des informations factuelles vérifiables sur des personnalités, des organisations ou des événements. Par exemple, une entité comme `Donald Trump` peut être enrichie par le triplet `<Donald Trump, IsA, US president>` issu de Wikidata. Pour chaque entité nommée e, le processus se déroule en trois étapes :

1. **Classification typologique** de l'entité parmi un ensemble de classes prédéfinies : {`"Person"`, `"Organization"`, `"Location"`, `"Event"`}. Cette étape conditionne les filtres de requêtes par la suite.
2. **Identification de la correspondance Wikidata** à l'aide de l'API `Wikibase`, afin de récupérer le `QID` correspondant à l'entité e.
3. **Requête SPARQL sur Wikidata**, définie dynamiquement en fonction du type d'entité. Par exemple, si l'entité est de type `"Person"`, on pourra extraire des propriétés telles que `conjoint/e` (`P26`) ou `parti politique` (`P102`), tandis qu'une entité de type `"Event"` donnera lieu à des requêtes sur les lieux ou dates associés, mais non sur des attributs relationnels personnels.

Cet enrichissement permet d'incorporer dans le graphe des informations factuelles validées (dates, catégories, affiliations, relations hiérarchiques), ce qui contribue à une meilleure structuration du graphe. Par ailleurs, il renforce la capacité du système à détecter des incohérences ou contradictions factuelles, comme par exemple l'attribution d'une citation ou d'un rôle politique à une personne qui n'a jamais occupé cette fonction.

Enrichissement des entités communes via ConceptNet Les entités *communes*, quant à elles, sont enrichies à l'aide de ressources de connaissances générales telles que ConceptNet, conçues pour représenter des relations de sens commun entre concepts. Ces entités, souvent absentes de bases de connaissances comme Wikidata, nécessitent un traitement complémentaire pour étoffer leur représentation sémantique.

Par exemple, l'entité `covid19` peut être reliée à `disease` par la relation `IsA`, donnant le triplet : `<covid19, IsA, disease>`. Le processus d'enrichissement se déroule comme suit :

— Pour chaque entité e, on interroge l'API de ConceptNet pour récupérer jusqu'à $n_{\max} = 5$ relations significatives et concepts associées, telles que `IsA`, `PartOf`, `UsedFor`, `HasProperty`, etc.

— Les connaissances extraites sont intégrées au graphe sous forme de triplets `<e, r, c>`, où c est un concept connexe ou dérivé.

Cet enrichissement permet d'introduire dans le graphe une couche de *connaissances de sens commun*, essentielle pour relier des concepts génériques à des catégories sémantiques plus larges ou pour détecter des associations douteuses, erronées ou détournées présentes dans les fake news.

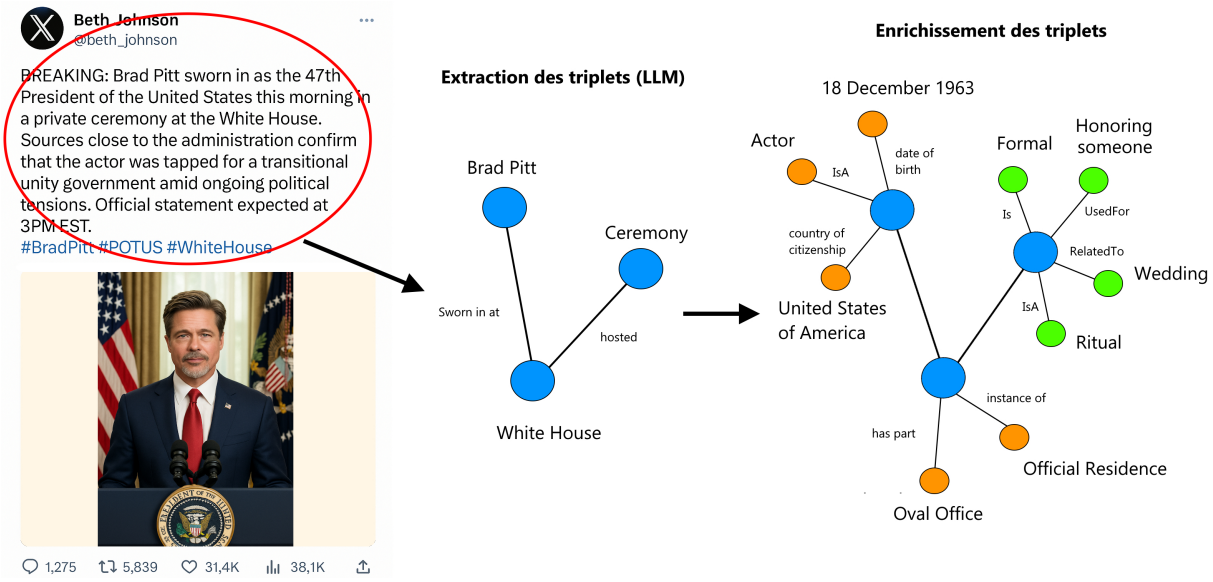


FIGURE 7.3 – Architecture de la génération du graphe de connaissances à partir du texte d'une actualité (avant normalisation des relations). En bleu sont représentées les entités extraites par le LLM, en orange les entités extraites de WikiData à partir d'entités nommées, et en vert les entités extraites de ConceptNet à partir des entités communes.

Finalement, on procède à un nettoyage du graphe : retrait des doublons, des entrées vides ou mal formattées. Nous disposons alors d'un graphe de connaissances enrichi par des bases de données externes et construit à partir du texte de la news.

7.3.2 Construction du graphe de connaissances à partir de l'image

Désormais, on souhaite construire un graphe de connaissances liées à l'image d'une actualité afin de traiter sa modalité visuelle. Une des difficultés de cette tâche est la nécessité d'établir une reconnaissance visuelle des entités nommées de l'image. Dans un premier temps, pour ne pas alourdir la méthode et parce que l'information transmise par une image passe beaucoup par les personnalités qu'elle met en jeu, nous nous concentrons sur la reconnaissance visuelle des personnalité célèbres. Si la détection d'entité commune

est largement documentée et accessible, la détection des personnalités est plus difficile, du fait notamment de la longue queue de distribution des entités qui rend difficile l'entraînement d'un modèle détectant toutes les personnalités célèbres. Nous proposons donc dans un premier temps une méthode de détection visuelle de célébrités, à partir d'un ensemble de données restreint. Dans un second temps, une fois les personnalités identifiées, nous construisons de manière analogue au texte (*cf.* Section 7.3.1) un graphe de connaissances pour chaque actualité visuelle $\mathcal{A}_{\text{img}}$.

Approche de reconnaissance visuelle de personnalité célèbres

On détaille ci-dessous notre méthode de reconnaissance visuelle de personnalités célèbres, en 5 étapes.

1. Constitution d'un référentiel de personnalités célèbres Afin de reconnaître des visages, on prépare un jeu de données d'images de célébrités, constitués de $N_{celeb} = 2173$ personnalités, liées à $N_{img} = 8940$ (soit une moyenne de 4.1 images par entité) :

- On récupère par scraping des noms des personnalités les plus populaires (définie par le nombre de pages auxquelles elles sont reliées) sur Wikipedia. Ces personnalités revêtent plusieurs fonctions : présidents d'instances internationales, personnalités politiques, acteurs, chanteurs... Du fait de leur popularité, ces célébrités sont plus susceptibles d'être présentes dans des actualités.
- Pour chaque nom, on réalise un scraping de plusieurs images représentatives depuis Wikipedia. Les images sont filtrées, celles ne contenant aucun visage sont supprimées, car elles sont trop peu informatives, celles en contenant plus de 3 sont également supprimées, car elles ajoutent trop de bruit aux futures représentations.
- Enfin, on calcule un vecteur d'embedding facial pour chaque personnalité à l'aide d'InsightFace [27][28].

2. Détection des visages dans l'image Pour une image $\mathcal{I}$ donnée, on applique un détecteur d'objets (YOLOv3 [118]) pour extraire l'ensemble des bounding boxes de l'image contenant des humains.

$$B(\mathcal{I}) = \{\text{bbox}_1, \text{bbox}_2, \ldots, \text{bbox}_k\}$$

3. Génération d'une légende visuelle (*caption*) Un Vision-Language Model (LLaVA [79]) produit une description textuelle $\mathcal{C}$ de l'image $\mathcal{I}$, par exemple :

$$\mathcal{C}(\mathcal{I}) = \textit{"A man is wearing a suit with a tie and has short gray hair"}$$

.

4. Alignement légende et bbox via CLIP Chaque entité humaine mentionnée dans la légende (par ex. *"a man is wearing a suit"*) est associée à la bbox la plus représentative au moyen de la similarité CLIP [114]. CLIP projette simultanément le contenu visuel (image définie par la bbox) et la description textuelle dans un espace vectoriel commun. L'alignement est réalisé en sélectionnant la bbox_i qui maximise la similarité cosinus entre les deux représentations, sous réserve que cette similarité dépasse un seuil fixé :

$$\text{sim}(\text{"a man is wearing a suit"}, \text{bbox}_i) \;\geq\; \tau_{\text{CLIP}}.$$

5. Reconnaissance faciale par comparaison d'embeddings Pour chaque bbox i alignée à un humain, on extrait son embedding via InsightFace et on le compare aux N_{celeb} embeddings de notre référentiel :

$$\max_{j=1...N_{celeb}} \text{sim}_{\cos}\left(\mathbf{f}_{\text{bbox}_i}, \mathbf{f}_{\text{celeb}_j}\right) \geq \tau_{\text{ID}}$$

Où $\mathbf{f}_{\text{bbox}_i}$ et $\mathbf{f}_{\text{celeb}_j}$ sont les vecteurs de features associés à la bbox i et à la célébrité j. Si le score dépasse le seuil τ_{ID}, on étiquette la bbox_i par le nom de la célébrité (ex. "Brad Pitt") et on remplace la mention générique (*"a man is wearing a suit"*) par l'entité nommée

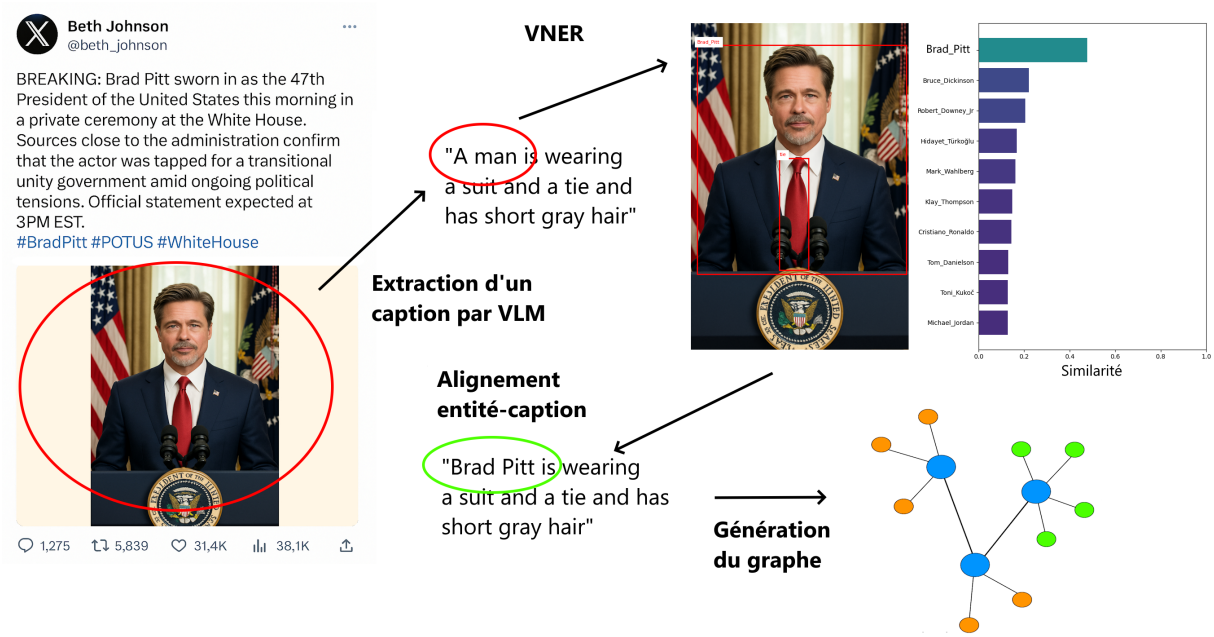


FIGURE 7.4 – Procédure de reconnaissance visuelle d'entités nommées et alignement avec le caption de l'image $\mathcal{A}_{img}$.

Génération du graphe de connaissances à partir de l'image Une fois les personnalités du caption VLM de l'image identifiées visuellement, on extrait un graphe de connaissance du caption de la même manière dont nous avons extrait le graphe de connaissances à partir de l'information textuelle. Cependant, nous ajustons l'ensemble des relations canoniques, afin d'intégrer des relations visuelles, telles que `NextTo` ou `In`.

7.4 Évaluation de la cohérence sémantique texte-image

Un autre signal exploité dans notre architecture est la cohérence sémantique entre le texte et l'image. En effet, de nombreuses fake news reposent non pas sur une falsification directe de l'image, mais sur une recontextualisation abusive, où une image authentique est associée à une description textuelle erronée, ou bien sur une interprétation trompeuse de son contenu visuel. Souvent, l'on observe dans le cadre d'une désinformation, une photo d'un évènement ancien, ou sans rapport avec celui décrit, qui peut renforcer la crédibilité d'une affirmation fausse.

Pour évaluer ce type de décalage, nous introduisons un module de similarité sémantique texte-image. Des travaux comme SAFE [175] ont montré que la seule mesure de cohérence multimodale pouvait déjà constituer un signal pertinent pour détecter des fake news, atteignant par exemple une accuracy de 0.874 sur Politifact [111]. Nous faisons le choix

d'une approche volontairement simple et générique, en utilisant directement la similarité produite par CLIP [114] comme score de cohérence.

Notre motivation n'est pas d'optimiser ce signal à lui seul, mais d'évaluer dans quelle mesure il peut complémenter les autres modules (visuel et multimodal-knowledge-based) de notre architecture. Ce choix est pragmatique : il s'agit de vérifier si l'ajout de ce signal, peu coûteux en calcul et largement réutilisable, apporte une information exploitable. Ce parti pris nous permet aussi d'analyser son effet dans l'étude d'ablation ultérieure, où nous discuterons sa réelle pertinence au sein du pipeline.

7.5 Apprentissage sur graphes et prédiction multimodale

Après avoir présenté les différents signaux exploités par notre pipeline (graphes de connaissances textuels et visuels, probabilité de transformation d'image, score de cohérence texte-image), nous détaillons dans cette section le module d'apprentissage et de fusion multimodale qui permet de produire la prédiction binaire finale.

Encodage des graphes par R-GCN. Chacun des graphes (textuel et visuel) est traité séparément par un encodeur basé sur un *Relational Graph Convolutional Network* (R-GCN) (*cf.* Section 3.6.3). Ces encodeurs propagent l'information entre nœuds en tenant compte de la nature des relations (arêtes multi-types). L'architecture retenue comprend deux couches R-GCN cachées avec activation ReLU et régularisation par dropout (qui limite le surapprentissage de certains nœuds), suivies d'une couche de sortie sans activation. Un pooling global (mean pooling (*cf.* Section 3.6.4))sur chaque graphe est ensuite appliqué afin d'agréger les représentations des nœuds en un unique vecteur représentant le graphe. Cette étape produit deux représentations vectorielles distinctes : $h_{\text{txt}} \in \mathbb{R}^d$ pour le graphe textuel et $h_{\text{img}} \in \mathbb{R}^d$ pour le graphe visuel.

Intégration des signaux supplémentaires. Aux représentations issues des deux R-GCNs sont adjoints les deux signaux scalaires décrits précédemment :

1. le score $h_{\text{fake-img}} \in [0, 1]$ de probabilité de transformation de l'image,
2. le score de cohérence sémantique texte-image h_{clip} issu du modèle CLIP.

Ces signaux sont pondérés par des coefficients d'importance $(\alpha_{\text{txt}}, \alpha_{\text{img}}, \alpha_{\text{clip}}, \alpha_{\text{fake-img}}) \in \mathbb{R}^4$, appris conjointement avec le reste du modèle, ce qui permet au réseau d'ajuster automatiquement la contribution relative de chaque modalité en fonction des données.

Fusion multimodale et classification. L'ensemble des représentations est concaténé en un vecteur joint :

$$h_{\text{joint}} = [\alpha_{\text{txt}} h_{\text{txt}}] \oplus [\alpha_{\text{img}} h_{\text{img}}] \oplus [\alpha_{\text{clip}} h_{\text{clip}}] \oplus [\alpha_{\text{fake-img}} h_{\text{fake-img}}] \tag{7.2}$$

Ce vecteur sert ensuite d'entrée pour un perceptron multi-couches (MLP) composé de deux couches linéaires séparées par une activation ReLU et une régularisation par dropout. La dernière couche applique une projection linéaire suivie d'une fonction sigmoïde, qui calcule la probabilité d'inauthenticité de l'information en entrée du modèle.

Les étapes de fusion et de classification constituent le charnière prédictive de notre architecture : elles combinent les signaux knowledge-based (graphes) qui encodent une information haut niveau, visuels (probabilité de transformation) qui reflètent des artefacts

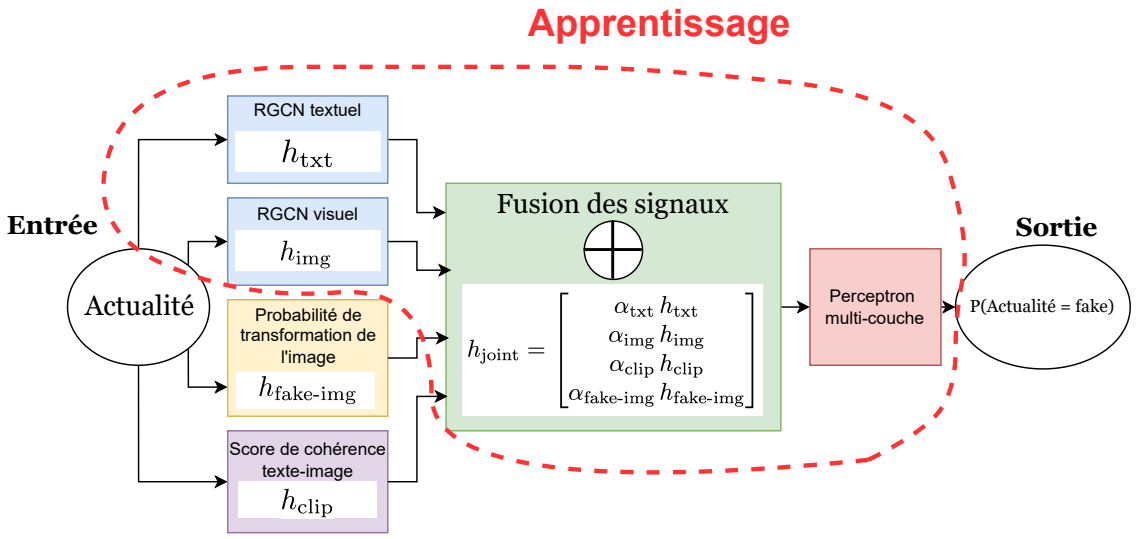


FIGURE 7.5 – Zoom sur le module de prédiction multimodale, mettant en évidence les modules entraînés de l'architecture. Les représentations issues des deux graphes de connaissances (encodés par des R-GCNs) et des signaux scalaires (probabilité de transformation visuelle et cohérence texte-image) sont intégrées dans un vecteur joint. Cette fusion multimodale est suivie d'un perceptron multi-couches, entraîné de manière supervisée, qui constitue la principale composante prédictive de notre modèle.

de falsification bas niveau, et intermodaux (cohérence texte-image), au sein d'un espace vectoriel commun. L'apprentissage conjoint des poids du réseau et des coefficients de pondération pour la fusion garantit une intégration flexible et adaptative de chaque signal.

En résumé, dans ce chapitre, nous avons présenté exhaustivement l'architecture multimodale proposée pour la détection des fake news, en explicitant les différents modules qui la composent. Nous avons montré comment les graphes de connaissances textuels et visuels, la probabilité de transformation d'image et le score de cohérence texte-image sont exploités et fusionnés dans un espace vectoriel commun, au moyen de deux R-GCNs et d'un module de classification supervisée. Cette conception modulaire combine des signaux complémentaires, allant des artefacts forensiques bas niveau jusqu'à des connaissances haut niveau, tout en garantissant une certaine capacité d'explicabilité.

Nous consacrons le chapitre qui suit à l'évaluation expérimentale de cette architecture. Nous y détaillerons les jeux de données utilisés, examinerons la qualité des graphes construits, les conditions d'entraînement, ainsi qu'une étude d'ablation permettant de mesurer la contribution de chacun des signaux. Ces expérimentations permettront de valider empiriquement la pertinence et les limites des choix de conception introduits dans ce chapitre.

8 Expérimentations

Dans ce chapitre, nous présentons les expérimentations réalisées afin d'évaluer l'architecture proposée dans ce travail de thèse, décrite au Chapitre 7, pour la détection de fake news multimodales enrichies par des connaissances externes.

Nous détaillons dans un premier temps les modalités d'entraînement du modèle et les hyperparamètres utilisés, avant de conduire une étude d'ablation exhaustive qui met en évidence la contribution de chaque signal d'entrée au sein du module de classification binaire.

Nous exploitons pour cela le jeu de données **DGM**^{4} [126], un corpus multimodal combinant texte et image, sur lequel nous entraînons et évaluons notre modèle.

Enfin, nous rapportons également des résultats concernant la qualité des graphes de connaissances extraits du texte et de l'image, qui constituent une composante centrale de notre pipeline et contribuent à l'explicabilité du processus de détection.

L'ensemble de ces expérimentations vise à (i) valider la faisabilité de l'approche proposée, (ii) mesurer précisément l'impact des différentes composantes et les performances du modèle, et (iii) mettre en évidence les limites et perspectives pour une intégration plus robuste de connaissances externes dans la détection de fake news multimodale.

8.1 Entraînement et optimisation du modèle

Dans cette section, nous présentons les modalités d'entraînement de notre architecture, depuis la constitution du jeu de données jusqu'au choix des hyperparamètres et des variantes architecturales évaluées par étude d'ablation.

8.1.1 Dataset d'entraînement

Nous avons retenu DGM^4 [126], un dataset spécifiquement conçu pour la détection de fake news multimodales, comme base d'entraînement. Ce corpus associe du texte et des images issues de médias traditionnels tels que *The Washington Post* ou *BBC*. Les données multimodales sont altérées de manière contrôlée : les images peuvent être modifiées par *face swapping* ou par altération d'attributs faciaux, tandis que les textes sont perturbés par substitution de segments textuels (*text swapping*) ou par modification d'attributs sémantiques. Cette diversité de manipulations reflète de manière réaliste les types d'altérations rencontrés dans des scénarios réels de désinformation.

Pour nos expérimentations, nous avons constitué un sous-échantillon équilibré de **1000 instances** (500 fake et 500 réelles) pour l'entraînement, accompagné d'un jeu de validation de 150 instances et d'un jeu de test de 80 instances. Chaque exemple requiert un temps de prétraitement estimé à $7 \sim 8$ minutes pour la construction des graphes, en utilisant un GPU Nvidia RTX 4060. Le temps de préparation complet de ce corpus correspond ainsi à environ une semaine de calcul continu.



FIGURE 8.1 – Exemple de données issues du dataset DGM[4]. En haut, l'information a subi une double altération (face swap + text swap). En bas, seule une modification d'attribut textuel est présente, ce qui rend la détection plus subtile.

Discussion des limites. La taille réduite de l'échantillon utilisé (1000 instances) constitue une limite importante pour l'apprentissage profond. Ce volume représente néanmoins un seuil minimal permettant d'obtenir des résultats significatifs et de comparer différentes configurations du modèle. En contrepartie, il restreint la capacité des couches GCN et de classification à exploiter un nombre élevé de paramètres, ce qui peut limiter les performances maximales atteignables. Une extension future des expériences devrait s'appuyer sur des corpus plus larges afin de renforcer la robustesse et la généralisation du modèle.

8.1.2 Embeddings et architectures explorés

Embeddings initiaux des graphes

Afin d'améliorer la qualité des représentations apprises par nos GNN, nous avons initialisé les représentations des entités avec des embeddings pré-entraînés de graphes de connaissances. Cette approche permet de fournir au modèle une structure sémantique initiale plus riche qu'un simple encodage aléatoire, et facilite l'apprentissage dans un contexte où le volume de données d'entraînement est limité. Nous nous appuyons pour cela sur la librairie Python LibKGE [12], conçue pour l'entraînement et l'évaluation d'embeddings de graphes de connaissances.

Aléatoire : Un premier scénario de référence consiste à initialiser les représentation des entités avec un embedding aléatoire. Cette configuration constitue la représentation la plus élémentaire, où l'intégralité de l'apprentissage des représentations pertinentes repose sur les couches GNNs. Étant donné la taille restreinte de notre corpus d'entraînement,

nous anticipons que les performances obtenues avec cette initialisation seront limitées, ce qui en fait un point de comparaison minimal.

TransE : Nous utilisons ensuite un embedding initial de type **TransE**, pré-entraîné sur le graphe **Freebase15K** [9]. Ce graphe contient environ 15 000 entités et 1 000 relations, pour un total de 600 000 triplets. TransE offre un compromis intéressant : les représentations ainsi obtenues sont informatives tout en restant relativement rapides à intégrer dans notre architecture, du fait du nombre limité d'entités.

ComplEx : Enfin, nous exploitons un embedding **ComplEx** pré-entraîné sur **Wikidata5M** [154], un sous-graphe de Wikidata comprenant environ 5 millions d'entités et 800 relations. ComplEx présente deux avantages majeurs : (i) sa capacité à modéliser efficacement les relations antisymétriques, et (ii) une couverture lexicale beaucoup plus large que TransE, ce qui permet de représenter efficacement chaque entité rencontrée dans la tâche de détection des fake news. Néanmoins, dans notre configuration expérimentale, nous ne tirons pas pleinement parti de la représentation complexe native de ComplEx : les embeddings sont projetés dans un espace réel $\mathbb{R}^d$, ce qui peut atténuer une partie de ses bénéfices théoriques.

Choix du GNN

Nous avons évalué deux types d'architectures de réseaux de neurones de graphes (GNN) pour modéliser les relations entre entités : (i) des **R-GCNs** (Relational Graph Convolutional Networks), qui tiennent explicitement compte des types de relations, et (ii) des **GCNs classiques**, qui ne distinguent pas la nature des arêtes.

R-GCN : Dans le cas des R-GCNs, la diversité des relations est un facteur crucial qui conditionne la qualité de l'apprentissage. Afin de limiter la complexité du modèle, particulièrement dans le contexte d'un corpus d'entraînement restreint, nous avons choisi de travailler avec deux ensembles réduits de relations canoniques :

$$\mathcal{R}_{\text{txt-can}} = \{\texttt{IsA}, \texttt{PartOf}, \texttt{RelatedTo}, \texttt{Causes}, \texttt{LocatedIn}, \texttt{MemberOf}, \texttt{UsedFor}\} \tag{8.1}$$

$$\mathcal{R}_{\text{vis-can}} = \{\texttt{IsA}, \texttt{PartOf}, \texttt{NextTo}, \texttt{On}, \texttt{In}, \texttt{Holding}, \texttt{Wearing}, \texttt{Behind}, \texttt{InFrontOf}\}. \tag{8.2}$$

Ces ensembles sont supposés couvrir un large spectre de relations hiérarchiques, fonctionnelles ou spatiales, permettant de contextualiser efficacement les entités entre elles. L'avantage d'une telle modélisation est de capturer plus finement les dépendances structurelles et contextuelles ; en revanche, cette approche accroît le coût computationnel et peut rendre l'apprentissage plus difficile à stabiliser.

GCN classique : À titre de comparaison, nous avons également testé des GCNs standards, qui appliquent les mêmes opérations à toutes les arêtes, sans distinction de relation. Ces architectures présentent l'avantage d'être plus simples et plus rapides à entraîner, ce qui est pertinent dans un contexte de données limitées. Cependant, cette simplification se fait au détriment de la contextualisation : en ne tenant pas compte de la nature des relations, le modèle perd une partie significative de l'information, ce qui peut limiter sa capacité à discriminer les entités dans des contextes complexes.

8.1.3 Stratégie d'entraînement et réglages d'hyperparamètres

On présente notre stratégie, classique, d'apprentissage profond pour l'entraînement du modèle.

Objectif et fonction de coût : La tâche est une classification binaire nous optimisons donc la fonction de coût cross entropique binaire BCE sur la sortie *logit* du classifieur.

Optimisation et arrêt anticipé. L'optimiseur Adam est utilisé avec un taux d'apprentissage initial fixé à 10^{-4}. Afin d'améliorer la convergence, nous l'associons à un scheduler *ReduceLROnPlateau*, qui réduit automatiquement le taux d'apprentissage global par un facteur 0.5 lorsque la F_1 de validation n'augmente plus pendant cinq époques consécutives. Nous appliquons un *gradient clipping* avec un seuil fixé à 1.0, ce qui consiste à contraindre la norme L_2 des gradients à ne pas dépasser cette valeur. Cette technique stabilise l'entraînement en limitant l'explosion de gradients lors de la rétropropagation. En outre, nous procédons à un arrêt anticipé avec patience 10 époques. Les expériences sont initialisées avec une graine fixe et un `batch_size`de 32. Enfin, un dropout de 25% est également appliqué sur les neurones pour limiter le surapprentissage.

Hyperparamètre	Valeur	Commentaire
`batch_size`	32	Taille des batchs
`epochs`	100	Nombre maximal d'époques
`lr`	10^{-4}	Taux d'apprentissage initial (Adam)
`wd`	0	Weight decay (régularisation L_2)
`couches GCN`	2	Nombre de couches des GNN
`joint-hidden MLP`	64	Dimension des couches cachées du MLP
`dim / in`	32	Dimension des embeddings d'entrée
`hidden`	64	Dimension des couches cachées des GNN
`out`	1	Dimension de sortie des encodeurs
$\lvert\mathcal{R}_{\text{text}}\rvert$	7	Nombre de relations textuelles canoniques
$\lvert\mathcal{R}_{\text{img}}\rvert$	9	Nombre de relations visuelles canoniques

TABLE 8.1 – Résumé des hyperparamètres utilisés pour l'entraînement du modèle.

Hyperparamètres : Les valeurs présentées dans le Tableau 8.1 correspondent aux paramètres par défaut retenus pour nos expériences. Elles ont été sélectionnées empiriquement après une série de tests exploratoires, de manière à assurer un compromis satisfaisant entre stabilité de l'entraînement et performance sur le jeu de validation. Le paramètre `weight decay`, qui correspond à une régularisation L_2 des poids, a été fixé à 0 dans nos expériences, après vérification que son activation n'apportait pas de gain notable sur le jeu de validation.

Protocole : Chaque batch contient 32 données multimodales complète (texte-graphe, image-graphe, probabilité d'image fake, score de cohérence CLIP), les encodeurs de graphes (GCN ou R-GCN) sont appris *de bout en bout* conjointement avec la tête de classification. Nous considérons toutes les variantes activant ou non les contributions `use_graph_txt`, `use_graph_img`, `use_clipscore`, `use_fake_img`, afin d'isoler l'apport de chaque signal (*cf.* étude d'ablation). Les métriques standard (Accuracy, Précision, Rappel, F_1) sont suivies à chaque époque sur le jeu de validation et enregistrées dans TensorBoard. Le meilleur

modèle, défini selon la F_1 de validation, est sauvegardé avec un identifiant de configuration haché.

Fusion multimodale. Les encodeurs de graphes (texte et image) produisent chacun un vecteur de dimension $d =$ `out` ; deux scalaires externes (score CLIP et score de falsification d'image) sont intégrés avec des poids apprenables α. Le vecteur concaténé $h_{\text{joint}} \in \mathbb{R}^{2d+2}$ est projeté par un MLP : `Linear(2d+2,` m)→ReLU→`Linear(`m`,1)` avec $m =$ `joint_hidden` $= 64$.

Justification architecturale. Compte tenu du faible volume d'entraînement (≈1000 exemples), nous avons volontairement adopté une tête de fusion compacte. Les encodeurs de graphes (R-GCN/GCN) opèrent dans un espace caché de dimension 64 au cours de la propagation, puis projettent leur représentation avec un mean-pooling en une dimension. Cette projection agit comme un goulot d'étranglement régularisant, qui limite le risque de sur-apprentissage dans notre contexte de données d'entraînement restreintes. L'initialisation par un embedding TransE pré-entraîné fournit en outre un a priori sémantique fort, réduisant l'espace de recherche effectif et donc limitant le risque de surapprentissage. Ainsi, bien que le nombre de paramètres du modèle dépasse le nombre d'exemples disponibles, l'utilisation d'embeddings pré-entraînés et des régularisations (dropout, early-stopping) permettent de tirer parti de la capacité du modèle. Cette configuration illustre le compromis en apprentissage profond entre la complexité du modèle et la disponibilité de données d'entraînement.

Après avoir détaillé le protocole d'entraînement et les réglages retenus, nous présentons à présent les résultats expérimentaux. Nous commençons par une étude d'ablation visant à quantifier la contribution de chaque signal et de chaque composant architectural, avant de comparer les variantes du modèle et d'analyser les performances du meilleur paramétrage retenu.

8.1.4 Étude d'ablation

Dans cette sous-section, nous présentons une étude d'ablation exhaustive. À l'aide des paramètres booléens `use_clip`, `use_fakeimg`, `use_graphimg`, et `use_graphtxt`, nous activons ou désactivons chacun des quatre signaux afin d'évaluer leur contribution respective pour la détection multimodale de fake news. Cette analyse permet de mettre en évidence les signaux les plus informationnels et les combinaisons les plus efficaces.

Liste des combinaisons. En considérant l'ensemble des activations possibles, $2^4 - 1 = 15$ combinaisons distinctes de signaux peuvent être définies. Chacune de ces configurations est évaluée avec deux initialisations différentes des embeddings de graphe : soit un embedding **TransE** pré-entraîné, soit un embedding **aléatoire**, comme décrit précédemment. Au total, 30 modèles sont donc testés dans cette expérience, tous implémentés avec l'architecture R-GCN (le GCN classique n'étant pas inclus dans cette analyse).

Analyse des courbes d'entraînement : La Figure 8.2 représente l'évolution de la fonction de perte au cours des époques. On y observe à la fois l'effet de l'early stopping et la capacité du modèle à apprendre, quelle que soit la configuration considérée.

La Figure 8.3 illustre les performances sur le jeu de validation, rapportées selon quatre métriques standard : Accuracy, Précision, Rappel et F_1-score. Dans notre cas, où les classes sont équilibrées, l'Accuracy et le F_1-score constituent les indicateurs principaux.

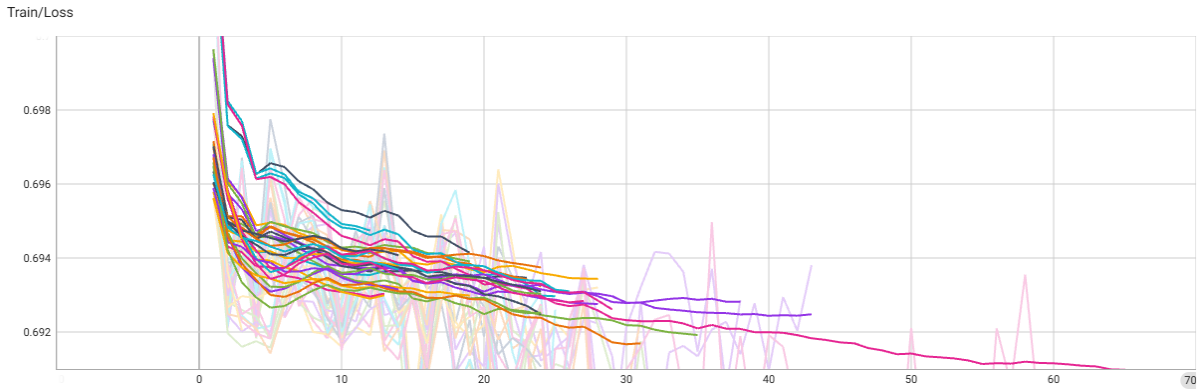
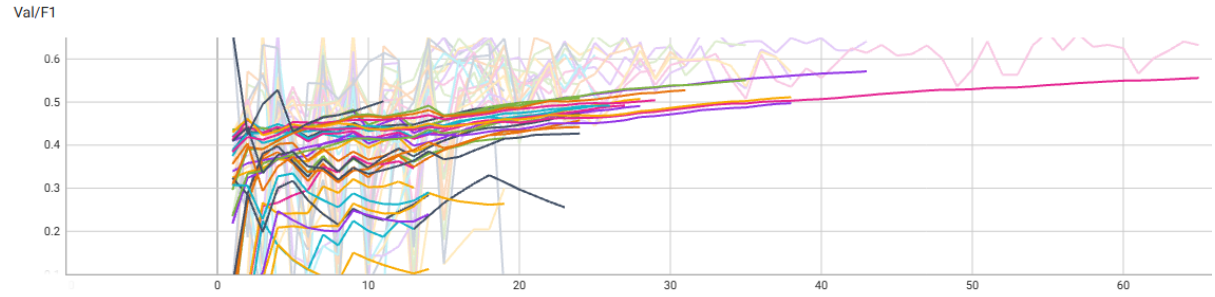


FIGURE 8.2 – Evolution de la loss pendant l'apprentissage des 30 modèles de l'étude d'ablation.

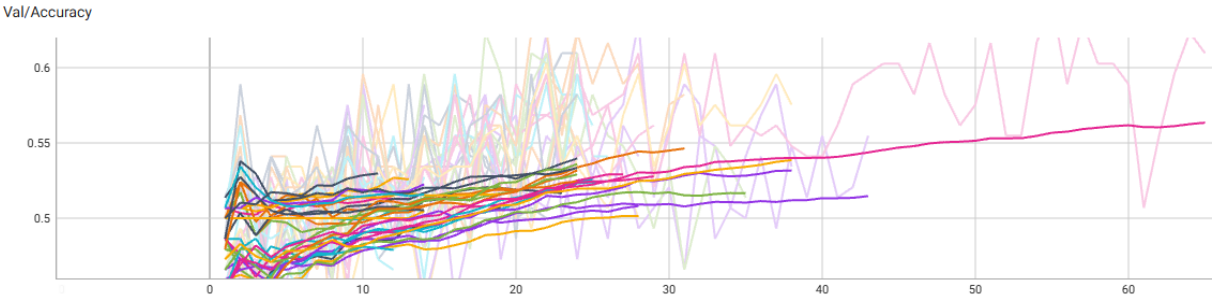
Cependant, la Précision et le Rappel restent essentiels pour analyser les compromis entre faux positifs et faux négatifs.

De manière générale, presque toutes les configurations conduisent à une amélioration des performances au cours de l'apprentissage, ce qui montre que chaque signal contient une part d'information exploitable pour la détection. Toutefois, la majorité des modèles convergent rapidement (moins de 30 époques) ou atteignent des performances maximales modestes (Accuracy et F_1 inférieurs à 0.55). Cela suggère une difficulté du modèle à extraire des critères discriminants robustes, probablement en raison de signaux trop faibles ou mal exploités par l'architecture.

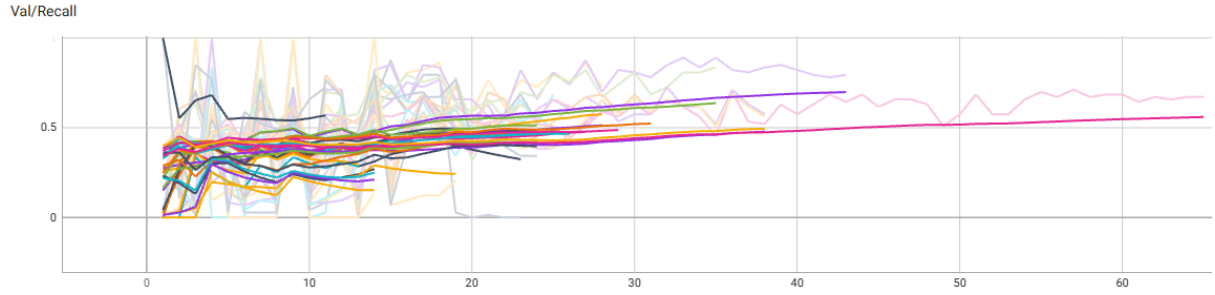
Les résultats quantitatifs présentés dans le Tableau 8.2, que nous commenterons par la suite, permettent d'émettre des hypothèses plus solides sur ces limites. Néanmoins, certaines courbes se distinguent nettement : par exemple, la courbe rose progresse jusqu'à l'époque 55 et surpasse largement les autres, indiquant que les signaux qu'elle exploite sont particulièrement pertinents.



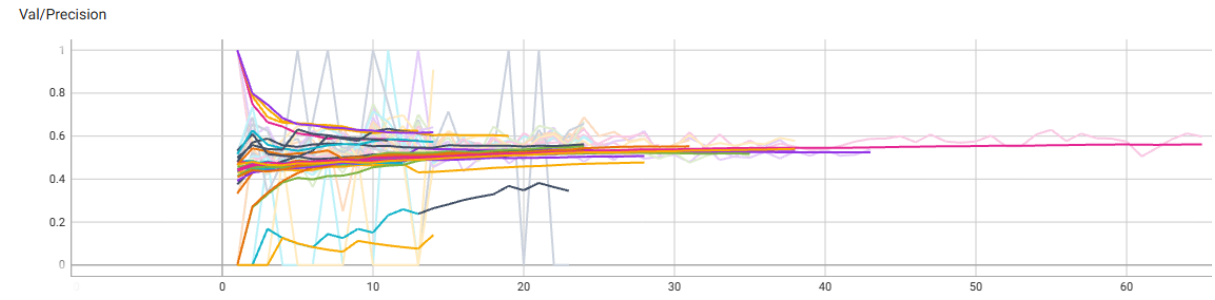
(a) Validation F1-score



(b) Validation Accuracy



(c) Validation Recall



(d) Validation Precision

FIGURE 8.3 – Étude d'ablation : évolution des métriques au cours de l'entraînement, pour différentes variantes du modèle.

Impact des différents signaux

Examen des résultats quantitatifs. Le Tableau 8.2 présente les performances maximales obtenues par chacun des modèles testés lors de l'étude d'ablation. Un ✔ indique qu'un signal est activé, tandis qu'un ✘ correspond à un signal désactivé.

Tout d'abord, on observe que l'embedding TransE surpasse systématiquement l'embedding aléatoire, ce qui était naturellement attendu. L'examen des signaux pris individuellement montre en revanche que les deux signaux scalaires (cohérence sémantique texte-image et probabilité d'image falsifiée) ne sont pas informatifs : ils conduisent toujours à une prédiction positive. Ce constat est cohérent avec nos observations antérieures sur la détection d'images falsifiées (*cf.* Chapitre 6) : si ce signal est pertinent sur d'autres de données, il peut l'être moins pris individuellement, sur DGM^4, où les altérations visuelles sont rares et souvent subtiles. Pour le score de cohérence sémantique, son caractère rudimentaire explique probablement son inefficacité : les dissimilarités entre texte et image dans ce corpus sont le plus souvent fines, et échappent donc à une mesure aussi simple.

À l'inverse, les graphes textuels et visuels seuls se révèlent significativement plus discriminants, avec par exemple une Accuracy atteignant 63% pour le graphe textuel pris isolément.

Emb.	ImgGraph	TxtGraph	Txt-Img	FakeImg	Acc.	Prec.	Rec.	F_1
Random	✔	✔	✔	✔	0.582	0.570	0.671	0.616
	✘	✔	✔	✔	0.582	0.558	0.795	0.655
	✔	✘	✔	✔	0.568	0.547	0.795	0.648
	✔	✔	✘	✔	0.534	0.524	0.740	0.614
	✔	✔	✔	✘	0.486	0.487	0.507	0.497
	✘	✘	✔	✔	0.500	0.500	1.000	0.667
	✘	✔	✘	✔	0.582	0.560	0.767	0.647
	✘	✔	✔	✘	0.548	0.528	0.890	0.663
	✔	✘	✘	✔	0.548	0.543	0.603	0.571
	✔	✘	✔	✘	0.555	0.538	0.767	0.633
	✔	✔	✘	✘	0.575	0.560	0.699	0.622
	✘	✘	✘	✔	0.500	0.500	1.000	0.667
	✘	✘	✔	✘	0.500	0.500	1.000	0.667
	✘	✔	✘	✘	0.555	0.534	0.863	0.660
	✔	✘	✘	✘	0.555	0.542	0.712	0.615
TransE	✔	✔	✔	✔	0.596	0.585	0.658	0.619
	✘	✔	✔	✔	0.623	0.607	0.699	0.650
	✔	✘	✔	✔	0.486	0.492	0.836	0.619
	✔	✔	✘	✔	**0.644**	**0.630**	0.699	0.662
	✔	✔	✔	✘	0.582	0.575	0.630	0.601
	✘	✘	✔	✔	0.500	0.500	1.000	0.667
	✘	✔	✘	✔	0.562	0.544	0.767	0.636
	✘	✔	✔	✘	0.555	0.537	0.795	0.641
	✔	✘	✘	✔	0.562	0.541	0.808	0.648
	✔	✘	✔	✘	0.473	0.483	0.795	0.601
	✔	✔	✘	✘	0.596	0.595	0.603	0.599
	✘	✘	✘	✔	0.500	0.500	1.000	0.667
	✘	✘	✔	✘	0.500	0.500	1.000	0.667
	✘	✔	✘	✘	0.630	0.623	0.658	0.640
	✔	✘	✘	✘	0.555	0.540	0.740	0.624

TABLE 8.2 – Étude d'ablation sur les quatre modules du pipeline, avec 15 × random et 15 × TransE : cases cochées = module activé, croisées = module désactivé.

Meilleur modèle : Le meilleur modèle du tableau (surligné en rouge), qui correspond également à la courbe rose dans la Figure 8.2 et présenté individuellement dans la Figure 8.4, exploite l'embedding TransE et trois signaux sur quatre : les graphes textuel et visuel, ainsi que la probabilité de falsification de l'image. Cela suggère que chacun de ces trois signaux est réellement utile à la détection, tandis que le signal de cohérence sémantique quant à lui n'apporte pas de gain de performance.

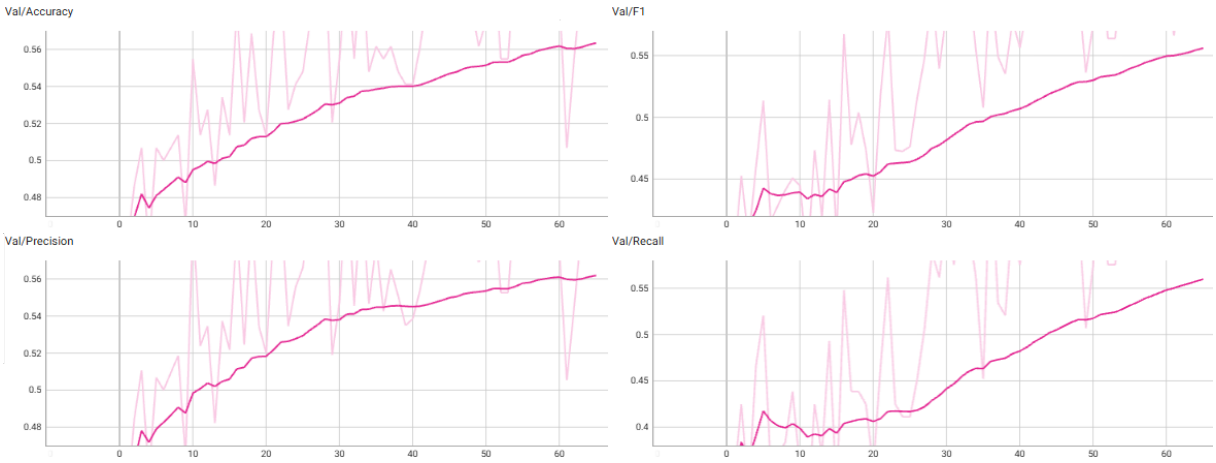


FIGURE 8.4 – Courbes de performance (Accuracy, F1-score, Précision et Rappel) pour le meilleur modèle utilisant l'embedding TransE, évalué sur le jeu de validation.

En résumé, cette étude met en évidence plusieurs enseignements. Premièrement, les signaux scalaires seuls se révèlent peu informatifs dans le corpus DGM[4], en raison du caractère subtil des altérations présentes. Deuxièmement, les graphes textuels et visuels constituent les signaux les plus discriminants, et leur combinaison améliore sensiblement la détection. Enfin, l'embedding TransE confirme son intérêt face à une initialisation aléatoire, en fournissant des représentations plus stables et performantes.

Dans l'ensemble, ces résultats suggèrent que la détection efficace de fake news multimodales repose principalement sur l'exploitation conjointe des graphes de connaissances et des signaux visuels, tandis que les mesures de cohérence texte-image doivent être enrichies pour contribuer réellement. Bien que les performances obtenues demeurent limitées, en partie à cause de la taille restreinte du jeu d'entraînement, cette étude démontre l'intérêt de la multimodalité d'un point de vue empirique et valide le bon fonctionnement de notre modèle.

Nous présentons ci-après des résultats complémentaires obtenus en remplaçant les R-GCN par des GCN classiques, et en utilisant l'embedding ComplEx à la place de TransE.

Comparaison avec les variantes du modèle

GCN simple. Bien que l'architecture R-GCN ait fourni des résultats concluants, il demeure pertinent de tester une variante plus simple, le GCN classique, qui ne prend pas en compte la nature des relations. Cette architecture présente l'avantage de réduire considérablement le nombre de paramètres du modèle, au prix d'une perte d'information contextuelle. Dans un scénario où la quantité de données d'entraînement est limitée, elle pourrait néanmoins mieux exploiter les ressources disponibles.

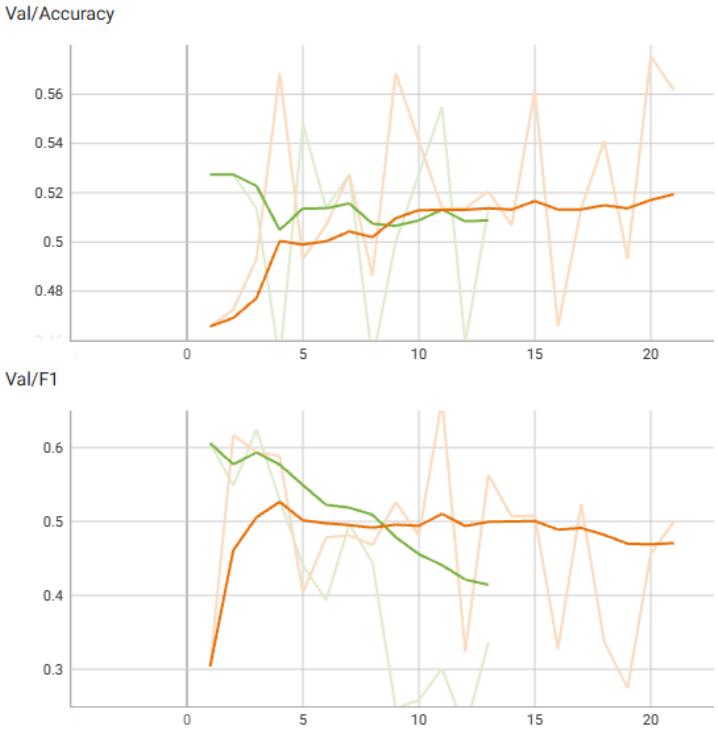


FIGURE 8.5 – Courbes d'accuracy et de F_1-score pour deux architectures GCN, avec trois signaux d'entrée, sur le jeu de validation. La configuration de la courbe orange comporte deux fois plus de couches GCN que celle de la courbe verte.

La Figure 8.5 rapporte deux courbes d'entraînement de notre modèle avec des GCN au lieu des R-GCN. La courbe verte correspond à la même configuration que notre meilleur modèle (Figure 8.4), c'est-à-dire avec les signaux graphe texte, graphe image et probabilité de transformation de l'image, ainsi que les mêmes hyperparamètres. Cet entraînement échoue rapidement et n'apprend rien au-delà de l'époque 3.

La courbe orange correspond à une configuration similaire, mais où la dimension des couches du GCN est doublée. L'hypothèse était que les paramètres libérés par l'absence de relations explicites pourraient être réinvestis dans la représentation des entités. Cependant, ce modèle échoue également, avec une dernière amélioration observée à l'époque 21 seulement.

Ces résultats nous amènent à conclure qu'une architecture GCN simple ne parvient pas à capturer l'information pertinente dans ce cadre, et renforcent notre choix de conserver une architecture R-GCN dans le pipeline final.

Embedding ComplEx. Nous avons également testé l'embedding ComplEx pré-entraîné sur WikiData5M, dont les résultats sont rapportés dans la Figure 8.6. Le GNN choisi est à nouveau un R-GCN. La courbe bleue correspond à la même configuration que notre meilleur modèle, à l'exception de l'utilisation de ComplEx à la place de TransE. Cet entraînement échoue rapidement et s'interrompt dès l'époque 11.

La courbe rose correspond à une variante plus complexe : nous avons doublé la profondeur des couches R-GCN afin d'exploiter davantage la richesse de l'embedding ComplEx par rapport à TransE, et augmenté la patience à 25 pour laisser davantage de temps au modèle avant l'arrêt anticipé. Cependant, cette configuration échoue également : l'apprentissage s'interrompt à l'époque 28, après une stagnation des performances observée dès l'époque 3.

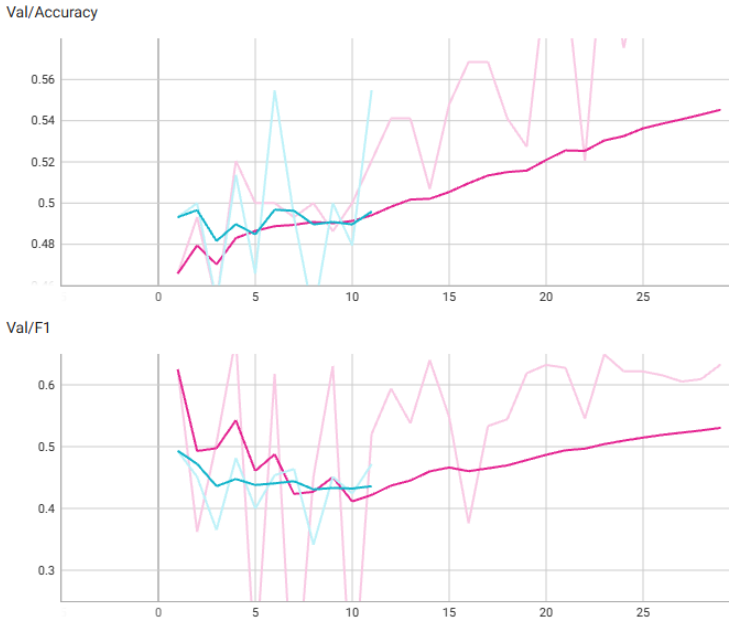


FIGURE 8.6 – Courbes d'accuracy et de F_1-score pour deux architectures R-GCN utilisant l'embedding ComplEx pré-entraîné sur WikiData5M, avec trois signaux d'entrée, sur le jeu de validation. La courbe rose correspond à une variante avec plus de couches R-GCN et une patience plus grande pour l'early stopping.

Temps de calcul pour l'entraînement. Nous mettons en évidence des différences significatives de temps de calcul selon l'embedding initial choisi pour les graphes. Sur un GPU Nvidia RTX4060, une époque d'entraînement prend en moyenne :

1. **20 secondes** avec un embedding aléatoire,
2. **2 minutes** avec un embedding TransE pré-entraîné,
3. **4 heures 20 minutes** avec un embedding ComplEx pré-entraîné sur WikiData5M.

Ainsi, l'utilisation de ComplEx se traduit par un coût de calcul extrêmement élevé, équivalant à plusieurs jours de calcul pour une seule architecture complète. Ce facteur constitue une contrainte pratique importante dans la conduite des expérimentations.

8.1.5 Tests sur d'autres jeux de données

Nous avons également évalué notre modèle sur deux jeux de données externes : (i) un sous-échantillon équilibré de 36 instances issues de Fakeddit [97], présenté dans le Chapitre 2, et (ii) 40 instances issues du jeu de données Twitter [7]. Ces deux corpus sont multimodaux et constitués de posts de réseaux sociaux annotés. Le jeu Twitter se distingue par sa plus grande complexité et sa proximité avec des conditions réelles, ce qui le rend a priori plus difficile à évaluer. Il traite d'évènements d'actualité comme les attentats de Charlie Hebdo ou le vol de la Malaysia AirLine, ou bien de faits plus marginaux.

Le Tableau 8.3 rapporte les résultats des trois meilleurs modèles (sur les datasets Fakeddit et Twitter) issus de l'entraînement, tous fondés sur une architecture R-GCN mais exploitant des combinaisons de signaux différentes. Le premier correspond au meilleur modèle validé sur DGM[4] : il exploite les graphes textuel et visuel ainsi que la probabilité de transformation de l'image. Le second plus performant est le modèle complet, intégrant l'ensemble des quatre signaux. Enfin, le troisième modèle, plus surprenant, n'exploite que le graphe visuel, associé à la probabilité de falsification et au score de cohérence CLIP, sans utiliser le graphe textuel. Ce tableau rappelle également les résultats obtenus sur le



FIGURE 8.7 – Deux exemples de fake news issues du dataset Twitter [7]. L'une est à visée profondément politique dans le contexte de la guerre en Ukraine, l'autre plus légère.

dataset DGM^4 et rapporte pour chaque dataset, des résultats obtenus par des méthodes de l'état de l'art, pour une comparaison adaptée.

L'analyse des performances montre que ce le modèle le moins performant des trois est biaisé : son Rappel est excessivement élevé mais son Accuracy très faible, car il prédit presque systématiquement la classe *fake* . Il s'avère donc inopérant en pratique. À l'inverse, le modèle qui obtenait les meilleurs résultats sur DGM^4 confirme sa robustesse en atteignant 0.61 d'Accuracy et 0.65 de F_1-score sur Fakeddit, ce qui constitue une performance correcte. Le modèle complet, légèrement moins performant sur Fakeddit, s'avère en revanche le plus efficace sur Twitter, où les autres modèles échouent largement. Ses résultats restent modestes (0.56 d'Accuracy, 0.59 de F_1, 0.63 d'AUC), mais ils reflètent la complexité intrinsèque du corpus Twitter, caractérisé par une grande hétérogénéité des falsifications et par des manipulations souvent subtiles. Cette complexité est marquée par les résultats obtenus dans l'état de l'art, plus faibles sur ce dataset que sur Fakeddit.

Nous notons que les modèles de l'état de l'art atteignent plus de 0.9 d'Accuracy sur Fakeddit, des performances largement supérieures aux nôtres. Néanmoins, notre approche se distingue par sa capacité d'explicabilité : au-delà de la classification brute, elle fournit des justifications visuelles et sémantiques aux prédictions. Cette dimension est essentielle dans le contexte de la détection de fake news, non seulement pour analyser les résultats a posteriori, mais également pour convaincre les utilisateurs trompés de l'inexactitude d'une information. La section suivante détaille précisément notre démarche pour intégrer l'explicabilité dans le processus de détection.

8.2 Explicabilité des Résultats

Dans un premier temps, nous cherchons à mesurer la contribution de chacun des signaux d'entrée à la prédiction finale. Pour ce faire, nous quantifions l'importance de chaque signal à la fois en entrée du MLP (pré-MLP) et dans la décision finale (post-MLP). Nous présentons ci-après la méthode retenue pour cette tâche.

Contexte : Notre modèle combine deux encodeurs de graphes (texte et image) et deux signaux scalaires (score CLIP et la probabilité d'image fake). Chaque donnée x est encodée

TABLE 8.3 – Synthèse des performances de nos modèles et de modèles de l'état de l'art sur *Fakeddit* et *Twitter* et *DGM*[4]. En rouge sont surlignées les meilleures configurations pour nos méthodes et pour celles de l'état de l'art, pour chaque dataset.

Dataset	Config	Acc	Prec	Rec	F1	AUC
Fakeddit	Graphe txt/img + prob fake img	**0.61**	0.62	0.68	0.65	**0.66**
	Graphe img + prob fake img + CLIP	0.50	0.51	**0.95**	**0.67**	0.50
	Graphe txt/img + prob fake img + CLIP	**0.61**	**0.65**	0.58	0.61	0.55
Fakeddit SoA	Fakeddit [97]	0.89	–	–	–	–
	GAMED [127]	**0.94**	0.95	0.94	0.95	–
Twitter	Graphe txt/img + prob fake img	0.46	0.46	0.60	0.52	0.50
	Graphe img + prob fake img + CLIP	0.44	0.46	**0.90**	0.61	0.45
	Graphe txt/img + prob fake img + CLIP	**0.56**	**0.54**	0.65	0.59	**0.63**
Twitter SoA	MVAE [64]	0.75	–	–	–	–
	JIN et al. [60]	0.68	–	–	0.68	–
	SpotFake [134]	0.78	–	–	–	–
	MCAN [161]	**0.81**	–	–	–	–
DGM[4] (Val)	Graphe txt/img + prob fake img	0.64	0.63	0.70	0.66	–
DGM[4] SoA	SDML [178]	0.75	–	–	–	–

en quatre tranches : $h_{\text{text}}(x) \in \mathbb{R}^{d_t}$, $h_{\text{img}}(x) \in \mathbb{R}^{d_i}$, $h_{clip}(x) \in \mathbb{R}$ (CLIP), $h_{fake-img}(x) \in \mathbb{R}$ (probabilité,fake image). Avant la tête de classification (MLP), nous appliquons des scalaires appris $\alpha_{\text{text}}, \alpha_{\text{img}}, \alpha_{\text{clip}}, \alpha_{\text{fake}}$ et concaténons :

$$h_{\text{joint}} = [\alpha_{\text{txt}} h_{\text{txt}}] \oplus [\alpha_{\text{img}} h_{\text{img}}] \oplus [\alpha_{\text{clip}} h_{\text{clip}}] \oplus [\alpha_{\text{fake-img}} h_{\text{fake-img}}]$$

La tête de classification $g(\cdot)$ renvoie un logit $\ell(x) = g(h_{joint}(x)) \in \mathbb{R}$ et une probabilité $p(x) = \sigma(\ell(x))$.

Mesure pré-MLP : Contribution effective

Nous mesurons pour chaque signal $s \in \{\text{text}, \text{img}, \text{clip}, \text{fake}\}$ l'énergie du signal fourni au MLP, définie comme sa norme ℓ_1 après sa pondération par α_s :

$$E_s(x) \;=\; \begin{cases} \|\alpha_{\text{text}}\, h_{\text{text}}(x)\|_1\,, & s = \text{text}, \\ \|\alpha_{\text{img}}\, h_{\text{img}}(x)\|_1\,, & s = \text{img}, \\ \|\alpha_{\text{clip}}\, h_{clip}(x)\|_1\,, & s = \text{clip}, \\ \|\alpha_{\text{fake}}\, h_{fake-img}(x)\|_1\,, & s = \text{fake-img} \end{cases}$$

Pour obtenir une décomposition relative par échantillon, on normalise en pourcentages :

$$\widetilde{E}_s(x) \;=\; \frac{E_s(x)}{\sum_j E_j(x)} \times 100.$$

Mesure post-MLP : Ablation par Signal (Δ logit)

Pour quantifier l'impact d'un signal sur la décision finale, on annule uniquement sa composante à l'entrée du MLP et on mesure la variation de logit. Soit M_s l'opérateur qui met à zéro les indices de h_{joint} correspondant à s. On définit, pour chaque échantillon,

$$\Delta_s(x) \;=\; \ell(s) \;-\; \ell\Big(M_s h_{joint}(x)\Big).$$

Nous travaillons ave le logit (plutôt qu'avec la probabilité) afin d'éviter les effets de normalisation du sigmoïde. Pour former des pourcentages robustes, on tronque les contributions négatives à 0 avant normalisation [1] :

$$\widetilde{\Delta}_s(x) \;=\; \frac{\max(0, \Delta_s(x))}{\sum_j \max(0, \Delta_j(x))} \times 100.$$

Interprétation et complémentarité des deux mesures : La mesure pré-MLP reflète la quantité d'information injectée par signal après pondération par α, tandis que l'ablation post-MLP mesure son impact causal sur la sortie du classifieur. Les deux vues sont complémentaires : un signal peut occuper une grande part de $h_{joint}(x)$, mais être peu utilisé par le MLP (poids faibles ou redondance), et inversement un petit signal scalaire peut avoir un fort effet sur le logit.

Analyse des contributions des signaux : La Figure 8.8 présente les contributions moyennes relatives de chaque signal, mesurées en entrée et en sortie du classifieur, sur le dataset Fakeddit. L'architecture considérée combine trois signaux : graphe textuel, graphe visuel et probabilité de transformation de l'image.

On observe d'abord que, bien que la probabilité de transformation de l'image ne représente qu'une faible part du signal en entrée (10.7%), elle devient le facteur dominant en sortie, contribuant en moyenne à 51.4% de la décision finale. Cela traduit une corrélation apprise par le modèle entre la probabilité de transformation visuelle et la présence d'une fake news, ce qui est cohérent avec les données.

Le graphe textuel constitue près de la moitié du signal en entrée (49.4%) et environ un tiers de la prédiction finale (32.9%). Sa contribution reste donc importante et relativement équilibrée, ce qui confirme son rôle discriminant dans la tâche.

1. Ainsi l'on observe seulement les contributions tirant la détection vers la classe fake

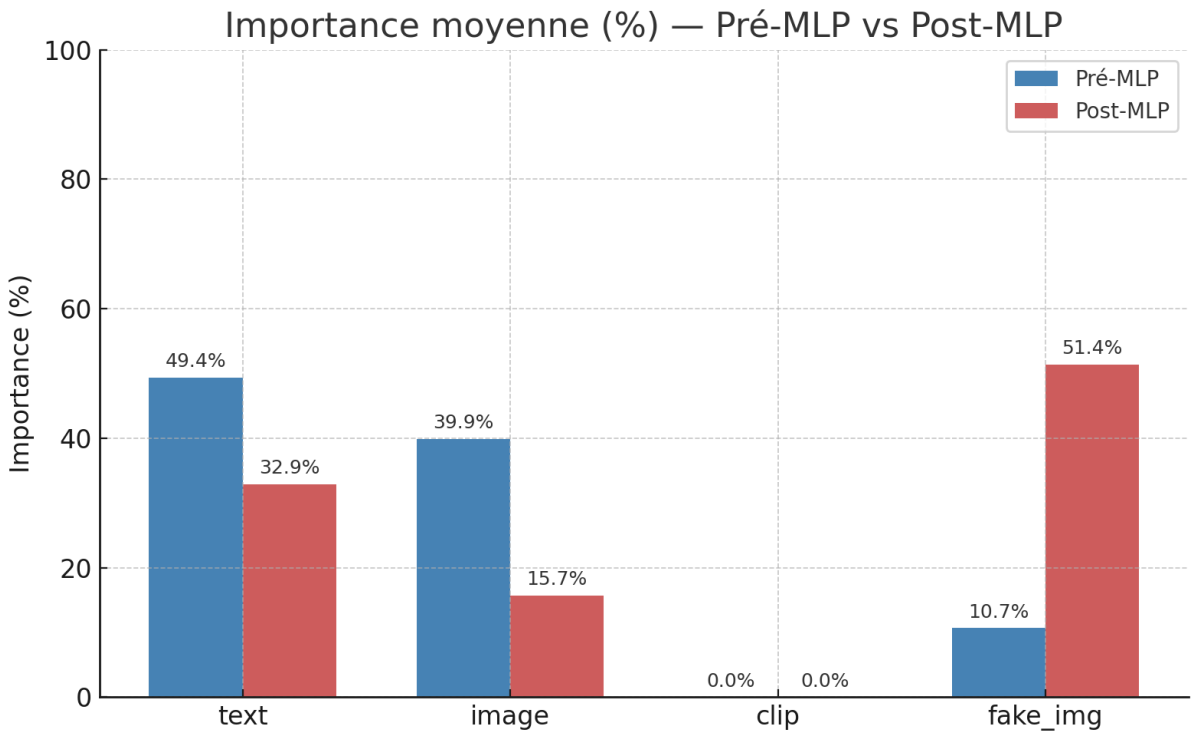


FIGURE 8.8 – Comparaison des contributions moyennes des signaux avant et après la classification avec le MLP sur le dataset Fakeddit.

À l'inverse, le graphe visuel contribue à 39.9% du signal en entrée, mais ne pèse plus que 15.7% en sortie. Cet écart suggère que l'information portée par le graphe visuel est redondante avec celle déjà encapsulée dans le signal de probabilité de transformation, qui prend alors le dessus dans la combinaison finale. Autrement dit, le modèle semble déléguer l'essentiel de la discrimination visuelle au score de falsification plutôt qu'au graphe image.

8.2.1 Analyses des prédictions sur deux exemples

Nous présentons dans cette sous-section deux exemples de prédiction avec explicabilité sur des instances issues de Fakeddit et de Twitter.

La Figure 8.9 illustre la prédiction de notre modèle sur un post Twitter affirmant que John Lennon et Che Guevara jouent de la guitare ensemble. Il s'agit d'une fake news. Notre modèle prédit une probabilité de fake news de 62%, supérieure au seuil $\tau = 50\%$, ce qui conduit correctement à une classification comme fake news, en accord avec la vérité terrain.

L'analyse des contributions montre toutefois que la décision repose uniquement sur le graphe textuel (79%) et, dans une moindre mesure, sur le graphe visuel (21%). Autrement dit, bien qu'il s'agisse manifestement d'un montage visuel, notre méthode ne le détecte pas directement : la classification finale provient essentiellement du texte. Cette incapacité peut avoir plusieurs causes : (i) une limite intrinsèque de notre méthode, qui n'est pas infaillible, ou (ii) une modification physique de l'image, qui ne laisse pas d'artefacts visuels bas niveau détectables.

La bonne prédiction malgré cette faiblesse illustre la robustesse d'une approche multimodale : la faillibilité d'un signal peut être compensée par la solidité des autres, garantissant ainsi une classification correcte dans des cas difficiles.

Quant à elle, la Figure 8.10 illustre une prédiction correcte sur une fake news issue du dataset Fakeddit, dans laquelle l'image est détectée comme falsifiée. L'analyse des contri-

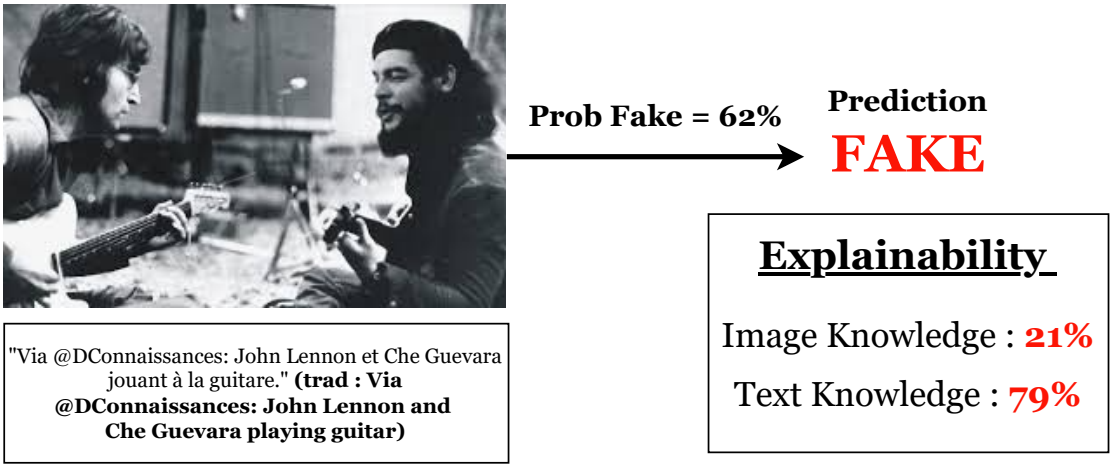


FIGURE 8.9 – Exemple de fake news du dataset Twitter [7] correctement prédite par notre modèle.

butions montre une compensation claire : le graphe visuel n'intervient que marginalement (6%), mais cette faiblesse est compensée par le score de probabilité de transformation de l'image (32%). Nous avions déjà mis en évidence ce phénomène dans le barplot (Figure 8.8), où l'information visuelle semblait palliée par ce score. Le graphe textuel reste le signal principal, contribuant à hauteur de 62% à la décision finale.

Dans ce cas, l'image étant reconnue par le modèle comme falsifiée, il est particulièrement pertinent d'enrichir l'explicabilité par une carte de localisation des zones transformées. L'exemple présenté est relativement simple, mais la carte met en évidence la transformation appliquée à la statuette des Oscars, illustrant l'intérêt de cette approche explicative.

8.2.2 Évaluation de la qualité des graphes extraits

Bien que la génération même des graphes textuels et visuels relève du prétraitement, nous choisissons de présenter ces résultats après l'entraînement du modèle afin d'éclaircir a posteriori les performances prédictives observées. Nous explorons les performances de la génération en terme d'hallucination et de variabilité sur la génération de chaque graphe. Ces expériences mettent en évidence les forces et faiblesses de la génération des graphes et ouvrent des perspectives d'amélioration de l'architecture globale.

8.2.3 Variabilité des graphes générés

Afin d'évaluer la stabilité et la robustesse de notre pipeline de génération de graphes, nous menons une série d'expériences sur un petit échantillon de 10 images aléatoires issues de DGM[4]. Pour chaque image, 50 graphes sont générés de la même manière que dans notre architecture (*cf.* Chapitre 7), puis comparés à la vérité terrain correspondante. Les graphes de vérité terrains ont été calculés par GPT4 puis corrigés manuellement. Dans cette partie, seule les entités sont étudiées et les graphes de vérité terrain correspondent donc simplement à une liste d'entités que nous ne normalisons pas, afin de garder un maximum d'information sémantique.

L'objectif est d'analyser la *variabilité intra-image* des graphes, c'est-à-dire la manière dont les entités et fluctuent d'une itération à l'autre sur une même image, et de quantifier la part d'*hallucinations* introduites par le modèle. Nous procédons sur 10 images pour

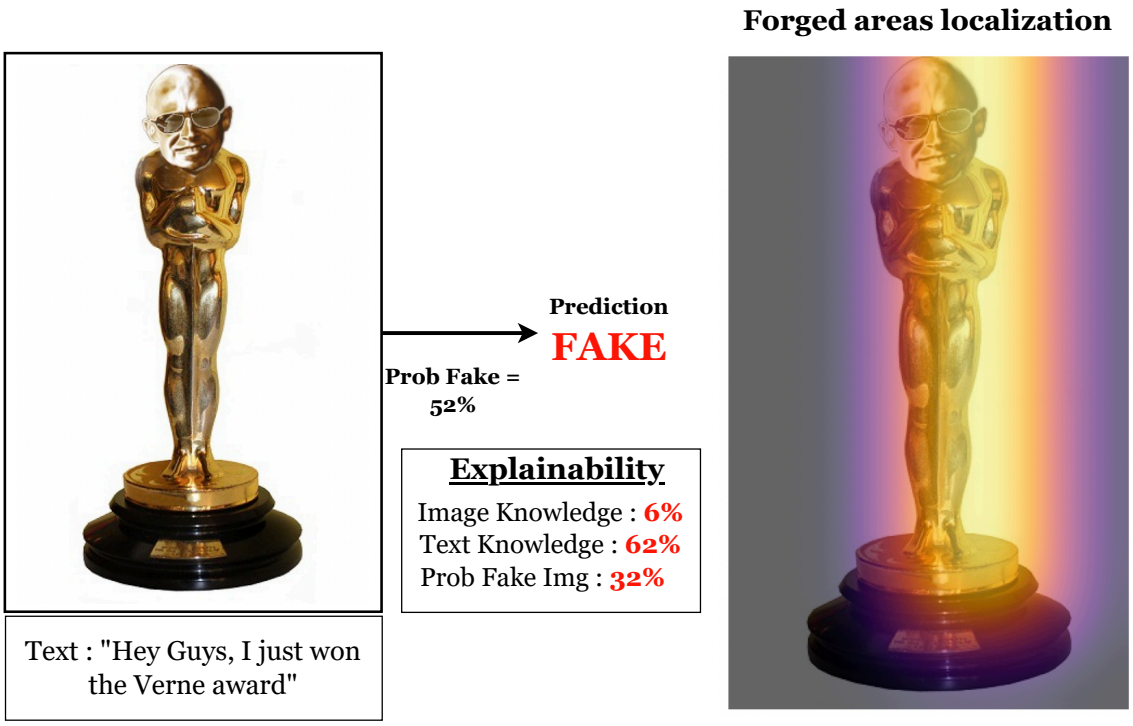


FIGURE 8.10 – Exemple de fake news du dataset Fakeddit [97] correctement prédite par notre modèle. La carte de localisation des transformations visuelles améliore l'explicabilité

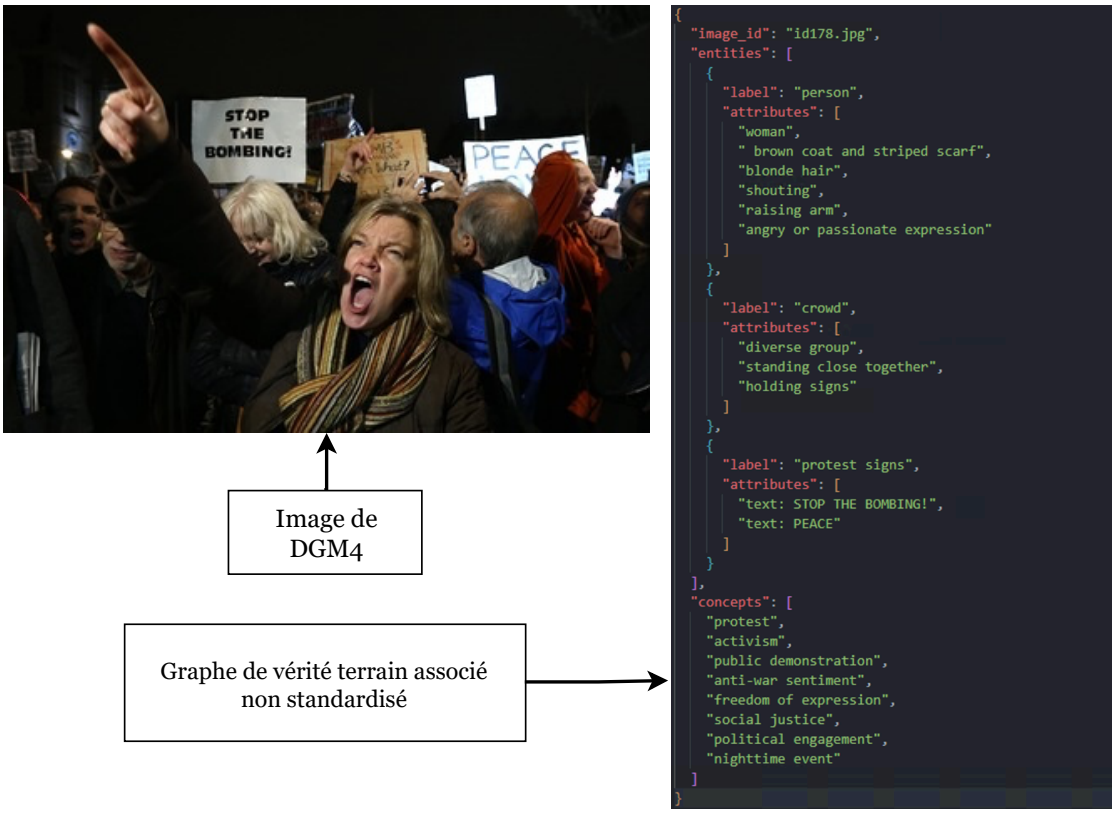


FIGURE 8.11 – Image de DGM[4] étudiée dans le cadre de l'étude de la variabilité des graphes, et sa liste d'entités vérifiées associée

éviter d'observer un phénomène trop spécifique à une scène donnée.

Métriques utilisées :

On note $X = (x_1, x_2, \ldots, x_{10})$ l'ensemble des images testées. Pour une image $x \in X$, on note $\hat{G}(x)$ le graphe généré et $G^\star(x)$ le graphe de vérité terrain. Les entités textuelles extraites sont comparées dans l'espace des représentations vectorielles fournies par `SentenceTransformer("all-MiniLM-L6-v2")`, qui permet de calculer une mesure de similarité cosinus (similarité sémantique) entre deux entités. Nous avons utilisé plusieurs indicateurs pour caractériser la variabilité des graphes générés :

Taux d'hallucination : On cherche à quantifier la proportion d'entités générées qui ne trouvent pas de correspondance dans le graphe de référence, c'est à dire hallucinée par le LLM. Soit $\tau \in [0, 1]$ un seuil de similarité cosinus. On définit l'ensemble des entités générées comme $E_{\hat{G}}(x)$ et celui des entités de référence comme $E_{G^\star}(x)$. Une entité $e \in E_{\hat{G}}(x)$ est dite *valide* si elle possède un appariement $e' \in E_{G^\star}(x)$ tel que :

$$\cos\Big(v(e), v(e')\Big) \geq \tau,$$

où $v(\cdot)$ désigne l'embedding issu du modèle `all-MiniLM-L6-v2`. Le taux d'hallucination est alors défini par :

$$H(x, \tau) = 1 - \frac{|\{e \in E_{\hat{G}}(x) \mid \exists e' \in E_{G^\star}(x),\ \cos(v(e), v(e')) \geq \tau\}|}{|E_{\hat{G}}(x)|}.$$

Cette métrique renseigne sur la part d'entités inventées par le modèle, ce qui est crucial pour évaluer la fiabilité des graphes dans une perspective de détection de fake news : un taux d'hallucination élevé peut introduire des biais et compromettre l'interprétation du graphe.

Similarité Jaccard cosinus (*Intersection over Union*) : On considère $E_{\hat{G}}(x)$ (entités générées) et $E_{G^\star}(x)$ (entités de référence). Chaque entité de $E_{G^\star}(x)$ joue le rôle de représentant de classe sémantique. On réalise un appariement des entités générées vers les entités de référence : en parcourant $e \in E_{\hat{G}}(x)$, on associe au plus une entité $e' \in E_{G^\star}(x)$ non encore utilisée telle que

$$\cos\Big(v(e), v(e')\Big) \ \geq \ \tau$$

où $v(\cdot)$ provient de `SentenceTransformer("all-MiniLM-L6-v2")` et τ est le seuil. On définit alors

$$J_\tau(x) \ = \ \begin{cases} 1, & \text{si } |E_{\hat{G}}(x)| = |E_{G^\star}(x)| = 0, \\ \dfrac{|\operatorname{match}_\tau(x)|}{|E_{\hat{G}}(x) \cup E_{G^\star}(x)|}, & \text{sinon.} \end{cases}$$

Intuitivement, on fusionne au maximum (greedy) les entités générées dans les classes sémantiques représentées par $E_{G^\star}(x)$, puis on rapporte la taille de cette intersection au nombre total d'entités distinctes (union).

Il est important de souligner que, dans notre définition, chaque entité de vérité terrain $e' \in E_{G^\star}(x)$ ne peut être appariée qu'à une seule entité générée $e \in E_{\hat{G}}(x)$. Ainsi, lorsqu'un modèle génère plusieurs variantes proches d'une même entité (par exemple *"Obama"*,

"Barack Obama", *"President Obama"*), une seule sera retenue dans le matching, tandis que les autres seront considérées comme non appariées.

La similarité de Jaccard $J_\tau(x)$ fournit une mesure normalisée du recouvrement entre le graphe généré et le graphe de référence. Elle capture à la fois la capacité du modèle à couvrir les entités attendues (intersection) et sa tendance à produire des entités supplémentaires (union). Un score élevé indique que la majorité des entités générées sont correctement appariées avec des entités de vérité terrain, tandis qu'un score faible révèle soit une proportion importante d'entités inventées, soit une production redondante de variantes lexicales qui ne trouvent pas de correspondance unique dans le graphe de référence. Ainsi, $J_\tau(x)$ doit être interprété comme un indicateur global de fidélité sémantique et de compacité des graphes générés.

Distribution intra-image : Pour une image x, on considère $n = 50$ réalisations du graphe généré $\{\hat{G}_k(x)\}_{k=1}^n$ (runs stochastiques). La distribution empirique des similarités Jaccard cosinus seuillées est

$$\mathcal{D}(x,\tau) \;=\; \Big\{\, J_\tau\Big(\hat{G}_k(x), G^\star(x)\Big) \;\Big|\; k = 1,\dots,n \,\Big\}.$$

On peut résumer la variabilité par des statistiques empiriques comme la moyenne, l'écart-type, la médiane. Une faible dispersion traduit une plus grande stabilité intra-image des graphes générés, et donc une plus grande stabilité dans la prédiction finale des fake news.

Cartographie des entités : Soit l'union des ensembles d'entités $\mathcal{E}(x) = E_{\hat{G}}(x) \cup E_{G^\star}(x)$. On projette chaque embedding de $\mathbb{R}^d$ (issu de `all-MiniLM-L6-v2`) vers $\mathbb{R}^2$ via une réduction de dimension $\phi : \mathbb{R}^d \to \mathbb{R}^2$ en python :

$$\tilde{v}(e) \;=\; \phi\Big(v(e)\Big) \in \mathbb{R}^2.$$

On définit l'ensemble des entités non hallucinées au seuil τ :

$$V(x,\tau) \;=\; \{\, e \in E_{\hat{G}}(x) \mid \exists\, e' \in E_{G^\star}(x),\ (e,e'), \cos(v(e), v(e')) \geq \tau \,\}$$

d'où l'ensemble des entités hallucinées au seuil τ :

$$H(x,\tau) \;=\; E_{\hat{G}}(x) \setminus V(x,\tau)$$

Nous pouvons alors visualiser en 2D ces ensembles en traçant $\{\tilde{v}(e) : e \in V(x,\tau)\}$ et et $\{\tilde{v}(e) : e \in H(x,\tau)\}$, ce qui permet d'inspecter visuellement la cohérence sémantique locale (amas de points compacts contre points dispersés) et d'identifier des zones propices aux hallucinations.

Résultats :

Hallucination en fonction du seuil : D'abord, nous quantifions avec la courbe 8.12 le taux d'hallucination des graphes générés en fonction de la sévérité du seuil de matching pour l'appariement des entités du graphe généré avec celles du graphe de vérité terrain. Au delà d'un seuil sévère de 0.85 le taux d'hallucination se stabilise à 93%, les 7% restant correspondent à un matching exact des entités. Lorsque le seuil est bas ($\tau = 0.5$), on observe tout de même un taux d'hallucination de 65%. Ce taux est assez élevé, malgré l'imprécision du processus de matching, cela indique du bruit non négligeable au sein des graphes visuels. Nous notons également que le taux d'hallucination en fonction du seuil montre une croissance quasi-linéaire pour des seuils compris entre 0.5 et 0.85.

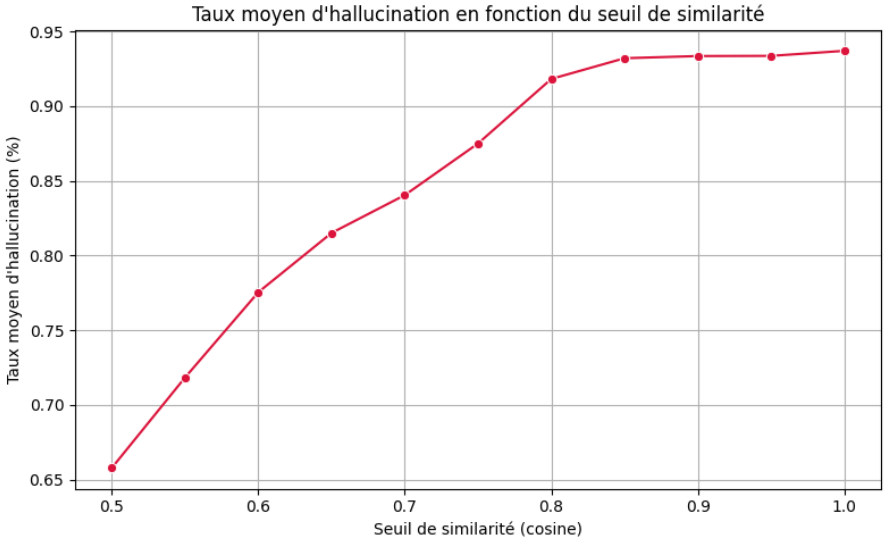


FIGURE 8.12 – Taux d'hallucination en fonction du seuil appliqué lors de la détection des entités visuelles.

Représentation 2D de l'hallucination : Nous représentons en deux dimensions sur la sous-figure 8.13b les entités de l'image 8.13a. Les points rouges correspondent aux entités hallucinées ($\tau = 0.5$) tandis que les points bleus correspondent aux entités matchées à la vérité terrain. Les points bleus s'amassent en clusters sémantiques, ici par exemple l'un est associé à la nature des personnes présentes : `gender, female, girl etc.`, ou un autre plutôt au décor et au contexte `library, computer, mouse etc.`. Ces regroupements sont attendus puisque le matching se fait sémantiquement, de même que la projection 2D.

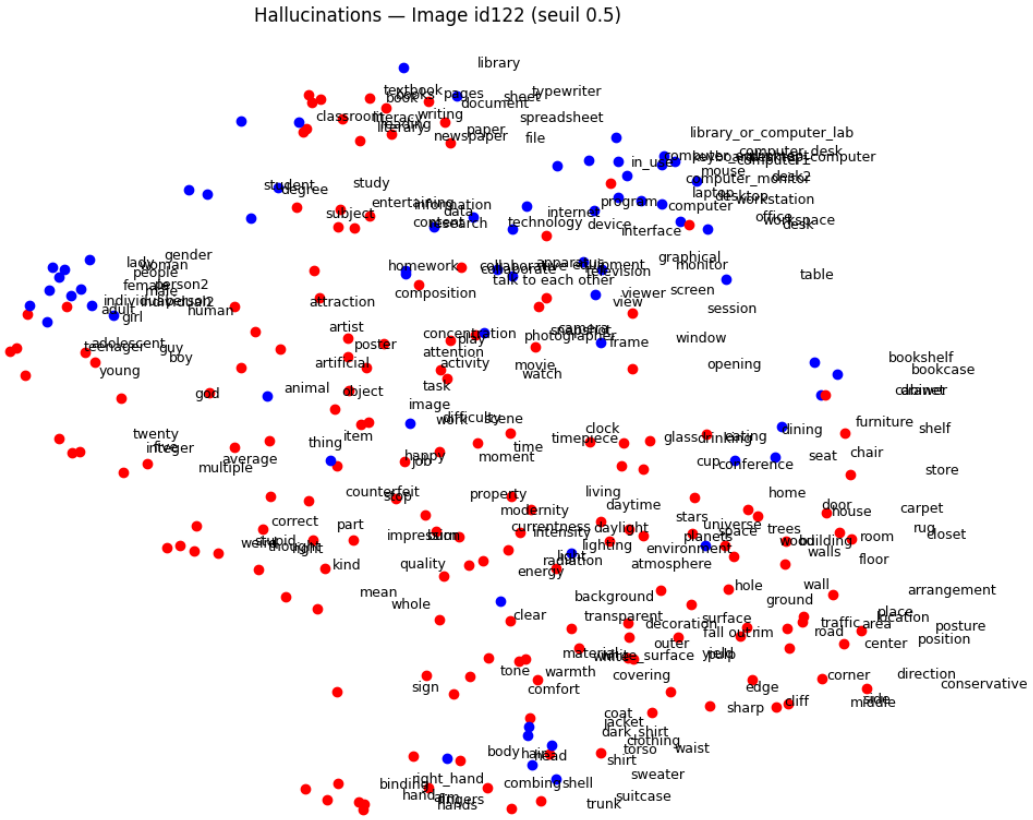
Parmi les entités hallucinées, on observe beaucoup de vocabulaire générique et non spécifique (ex. `quality, part, room, sign`). Ces entités sont du bruit informationnel qui peuvent noyer le signal utile pour la détection. Cependant il demeure assez difficile à trier de manière automatisée.

Distribution de la similarité Jaccard : La Figure 8.14 illustre la distribution des similarités Jaccard par image. On observe la variabilité *intra-image*, à partir l'étendue de chaque boîte et la présence éventuelle de valeurs extrêmes : certaines images (IDs 174, 228, 53) présentent une dispersion importante, indiquant une forte hétérogénéité des similarités, tandis que d'autres (IDs 122, 240) apparaissent plus concentrées. Cette différence de variabilité *intra-images* entre les images indique une certaine hétérogénéité dans la fiabilité du modèle selon les images. On peut aussi observer la variabilité *inter-images*, qui se manifeste par les différences de hauteur de la médiane : les images (IDs 122, 174, 178) affichent des similarités globalement plus élevées, alors que d'autres (ID 367 notamment) présentent des valeurs faibles et homogènes. Ces observations suggèrent que la stabilité des graphes générés dépend fortement du contenu visuel initial.

Synthèse sur la variabilité des graphes générés : Ces expériences mettent en évidence que notre génération de graphes est sujette à une variabilité importante, qui affecte négativement la détection de fake news dans sa robustesse et sa fiabilité. La présence d'hallucinations, peut biaiser l'interprétation du graphe, en particulier dans le cadre des fake news, et affecter la classification. Cependant, nos graphes contiennent tout de même une quantité non négligeables d'informations utiles, et d'ailleurs exploitée avec succès par le modèle final. En effet celui-ci performe mieux en usant des graphes que sans, nous nous



(a)



(b) Visualisation des entités détectées et hallucinations pour une image (ID 122) au seuil 0.5.

FIGURE 8.13 – Exemple de visualisation des entités hallucinées dans le graphe généré d'une image de DGM[4].

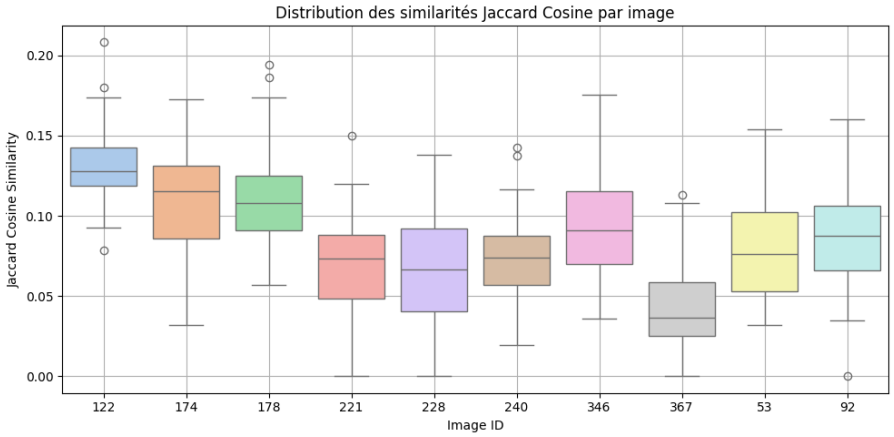


FIGURE 8.14 – Distribution des scores de similarité Jaccard et cosinus par image dans l'évaluation des graphes extraits.

appuyons également dessus pour justifier nos prédictions.

8.3 Synthèse des expérimentations

Les expérimentations menées dans ce chapitre permettent de tirer plusieurs enseignements majeurs sur la faisabilité et les limites de notre approche pour la détection multimodale de fake news enrichie par des graphes de connaissances.

Qualité de l'entraînement du modèle et rôle des embeddings : Nos résultats montrent que l'utilisation d'embeddings pré-entraînés (TransE) pour faciliter l'apprentissage des GNN apporte un gain systématique par rapport à une initialisation aléatoire, en fournissant une base sémantique stable aux R-GCN. À l'inverse, l'exploitation d'embeddings plus complexes (ComplEx) s'avère inadaptée dans notre configuration : leur coût computationnel élevé et la taille réduite des données d'entraînement entraînent un échec de l'apprentissage. Ce constat illustre le compromis entre richesse des représentations et capacité d'entraînement effectif lorsque les données sont limitées.

Impact des différents signaux : L'étude d'ablation a montré que les signaux scalaires (score CLIP et probabilité de falsification visuelle) sont peu informatifs pris isolément, mais peuvent renforcer la robustesse de la fusion multimodale. Les graphes textuels et visuels constituent en revanche les signaux les plus discriminants isolément. Leur combinaison, enrichie par la probabilité d'image falsifiée, offre la meilleure configuration observée. Cette observation confirme que la structure sémantique apportée par les graphes est centrale pour la tâche, en accord avec l'hypothèse de départ de ce travail.

Comparaison des architectures : Les tests ont mis en évidence la supériorité nette des R-GCN sur les GCN simples. Ces derniers, incapables d'exploiter la diversité relationnelle, échouent à capturer des indices pertinents dans le corpus. Cela justifie pleinement l'intégration d'une modélisation relationnelle explicite au sein de notre pipeline.

Évaluation et comparaison sur d'autres dataset : L'application du modèle sur Fakeddit et Twitter a montré des résultats modestes mais cohérents : bien que les performances quantitatives restent inférieures à l'état de l'art (accuracy ≈ 0.9 sur Fakeddit),

notre modèle conserve une robustesse appréciable (accuracy = 0.61, F1 = 0.65 sur Fakeddit), et démontre sa transférabilité partielle sur Twitter, malgré la complexité de ce corpus. Ces expériences confirment que la généralisation reste limitée par la taille réduite du dataset d'entraînement et la complexité des falsifications rencontrées.

Explicabilité : La quantification des contributions des trois signaux a montré que le modèle ne s'appuie pas uniformément ceux-ci. Lorsque l'image est détectée comme fake, la probabilité de falsification d'image domine la décision finale, tandis que le graphe textuel reste une source d'information stable et robuste. Le graphe visuel, quant à lui, tend à être redondant et voit son impact atténué. Cette complémentarité, bien qu'imparfaite, constitue un atout majeur pour la fiabilité du modèle. De plus, la localisation des transformation renforce la dimension explicative de notre modèle.

Variabilité des graphes générés : L'évaluation spécifique de la génération de graphes met en lumière une forte variabilité intra et inter images. Le taux d'hallucination, qui atteint jusqu'à 65% à seuil faible et 93% à seuil strict, illustre la difficulté à contrôler la qualité des entités extraites. La similarité Jaccard (IoU) et la visualisation des hallucination montrent que, malgré ce bruit, une part significative des entités utiles est préservée, ce qui explique que les graphes contribuent effectivement à la détection finale. Ces résultats soulignent toutefois l'intérêt de mécanismes de filtrage ou de normalisations pour limiter l'impact des redondances et hallucinations lors de la génération des graphes.

Conclusion de la synthèse. Dans l'ensemble, ces expériences confirment la validité de notre pipeline, en particulier de l'intérêt des graphes de connaissances textuels et visuels pour améliorer la détection de fake news multimodales. Néanmoins, plusieurs limites apparaissent : (i) La taille réduite du corpus d'entraînement, (ii) La variabilité des graphes générés, (iii) La dépendance à quelques signaux dominants. Ces limites ouvrent des perspectives claires : Elargissement des données d'entraînement, amélioration de la génération de graphes (désambiguïsation, filtrage des hallucinations), et intégration de signaux multimodaux plus riches.

Ce chapitre a présenté l'ensemble des expérimentations menées afin d'évaluer les différentes composantes de notre approche. Le Chapitre 9 conclut ce manuscrit en proposant une synthèse des contributions, une discussion des limites, ainsi que des perspectives de recherche ouvertes par ce travail.

9 Conclusion, Discussion, Perspectives

Au cours de la dernière décennie, la prolifération massive des fake news s'est imposée comme un enjeu majeur pour l'ensemble des citoyens. La viralité et la sophistication croissante de ces actualités multimédias trompeuses, amplifiées par l'émergence des modèles génératifs, rendent la tâche de leur détection particulièrement complexe. Dans ce contexte, la recherche scientifique tente de proposer des solutions techniques fiables et explicables, tout en s'inscrivant dans une démarche éthique plus large de préservation de la confiance dans l'information.

Le travail présenté au long de ce manuscrit suit pleinement cette perspective. Nous avons proposé une approche automatique de détection des fake news, multimodale, s'appuyant sur des représentations visuelles et textuelles, et enrichies de connaissances externes. Nous prêtons de plus un intérêt certain, dans notre méthode, à l'explicabilité des prédictions, dimension indispensable pour favoriser l'acceptabilité et la compréhension des systèmes de détection par les différents acteurs (chercheurs, journalistes, citoyens).

9.1 Synthèse des contributions

Les contributions principales de cette thèse peuvent être résumées ainsi :

Contribution dans l'état de l'art : Nous avons établi un état de l'art approfondi sur la détection des fake news, en distinguant les approches basées sur le contenu, sur le contexte social et sur la source de l'information. Cette analyse a mis en lumière un manque notable dans la littérature : l'absence de méthodes de détection automatique multimodales, capables d'intégrer à la fois le contenu et des connaissances externes, tout en fournissant une explicabilité développée des prédictions. Une telle combinaison permet d'adresser le problème des fake news dans une logique proche du fact-checking journalistique. La méthode proposée dans ce manuscrit vise précisément à combler ce manque identifié.

Contribution sur la localisation de transformations d'images : Nous avons proposé l'utilisation du clustering SNMF pour localiser les zones transformées dans les images, en supposant la disponibilité d'une mesure de similarité forensique entre images. Cette contribution s'inscrit dans le contexte de détection explicable des fake news visuelles unimodales, connexe à l'objectif final de détection multimodale. Les résultats expérimentaux ont montré une certaine robustesse face à des transformations variées, notamment des manipulations de type *deepfake*.

Contribution sur l'exploration de nouvelles méthodes de détection d'images falsifiées : Nous avons exploré une approche *pixel-to-image*, reposant sur l'exploitation des cartes de localisation des transformations produites par un réseau de neurones convolutionnel pré-entraîné à cette fin. En formulant une hypothèse de connexité des zones

transformées, cette méthode permet d'estimer un score de probabilité de transformation de l'image, fournissant ainsi une probabilité de falsification exploitable dans le cadre de la détection automatique.

Contribution sur les données disponibles pour la localisation de transformations d'images : Nous avons développé et mis à disposition une application web permettant la génération automatique d'images falsifiées, tout en garantissant leur cohérence sémantique. Cet outil constitue une ressource originale pour l'évaluation des modèles de détection de transformations d'images, dans un contexte où les bases de données publiques demeurent limitées. Il facilite la création de scènes variées, offrant ainsi un support utile à la communauté scientifique.

Contribution sur la génération de graphes de scènes par LLM : Nous avons proposé une méthode *end-to-end* pour la génération de graphes de scènes, afin de structurer l'information sémantique contenue dans les images et d'en faciliter la compréhension, notamment pour améliorer les entrées de tâches en aval telles que la détection de fake news. Ce travail constitue également une analyse exploratoire des performances des LLMs dans la génération *zero-shot* de graphes de scènes. Une attention importante a été portée au choix de la granularité des graphes générés, afin d'adapter leur niveau de détail en fonction de la tâche aval considérée.

Dans cette optique, nous avons mobilisé à la fois des *Large Language Models* (LLMs) et des *Vision-Language Models* (VLMs) pour la génération de graphes de scènes. Cette approche permet d'exploiter conjointement la structure sémantique et l'ancrage visuel des entités en jeu pour une compréhension approfondie des images. Une évaluation rigoureuse a été menée sur plusieurs jeux de données de référence (Visual Genome, OpenImages), montrant le potentiel des approches granulaires *LLM-based* et zero-shot pour la génération de graphes de scènes. Bien que les performances quantitatives restent en deçà de ceux obtenus par des méthodes spécialisées et *fine-tuned*, ils ouvrent la voie à des scénarios plus génériques et transférables, dans lesquels l'adaptabilité et la modularité priment sur l'optimisation spécifique.

Contribution sur la détection explicable de fake news multimodales : Nous avons proposé une architecture multimodale explicable pour la détection automatique des fake news, intégrant plusieurs modalités complémentaires : des connaissances issues du contenu textuel et du contenu visuel, et des signaux issus de l'analyse de falsifications d'images. L'approche repose sur une fusion pondérée des représentations des connaissances apprises par des R-GCN et qui permet de combiner des signaux hétérogènes tout en offrant un mécanisme d'explicabilité de la prédiction.

Nous avons conduit une série d'expérimentations sur plusieurs jeux de données de référence (DGM[4], Fakeddit, Twitter), afin d'évaluer la faisabilité et l'efficacité de notre approche. Ces expériences ont montré que l'architecture proposée est pleinement fonctionnelle et qu'elle apprend effectivement à exploiter les signaux multimodaux et enrichis de connaissances extraits des données. Si les performances quantitatives demeurent inférieures à celles des meilleurs modèles de l'état de l'art, nos résultats démontrent néanmoins le potentiel de cette architecture et soulignent son intérêt pour des approches explicables et généralisables.

Une attention particulière a été portée à l'explicabilité : nous avons proposé un cadre original permettant de visualiser et quantifier l'apport de chaque modalité dans la décision finale, illustré par des résultats qualitatifs sur des exemples concrets. Par ailleurs, nous avons étudié la qualité des graphes de connaissances générés (en termes de redondance,

de variabilité et d'hallucinations). Si ces graphes présentent encore un certain niveau de bruit, leur utilisation contribue néanmoins à améliorer les performances globales de détection, confirmant l'intérêt d'intégrer ce type de connaissances dans un cadre multimodal explicable.

9.2 Discussion des limites

Malgré ces contributions, plusieurs limites subsistent :

Données : Les jeux de données utilisés dans nos expérimentations, bien que déjà variés et complexes, demeurent limités par leur taille, leur diversité contextuelle et leur représentativité des informations mondiales. En effet, ils sont en grande majorité rédigés en anglais et portent principalement sur des actualités occidentales. Une telle restriction thématique et linguistique peut réduire la capacité de généralisation du modèle, et son application à des contextes géopolitiques ou culturels élargis peut conduire à une baisse significative de performance.

Méthodologie : La modularité de notre architecture, bien qu'elle soit au service du traitement multimodal et de l'explicabilité, présente également des limites. Ainsi, chaque module peut nécessiter un calibrage plus fin, et la fusion multimodale peut s'avérer imparfaite, certains signaux prenant parfois le pas sur d'autres dans le processus de décision.

Quant à elle, la génération des graphes de connaissances dépend en grande partie des performances intrinsèques des LLMs et VLMs employés, eux-mêmes sensibles aux biais présents dans leurs données d'entraînement et particulièrement coûteux en ressources computationnelles. Leur taille et leur complexité rendent par ailleurs leur entraînement ou leur adaptation (par *fine-tuning*) difficile à mettre en œuvre.

De plus, le signal exploité pour la détection des transformations d'images n'est pas exempt de biais et sa robustesse n'est pas garantie face à l'évolution rapide des techniques de falsification, notamment l'émergence de *deepfakes* toujours plus trompeurs.

Explicabilité : Bien que notre approche propose un score de contribution par signal, ainsi qu'une carte de localisation des transformations de l'image lorsqu'elles existent, l'explicabilité est perfectible. Elle pourrait être renforcée en identifiant de manière explicite les faits ou relations précises des graphes de connaissances ayant conduit à la prédiction finale.

9.3 Perspectives de recherche

Ce travail de thèse ouvre plusieurs pistes de recherche :

Amélioration méthodologique : Chacun des signaux exploités dans notre approche peut faire l'objet d'améliorations spécifiques. La détection des transformations d'images pourrait être rendue plus robuste.

La génération de graphes de connaissances, qu'ils soient visuels ou textuels, pourrait être renforcée en mobilisant des LLMs plus performants ou d'autres techniques de construction d'ontologies. Des études approfondies seraient nécessaires pour déterminer les méthodes d'extraction de connaissances les plus pertinentes, ainsi que le niveau optimal d'intégration d'informations issues de bases externes.

Enfin, la fusion multimodale pourrait être affinée par l'introduction de véritables mécanismes d'attention, permettant une pondération dynamique et plus fine des différents signaux.

Explicabilité avancée : Il serait pertinent de développer des approches d'explication plus fines, capables de générer des justifications explicites des décisions et d'identifier les faits précis responsables de la prédiction. Parallèlement, le déploiement d'une application web interactive permettrait de rendre ces mécanismes explicatifs accessibles à un public plus large, favorisant ainsi une utilisation démocratisée et compréhensible par des journalistes ou n'importe quel citoyen.

Élargissement des modalités : Les prochaines années devraient voir se démocratiser de nouvelles formes de fake news, notamment la vidéo, rendue accessible par les performances croissantes des modèles génératifs spécialisés. Dans ce contexte, l'intégration de signaux issus d'autres modalités, telles que l'audio ou la vidéo, semble être une étape nécessaire pour lutter efficacement contre la désinformation.

Par ailleurs, l'exploitation de signaux complémentaires, tels que des informations liées à la source ou au contexte de publication, pourrait enrichir à la fois la performance du modèle et la qualité de son explicabilité.

9.4 Conclusion

Cette thèse a apporté des contributions à la compréhension et à la détection automatique, basée sur le contenu, des fake news multimodales, en mobilisant à la fois des approches fondées sur les connaissances et l'exploitation d'artefacts visuels de bas niveau, dans une perspective résolument orientée vers l'explicabilité. Les perspectives identifiées ouvrent la voie à de nombreux travaux futurs, qu'il s'agisse de tâches amont, comme la constitution de jeux de données plus divers contextuellement, d'améliorations méthodologiques de l'architecture elle-même, ou encore de tâches aval, telles que le déploiement d'applications interactives permettant de démocratiser l'usage de ces modèles.

Ce travail souligne l'importance de poursuivre les recherches vers des systèmes de détection plus robustes et plus transparents, capables de rétablir la confiance entre les citoyens et l'information, et de contribuer durablement à la lutte contre la désinformation dans nos sociétés numériques.

Bibliographie

[1] Meta AI. *LLaMA 3.2 11B Vision*. Accessed April 2025. 2024. URL : https://huggingface.co/meta-llama/Llama-3.2-11B-Vision.

[2] Jawaher Alghamdi, Suhuai Luo et Yuqing Lin. « A comprehensive survey on machine learning approaches for fake news detection ». In : *Multimedia Tools and Applications* 83.17 (2024), p. 51009-51067.

[3] Belhassen Bayar et Matthew C. Stamm. « Constrained Convolutional Neural Networks : A New Approach Towards General Purpose Image Manipulation Detection ». In : *IEEE Transactions on Information Forensics and Security* 13.11 (2018), p. 2691-2706.

[4] Catherine Beauvais. « Fake news : Why do we believe it ? » In : *Joint Bone Spine* 89.4 (2022), p. 105371. ISSN : 1297-319X. DOI : https://doi.org/10.1016/j.jbspin.2022.105371. URL : https://www.sciencedirect.com/science/article/pii/S1297319X22000306.

[5] Jonathan Berant et al. « Semantic parsing on freebase from question-answer pairs ». In : *Proceedings of the 2013 conference on empirical methods in natural language processing*. 2013, p. 1533-1544.

[6] Gajanan K Birajdar et Vijay H Mankar. *Digital image forgery detection using passive techniques : A survey*. 2013.

[7] Christina Boididou et al. « Detection and visualization of misleading content on Twitter ». In : *International Journal of Multimedia Information Retrieval* 7.1 (1er mars 2018), p. 71-86. DOI : 10.1007/s13735-017-0143-x. URL : https://doi.org/10.1007/s13735-017-0143-x.

[8] David Boissonneault et Emily Hensen. *Fake News Detection with Large Language Models on the LIAR Dataset*. Mai 2024. DOI : 10.21203/rs.3.rs-4465815/v1.

[9] Kurt Bollacker et al. « Freebase : A collaboratively created graph database for structuring human knowledge ». In : juin 2008, p. 1247-1250. DOI : 10.1145/1376616.1376746.

[10] Antoine Bordes et al. « Translating Embeddings for Modeling Multi-relational Data ». In : *Advances in Neural Information Processing Systems*. Sous la dir. de C.J. Burges et al. T. 26. Curran Associates, Inc., 2013.

[11] Michael Brennan et Rachel Greenstadt. « Practical Attacks Against Authorship Recognition Techniques ». English. In : *Proceedings of the 21st Innovative Applications of Artificial Intelligence Conference, IAAI-09*. 21st Innovative Applications of Artificial Intelligence Conference, IAAI-09 ; Conference date : July 14–16, 2009. 2009, p. 60-65. ISBN : 9781577354239. URL : https://cdn.aaai.org/ocs/i/iaai0020/257-3903-1-PB.pdf.

[12] Samuel Broscheit et al. « LibKGE - A Knowledge Graph Embedding Library for Reproducible Research ». In : *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*. 2020, p. 165-174. URL : https://www.aclweb.org/anthology/2020.emnlp-demos.22.

[13] Matt Burgess. *Elon Musk's 'Rate Limits' Are Slowing Twitter to a Crawl*. 2023. URL : https://www.wired.com/story/elon-musk-twitter-rate-limits-extremely-hardcore/.

[14] Juan Cao et al. « Exploring the role of visual content in fake news detection ». In : *Disinformation, Misinformation, and Fake News in Social Media : Emerging Research Challenges and Opportunities* (2020), p. 141-161.

[15] Tiago José de Carvalho et al. « Exposing Digital Image Forgeries by Illumination Color Classification ». In : *IEEE Transactions on Information Forensics and Security* 8.7 (2013), p. 1182-1194. DOI : 10.1109/TIFS.2013.2265677.

[16] V. Chandola, A. Banerjee et V. Kumar. « Anomaly Detection : A Survey ». In : *ACM Comput. Surv.* 41.3 (2009).

[17] Xiaojun Chang et al. « Scene Graphs : A Survey of Generations and Applications ». In : *CoRR* abs/2104.01111 (2021). arXiv : 2104.01111. URL : https://arxiv.org/abs/2104.01111.

[18] Vinay Chaudhri et al. « Knowledge Graphs : Introduction, History and, Perspectives ». In : *AI Magazine* 43.1 (mars 2022), p. 17-29. DOI : 10.1002/aaai.12033. URL : https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/19119.

[19] Can Chen, Luca Zanotti Fragonara et Antonios Tsourdos. « GAPointNet : Graph attention based point neural network for exploiting local feature of point cloud ». In : *Neurocomputing* 438 (2021), p. 122-132.

[20] Jie Chen, Yousef Saad et Zechen Zhang. « Graph coarsening : from scientific computing to machine learning ». In : *SeMA Journal* 79.1 (2022), p. 187-223. ISSN : 2281-7875. DOI : 10.1007/s40324-021-00282-x. URL : https://doi.org/10.1007/s40324-021-00282-x.

[21] Tianshui Chen et al. *Knowledge-Embedded Routing Network for Scene Graph Generation*. 2019. arXiv : 1903.03326 [cs.CV]. URL : https://arxiv.org/abs/1903.03326.

[22] Giovanni Luca Ciampaglia et al. « Computational Fact Checking from Knowledge Networks ». In : *PLOS ONE* 10.6 (juin 2015), p. 1-13. DOI : 10.1371/journal.pone.0128193. URL : https://doi.org/10.1371/journal.pone.0128193.

[23] Weilin Cong, William Wang et Wang-Chien Lee. « Scene Graph Generation via Conditional Random Fields ». In : *CoRR* abs/1811.08075 (2018). Withdrawn. arXiv : 1811.08075. URL : http://arxiv.org/abs/1811.08075.

[24] contact@dalle2.gallery. *DallůE 2 largest image database*. https://dalle2.gallery.

[25] Limeng Cui et Dongwon Lee. *CoAID : COVID-19 Healthcare Misinformation Dataset*. 2020. arXiv : 2006.00885 [cs.SI]. URL : https://arxiv.org/abs/2006.00885.

[26] Michaël Defferrard, Xavier Bresson et Pierre Vandergheynst. « Convolutional neural networks on graphs with fast localized spectral filtering ». In : *Advances in neural information processing systems* 29 (2016).

[27] Jiankang Deng et al. « ArcFace : Additive Angular Margin Loss for Deep Face Recognition ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[28] Jiankang Deng et al. « RetinaFace : Single-Shot Multi-Level Face Localisation in the Wild ». In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[29] Leon Derczynski et al. « SemEval-2017 Task 8 : RumourEval : Determining rumour veracity and support for rumours ». In : *arXiv preprint arXiv:1704.05972* (2017).

[30] Jacob Devlin et al. « Bert : Pre-training of deep bidirectional transformers for language understanding ». In : *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics : human language technologies, volume 1 (long and short papers)*. 2019, p. 4171-4186.

[31] Jing Dong, Wei Wang et Tieniu Tan. « CASIA Image Tampering Detection Evaluation Database ». In : *2013 IEEE China Summit and International Conference on Signal and Information Processing*. IEEE, juill. 2013. doi : 10.1109/chinasip.2013.6625374. url : https://doi.org/10.1109/chinasip.2013.6625374.

[32] Xin Dong et al. « Knowledge vault : A web-scale approach to probabilistic knowledge fusion ». In : *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, p. 601-610.

[33] Alexey Dosovitskiy et al. « An image is worth 16x16 words : Transformers for image recognition at scale ». In : *arXiv preprint arXiv:2010.11929* (2020).

[34] Lisa Ehrlinger et Wolfram Wöß. « Towards a definition of knowledge graphs. » In : *SEMANTiCS (Posters, Demos, SuCCESS)* 48.1-4 (2016), p. 2.

[35] *Fake News Challenge (FNC-1) Stance Detection Dataset*. 2018. url : http://www.fakenewschallenge.org/.

[36] Tharindu Fernando et al. « Detection of fake and fraudulent faces via neural memory networks ». In : *IEEE Transactions on Information Forensics and Security* 16 (2020), p. 1973-1988.

[37] France Culture. *FranceSoir, grandeur et déliquescence d'un journal devenu ń anti-journalistique ż*. 2022. url : https://www.radiofrance.fr/franceculture/france-soir-grandeur-et-deliquescence-d-un-journal-devenu-anti-journalistique-9661755.

[38] Shane Frederick. « Cognitive Reflection and Decision Making ». In : *Journal of Economic Perspectives* 19.4 (déc. 2005), p. 25-42. doi : 10.1257/089533005775196732.

[39] Alberto García-Durán, Antoine Bordes et Nicolas Usunier. « Effective blending of two and three-way interactions for modeling multi-relational data ». In : *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2014, p. 434-449.

[40] Akansha Gautam, Venktesh V et Sarah Masud. *Fake News Detection System using XLNet model with Topic Distributions : CONSTRAINT@AAAI2021 Shared Task*. 2021. arXiv : 2101.11425 [cs.CL]. url : https://arxiv.org/abs/2101.11425.

[41] Anastasia Giachanou, Guobiao Zhang et Paolo Rosso. « Multimodal multi-image fake news detection ». In : *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)*. IEEE. 2020, p. 647-654.

[42] Sherry Girgis, Eslam Amer et Mahmoud Gadallah. « Deep learning algorithms for detecting fake news in online text ». In : *2018 13th international conference on computer engineering and systems (ICCES)*. IEEE. 2018, p. 93-97.

[43] Mohammad Hadi Goldani, Saeedeh Momtazi et Reza Safabakhsh. « Detecting fake news with capsule neural networks ». In : *Applied Soft Computing* 101 (2021), p. 106991.

[44] Thierry Grillet et al. *Histoire de fausses nouvelles*. BnF Essentiels, 2022. URL : https://essentiels.bnf.fr/fr/enseignants/b64967cc-0065-4fe1-9490-9b1c7d51250b-fausses-nouvelles.

[45] J. Grim, P. Somol et P.Pudil. « Digital Image Forgery Detection by Local Statistical Models ». In : *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (2010), p. 579-582. DOI : 10.1109/IIHMSP.2010.147.

[46] Shu Guo et al. « Semantically Smooth Knowledge Graph Embedding ». In : *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*. Sous la dir. de Chengqing Zong et Michael Strube. Beijing, China : Association for Computational Linguistics, juill. 2015, p. 84-94. DOI : 10.3115/v1/P15-1009. URL : https://aclanthology.org/P15-1009.

[47] Xiao Guo et al. *Hierarchical Fine-Grained Image Forgery Detection and Localization*. 2023. arXiv : 2303.17111 [cs.CV]. URL : https://arxiv.org/abs/2303.17111.

[48] William L. Hamilton, Rex Ying et Jure Leskovec. *Inductive Representation Learning on Large Graphs*. 2018. arXiv : 1706.02216 [cs.SI]. URL : https://arxiv.org/abs/1706.02216.

[49] Yi Han et al. « Knowledge enhanced multi-modal fake news detection ». In : *arXiv preprint arXiv:2108.04418* (2021).

[50] Shizhu He et al. « Learning to represent knowledge graphs with gaussian embedding ». In : *Proceedings of the 24th ACM international on conference on information and knowledge management*. 2015, p. 623-632.

[51] Aidan Hogan et al. « Knowledge Graphs ». In : *ACM Comput. Surv.* 54.4 (juill. 2021). ISSN : 0360-0300. DOI : 10.1145/3447772. URL : https://doi.org/10.1145/3447772.

[52] Yu-Feng Hsu et Shih-Fu Chang. « Detecting Image Splicing Using Geometry Invariants And Camera Characteristics Consistency ». In : *Interational Conference on Multimedia and Expo (ICME)*. Toronto, Canada, 2006.

[53] Xuefeng Hu et al. « SPAN : Spatial pyramid attention network for image manipulation localization ». In : *European conference on computer vision*. Springer. 2020, p. 312-328.

[54] Fangjun Huang, Jiwu Huang et Yun Qing Shi. « Detecting Double JPEG Compression With the Same Quantization Matrix ». In : *IEEE Transactions on Information Forensics and Security* 5.4 (2010), p. 848-856. DOI : 10.1109/TIFS.2010.2072921.

[55] HuggingFaceH4. *Zephyr-7B-Alpha*. Accessed April 2025. 2023. URL : https://huggingface.co/HuggingFaceH4/zephyr-7b-alpha.

[56] Guoliang Ji et al. « Knowledge graph completion with adaptive sparse transfer matrix ». In : *Proceedings of the AAAI conference on artificial intelligence*. T. 30. 1. 2016.

[57] Guoliang Ji et al. « Knowledge graph embedding via dynamic mapping matrix ». In : *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1 : Long papers)*. 2015, p. 687-696.

[58] Ziwei Ji et al. « Survey of Hallucination in Natural Language Generation ». In : *ACM Comput. Surv.* 55.12 (mars 2023). ISSN : 0360-0300. DOI : 10.1145/3571730. URL : https://doi.org/10.1145/3571730.

[59] Yuheng Jia et al. *Ensemble Clustering via Co-association Matrix Self-enhancement*. 2023.

[60] Zhiwei Jin et al. « Multimodal Fusion with Recurrent Neural Networks for Rumor Detection on Microblogs ». In : *Proceedings of the 25th ACM International Conference on Multimedia*. MM '17. Mountain View, California, USA : Association for Computing Machinery, 2017, p. 795-816. ISBN : 9781450349062. DOI : 10.1145/3123266.3123454. URL : https://doi.org/10.1145/3123266.3123454.

[61] Zhiwei Jin et al. « Novel Visual and Statistical Image Features for Microblogs News Verification ». In : *IEEE Transactions on Multimedia* 19.3 (2017), p. 598-608. DOI : 10.1109/TMM.2016.2617078.

[62] Heejung Jwa et al. « exBAKE : Automatic Fake News Detection Model Based on Bidirectional Encoder Representations from Transformers (BERT) ». In : *Applied Sciences* 9.19 (2019). ISSN : 2076-3417. DOI : 10.3390/app9194062. URL : https://www.mdpi.com/2076-3417/9/19/4062.

[63] Ravpreet Kaur et Sarbjeet Singh. « A comprehensive review of object detection with deep learning ». In : *Digital Signal Processing* 132 (2023), p. 103812. ISSN : 1051-2004. DOI : https://doi.org/10.1016/j.dsp.2022.103812. URL : https://www.sciencedirect.com/science/article/pii/S1051200422004298.

[64] Dhruv Khattar et al. « MVAE : Multimodal Variational Autoencoder for Fake News Detection ». In : *The World Wide Web Conference*. WWW '19. San Francisco, CA, USA : Association for Computing Machinery, 2019, p. 2915-2921. ISBN : 9781450366748. DOI : 10.1145/3308558.3313552. URL : https://doi.org/10.1145/3308558.3313552.

[65] Kibum Kim et al. *LLM4SGG : Large Language Models for Weakly Supervised Scene Graph Generation*. 2024. arXiv : 2310.10404 [cs.CV]. URL : https://arxiv.org/abs/2310.10404.

[66] TN Kipf. « Semi-Supervised Classification with Graph Convolutional Networks ». In : *arXiv preprint arXiv:1609.02907* (2016).

[67] Elena Kochkina, Maria Liakata et Arkaitz Zubiaga. *PHEME dataset for Rumour Detection and Veracity Classification*. Dataset. 2018. DOI : 10.6084/m9.figshare.6392078.v1. URL : https://doi.org/10.6084/m9.figshare.6392078.v1.

[68] Chenqi Kong et al. « Pixel-inconsistency modeling for image manipulation localization ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025).

[69] Ranjay Krishna et al. *Visual Genome : Connecting Language and Vision Using Crowdsourced Dense Image Annotations*. 2016. arXiv : 1602.07332 [cs.CV]. URL : https://arxiv.org/abs/1602.07332.

[70] Alex Krizhevsky, Ilya Sutskever et Geoffrey E Hinton. « ImageNet Classification with Deep Convolutional Neural Networks ». In : *Advances in Neural Information Processing Systems*. Sous la dir. de F. Pereira et al. T. 25. Curran Associates, Inc., 2012. URL : https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[71] Da Kuang, Chris Ding et Haesun Park. *Symmetric nonnegative matrix factorization for graph clustering*. 2012.

[72] Alina Kuznetsova et al. « The Open Images Dataset V4 : Unified image classification, object detection, and visual relationship detection at scale ». In : *IJCV* (2020).

[73] Junnan Li et al. *BLIP-2 : Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*. 2023. arXiv : 2301.12597 [cs.CV]. URL : https://arxiv.org/abs/2301.12597.

[74] Xin Li et Dan Roth. « Learning question classifiers ». In : *COLING 2002 : The 19th International Conference on Computational Linguistics*. 2002.

[75] Xinyi Li, Yongfeng Zhang et Edward C Malthouse. « Large language model agent for fake news detection ». In : *arXiv preprint arXiv:2405.01593* (2024).

[76] Yankai Lin et al. « Learning entity and relation embeddings for knowledge graph completion ». In : *Proceedings of the AAAI conference on artificial intelligence*. T. 29. 1. 2015.

[77] Chuang Liu et al. « Graph pooling for graph neural networks : Progress, challenges, and opportunities ». In : *arXiv preprint arXiv:2204.07321* (2022).

[78] Hanxiao Liu, Yuexin Wu et Yiming Yang. « Analogical inference for multi-relational embeddings ». In : *International conference on machine learning*. PMLR. 2017, p. 2168-2178.

[79] Haotian Liu et al. « Visual Instruction Tuning ». In : *NeurIPS*. 2023.

[80] Shilong Liu et al. « Grounding dino : Marrying dino with grounded pre-training for open-set object detection ». In : *European conference on computer vision*. Springer. 2024, p. 38-55.

[81] Xuannan Liu et al. « FakeNewsGPT4 : Advancing Multimodal Fake News Detection through Knowledge-Augmented LVLMs ». In : *CoRR* abs/2403.01988 (2024). URL : https://doi.org/10.48550/arXiv.2403.01988.

[82] Yang Liu et Yi-Fang Brook Wu. « Early Detection of Fake News on Social Media through Propagation Path Classification with Recurrent and Convolutional Networks ». In : *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI'18/IAAI'18/EAAI'18. New Orleans, Louisiana, USA : AAAI Press, 2018. ISBN : 978-1-57735-800-8.

[83] Yinhan Liu et al. « Roberta : A robustly optimized bert pretraining approach ». In : *arXiv preprint arXiv:1907.11692* (2019).

[84] Ulrike von Luxburg. « A Tutorial on Spectral Clustering ». In : (2007). DOI : 10.48550/ARXIV.0711.0189. URL : https://arxiv.org/abs/0711.0189.

[85] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[86] Elio Masciari et al. « Detecting fake news by image analysis ». In : *Proceedings of the 24th symposium on international database engineering & Applications*. 2020, p. 1-5.

[87] Mohit Mayank, Shakshi Sharma et Rajesh Sharma. « DEAP-FAKED : Knowledge Graph based Approach for Fake News Detection ». In : *CoRR* abs/2107.10648 (2021). arXiv : 2107.10648. URL : https://arxiv.org/abs/2107.10648.

[88] Owen Mayer, Belhassen Bayar et Matthew C. Stamm. « Learning Unified Deep-Features for Multiple Forensic Tasks ». In : *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*. New York, NY, USA : Association for Computing Machinery, 2018, p. 79-84.

[89] Owen Mayer et Matthew C. Stamm. *Exposing Fake Images with Forensic Similarity Graphs*. 2019.

[90] Owen Mayer et Matthew C. Stamm. « Forensic Similarity for Digital Images ». In : *IEEE Transactions on Information Forensics and Security* 15 (2020), p. 1331-1346. ISSN : 1556-6021. DOI : 10.1109/tifs.2019.2924552.

[91] Media Bias/Fact Check. *Le Monde Bias and Reliability*. 2024. URL : https://mediabiasfactcheck.com/le-monde-bias/.

[92] Meta AI. *Meta-Llama-3-8B*. https://huggingface.co/meta-llama/Meta-Llama-3-8B. Accessed : April 2025. 2024.

[93] Tobias Mettler et Robert Winter. « Are business users social ? A design experiment exploring information sharing in enterprise social systems ». In : *Journal of Information Technology* 31.2 (2016), p. 101-114. DOI : 10.1057/jit.2015.28. eprint : https://doi.org/10.1057/jit.2015.28. URL : https://doi.org/10.1057/jit.2015.28.

[94] Tomas Mikolov. « Efficient estimation of word representations in vector space ». In : *arXiv preprint arXiv:1301.3781* 3781 (2013).

[95] George A Miller. « WordNet : a lexical database for English ». In : *Communications of the ACM* 38.11 (1995), p. 39-41.

[96] Eni Mustafaraj et Panagiotis Takis Metaxas. *The Fake News Spreading Plague : Was it Preventable ?* 2017. arXiv : 1703.06988 [cs.SI].

[97] Kai Nakamura, Sharon Levy et William Yang Wang. « r/Fakeddit : A New Multimodal Benchmark Dataset for Fine-grained Fake News Detection ». In : *arXiv preprint arXiv:1911.03854* (2019).

[98] Maximilian Nickel, Lorenzo Rosasco et Tomaso Poggio. « Holographic embeddings of knowledge graphs ». In : *Proceedings of the AAAI conference on artificial intelligence*. T. 30. 1. 2016.

[99] Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel et al. « A three-way model for collective learning on multi-relational data. » In : *Icml*. T. 11. 2011, p. 3104482-3104584.

[100] Femi Olan et al. « Fake news on Social Media : the Impact on Society ». In : *Information Systems Frontiers* (jan. 2022). ISSN : 1572-9419. DOI : 10.1007/s10796-022-10242-z. URL : https://doi.org/10.1007/s10796-022-10242-z.

[101] Jeff Z. Pan et al. « Content Based Fake News Detection Using Knowledge Graphs ». In : *The Semantic Web – ISWC 2018*. Sous la dir. de Denny Vrandei et al. Cham : Springer International Publishing, 2018, p. 669-683.

[102] Bo Pang et Lillian Lee. « Seeing stars : Exploiting class relationships for sentiment categorization with respect to rating scales ». In : *arXiv preprint cs/0506075* (2005).

[103] Guansong Pang et al. « Deep Learning for Anomaly Detection ». In : *ACM Computing Surveys* 54.2 (mars 2021), p. 1-38.

[104] Parth Patwa et al. « Fighting an Infodemic : COVID-19 Fake News Dataset ». In : *Combating Online Hostile Posts in Regional Languages during Emergency Situation*. Sous la dir. de Tanmoy Chakraborty et al. Cham : Springer International Publishing, 2021, p. 21-29.

[105] Jeffrey Pennington, Richard Socher et Christopher D Manning. « Glove : Global vectors for word representation ». In : *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, p. 1532-1543.

[106] Gordon Pennycook et David G. Rand. « The Psychology of Fake News ». In : *Trends in Cognitive Sciences* 25.5 (2021), p. 388-402. issn : 1364-6613. doi : https://doi.org/10.1016/j.tics.2021.02.007. url : https://www.sciencedirect.com/science/article/pii/S1364661321000516.

[107] Gordon Pennycook et al. « Shifting attention to accuracy can reduce misinformation online ». In : *Nature* 592 (avr. 2021), p. 1-6. doi : 10.1038/s41586-021-03344-2.

[108] Nam Thanh Pham et al. « Hybrid Image-Retrieval Method for Image-Splicing Validation ». In : *Symmetry* 11.1 (2019), p. 83.

[109] Huyen Trang Phan, Ngoc Thanh Nguyen et Dosam Hwang. « Fake news detection : A survey of graph neural network methods ». In : *Applied Soft Computing* 139 (2023), p. 110235.

[110] Dustin Podell et al. *SDXL : Improving Latent Diffusion Models for High-Resolution Image Synthesis*. 2023. arXiv : 2307.01952 [cs.CV].

[111] PolitiFact. *PolitiFact : an American nonprofit factchecking website*. https://www.politifact.com/.

[112] Peng Qi et al. « Exploiting multi-domain visual information for fake news detection ». In : *2019 IEEE international conference on data mining (ICDM)*. IEEE. 2019, p. 518-527.

[113] Xuebin Qin et al. « U2-Net : Going deeper with nested U-structure for salient object detection ». In : *Pattern Recognition* 106 (oct. 2020), p. 107404. issn : 0031-3203. doi : 10.1016/j.patcog.2020.107404. url : http://dx.doi.org/10.1016/j.patcog.2020.107404.

[114] Alec Radford et al. « Learning transferable visual models from natural language supervision ». In : *International conference on machine learning*. PmLR. 2021, p. 8748-8763.

[115] Aditya Ramesh et al. *DallůE 2*.

[116] Yuan Rao, Jiangqun Ni et Hao Xie. « Multi-semantic CRF-based attention model for image forgery detection and localization ». In : *Signal Processing* 183 (2021), p. 108051. issn : 0165-1684. doi : https://doi.org/10.1016/j.sigpro.2021.108051. url : https://www.sciencedirect.com/science/article/pii/S0165168421000906.

[117] Harita Reddy et al. « Text-mining-based Fake News Detection Using Ensemble Methods ». In : *International Journal of Automation and Computing* 17.2 (avr. 2020), p. 210-221.

[118] Joseph Redmon et Ali Farhadi. « Yolov3 : An incremental improvement ». In : *arXiv preprint arXiv:1804.02767* (2018).

[119] Nils Reimers et Iryna Gurevych. « Sentence-bert : Sentence embeddings using siamese bert-networks ». In : *arXiv preprint arXiv:1908.10084* (2019).

[120] Andreas Rossler et al. « FaceForensics++ : Learning to Detect Manipulated Facial Images ». In : *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[121] Victoria L Rubin et al. « Fake news or truth ? using satirical cues to detect potentially misleading news ». In : *Proceedings of the second workshop on computational approaches to deception detection*. 2016, p. 7-17.

[122] Ekraam Sabir et al. « Recurrent convolutional strategies for face manipulation detection in videos ». In : *Interfaces (GUI)* 3.1 (2019), p. 80-87.

[123] Sara Sabour, Nicholas Frosst et Geoffrey E. Hinton. « Dynamic Routing Between Capsules ». In : *CoRR* abs/1710.09829 (2017). arXiv : 1710.09829. URL : http://arxiv.org/abs/1710.09829.

[124] Mohammad Amin Sadeghi et Ali Farhadi. « Recognition using Visual Phrases ». In : (2011).

[125] Michael Schlichtkrull et al. « Modeling relational data with graph convolutional networks ». In : *European semantic web conference*. Springer. 2018, p. 593-607.

[126] Rui Shao, Tianxing Wu et Ziwei Liu. « Detecting and Grounding Multi-Modal Media Manipulation ». In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.

[127] Lingzhi Shen et al. « Gamed : Knowledge adaptive multi-experts decoupling for multimodal fake news detection ». In : *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*. 2025, p. 586-595.

[128] Jianbo Shi et J. Malik. « Normalized cuts and image segmentation ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), p. 888-905. DOI : 10.1109/34.868688.

[129] Kai Shu et al. *Fake News Detection on Social Media : A Data Mining Perspective*. 2017. arXiv : 1708.01967 [cs.SI]. URL : https://arxiv.org/abs/1708.01967.

[130] Kai Shu et al. « FakeNewsNet : A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media ». In : *CoRR* abs/1809.01286 (2018). arXiv : 1809.01286. URL : http://arxiv.org/abs/1809.01286.

[131] Kai Shu et al. *Hierarchical Propagation Networks for Fake News Detection : Investigation and Exploitation*. 2019. arXiv : 1903.09196 [cs.SI]. URL : https://arxiv.org/abs/1903.09196.

[132] Karen Simonyan et Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv : 1409.1556 [cs.CV]. URL : https://arxiv.org/abs/1409.1556.

[133] Rahul Singh. *Convert Corpus of text into Graph of Knowledge*. https://github.com/rahulnyk/knowledge_graph. Accessed : 2025-04-11. 2024.

[134] Shivangi SINGHAL et al. « Spotfake : A multi-modal framework for fake news detection ». In : *2019 IEEE fifth international conference on multimedia big data BigMM*. IEEE. 2019, p. 39-47.

[135] Niraj SITAULA et al. « Credibility-based Fake News Detection ». In : *CoRR* abs/1911.00643 (2019). eprint : 1911.00643. URL : http://arxiv.org/abs/1911.00643.

[136] Richard SOCHER et al. « Reasoning with neural tensor networks for knowledge base completion ». In : *Advances in neural information processing systems* 26 (2013).

[137] Robyn SPEER, Joshua CHIN et Catherine HAVASI. « ConceptNet 5.5 : An Open Multilingual Graph of General Knowledge ». In : *CoRR* abs/1612.03975 (2016). arXiv : 1612.03975. URL : http://arxiv.org/abs/1612.03975.

[138] Alexander STREHL et Joydeep GHOSH. « Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions. » In : *J. Mach. Learn. Res.* 3 (2002), p. 583-617.

[139] Charles SUTTON et Andrew MCCALLUM. « An Introduction to Conditional Random Fields ». In : *Found. Trends Mach. Learn.* 4.4 (avr. 2012), p. 267-373. ISSN : 1935-8237. DOI : 10.1561/2200000013. URL : https://doi.org/10.1561/2200000013.

[140] Eugenio TACCHINI et al. *Some Like it Hoax : Automated Fake News Detection in Social Networks*. 2017. arXiv : 1704.07506 [cs.LG].

[141] Ollama TEAM. *Ollama : Run open-source large language models locally*. Accessed : 2025-04-12. 2023. URL : https://ollama.com.

[142] James THORNE et al. *FEVER : a large-scale dataset for Fact Extraction and VERification*. 2018. arXiv : 1803.05355 [cs.CL]. URL : https://arxiv.org/abs/1803.05355.

[143] Petter TÖRNBERG. « Echo chambers and viral misinformation : Modeling fake news as complex contagion ». In : *PLOS ONE* 13.9 (sept. 2018), p. 1-21. DOI : 10.1371/journal.pone.0203. URL : https://ideas.repec.org/a/plo/pone00/0203958.html.

[144] Théo TROUILLON et al. « Complex Embeddings for Simple Link Prediction ». In : *CoRR* abs/1606.06357 (2016). arXiv : 1606.06357. URL : http://arxiv.org/abs/1606.06357.

[145] Nikhita VEDULA et Srinivasan PARTHASARATHY. « FACE-KEG : Fact Checking Explained using KnowledgE Graphs ». In : New York, NY, USA : Association for Computing Machinery, 2021. ISBN : 9781450382977. DOI : 10.1145/3437963.3441828. URL : https://doi.org/10.1145/3437963.3441828.

[146] Petar VELIKOVI et al. *Graph Attention Networks*. 2018. arXiv : 1710.10903 [stat.ML]. URL : https://arxiv.org/abs/1710.10903.

[147] Pawan Kumar VERMA et al. « WELFake : Word Embedding Over Linguistic Features for Fake News Detection ». In : *IEEE Transactions on Computational Social Systems* 8.4 (2021), p. 881-893. DOI : 10.1109/TCSS.2021.3068519.

[148] Nathalie Diane WANDJI, Xingming SUN et Moise Fah KUE. « Detection of copy-move forgery in digital images based on DCT ». In : *CoRR* abs/1308.5661 (2013).

[149] Jianfeng WANG et al. *GIT : A Generative Image-to-text Transformer for Vision and Language*. 2022. arXiv : 2205.14100 [cs.CV].

[150] Jingyi WANG et al. *LLaVA-SG : Leveraging Scene Graphs as Visual Semantic Expression in Vision-Language Models*. 2024. arXiv : 2408.16224 [cs.CV]. URL : https://arxiv.org/abs/2408.16224.

[151] Junke WANG et al. « Objectformer for image manipulation detection and localization ». In : *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, p. 2364-2373.

[152] Quan WANG et al. « Knowledge Graph Embedding : A Survey of Approaches and Applications ». In : *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017), p. 2724-2743. DOI : 10.1109/TKDE.2017.2754499.

[153] William Yang WANG. « "Liar, Liar Pants on Fire" : A New Benchmark Dataset for Fake News Detection ». In : *CoRR* abs/1705.00648 (2017). arXiv : 1705.00648. URL : http://arxiv.org/abs/1705.00648.

[154] Xiaozhi WANG et al. *KEPLER : A Unified Model for Knowledge Embedding and Pre-trained Language Representation*. 2020. arXiv : 1911.06136 [cs.CL].

[155] Zhen WANG et al. « Knowledge Graph and Text Jointly Embedding ». In : *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Sous la dir. d'Alessandro MOSCHITTI, Bo PANG et Walter DAELEMANS. Doha, Qatar : Association for Computational Linguistics, oct. 2014, p. 1591-1601. DOI : 10.3115/v1/D14-1167. URL : https://aclanthology.org/D14-1167.

[156] Zhen WANG et al. « Knowledge graph embedding by translating on hyperplanes ». In : *Proceedings of the AAAI conference on artificial intelligence*. T. 28. 1. 2014.

[157] Apurva WANI et al. « Evaluating deep learning approaches for covid19 fake news detection ». In : *Combating Online Hostile Posts in Regional Languages during Emergency Situation : First International Workshop, CONSTRAINT 2021, Collocated with AAAI 2021, Virtual Event, February 8, 2021, Revised Selected Papers 1*. Springer. 2021, p. 153-163.

[158] Claire WARDLE. *Fake news. It's complicated*. First Draft News. Publié le 16 février 2017, consulté en ligne sur First Draft News. Fév. 2017. URL : https://firstdraftnews.org/articles/fake-news-complicated/.

[159] Andrew P. WEISS et al. « Toward a Comprehensive Model of Fake News : A New Approach to Examine the Creation and Sharing of False Information ». In : *Societies* 11.3 (juill. 2021), p. 1-17. URL : https://ideas.repec.org/a/gam/jsoctx/v11y2021i3p82-d595397.html.

[160] Janyce WIEBE, Theresa WILSON et Claire CARDIE. « Annotating expressions of opinions and emotions in language ». In : *Language resources and evaluation* 39.2 (2005), p. 165-210.

[161] Yang WU et al. « Multimodal Fusion with Co-Attention Networks for Fake News Detection ». In : *Findings of the Association for Computational Linguistics : ACL-IJCNLP 2021*. Sous la dir. de Chengqing ZONG et al. Online : Association for Computational Linguistics, août 2021, p. 2560-2569. DOI : 10.18653/v1/2021.findings-acl.226. URL : https://aclanthology.org/2021.findings-acl.226/.

[162] Han XIAO, Minlie HUANG et Xiaoyan ZHU. « From one point to a manifold : Knowledge graph embedding for precise link prediction ». In : *arXiv preprint arXiv:1512.04792* (2015).

[163] Danfei Xu et al. *Scene Graph Generation by Iterative Message Passing*. 2017. arXiv : 1701.02426 [cs.CV]. URL : https://arxiv.org/abs/1701.02426.

[164] Keyulu Xu et al. « How powerful are graph neural networks ? » In : *arXiv preprint arXiv:1810.00826* (2018).

[165] An Yang et al. *Qwen2 Technical Report*. 2024. arXiv : 2407.10671 [cs.CL]. URL : https://arxiv.org/abs/2407.10671.

[166] Bishan Yang et al. « Embedding entities and relations for learning and inference in knowledge bases ». In : *arXiv preprint arXiv:1412.6575* (2014).

[167] Ling Yang et al. *Diffusion Models : A Comprehensive Survey of Methods and Applications*. 2023. arXiv : 2209.00796 [cs.LG].

[168] Yang Yang et al. *TI-CNN : Convolutional Neural Networks for Fake News Detection*. 2023. arXiv : 1806.00749 [cs.CL]. URL : https://arxiv.org/abs/1806.00749.

[169] Zhilin Yang et al. « Xlnet : Generalized autoregressive pretraining for language understanding ». In : *Advances in neural information processing systems* 32 (2019).

[170] Qifan Yu et al. *Visually-Prompted Language Model for Fine-Grained Scene Graph Generation in an Open World*. 2023. arXiv : 2303.13233 [cs.CV]. URL : https://arxiv.org/abs/2303.13233.

[171] Wael AbdAlmageed Yue Wu et Premkumar Natarajan. « ManTra-Net : Manipulation Tracing Network For Detection And Localization of Image ForgeriesWith Anomalous Features ». In : 2019.

[172] Marcello Zanardelli et al. « Image forgery detection : a survey of recent deep-learning approaches ». In : *Multimedia Tools and Applications* 82.12 (mai 2023), p. 17521-17566. ISSN : 1573-7721. DOI : 10.1007/s11042-022-13797-w. URL : https://doi.org/10.1007/s11042-022-13797-w.

[173] Alireza Zareian, Svebor Karaman et Shih-Fu Chang. *Bridging Knowledge Graphs to Generate Scene Graphs*. 2020. arXiv : 2001.02314 [cs.CV]. URL : https://arxiv.org/abs/2001.02314.

[174] Tianfei Zhou et al. « Face Forensics in the Wild ». In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2021, p. 5778-5788.

[175] Xinyi Zhou, Jindi Wu et Reza Zafarani. *SAFE : Similarity-Aware Multi-Modal Fake News Detection*. 2020. URL : https://arxiv.org/abs/2003.04981.

[176] Zijian Zhou, Miaojing Shi et Holger Caesar. *VLPrompt : Vision-Language Prompting for Panoptic Scene Graph Generation*. 2024. arXiv : 2311.16492 [cs.CV]. URL : https://arxiv.org/abs/2311.16492.

[177] Guangming Zhu et al. *Scene Graph Generation : A Comprehensive Survey*. 2022. arXiv : 2201.00443 [cs.CV].

[178] Ye Zhu, Yunan Wang et Zitong Yu. « Multimodal Fake News Detection : MFND Dataset and Shallow-Deep Multitask Learning ». In : *arXiv preprint arXiv:2505.06796* (2025).