Factorisation de matrices symétriques non négatives appliquée à la détection de communautés dans des graphes et l'analyse forensique d'images

Gaël Marec[1], Nédra Mellouli[1,2]

[1]LIASD, Université Paris 8 Vincennes Saint-Denis,
[2] DVRC, Groupe de Vinci. Paris, France

Résumé. Avec la massification du volume de données, notamment sur les réseaux sociaux, la véracité des informations devient incertaine. Dans ce contexte, un défi majeur est la détection des falsifications d'images, où des transformations sont réalisées sur des images pour tromper les observateurs. En s'inscrivant dans le sillage du problème de détection d'anomalies, des méthodes récentes abordent la question de la détection des transformations d'images comme un problème de détection de communautés dans des graphes associés aux images. Dans ce travail nous proposons d'utiliser une méthode de clustering de communautés basée sur la factorisation de matrices symétriques non négatives. À travers l'examen de plusieurs expériences de détection de transformation sur des images falsifiées, nous évaluons la robustesse de la méthode et discutons des améliorations possibles.

1 Introduction

Les informations sur les réseaux sociaux contiennent souvent des images et, en particulier, les fausses informations sont souvent accompagnées d'images falsifiées. Nous entendons par fausses informations des contenus volontairement trompeurs. En effet, une image est souvent porteuse d'un contenu informationnel, déformable par des transformations telles que le splicing, warping ou copy-move... (exemple : introduction d'un chef d'état sur une photo d'une réunion diplomatique). De plus, certains domaines (journalisme, criminalistique etc) nécessitent la détection de falsifications d'images. C'est pourquoi, le défi de la détection et de la localisation des transformations d'images a gagné en popularité ces dernières années. La nature des transformations d'images pouvant être diverse, les recherches antérieures se sont concentrées sur la détection de transformations spécifiques telles que "copy-move" Wandji et al. (2013), compression JPEG, splicing (incorporation d'un bout d'image sur une autre), en usant de méthodes traditionnelles dans le domaine de la vision par ordinateur Birajdar et Mankar (2013). Cependant, certaines transformations peuvent être difficiles à détecter et à définir mathématiquement. Par conséquent, les approches d'apprentissage profond ont été pertinentes pour relever ce défi au cours de la dernière décennie. La détection de transformations locales sur une image est un exemple d'une problématique courante de détection d'anomalies Chandola et al. (2009) en machine learning. La détection d'anomalie consiste à trouver un moyen

automatique d'identifier des données anormales parmi un ensemble de données. Ici, il s'agit de détecter un ensemble de pixels anormaux sur l'image, correspondant aux zones transformées. Actuellement plusieurs approches ont fait leurs preuves, notamment l'usage d'une mesure de similarité qui quantifie la proximité des données entre elles. Une donnée éloignée des autres au regard de cette mesure étant alors considérée comme anormale. Une étude récente Mayer et Stamm (2019) réinterprète la détection et la localisation de la falsification 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 "Graphes de Similarité Forensique" (GSF). Ces graphes contiennent des informations bas niveau sur les transformations subies par les images associées. Chaque noeud du GSF représente une petite région carrée de l'image appelée *patch*. Ces noeuds sont reliés entre eux par des arêtes pondérées par un score de similarité caractérisant la proximité des patchs au regard des transformations subies. De plus, chaque GSF est identifiable à sa matrice d'adjacence, appelée *matrice de similarité*. La détection des transformations sur une image correspond alors à une partition des noeuds du GSF représentant l'image Mayer et Stamm (2019), c'est-à-dire une détection de communautés au sein des graphes. Cette approche a montré de meilleures performances que plusieurs méthodes de l'état de l'art pour la localisation des transformations Mayer et Stamm (2019). 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. C'est une tâche complexe, particulièrement adaptée à des modèles puissants tel que les réseaux de neurones. Dans ce travail nous utilisons une mesure de similarité préentraînée Mayer et Stamm (2020), reposant sur un réseau de neurones à couches de convolutions (CNN). La méthode de clustering choisie permet d'exploiter à des fins applicatives l'information encodée par la mesure de similarité. Il est donc primordial d'utiliser une méthode de clustering adaptée au problème considéré. Le clustering spectral von Luxburg (2007), est une approximation d'une solution du problème *min-cut* (problème classique de détection de communautés dans un graphe) et offre de bons résultats Mayer et Stamm (2019) pour la détection de transformations sur une image. Dans ce travail nous utilisons une autre méthode de clustering, le clustering SNMF Kuang et al. (2012) (Symmetric non Negative Matrix Factorisation). Nous obtenons avec ce dernier de meilleurs résultats pour la détection de transformations qu'avec le clustering spectral.

2 Approche proposée pour la détection de communauté

2.1 Procédure générale

Le travail d'O.Mayer et M.Stamm Mayer et Stamm (2019) présente une méthode facile d'usage 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 des traces forensiques (des informations bas niveau) laissées sur les images par les transformations. Cette mesure est calculée par un CNN entraîné sur des millions d'images Mayer et Stamm (2020). La procédure complète appliquée à chaque image est décrite par le pseudo-code (*cf.* Algorithme 1).

Algorithme 1 Procédure pour la localisation de transformation d'image

Entrée : paramètre de clustering θ, méthode de clustering C_θ, mesure de similarité S, image X, nombre de patchs n

1: Séparer X en patchs $X_1, \ldots, X_n$
2: Calcul des $\frac{n^2-n}{2}$ similarités entre chaque couple de patchs.
3: Construction de la matrice de similarité $M(S, X)$
4: Determine clusters $Y = (Y_1, \ldots, Y_n) \leftarrow C_\theta(M)$

La décomposition en patch de l'image se fait selon un quadrillage régulier. De plus, les patchs se chevauchent de 50% dans le but d'augmenter la précision de la détection, et d'étendre celle ci au niveau du pixel.

2.2 Hypothèses sur la mesure de similarité

Dans ce travail nous utilisons une mesure de similarité entre images construite et entraînée par O.Mayer et M.C.Stamm Mayer et Stamm (2020). Cette mesure est calculée par un réseau de neurone composé d'un CNN extracteur de features adapté du réseau MISLnet Bayar et Stamm (2018) et un réseau de neurones de trois couches appelé réseau de similarité Mayer et Stamm (2020). Le CNN permet d'extraire des features bas niveau qui font abstraction du contenu des images et est entraîné spécifiquement pour la détection des éditions et manipulations d'images Bayar et Stamm (2018); Mayer et Stamm (2020). Ces features représentent les *traces forensiques* des images. Les traces forensiques correspondent aux informations contenues dans l'image et qui caractérisent les transformations que celle-ci a subi ainsi que sa source. En particulier la *mesure de similarité forensique* introduite Mayer et Stamm (2020) prend en compte le modèle de caméra, le type de transformation et les paramètres de ces transformations. De plus elle s'adapte bien à des modèles de caméras et transformations inconnus. Bien que le CNN soit entraîné sur un ensemble fermé de traces forensiques, il a été montré Mayer et al. (2018) que les représentations en features apprises à partir de traces spécifiques se généralisent bien à de nouvelles traces inconnues. Cette mesure présente donc l'avantage de ne pas être spécifique à un seul type de transformation ou de caméra. L'extracteur de features a été entraîné sur 2×10^6 patchs d'images de taille 128×128 et 256×256 pixels labellisés par 50 modèles de caméras Mayer et Stamm (2020). Quant à lui le réseau de similarité évalue la similarité de deux images au regard de la proximité des *traces forensiques* qu'elles contiennent en associant un score à des paires de vecteurs de features. Ce dernier contient deux couches entièrement connectées ainsi qu'une couche dont le rôle est d'agréger les vecteurs de features des deux images. Enfin il est complété par un unique neurone dit de *similarité* avec une fonction d'activation sigmoïde calculant un score de similarité compris entre 0 et 1 (*cf.* Figure 1).

2.3 Méthode de clustering

Afin d'exploiter au mieux l'information contenue dans la mesure de similarité, il est nécessaire d'établir une méthode performante de clustering sur les matrices de similarité. La performance se traduit par un clustering rapide à calculer car il est possible de faire face à de grandes quantités d'images à traiter, facile à mettre en place, c'est-à-dire ne pas dépendre

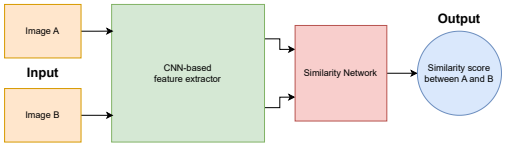


FIG. 1 – Architecture schématique du réseau de neurone calculant la mesure de similarité entre deux images Mayer et Stamm (2020). Le réseau est composé d'une première partie CNN extracteur de features, qui convertit les images en vecteurs de features de petite dimension. Puis ces vecteurs deviennent les entrées du réseau de similarité, qui calcule un score compris entre 0 et 1, indiquant si les images contiennent les mêmes traces forensiques.

fortement de nombreux hyperparamètres et enfin, offre une bonne capacité de localisation des transformations. Le clustering spectral remplit tous ces rôles, cependant il ne permet pas de capturer naturellement la structure des clusters contenus dans les données Kuang et al. (2012). En effet, pour ce faire, le clustering spectral nécessite l'usage d'une méthode de clustering auxiliaire. Bien que le choix de ce clustering auxiliaire soit arbitraire, l'algorithme *k-means* est souvent choisi pour sa simplicité von Luxburg (2007). La méthode SNMF Kuang et al. (2012) présente l'avantage de séparer les données en clusters sans imposer le recours à une méthode auxiliaire telle que k-means.

2.3.1 Clustering SNMF

Les matrices de similarité sont symétriques et nécessitent des méthodes de factorisation adaptées telle que la méthode SNMF Kuang et al. (2012). Le clustering SNMF permet de compresser la matrice de similarité dans un espace latent s'interprétant naturellement comme un espace de clusters. Etant donné une matrice de similarité $A \in \mathcal{S}_{n,n}(\mathbb{R})$ ($\mathcal{S}_{n,n}(\mathbb{R})$ étant l'ensemble des matrices symétriques réelles de taille n), sa factorisation symétrique non négative peut s'exprimer comme la solution du problème suivant :

$$\min_{H \geq 0} ||A - HH^T||_F^2 \tag{1}$$

avec F la norme de Frobenius, H une matrice non négative de dimension $n \times k$ et k le nombre de clusters souhaité. Ce problème (*cf.* equation 1) s'écrit sous une forme proche du problème du clustering spectral 2 reposant sur la détection de communautés dans des graphes von Luxburg (2007). Le clustering SNMF est donc intuitivement lié à la détection de communauté dans des graphes. Usuellement on choisit $k \ll n$ de sorte que cette factorisation corresponde à une compression des données. Le paramètre k peut être défini par l'utilisateur (en fonction du contexte d'applications) afin de prendre en compte la structure des données de la matrice de similarité A. En effet, une solution H satisfait $A \approx HH^T$ et correspond donc à une compression de n^2 données en $n \times k$ valeurs. En écrivant $A = [a_1, \ldots, a_n]$, avec $(a_i)_{1 \leq i \leq n}$ les vecteurs colonnes de A, on a en particulier la relation $a_i \approx H \cdot [h_{i,1} \ldots h_{i,k}]^T$, avec les $h_{i,j}$ les coordonnées de la matrice H. Alors le vecteur $[h_{i,1} \ldots h_{i,k}]^T$ s'interprète comme les coordonnées de la donnée a_i dans un espace latent à k-dimensions, et H comme la matrice de changement de base de l'espace des données à cet espace latent. De plus, la non négativité de la matrice H permet d'interpréter naturellement l'espace latent à k-dimensions comme un espace

de clusters : une donnée a_i appartient au cluster C_ℓ correspondant à la plus grande coordonnée $h_{i,\ell}$ du vecteur représentant a_i dans l'espace des clusters. Une solution H au problème 1 est déterminée à l'aide d'un algorithme de minimisation. Nous avons choisi d'utiliser ici la méthode de gradient projeté qui offre un bon compromis entre qualité de la minimisation et temps de calcul Kuang et al. (2012). Décrivons plus précisément la procédure d'optimisation suivie. Soit A la matrice de similarité de taille $n \times n$. Nous cherchons à résoudre le problème 1. Du fait de l'existence de la bijection $\phi : H = [h_1 \dots h_k] \in \mathcal{M}_{n,k}(\mathbb{R}) \mapsto x = [h_1^T, \dots, h_k^T] \in \mathbb{R}_{n\times k}$ on peut considérer indifféremment x et H. On optimise donc la fonction perte suivante $f : x \mapsto ||A - HH^T||_F^2$ de gradient $\nabla f : x \mapsto \phi^{-1}(4(HH^T - A) \cdot H)$ en considérant les itérations successives $(x_{k+1})_{i,j} = \max((x_k)_{i,j} - \alpha(\nabla f(x_k))_{i,j}, 0),\ \forall i, j \in \{1, \dots, n\} \times \{1, \dots, k\}$, où $\alpha > 0$ représente le pas du gradient. On arrête l'optimisation après un certain nombre maximum d'itérations fixé de manière empirique.

2.3.2 Clustering Spectral

Quant à lui le clustering spectral repose sur l'approximation du problème *mincut* qui est un problème classique de clustering de graphe Liu et al. (2022). De la même manière que dans la méthode SNMF, le but du clustering spectral est de compresser les données contenues dans la matrice de similarité A, mais pas sous les mêmes contraintes :

$$\min_{H^T H = I} ||A - HH^T||_F^2 \tag{2}$$

Ce problème présente l'avantage de posséder une solution globale et optimale déduite des vecteurs propres de A von Luxburg (2007). Cependant, en perdant la contrainte de non négativité sur H, la structure de clusters n'en est plus immédiatement déductible (la contrainte d'orthogonalité étant plus forte que la contrainte de positivité). Il est nécessaire d'utiliser une méthode de clustering auxiliaire sur les lignes de H afin d'extraire la structure de cluster. Le clustering spectral est donc moins naturel que la méthode SNMF pour le clustering de la matrice de similarité entre les données.

3 Résultats Expérimentaux

On s'intéresse à comparer l'efficacité des méthodes de clustering spectral et de clustering SNMF. Le nombre de clusters k est fixé à 2 afin de faire la distinction simple entre zones authentiques et zones transformées sur les images. Il est possible d'utiliser plus de deux classes en conservant les mêmes méthodes de clustering pour distinguer au sein d'une même image différents types de transformations. Cependant nous de disposons pas actuellement de jeux de données adaptés et de la vérité terrain pour ce faire. Ces expérimentations sont élaborées sur des jeux de données de la littérature. à chaque itération, le pas du gradient α dans le clustering SNMF est géométriquement décroissant (de raison $\beta = 0.1$) de valeur initiale $\alpha_0 = 1$. Intuitivement, à chaque itération α est choisi pour garantir que la fonction perte se rapproche d'un minimum, mais pas trop petit pour conserver une convergence efficace. Les tailles de patchs choisies pour chaque jeu de données sont résumées dans la Figure 2. Le calcul des métriques s'effectue au niveau du pixel (la conversion des classes des patchs au niveau des pixels se fait en suivant la procédure décrite par O.Mayer et M.Stamm Mayer et Stamm (2019), avec un seuil fixé à 0.5)

3.1 Jeux de données d'images

Dans cet article nous utilisons trois jeux de données, Columbia Hsu et Chang (2006), DSO-1 de Carvalho et al. (2013) et un dataset composé pour cet article que nous appellerons DALLE2. Columbia et DSO-1 sont des jeux de données déjà utilisés dans d'autres travaux Mayer et Stamm (2019, 2020) et qui présentent l'avantage d'avoir un historique de transformation connu. Ils servent donc de référence pour déterminer que la méthode fonctionne correctement. Le jeu de données DALLE2 sert à expérimenter la méthode sur des transformations plus complexes et des images dont l'historique de transformation est incertain. C'est pourquoi la soumission de DALLE2 à une évaluation par des métriques quantitatives est insuffisante à la compréhension des résultats.

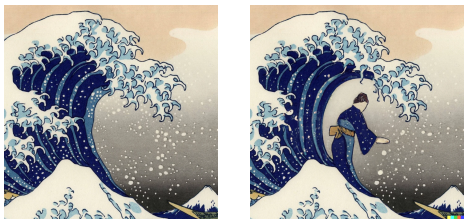
3.1.1 Columbia et DSO-1

Columbia contient 180 images transformées, ayant subi du splicing 1 fait à la main. Ces images représentent en majorité des scènes d'intérieur. Elles sont relativement petites (moins d'1 millon pixels), et les transformations sont grossières. Par conséquent, le temps de calcul pour chaque image est assez court (de l'ordre de la minute) et les résultats meilleurs que sur DSO-1 et DALLE2.

DSO-1 contient 200 images de résolution 1536×2048 dont 100 authentiques et 100 transformées. Les images transformées contiennent un individu qui a été rajouté à une scène contenant préalablement au moins une autre personne. Du fait de la cohérence visuelle et sémantique de la transformation, les images transformées de DSO-1 peuvent facilement passer pour authentiques pour un observateur humain. Leur résolution étant plus grande que pour les images de Columbia, les temps de calculs pour une seule image sont plus longs (de l'ordre de la dizaine de minutes). Bien que les transformations soient de même nature au sein de DSO-1 et Columbia, DSO-1 est un jeu de données de difficulté supérieure à Columbia pour la localisation des transformations. Les expériences sur Columbia et DSO-1 permettent de calibrer les hyper-paramètres et d'assurer que la méthode est fonctionnelle.

3.1.2 DALLE2

DALLE2, est un jeu de données construit pour ce travail. Il contient 60 images authentiques et 60 transformées de taille 1024×1024. Chaque image est générée par l'algorithme DALL·E 2 d'OpenAI Ramesh et al. de la manière suivante : une image ainsi qu'une description textuelle sont fournis à DALL·E 2 puis une portion de l'image définie par l'utilisateur est remplacée par DALL·E 2 en respectant la description. Des images construites de cette façon ont été récoltée depuis une base de données en libre accès contenant des images générées par DALL·E 2 contact@dalle2.gallery. Nous n'avons pas d'informations sur l'historique de transformation des images authentiques, elles peuvent très bien avoir subies des transformations locales préalablement. Cela peut poser problème pour l'évaluation du modèle car notre méthode peut détecter plusieurs transformations différentes et nous n'avons pas accès à la vérité terrain mais seulement à la transformation de DALL·E 2. L'architecture de DALL·E 2 repose sur un modèle de diffusion. Les modèles de diffusions sont des modèles génératifs profonds qui ont fait leurs preuves dans de nombreux domaines d'applications comme la génération d'images et de vidéos Yang et al. (2023). Les images construites par DALL·E 2 atteignent



(a) Image authentique (b) Image transformée

FIG. 2 – Exemple issu du jeu de données DALLE2, l'image (b) a été transformée à partir de l'image (a) suivant 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"*.

parfois un réalisme étonnant qui rend la détection de falsifications souvent impossible pour un humain. Le jeu de données DALLE2 semble a priori plus difficile à traiter, en comparaison de Columbia et DSO-1. En effet, d'une part la transformation subie est plus subtile qu'un simple déplacement de pixels, d'autre part on peut s'interroger sur l'efficacité de la mesure de similarité que nous employons face à des images générées par des réseaux de neurones.

3.2 Résultats

Métrique	Méthode	Columbia	DSO-1	DALLE2
F1 score	Spectral Clustering	0.93	0.91	**0.85**
	SNMF	**0.94**	**0.94**	0.74
MCC	Spectral Clustering	0.74	0.64	**-0.03**
	SNMF	**0.78**	**0.71**	0.01
IoU	Spectral Clustering	0.87	0.84	**0.73**
	SNMF	**0.89**	**0.89**	0.59

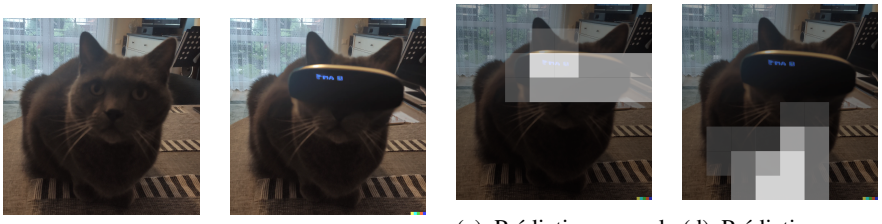
TAB. 1 – Quelques métriques d'évaluation.

Pour Columbia, nous avons testé notre méthode sur 30 images, correspondant à des images prises avec un appareil photo Canon G3 sur lesquelles ont été ajoutées des portions d'images prises avec un appareil Nikon D70. Les images de Columbia étant petites nous avons choisi d'utiliser des patchs de taille 128×128 pixels. Comme nous pouvions nous y attendre, les meilleurs résultats sont obtenus sur Columbia, au regard des trois métriques utilisées. De plus, sur Columbia, nous obtenons de meilleurs résultats sur toutes les métriques avec le clustering SNMF, en comparaison avec le clustering spectral. Ces résultats sont très légèrement meilleurs, entre 1 et 5% d'augmentation par rapport au clustering spectral. Seule une image correspond à un IoU inférieur à 0.5 8 pour le clustering spectral. Nous observons également que les clustering SNMF et Spectral coïncident pour certaines images. Le MCC est nettement supérieur à 0.5 1 pour les deux méthodes de clustering ce qui confirme que les deux méthodes sont efficientes.

Les résultats obtenus sur DSO-1 sont assez similaires à ceux de Columbia bien qu'ils soient légèrement moins bons. En raison du temps de calcul, nous n'avons testé que 10 images de DSO-1 avec des patchs de taille 256×256. Pour Columbia, presque toutes les images possé-

daient un IoU supérieur à 0.8 contre 0.5 pour DSO-1. La détection échoue totalement pour une seule image avec le clustering spectral, mais bien réussie par le clustering SNMF. Comme pour Columbia le MCC est supérieur à 0.5, les deux méthodes employées sont fiables. A nouveau, le clustering SNMF fait des meilleures performances que le clustering spectral.

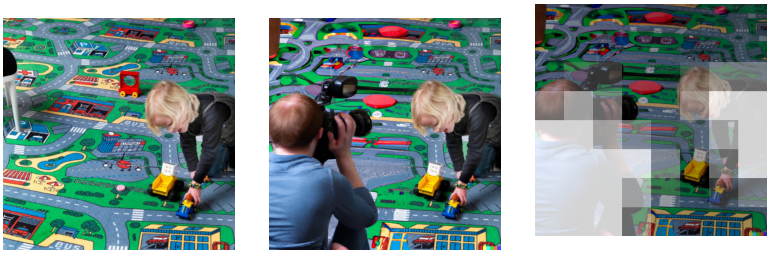
Comme attendu, les résultats sur DALLE2 sont plus mitigés. Que ce soit avec le clustering spectral, ou le clustering SNMF, le MCC est proche de 0 ce qui indique que les prédictions sont proches de l'aléatoire. Du fait de l'absence d'une vérité terrain fiable, les métriques sont difficilement interprétables en l'état. Cependant, en observant chaque prédicition une à une, nous pouvons faire les remarques suivantes : Pour le clustering spectral, *aucune* prédiction n'est correcte, le clustering spectral échoue à interpréter les matrices de similarité. Quant au clustering SNMF les prédictions semblent sensées pour 19 images parmi les 60 testées (les images en questions sont les n°3, 5, 11, 18, 20, 21, 26, 27, 28, 31, 32, 36, 37, 38, 41, 46, 48, 52, 59). Par exemple, le chat 3 est une de ces images. Toutes ces prédictions ne sont pas parfaites, mais le clustering SNMF semble effectivement réussir à détecter les transformations 4 effectuées par DALL·E 2. Parfois la méthode semble déceler des transformations supplémentaires 4 sur les images authentiques. Ces 19 images semblent ne rien avoir en commun, certaines représentent des dessins, d'autres sont photoréalistes.De plus, elles ne représentent pas non plus les mêmes thèmes (un village, un chat, une montagne...).



(a) Image authentique (b) Image transformée (c) Prédiction avec le clustering SNMF (d) Prédiction avec le clustering spectral

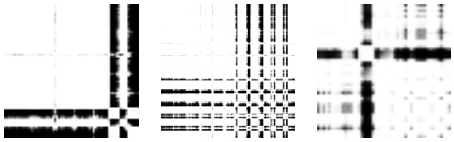
FIG. 3 – Exemple d'image issue du jeu de données construit avec DALLE2. Plus les zones grisées sont claires plus la transformation est détectée comme forte.

Les raisons de l'échec du modèle à détecter correctement les transformations de DALLE2 sont multiples. D'abord comme expliqué précédemment, le manque d'informations dont l'on dispose sur les images authentiques n'aide pas à mesurer les performances du modèle. Ensuite, il est possible d'ajuster le clustering SNMF afin qu'il soit plus rapide ou fournisse de meilleurs résultats Kuang et al. (2012). Mais le plus probable est que la mesure de similarité dont l'on dispose n'est pas adaptée à la détection de telles transformations. Cette mesure de similarité repose sur la similarité des images au regard de leur source et des transformations subies. Nous pouvons donc nous interroger sur la capacité de la mesure à détecter la dissimilarité entre des zones d'image authentiques et des zones générées par DALL·E 2. En effet, a priori la capacité de la mesure de similarité à distinguer deux modèles de caméras n'induit pas nécessairement sa capacité à distinguer une caméra d'un modèle de diffusion. C'est pourquoi, un réentraînement partiel et spécifique aux modèles de diffusion tels que DALL·E du réseau de neurone calculant la mesure de similarité pourrait améliorer les performances.



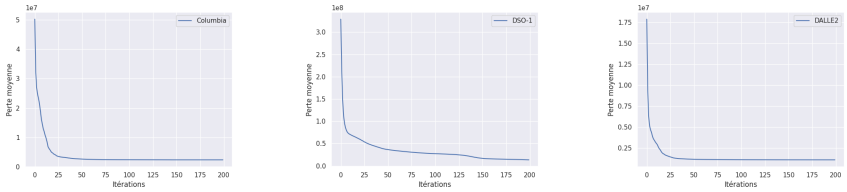
(a) Image authentique (b) Image transformée (c) Prédiction avec le clustering SNMF

FIG. 4 – Sur cette image issue de DALLE2, notre méthode détecte le caméraman ainsi que l'enfant comme zone transformée. La zone supérieure de l'image, ayant été manifestement transformée a été ignorée par la méthode. Cependant le caméraman a été correctement détecté. La détection de l'enfant est difficile à interpréter, nous n'avons pas de contrôle sur l'historique de ces images, il pourrait s'agir d'une erreur de détection ou au contraire, révéler une transformation préalable de l'image authentique.



(a) (b) (c)

FIG. 5 – Trois matrices de similarité correspondant chacune à une image de (a) Columbia, (b) DSO-1, (c) DALLE2 3. Dans les trois cas, les matrices de similarité présentent un fort contraste, cela indique que les zones similaires sont bien discriminées des zones dissimilaires. L'interprétation par clustering en est facilitée. En effet, une matrice de similarité parfaite serait binaire.



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

FIG. 6 – Représentation de la perte moyenne $||A - HH^T||_F^2$ (*cf.* section 2.3.1) à chaque itération de la descente de gradient utilisée pour le clustering SNMF, pour chaque jeu de données. En moyenne, toutes les optimisations convergent en 200 itérations.

Pour conclure, nos expériences ont montré que le clustering SNMF performe mieux que le clustering spectral (d'environ 5% 1) sur deux jeux de données. Sur DALLE2 les deux méthodes

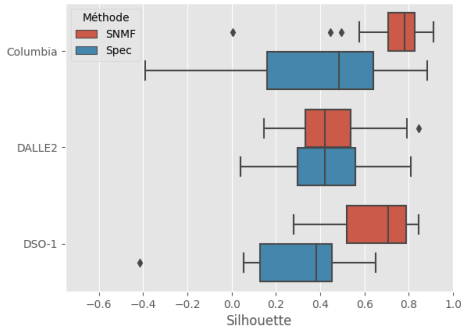


FIG. 7 – Boîte à moustaches représentant chaque indice de silhouette pour chaque jeu de données et chaque méthode. Pour DSO-1 et Columbia, la méthode SNMF surpasse le clustering spectral, pour DALLE2 les deux clusterings sont de qualité semblables au regard de l'indice de silhouette.

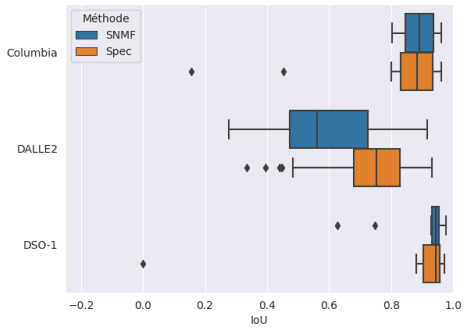


FIG. 8 – Boîtes à moustaches représentant chaque score IoU pour chaque jeu de données et chaque méthode. Les résultats sont semblables pour Columbia et DSO-1 et la méthode SNMF performe moins bien sur DALLE2.

	Columbia	DSO	DALLE2	Complexité algorithmique
Nombre de patchs par image	$n = 70$	$n = 165$	$n = 49$	
Résolution des images	568×757	1536×2048	1024×1024	
Taille des patchs	128×128	256×256	256×256	
Calcul de la matrice de similarité	$18.6s$	$227s$	$19.6s$	$O(n^2 \times$ nombre d'opérations d'un calcul de similarité$)$
Clustering spectral	$0.01s$	$0.06s$	$0.002s$	$O(n^3)$
Clustering SNMF	$1.09s$	$2.63s$	$0.523s$	$O(n^2 \times k)/$ itération
Localisation au niveau du pixel	$81.7s$	$1314s$	$259s$	$O(n \times nombre\ de\ pixels)$

TAB. 2 – Temps de calculs moyens pour une seule image, pour différentes étapes, pour différents jeux de données. Expérimentations sur une machine dotée d'un CPU Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz, et d'un GPU GeForce GTX 1080Ti.

échouent quantitativement. Cependant visuellement seul le clustering SNMF semble déceler des informations sur les transformations 3. Les limites aux résultats sont doubles, d'une part il y a un manque de contrôle sur l'historique de transformation des images de DALLE2, d'autre part, il y a un probable manque d'extension de la mesure de similarité à des transformations complexes comme celles issues de modèles génératifs récents.

3.3 Perspectives

Tout d'abord, nous avons confiance dans la procédure générale décrite par l'algorithme 1. Il s'agit d'une approche intuitive et pertinente du problème de détection des transformations sur une image. Dans cette perspective, nous proposons des améliorations potentielles, qui conserveraient la nature de la méthode. Il serait utile, dans un soucis d'investigation forensique, de mener des expériences avec plus que deux classes. En effet cela permettrait de distinguer différents types de transformations, au delà de simplement distinguer zones transformées et zones authentiques (cela serait particulièrement utile pour les images de DALLE2 4 dont l'historique de transformation est inconnu). Algorithmiquement utiliser plus de 2 classes ne change rien au clustering des patchs des images. Nous pourrions également évaluer le modèle sur plus de données, et notamment construire un jeu de données d'images tranformées par un modèle de diffusion récent tel que DALL·E 2 mais en garantissant que les images ne subissent qu'une seule transformation et provenant de ce modèle. Cela serait néanmoins coûteux en temps et en argent car à l'heure actuelle ces modèles ne sont pas en libre accès gratuit et illimité. Une des améliorations possible du modèle reste de travailler sur le clustering pour mieux interpréter les similarités. Il est parfois difficile de choisir entre différentes méthodes de clustering qui présentent chacunes des intérêts. C'est pourquoi il est également possible d'adopter une approche de type *ensemble clustering* Strehl et Ghosh (2002); Jia et al. (2023) qui permet d'agréger plusieurs méthodes de clustering dans le but d'obtenir un clustering de consensus meilleur que chacun pris individuellement.

Références

Bayar, B. et M. C. Stamm (2018). Constrained convolutional neural networks : A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security 13*(11), 2691–2706.

Birajdar, G. K. et V. H. Mankar (2013). Digital image forgery detection using passive techniques : A survey.

Chandola, V., A. Banerjee, et V. Kumar (2009). Anomaly detection : A survey. *ACM Comput. Surv. 41*(3).

contact@dalle2.gallery. Dall·e 2 largest image database. https ://dalle2.gallery.

de Carvalho, T. J., C. Riess, E. Angelopoulou, H. Pedrini, et A. de Rezende Rocha (2013). Exposing digital image forgeries by illumination color classification. *IEEE Transactions on Information Forensics and Security 8*(7), 1182–1194.

Hsu, Y.-F. et S.-F. Chang (2006). Detecting image splicing using geometry invariants and camera characteristics consistency. In *Interational Conference on Multimedia and Expo (ICME)*, Toronto, Canada.

Jia, Y., S. Tao, R. Wang, et Y. Wang (2023). Ensemble clustering via co-association matrix self-enhancement.

Kuang, D., C. Ding, et H. Park (2012). Symmetric nonnegative matrix factorization for graph clustering.

Liu, Y., J. Xia, S. Zhou, S. Wang, X. Guo, X. Yang, K. Liang, W. Tu, S. Z. Li, et X. Liu (2022). A survey of deep graph clustering : Taxonomy, challenge, and application.

Mayer, O., B. Bayar, et M. C. Stamm (2018). 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, pp. 79–84. Association for Computing Machinery.

Mayer, O. et M. C. Stamm (2019). Exposing fake images with forensic similarity graphs.

Mayer, O. et M. C. Stamm (2020). Forensic similarity for digital images. *IEEE Transactions on Information Forensics and Security 15*, 1331–1346.

Ramesh, A., P. Dhariwal, A. Nichol, C. Chu, et M. Chen. Dall·e 2.

Strehl, A. et J. Ghosh (2002). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res. 3*, 583–617.

von Luxburg, U. (2007). A tutorial on spectral clustering. *CoRR abs/0711.0189*.

Wandji, N. D., X. Sun, et M. F. Kue (2013). Detection of copy-move forgery in digital images based on DCT. *CoRR abs/1308.5661*.

Yang, L., Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, et M.-H. Yang (2023). Diffusion models : A comprehensive survey of methods and applications.

Summary

With the proliferation of data, particularly on social networks, the accuracy of information becomes uncertain. In this context, a major challenge lies in detecting image manipulations, where alterations are made to deceive observers. Aligning with the anomaly detection issue, recent methods approach the detection of image transformations as a community detection problem within graphs associated with the images. In this study, we propose using a community clustering method based on non-negative symmetric matrix factorization. By examining several experiments detecting alterations in manipulated images, we assess the method's robustness and discuss potential enhancements.