GRAFT : Adaptation non-supervisée au re-dimensionnement pour la détection de manipulation d'image

Ludovic DARMET, Kai WANG, François CAYRE

Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab
11 rue des Mathématiques, BP 46, 38402 Saint Martin d'Hères Cedex, France
Ludovic.Darmet@gipsa-lab.fr, Kai.Wang@gipsa-lab.fr
Francois.Cayre@gipsa-lab.fr

Résumé – Contrairement à la vision par ordinateur classique où ce sont les textures, frontières, formes, *etc.* qui sont discriminantes, en détection criminalistique pour les images, ce sont les bruits et résidus de l'image qui vont servir à la classification. C'est pourquoi la détection criminalistique pour les images est bien plus sensible aux bruits induits par la chaîne d'acquisition de l'image ou par le post-traitement appliqué. Nous nous intéressons ici au cas où le jeu de test a subi un re-dimensionnement préalable aux modifications, contrairement au jeu d'entraînement qui n'a subi que les modifications. Nous rapportons pour commencer des chutes de performances pour deux techniques de classification, avant d'introduire notre nouvelle méthode et dans un cadre non-supervisé : Gaussian mixture model Resizing Adaptation by Fine-Tuning (GRAFT). Finalement nous montrons comment cette stratégie permet d'augmenter les performances.

Abstract – Image forensics aims at classifying residuals or noises of images, in contrast with classic computer vision that deals with image content, object shapes, *etc.* Therefore, methods in image forensics are much more sensitive to noises generated by the acquisition process as well as any pre-processing. In this paper, we focus on one particular pre-processing operation: resizing. We first report losses in performances for two different pipelines of classification. Then we propose a new and successful method, GRAFT, to adapt directly the model, in an unsupervised setting, to the test domain. Finally we show some results on how our strategy reduces drops in performance.

1 Introduction

Avec l'accessibilité des smartphones et des caméras numériques, le partage de photo n'a jamais été aussi facile. Suivant cette même tendance, on trouve maintenant des outils d'édition de photographies, comme Photoshop, sur smartphone et en moins de 5 minutes une photo peut être capturée, éditée et partagée. C'est une formidable révolution pour la communication mais cela diminue par ailleurs la crédibilité que l'on accorde aux images. C'est pourquoi nous avons besoin d'outils d'analyse automatique permettant de vérifier l'intégrité d'une image, ce qui est le but de la criminalistique pour les images.

On peut définir un certain nombre d'opérations élémentaires en traitement d'image (voir Tableau 1, Section 3.1). Dans le cas d'une manipulation complexe comme un copier-coller, ces opérations sont effectuées afin de couvrir les traces laissées. Par exemple l'outil « Photoshop Clone Stamp Tool » utilise un flou gaussien pour avoir une transition plus douce dans les zones modifiées. Nous avons donc choisi de nous concentrer sur ces opérations de base. Par ailleurs, nous pensons qu'être seulement capable de classer une image comme manipulée ou non n'est pas suffisant. Il faut pouvoir localiser précisément la modification afin de quantifier à quel point l'intégrité de l'image a été compromise. C'est pourquoi nous faisons une classification sur des *patches* de petite taille (8×8).

Dans ce cadre, il existe dans la littérature trois types de méthodes. Un premier type s'appuie sur la modélisation statistique explicite des *patches*. Une vraisemblance trop faible par rapport à un modèle sur les *patches* originaux, ou trop élevée par rapport à un modèle sur les *patches* modifiés, seront des indications d'une possible modification. On peut citer ici la méthode de Fan *et al.* [1] qui utilise des mixtures de gaussiennes.
Il existe aussi de nombreuses méthodes se basant sur l'extraction de caractéristiques, qui sont des statistiques d'une modélisation implicite de l'image. Ces caractéristiques sont ensuite utilisées pour entraîner un classifieur. L'une des caractéristiques les plus populaires et efficaces sont les caractéristiques SPAM [2]. Plusieurs filtrages spatiaux de l'image sont effectués avant de compter les co-occurrences entre pixels voisins. Dans ce cas, le modèle implicite est un champ de Markov d'où l'on extrait des statistiques sur les probabilités de transition. Enfin il existe toute une famille de méthodes d'*apprentissage profond*. Ces méthodes utilisent des réseaux comparables à la vision par ordinateur classique, avec cependant une attention particulière sur les premières couches pour les fixer ou contraindre à extraire les résidus de l'image [3]. Les résultats de ces méthodes sont extrêmement compétitifs mais ne permettent pas de traiter des patchs de petite taille (taille minimum de 32×32 dans la littérature). C'est pourquoi nous nous concentrons sur les deux premiers types de méthodes.

Des pré-traitements différents entre les échantillons d'entraî-

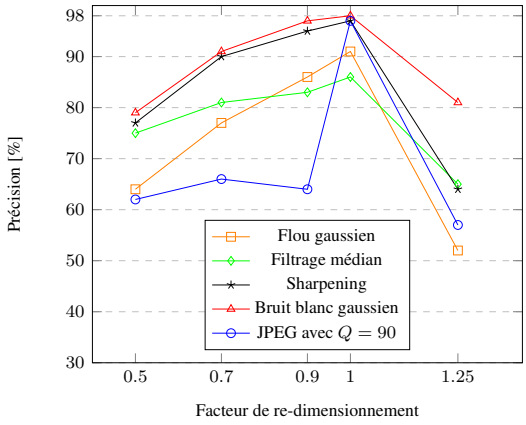

FIGURE 1 – Précision avec la méthode mixture de gaussiennes selon différents facteurs de re-dimensionnement (interpolation bicubique).

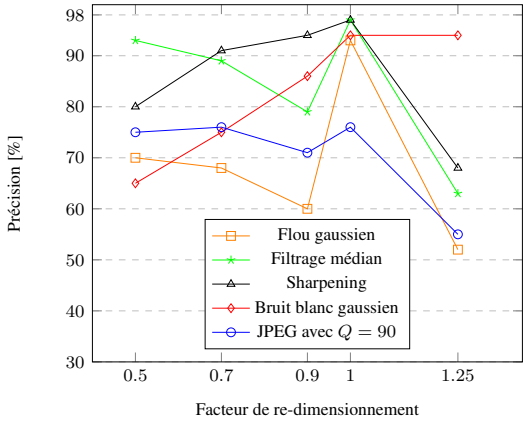

FIGURE 2 – Précision avec les caractéristiques SPAM selon différent facteur de re-dimensionnement (interpolation bicubique).

nement et de test vont induire des différences dans les statistiques locales de l'image. Ainsi, une image re-dimensionnée vers une taille inférieure aura localement des transitions plus abruptes que l'image en pleine taille, sans rien vouloir dire sur une modification ou non de l'image. Les probabilités de transitions entre voisins, même dans le domaine filtré, vont aussi être impactées. On peut donc s'attendre à des chutes de performances des deux types de détecteurs introduits précédemment. En effet nous avons relevé jusqu'à -35% de précision pour la méthode de Fan *et al.* (voir Figure 1) pour la détection de compression JPEG ($Q = 90$) dans le cas où les échantillons de test ont subi comme pré-traitement un re-dimensionnement de facteur 0.5 avec une interpolation bi-cubique. Dans les mêmes conditions, nous avons relevé -39% de précision (voir Figure 2) pour la détection d'ajout de bruit blanc gaussien à l'aide de caractéristiques SPAM. Des problèmes similaires ont également été remarqués dans le domaine voisin de la stéganalyse sous le nom de « cover-source mismatch » [4].

Une réponse très simple à ce problème pourrait être de ré-entraîner un classifieur à partir de zéro sur des données cibles ayant subi un re-dimensionnement préalable. Cependant, de telles données ne sont pas toujours disponibles en quantité suffisante (il faut environ 400000 échantillons par classe pour entraîner une mixture de gaussiennes et obtenir le score maximum atteignable avec cette méthode) ou alors on a difficilement accès aux labels pour l'entraînement. De plus, l'extraction de caractéristiques et l'entraînement sont généralement très consommateurs en temps de calcul. Idéalement, ces chutes de performances devraient pouvoir être amorties, au moins en partie. En effet, malgré ces différences de distribution statistiques, qui restent limitées, les traces laissées par les modifications sont toujours présentes. C'est le cadre de « l'adaptation de domaine ». Dans notre cas, nous cherchons à adapter un modèle appris sur un domaine sans pré-traitement vers un domaine avec. A cette fin, il est possible d'adapter soit directement les caractéristiques, soit le modèle. Nous nous intéressons ici à l'adaptation de modèle et plus particulièrement le modèle de mixture de gaussiennes.

Les contributions de ce travail sont :

— La mise en avant d'un nouveau problème : la vulnérabilité face au re-dimensionnement en pré-traitement pour deux méthodes de l'état de l'art ;

— Une nouvelle méthode efficace et rapide permettant de limiter ce problème, de manière non-supervisée, pour les méthodes s'appuyant sur des mixtures de gaussiennes.

2 Approche proposée

2.1 Processus de classification

Notre classifieur est largement inspiré de celui de Fan *et al.* [1] s'appuyant sur des mixtures de gaussiennes. L'image est découpée en *patches* de taille 8×8, chaque *patch* est centré (la composante continue est soustraite). Nous avons choisi cette taille afin de s'affranchir au maximum du contenu de l'image (nos images ont pour taille approximative 2000×3000). Les *patches* sont centrés afin de ne capturer que les variations locales et, encore une fois, de s'affranchir du contenu.

Un modèle est entraîné uniquement sur des *patches* originaux afin de modéliser au mieux cette classe, puis un deuxième modèle est entraîné uniquement sur des *patches* modifiés. La procédure « Expectation-Maximization » (EM) est utilisée pour l'entraînement, en maximisant la vraisemblance pour chaque mixture. Pour la phase de test, un ratio de vraisemblance est formé :

$$r(x_i) = \frac{\mathcal{L}_{GMM_{manip}}(x_i)}{\mathcal{L}_{GMM_{ori}}(x_i)} \quad (1)$$

Si $r(x_i) > 1$ le *patch* est classé comme manipulé.

2.2 GRAFT : GMM Resizing Adaptation by Fine-Tuning

Nous avons choisi de nous concentrer sur « l'adaptation de modèle » et non pas « l'adaptation de caractéristiques ». C'est-

à-dire que nous cherchons à adapter les paramètres explicites d'un modèle entraîné sur des échantillons n'ayant pas subi de re-dimensionnement. Ainsi, les paramètres pour une mixture de gaussiennes sont les matrices de covariances, les poids et les moyennes. Ici, les moyennes sont à 0 et ne présentent donc pas d'intérêt. Nous avons choisi de nous concentrer sur les matrices de covariance car d'après l'étude réalisée par Fan *et al.* [1] ce sont elles qui portent le plus d'information.

Nous avons $\mathcal{C}_1$, un ensemble provenant de la concaténation des matrices de covariance des deux mixtures, entraînées sur les données sources originales et données sources modifiées. Nous pouvons aussi calculer plusieurs estimations de la matrice de covariance des données cibles, ce qui forme un second ensemble : $\mathcal{C}_2$. Nous sommes alors à la recherche d'une transformation permettant de rapprocher $\mathcal{C}_1$ de $\mathcal{C}_2$. Nous avons été inspirés ici par les travaux de Rodrigues *et al.* : « Riemannian Procrustes Analysis (RPA) » [5] qui permettent de réaliser un rapprochement entre ensembles de matrices de covariances. Pour eux, les matrices de covariances sont des caractéristiques (contrairement à notre étude, où ce sont des paramètres), utilisées pour une classification semi-supervisée. Les deux ensembles sont rapprochés afin qu'un classifieur appris sur les matrices sources puisse encore être performant sur les matrices cibles. RPA se décompose en trois étapes principales : une translation vers l'identité des deux ensembles, une mise à l'échelle et un alignement des deux ensembles par une rotation.

À la fin de la procédure RPA, $\mathcal{C}_1^{RPA}$ a pour moyenne géométrique l'identité : $\mathcal{I}_n$. Il faudrait pour coller aux données alors effectuer une translation de $\mathcal{C}_1^{RPA}$ afin d'avoir le même barycentre géométrique (en prenant comme mesure distance la longueur de la la géodésique) que $\mathcal{C}_2$. Néanmoins, si cette méthode RPA permet à nos modèles de mieux coller aux données cibles, elle fait chuter le pouvoir discriminant et donc les résultats ne s'en trouvent pas améliorés. Ce compromis à atteindre entre pouvoir discriminant et rapprochement entre la source et la cible est discuté dans l'excellent travail de Ben-David *et al.* [6]. C'est pour augmenter ce pouvoir discriminant tout en rapprochant les ensembles que nous avons ajouté une interpolation entre chaque élément de $\mathcal{C}_1^{RPA}$ et $\mathcal{C}_1^{trg}$. $\mathcal{C}_1^{trg}$ est l'ensemble $\mathcal{C}_1$ translaté de manière à avoir même barycnetre géométrique que $\mathcal{C}_2$. Deux coefficients d'interpolation différents α_1 et α_2 sont choisis pour chacun des deux sous-ensembles dans $\mathcal{C}_1$. Nous signalons ici au lecteur, que expérimentalement, sans doute du à la faible courbure de l'espace à cet endroit là ou bien peut-être les petites distances parcourues, les distances euclidienne et géodésique sont quasi-égales ici. Donc parcourir la géodésique ou une interpolation sont presque équivalents. Cependant l'interpolation est beaucoup plus rapide à calculer. À la fin, nous obtenons donc, comme l'illustre l'Equation (2), des matrices de covariances adaptées pour les deux mixtures : $\mathcal{C}_{adp}^{ori}$ et $\mathcal{C}_{adp}^{mnp}$.

$$\begin{aligned} \mathcal{C}_{adp}^{ori} &= \mathcal{C}_1^{trg,ori} * (1-\alpha_1) + \mathcal{C}_1^{RPA,ori} \times \alpha_1, \\ \mathcal{C}_{adp}^{mnp} &= \mathcal{C}_1^{trg,mnp} * (1-\alpha_2) + \mathcal{C}_1^{RPA,mnp} \times \alpha_2. \end{aligned} \tag{2}$$

Il reste alors à déterminer les α. Comme indiqué dans la Figure 1, la précision chute en cas de re-dimensionnement mais ne descend pas jusqu'à 50% (tirage aléatoire). Ce qui veut dire qu'il existe des échantillons pour lesquels il est encore possible de classifier correctement. C'est avec cette intuition que nous extrayons deux sous-échantillons « presque-sûrement » originaux et modifiés. Ce sont les 30% d'échantillons (15% + 15%) pour lesquels le ratio (Équation (1)) est plus éloigné de 1. Notre algorithme n'est pas sensible à ce paramètre de 30%, il ne doit juste pas être trop grand pour garder une précision suffisante de pseudo-labels. De manière à ne pas avoir non plus des échantillons trop similaires aux échantillons source et donc de s'attacher plus fidèlement aux données cibles, nous excluons les 10% (5% + 5%) les plus sûrement classés. Nous obtenons donc à la fin deux sous-ensembles qui représentent 20% (10% + 10%) des données cibles. Cela pourrait correspondre à une phase d'Expectation dans l'algorithme EM. Nous allons maximiser la log-vraisemblance sur ces sous-ensembles afin de déterminer les α_1 et α_2 optimaux.

Algorithm 1 Algorithme GRAFT

Input : Matrices de covariances des deux mixtures : $\mathcal{C}_1$, les données cibles
Output : Matrices de covariances des mixtures adaptées

1: Calcul de « pseudo-labels » basé sur le ratio de vraisemblance avec $\mathcal{C}_1$ comme matrices de covariances ;
2: Utilisation de ces pseudo-labels pour calculer un ensemble d'estimations de la matrice de covariances empirique des données cibles : $\mathcal{C}_2$
3: Procédure RPA : $\mathcal{C}_1^{RPA}$ et $\mathcal{C}_2^{RPA}$
4: Translation de $\mathcal{C}_1$ pour avoir la même moyenne géométrique que $\mathcal{C}_2$: $\mathcal{C}_1^{trg}$
5: Initialisation de $\alpha_1 \sim \mathcal{U}[0.5, 0.9]$ et $\alpha_2 \sim \mathcal{U}[0.1, 0.9]$
6: Première interpolation sous-optimale entre $\mathcal{C}_1^{RPA}$ et $\mathcal{C}_1^{trg}$, pour trouver des pseudo-labels plus précis
7: Recherche de α_1 et α_2 qui maximisent la somme des vraisemblances sur les échantillons pseudo-labélisés : $\mathcal{C}_{adp}^{ori}$ et $\mathcal{C}_{adp}^{mnp}$
8: 5 itérations des étapes 5 à 7 pour sélectionner celle qui produit la vraisemblance maximum
9: **return** Matrices de covariances adaptées $\mathcal{C}_{adp}^{ori}$ et $\mathcal{C}_{adp}^{mnp}$

Cette étape de détermination des « pseudo-labels » est réalisée avec une version sous-optimale de l'interpolation. En effet, nous initialisons $\alpha_1 \sim \mathcal{U}[0.5, 0.9]$ et $\alpha_2 \sim \mathcal{U}[0.1, 0.9]$. L'intuition est que même en étant sous-optimale, cette première interpolation reste meilleure que sans (ce que nous avons vérifié par ailleurs). Les intervalles correspondent aux valeurs typiques prises par les paramètres, selon les différentes opérations et les différents facteurs de re-dimensionnement. Nous avons vérifié expérimentalement la précision des pseudo-labels qui se trouve au-dessus de 95%, ce qui a confirmé nos intuitions. Nous répétons cette initialisation de α_1 et α_2 5 fois, puis sélectionnons l'itération produisant la log-vraisemblance la plus élevée sur les échantillons pseudo-labellisés. Toutes ces étapes sont résumées dans l'Algorithme 1.

3 Résultats expérimentaux

3.1 Données et implémentation

Le code des expériences est disponible sur `https://forge.uvolante.org/darmet/GRAFT`. Nous avons utilisé des *patches* 8x8 en nuances de gris. Les images brutes proviennent

de la base DRESDEN [7]. Elle est composée de 1200 images au format RAW, provenant de 5 appareils photo différents et en grande résolution. Les images sont diverses quant au contenu et à l'exposition.

Nous avons mis de côté 30% des images pour la phase de test, elles subissent un re-dimensionnement avant modification. Dans les 70 autres pourcents, nous avons extrait 400000 *patches* pour chaque classe, ce qui fait 800000 *patches* pour chaque problème binaire. Nous avons utilisé l'implémentation de Scikit-learn pour les mixtures de gaussiennes, avec des matrices de covariances pleines et 75 composantes. Le nombre de composantes est un compromis entre précision et temps d'entraînement du modèle.

Les opérations considérées sont récapitulées dans le Tableau 1.

Tableau 1 – Liste des opérations appliquées aux images

ORI	Pas de modification
FG	Filtrage gaussien avec un noyau 3×3 et $\sigma = 0.5$
FM	Filtrage médian avec un noyau 3×3
USM	*Unsharp masking* fenêtre 3×3, et $amount = 0.5$
BBG	Addition de bruit blanc avec $\sigma = 2$
JPEG	Compression JPEG avec $Q = 90$

3.2 Résultats de notre méthode

Dans le Tableau 2, se trouvent les résultats de notre méthode pour un facteur de re-dimensionnement de 0.5. Nous nous comparons avec un apprentissage de zéro sur 10% (5% + 5%) des données de test, qui utiliserait donc des informations sur les labels des images de test. Comme nous pouvons le constater dans ce tableau, notre méthode est compétitive avec ce ré-entraînement et elle est aussi bien plus rapide et ne nécessite aucune information de labels. On peut remarquer que notre méthode fonctionne mieux pour les opérations où la chute de performance est grande (ex : flou gaussien ou compression JPEG) que là où la chute est plus limitée (filtrage médian).

La méthode se comporte également très bien face à l'agrandissement, nous avons pu atteindre jusqu'à +38% pour la détection de compression JPEG ($Q = 90$) pour un facteur d'agrandissement de 1.25 (voir Tableau 3). Par ailleurs, les améliorations sont plus limitées à mesure que le facteur d'interpolation s'approche de $\times 1$, là où les chutes de performances sont plus faibles. Par exemple, les augmentations vont seulement jusqu'à $+8\%$ pour un facteur de re-dimensionnement de $\times 0.7$. Dans le même temps, les chutes de performances sont au maximum de -14%. Dans un souci de place, nous n'avons pas inclus de tableaux avec ces facteurs d'interpolation proches de 1.

4 Conclusions et perspectives

Nous avons introduit ici de nouvelles préoccupations pour les méthodes classiques de détection criminalistique pour les images, et en particulier pour deux des plus puissantes fonctionnant sur des *patches*, concernant les étapes de pré-traitement que peuvent subir les images avant d'être effectivement altérées. Nous proposons une méthode rapide et flexible permettant d'amortir au moins partiellement ce problème dans le cas où le détecteur s'appuie sur des mixtures de gaussiennes. Nous avons pu améliorer les résultats jusqu'à $+38\%$ (détection de compression JPEG avec facteur de re-dimensionnement de 1.25), le tout dans un contexte non-supervisé.

Nous nous sommes concentrés ici sur une opération de prétraitement particulière, le re-dimensionnement, d'autres opérations doivent être considérées avec des solutions adaptées. Par ailleurs, cette étude pose de nouvelles questions sur le pouvoir de généralisation dans un sens large en détection criminalistique pour les images. Développer des méthodes non ou faiblement supervisées pourraient être une solution intéressante.

Tableau 2 – Précision de test (en $\%$) pour GRAFT (re-dimensionnement de $\times 0.5$). L'amélioration, comparée au cas « sans adaptation », est donnée entre parenthèses.

	FG	FM	USM	BBG	JPEG
Sans adaptation	64	75	77	79	62
Entraînement avec 10%	77 (+13)	79 (+4)	82 (+5)	84 (+5)	81 (+19)
GRAFT	79 (+15)	79 (+4)	88 (+11)	89 (+10)	73 (+11)

Tableau 3 – Précision de test (en $\%$) pour GRAFT (re-dimensionnement de $\times 1.25$).

	FG	FM	USM	BBG	JPEG
Sans adaptation	52	65	64	81	57
Entraînement avec 10%	90 (+38)	97 (+32)	97 (+33)	99 (+18)	98 (+41)
GRAFT	83 (+31)	74 (+9)	83 (+19)	96 (+15)	95 (+38)

Références

[1] W. Fan, K. Wang et F. Cayre. *General-Purpose Image Forensics using Patch Likelihood under Image Statistical Models*. Proceedings of the IEEE International Workshop on Information Forensics and Security (2015), 1–6.

[2] T. Pevný, P. Bas et J. Fridrich. *Steganalysis by Subtractive Pixel Adjacency Matrix*. IEEE Transactions on Information Forensics and Security (2010), 215–224.

[3] B. Bayar et M.C. Stamm *A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer*. Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security (2010).

[4] J. Kodovský, V. Sedighi et J. Fridrich. *Study of Cover Source Mismatch in Steganalysis and Ways to Mitigate its Impact*. Proceedings of the SPIE Media Watermarking, Security, and Forensics (2014), 1–12.

[5] P. L. C. Rodrigues, C. Jutten et M. Congedo. *Riemannian Procrustes Analysis : Transfer Learning for Brain-Computer Interfaces*. IEEE Transactions on Biomedical Engineering (2019), 1–12

[6] S. Ben-David, J. Blitzer, K. Crammer et F. Pereira. *Analysis of Representations for Domain Adaptation*. Proceedings of the International Conference on Neural Information Processing Systems (2006), 137–144

[7] T. Gloe et R. Böhme. *The Dresden Image Database for Benchmarking Digital Image Forensics*. Proceedings of the ACM Symposium on Applied Computing (2010), 1585–1591