Analyse de l'évolutivité d'un réseau d'apprentissage profond pour la stéganalyse d'images

Hugo RUIZ
Équipe ICAR, LIRMM,
Univ Montpellier, CNRS
Email : hugo.ruiz@lirmm.fr

Marc CHAUMONT
Équipe ICAR, LIRMM,
Univ Montpellier, CNRS, Univ Nîmes
Email : marc.chaumont@lirmm.fr

Mehdi YEDROUDJ
Équipe ICAR, LIRMM,
Univ Montpellier, CNRS
Email : mehdi.yedroudj@lirmm.fr

Frédéric COMBY
Équipe ICAR, LIRMM,
Univ Montpellier, CNRS
Email : frederic.comby@lirmm.fr

Gérard SUBSOL
Équipe ICAR, LIRMM,
Univ Montpellier, CNRS
Email : gerard.subsol@lirmm.fr

Résumé : *Depuis l'émergence de l'apprentissage profond et son utilisation dans le domaine de la stéganalyse, la plupart des travaux ont continué à utiliser des CNNs de taille petite à moyenne, et à les faire apprendre sur des bases de données relativement petites.*
De même, les benchmarks et les comparaisons entre les différents algorithmes de stéganalyse basés sur des CNNs, sont effectués sur des bases de données de petite à moyenne taille. Ceci ne permet pas de savoir, 1) *si le classement entre algorithmes, avec un critère comme l'« accuracy », reste le même si la base de données d'apprentissage est plus grande,* 2) *si l'efficacité des CNNs s'effondre quand la taille de la base d'apprentissage augmente d'un ordre de grandeur,* 3) *et enfin, la taille minimale requise pour obtenir un résultat meilleur que celui obtenu par une prédiction aléatoire.*
Dans cet article, après une discussion sur le comportement observé des CNNs en fonction de leur taille et de la taille de la base de données, nous confirmons que la loi de puissance de l'erreur est également valable en stéganalyse, et ce dans le cas limite d'un réseau de taille moyenne, sur une base de données diverse, de grande taille, et dont le développement est « contrôlé ».

Mots-clés : stéganalyse, passage à l'échelle, million d'images, développement « contrôlé »

1 Introduction

La stéganographie est l'art de dissimuler des informations dans un support anodin de sorte que l'existence même du message secret soit cachée à tout observateur non averti. Inversement, la stéganalyse est l'art de détecter la présence de données cachées dans de tels supports [7]. Tout au long de cet article, les images dites « cover » font références à des images originales et les images dites « stego » sont des images ayant été altérées.

Depuis 2015, grâce à l'utilisation du l'apprentissage profond, les performances de la stéganalyse se sont considérablement améliorées [4]. Néanmoins, dans de nombreux cas, ces performances dépendent de la taille de la base de données d'apprentissage. Il est en effet communément admis que, globalement, plus l'ensemble de données est grand, meilleurs sont les résultats [20].

L'objectif de cet article est de mettre en évidence l'amélioration des performances d'un algorithme de stéganalyse basé sur l'apprentissage profond lorsque la taille de l'ensemble d'apprentissage augmente.

Dans la section 2.1, nous discutons de ces questions et des lois ou modèles qui ont été proposés par la communauté scientifique. Ensuite, dans la section 2.2, nous présentons les tests réalisés pour évaluer la loi de puissance de l'erreur. Nous justifions et discutons les différents choix et paramétrages nécessaires à l'exécution des expériences. Dans la section 3, nous présentons le protocole expérimental et décrivons les expériences menées. Nous analysons ensuite l'évolution de l'*accuracy* en fonction de la taille de l'ensemble d'apprentissage. Enfin, nous concluons et donnons quelques perspectives.

2 Travail proposé

2.1 Passage à l'échelle du modèle et des données

De nombreux articles théoriques et pratiques tentent de mieux comprendre le comportement des réseaux de neurones lorsque leur dimension augmente [3, 16, 2, 11, 11] ou lorsque le nombre d'exemples augmente [8, 15, 12]. De nombreuses expériences sont réalisées afin d'observer l'évolution de l'erreur de test en fonction de la *taille du modèle*, ou en fonction de la *taille de l'ensemble d'apprentissage*. Ces recherches sont essentielles car la découverte de lois génériques pourrait confirmer que les utilisateurs de CNNs appliquent les bonnes méthodologies.

Dans les études sur *la mise à l'échelle du modèle*, les chercheurs ont observé trois régions en fonction de la taille du modèle. Il y a la région de *sous-apprentissage* du modèle, la région de *sur-apprentissage* du modèle, et enfin la région de *sur-paramétrage* du modèle. Le point de transition vers la région sur-paramétrée est appelé le *seuil d'interpolation* (Voir figure 1 dans [13]).

En général, la conclusion est que les réseaux sur-paramétrés (possédant des millions de paramètres) peuvent être en pratique utilisés pour n'importe quelle tâche. On peut citer, par exemple, le réseau EfficientNet [17]). Ce réseau a été massivement utilisé par les compétiteurs [22, 5] du concours Alaska#2 [6].

Dans les études sur la mise à l'échelle des données, les chercheurs ont observé qu'il existe trois régions en fonction

de la taille de l'ensemble de données [8]. Il s'agit de la région à *faible quantité de données*, de la région de la *loi de puissance* (*power law*) et enfin de la région d'*erreur irréductible* (*irreducible error*) (Voir figure 2 dans [13]).. Dans la région de la loi de puissance, plus les données sont nombreuses, meilleurs sont les résultats [15, 19].

Récemment, les auteurs de [12] ont proposé une loi générique qui modélise le comportement lors de la mise à l'échelle intégrant à la fois de la taille du modèle et la taille de l'ensemble de données. Le premier terme est fonction de la taille de l'ensemble de données, notée n, et le second est fonction de la taille du modèle, notée m :

$$\begin{aligned} \epsilon : \mathbb{R} \times \mathbb{R} &\rightarrow [0,1] \\ \epsilon(m,n) &\rightarrow \underbrace{a(m)n^{-\alpha(m)}}_{\text{données}} + \underbrace{b(n)m^{-\beta(n)}}_{\text{modèle}} + c_\infty \end{aligned} \tag{1}$$

avec $\alpha(m)$ et $\beta(n)$ contrôlant le taux de décroissance de l'erreur, dépendant respectivement de m et n, et c_∞ l'erreur irréductible, une constante réelle positive, indépendante de m et n.

Ensuite, les auteurs proposent une simplification de l'expression en :

$$\tilde{\epsilon}(m,n) = an^{-\alpha} + bm^{-\beta} + c_\infty \tag{2}$$

avec a, b, α, et β constantes positives réelles.

Avec un réseau efficace, ayant un nombre conséquent de paramètres, et avec suffisament de données d'apprentissage, on atteind la région de la région de la loi de puissance et ainsi l'équation 2 peut être simplifiée, comme dans [8] :

$$\epsilon(n) = a'n^{-\alpha'} + c'_\infty \tag{3}$$

Dans la suite de notre article, nous observons, dans le contexte de la stéganalyse JPEG, le comportement d'un réseau moyen lorsque la taille du jeu de données augmente.

2.2 Conception des tests de référence

Choix du réseau : Notre objectif est d'évaluer l'*accuracy* (ou de manière équivalente la probabilité d'erreur) en fonction de l'augmentation de la taille du jeu de données. Étant limités par les ressources informatiques, nous avons donc besoin d'un réseau de faible complexité et nous avons donc sélectionné le réseau LC-Net [10], ayant seulement 300 000 paramètres, et reconnu comme l'un des meilleurs CNN en stéganalyse JPEG à la date à laquelle nous avons réalisé les expériences (entre septembre 2019 et août 2020).

Choix de la charge utile : L'objectif, ici, est d'obtenir une « accuracy » comprise entre 60 et 70% [1] pour une petite base de données [2], afin d'observer la progression lorsque l'ensemble de données est échelonné. Après de nombreux ajustements expérimentaux, nous avons trouvé que 0,2 bits par coefficient AC non nul (bpnzacs) était une bonne charge utile pour une image JPEG 256×256 pixels en niveau de gris avec un facteur de qualité JPEG de 75.

Choix relatifs à la base de données : Nous avons décidé de travailler sur des images JPEG en niveaux de gris afin de mettre de côté la stéganalyse couleur, qui est encore récente et pas encore assez comprise théoriquement [1].

Nous avons également décidé de travailler uniquement sur des images de taille 256×256 avec un facteur de qualité 75. Les conclusions obtenues dans ce qui suit s'étendent vraissemblablement sur un petit intervalle autour du facteur de qualité 75 comme observé dans [23].

3 Résultats

3.1 Base de données & matériel utilisés

Les expériences ont été menées sur la base de données LSSD [14]. La base LSSD a l'avantage d'être séparée en plusieurs tailles différentes (10k, 50k, 100k, 500k, 1M et 2M) permettant l'étude du passage à l'échelle. Les images de LSSD ont été obtenues en développant les images RAW [3] des différentes bases de données : Alaska#2, BOSS, Dresden, RAISE, Stego App, et Wesaturate. Dans nos expériences, nous n'avons utilisé que les versions 10k jusqu'à 500k de la base de données « cover » en raison du temps d'apprentissage excessivement long pour les versions 1M et 2M d'images.

La base de données « cover » utilisée pour la phase de test est composée de 100 000 images et sera toujours la même, quelles que soient les expériences. Cette base de données de test est obtenue en développant des images RAW qui n'étaient pas présentes dans la base de données prévue pour l'apprentissage et conserve quasiment la même distribution des bases de données initiales. Ainsi, le scénario de stéganalyse est proche d'un scénario clairvoyant, où le jeu de test et le jeu d'apprentissage sont statistiquement très proches.

L'étude a été réalisée sur un serveur IBM grâce à un docker ayant accès à 144 processeurs Altivec POWER9 supportés (MCP) et à deux cartes graphiques GV100GL (Tesla V100 SXM2 16Go).

3.2 Entraînement, validation et test

Le processus d'insertion a été réalisé par une l'implémentation Matlab de l'algorithme J-UNIWARD [9], avec une charge utile de 0,2 bpnzacs. Il a fallu près de trois jours (2 jours et 20 heures) pour l'incorporation sur un Intel Xeon W2145 (8 cœurs, 3.7 GHz Turbo (max 4.5 GHz), 11M de cache).

Avant d'alimenter le réseau de neurones, les images JPEG doivent être décompressées afin d'obtenir des images spatiales non arrondies en « valeurs réelles ». Notons que l'espace de stockage nécessaire devient important [4]. Afin d'éviter de stocker toutes les images décompressées, il faudrait effectuer une décompression « en ligne » de manière asynchrone couplée à une construction de mini-batch « en ligne », pour alimenter le réseau neuronal « à la volée ».

L'ensemble d'apprentissage est divisé en deux ensembles : 90% pour l'ensemble d'apprentissage « réel » et 10% pour la validation. Comme dit précédemment, l'ensemble de test est toujours le même et est composé de 200k images (« cover » et « stegos »).

3.3 Hyper-paramètres

Pour entraîner notre CNN, nous avons utilisé une descente de gradient stochastique en mini-batch sans dro-

1. « Accuracy » assez éloignée de la région de prédiction aléatoire.
2. Une base de données trop petite pourrait biaiser l'analyse puisqu'il existe une région où l'erreur augmente lorsque l'ensemble de données augmente (voir [11]).
3. Données issues des capteurs de l'appareil photo.
4. Pour une image en niveaux de gris 256×256, la taille du fichier est d'environ 500 kB lorsqu'il est stocké au format MAT en *double*.

pout. Nous avons utilisé la majorité des hyper-paramètres de l'article [10]. Le taux d'apprentissage, pour tous les paramètres, a été fixé à 0,002 et est diminué aux époques 130 et 230, avec un facteur égal à 0,1. L'optimiseur est Adam, et la décroissance des poids est de 5.10^{-4}. La taille du batch est fixée à 100, ce qui correspond à 50 paires cover/stego. Afin d'améliorer la généralisation du CNN, nous avons mélangé l'ensemble de la base d'entraînement au début de chaque époque. La première couche a été initialisée avec les 30 filtres passe-haut de base de SRM, sans normalisation, et le seuil de la couche TLU est égal à 31 comme dans [18, 21]. Nous avons effectué un arrêt précoce après 250 époques comme dans [10]. Une partie du matériel est dispobible ici : http://www.lirmm.fr/ chaumont/LSSD.html.

3.4 Résultats et discussions

Les différents ensembles d'apprentissage, de 20k à 1M d'images (« cover » et « stegos »), ont été utilisés pour tester le LC-Net. Le tableau 1 donne les performances du réseau lorsqu'il a été évalué sur la base de test (200k images) suite à l'apprentissage. Notez que plusieurs tests ont été effectués pour chaque taille de l'ensemble d'apprentissage et que les « accuracys » affichées représentent une moyenne calculée sur les 5 meilleurs modèles sélectionnés grâce à l'ensemble de validation.

TABLE 1 – Accuracy moyenne évaluée sur l'ensemble de test de 200 000 images de cover/stego, en fonction de la taille de la base de données d'apprentissage.

Images	Tests	Accuracy	Std. dev.	Durée
20,000	5	62.33%	0.84%	2h 21
100,000	5	64.78%	0.54%	11h 45
200,000	5	65.99%	0.09%	23h 53
1,000,000	1	68.31%	/	10j

Il faut noter que les temps d'apprentissage deviennent importants (10 jours) dès que le nombre d'images dépasse 1 million. Il s'agit d'un problème important qui ne nous a pas permis de réaliser une évaluation sur les bases de données 2M (1M de « cover » + 1M de « stegos ») et 4 millions.

Les résultats du tableau 1, obtenus pour la charge utile 0.2 bpnzacs, confirment que plus l'ensemble d'apprentissage est grand (100k, 200k, 1M), meilleure est l'« accuracy ». Pour la base de données 20k, l'« accuracy » est de 62% et croit de presque 2% à chaque fois que la taille de l'ensemble d'apprentissage augmente. De plus, l'écart-type devient de plus en plus petit, ce qui souligne que le processus d'apprentissage est de plus en plus stable à mesure que la base de données augmente.

Ces premiers résultats signifient que la plupart des expériences de stéganalyse menées par la communauté, en utilisant un réseau d'apprentissage profond de taille moyenne (mais aussi de grande taille), ne sont pas réalisées avec suffisamment d'exemples pour atteindre la performance optimale, puisque la plupart du temps la base de données est comprise entre 10 000 (ensemble d'apprentissage BOSS) et 150 000 images (ensemble d'apprentissage Alaska#2 avec un seul algorithme d'insertion). Ainsi, dans nos expériences, l' « accuracy » est déjà améliorée de 6% lorsque la base de données passe de 20 000 à 1 million d'images et l' « accuracy » peut probablement être améliorée en augmentant la taille de l'ensemble de données puisque la région d'erreur irréductible n'est probablement pas atteinte.

Ces résultats confirment également qu'un réseau de taille moyenne tel que LC-Net ne voit pas ses performances s'effondrer lorsque la taille de la base de données augmente.

Plus intéressant encore, à partir de ces premiers résultats, nous pouvons estimer la loi de puissance suivante $\epsilon(n) = 0.492415n^{-0.086236} + 0.168059$. Ainsi, si nous choisissons n = 20M d'images, cette loi de puissance prédit une probabilité d'erreur de 28,3%. Si l'on considère une probabilité d'erreur de 28,3% pour 20M d'images, le gain obtenu par rapport à la probabilité d'erreur de 37,7% avec 20k images, correspond à une augmentation de 9% ce qui est une amélioration considérable dans le domaine de la stéganalyse.

En conclusion, la loi de puissance de l'erreur est confirmée pour la stéganalyse avec le Deep Learning, et ce même lorsque les réseaux ne sont pas très grands (300 000 paramètres), même en commençant avec une base de données de taille moyenne (ici, seulement 20 000 images), et même si la base de données est diversifiée. De plus grandes bases de données sont nécessaires pour un apprentissage optimal, et l'utilisation de plus d'un million d'images est probablement nécessaire avant d'atteindre la région d'erreur irréductible [8].

4 Conclusion et perspectives

Dans cet article, nous avons d'abord rappelé les résultats récents obtenus par la communauté travaillant sur l'apprentissage profond, et relatifs au comportement des réseaux d'apprentissage profond lorsque la taille du modèle ou de la base de données augmente. Nous avons ensuite proposé un dispositif expérimental afin d'évaluer le comportement d'un stéganalyseur CNN de taille moyenne (LC-Net) lorsque la taille de la base de données est augmentée.

Les résultats obtenus montrent qu'un réseau de taille moyenne ne s'effondre pas lorsque la taille de la base de données augmente (jusqu'à 1M), malgré une certaine diversité. De plus, ses performances sont accrues avec l'augmentation de la taille de la base de données. Enfin, nous avons observé que la loi de puissance de l'erreur est également valable pour le domaine de la stéganalyse.

Les travaux futurs devront être réalisés sur une base de données encore plus diverse (facteurs de qualité, taille de la charge utile, algorithme d'insertion, couleur, base de données moins contrôlée...), et également avec d'autres réseaux. Plus pratiquement, un effort devra être fait afin de réduire le temps d'apprentissage, et surtout la gestion de la mémoire. Enfin, il reste des questions ouvertes à résoudre telles que : trouver une valeur d'erreur irréductible plus précise, trouver la pente de la loi de puissance en fonction du point de départ du CNN (utilisation du transfert, utilisation du curriculum, utilisation de l'augmentation des données comme les pixels-off [20]).

Remerciements

Les auteurs tiennent à remercier la Direction Générale de l'Armement (DGA) pour son soutien dans le cadre du projet ANR Alaska (ANR-18-ASTR-0009). Nous remercions également IBM Montpellier et l'Institut de Dévelop-

pement et de Ressources en Calcul Scientifique Intensif (IDRISS/CNRS) pour nous avoir donné accès à des ressources de Calcul Haute Performance.

Références

[1] Hasan Abdulrahman, Marc Chaumont, Philippe Montesinos, and Baptiste Magnier. Color Images Steganalysis Using RGB Channel Geometric Transformation Measures. *Security and Communication Networks*, 9(15) :2945–2956, 2016.

[2] Madhu S. Advani, Andrew M. Saxe, and Haim Sompolinsky. High-Dimensional Dynamics of Generalization Error in Neural Networks. *Neural Networks*, 132 :428–446, 2020.

[3] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling Modern Machine-Learning Practice and the Classical Bias–variance Trade-off. *Proceedings of the National Academy of Sciences*, 116(32) :15849–15854, 2019.

[4] Marc Chaumont. Deep Learning in steganography and steganalysis. In M. Hassaballah, editor, *Digital Media Steganography : Principles, Algorithms, Advances*, chapter 14, pages 321–349. Elsevier, July 2020.

[5] Kaizaburo Chubachi. An Ensemble Model using CNNs on Different Domains for ALASKA2 Image Steganalysis. In *Proceedings of the IEEE International Workshop on Information Forensics and Security, WIFS'2020*, Virtual Conference due to Covid (Formerly New-York, NY, USA), December 2020.

[6] Rémi Cogranne, Quentin Giboulot, and Patrick Bas. Challenge Academic Research on Steganalysis with Realistic Images. In *Proceedings of the IEEE International Workshop on Information Forensics and Security, WIFS'2020*, Virtual Conference due to Covid (Formerly New-York, NY, USA), December 2020.

[7] Jessica Fridrich. *Steganography in Digital Media*. Cambridge University Press, 2009. Cambridge Books Online.

[8] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep Learning Scaling is Predictable, Empirically. In *Unpublished - ArXiv*, volume abs/1712.00409, 2017.

[9] Vojtech Holub, Jessica Fridrich, and Tomas Denemark. Universal Distortion Function for Steganography in an Arbitrary Domain. *EURASIP Journal on Information Security, JIS*, 2014(1), 2014.

[10] Junwen Huang, Jiangqun Ni, Linhong Wan, and Jingwen Yan. A Customized Convolutional Neural Network with Low Model Complexity for JPEG Steganalysis. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec'2019, pages 198–203, Paris, France, July 2019.

[11] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep Double Descent : Where Bigger Models and More Data Hurt. In *Proceedings of the Eighth International Conference on Learning Representations, ICLR'2020*, Virtual Conference due to Covid (Formerly Addis Ababa, Ethiopia), April 2020.

[12] Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A Constructive Prediction of the Generalization Error Across Scales. In *Proceedings of the Eighth International Conference on Learning Representations, ICLR'2020*, Virtual Conference due to Covid (Formerly Addis Ababa, Ethiopia), April 2020.

[13] Hugo Ruiz, Marc Chaumont, Mehdi Yedroudj, Ahmed Oulad Amara, Frédéric Comby, and Gérard Subsol. Analysis of the Scalability of a Deep-Learning Network for Steganography "Into the Wild". *Lecture Notes in Computer Science, Springer LNCS*, 12666 :439 –452, January 2021.

[14] Hugo Ruiz, Mehdi Yedroudj, Marc Chaumont, Frédéric Comby, and Gérard Subsol. LSSD : a Controlled Large JPEG Image Database for Deep-Learning-based Steganalysis "Into the Wild". *Lecture Notes in Computer Science, Springer LNCS*, 12666 :470 – 483, January 2021.

[15] Vittorio Sala. Power Law Scaling of Test Error Versus Number of Training Images for Deep Convolutional Neural Networks. In *Proceedings of the Multimodal Sensing : Technologies and Applications*, volume 11059, pages 296 – 300, Munich, 2019. International Society for Optics and Photonics, SPIE.

[16] S Spigler, M Geiger, S d'Ascoli, L Sagun, G Biroli, and M Wyart. A Jamming Transition from Under- to Over-parametrization Affects Generalization in Deep Learning. *Journal of Physics A : Mathematical and Theoretical*, 52(47) :474001, oct 2019.

[17] Mingxing Tan and Quoc Le. EfficientNet : Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning, PMLR'2019*, volume 97, pages 6105–6114, Long Beach, California, USA, June 2019.

[18] Jian Ye, Jiangqun Ni, and Y. Yi. Deep Learning Hierarchical Representations for Image Steganalysis. *IEEE Transactions on Information Forensics and Security, TIFS*, 12(11) :2545–2557, November 2017.

[19] Mehdi Yedroudj, Marc Chaumont, and Frédéric Comby. How to Augment a Small Learning Set for Improving the Performances of a CNN-Based Steganalyzer ? In *Proceedings of Media Watermarking, Security, and Forensics, MWSF'2018, Part of IS&T International Symposium on Electronic Imaging, EI'2018*, page 7, Burlingame, California, USA, 28 January - 2 February 2018.

[20] Mehdi Yedroudj, Marc Chaumont, Frederic Comby, Ahmed Oulad Amara, and Patrick Bas. Pixels-off : Data-Augmentation Complementary Solution for Deep-Learning Steganalysis. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, IHMSec '20, page 39–48, Virtual Conference due to Covid (Formerly Denver, CO, USA), June 2020.

[21] Mehdi Yedroudj, Frédéric Comby, and Marc Chaumont. Yedrouj-Net : An Efficient CNN for Spatial Steganalysis. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'2018*, pages 2092–2096, Calgary, Alberta, Canada, April 2018.

[22] Yassine Yousfi, Jan Butora, Eugene Khvedchenya, and Jessica Fridrich. ImageNet Pre-trained CNNs for JPEG Steganalysis. In *Proceedings of the IEEE International Workshop on Information Forensics and Security, WIFS'2020*, Virtual Conference due to Covid (Formerly New-York, NY, USA), December 2020.

[23] Yassine Yousfi and Jessica Fridrich. JPEG Steganalysis Detectors Scalable With Respect to Compression Quality. In *Proceedings of Media Watermarking, Security, and Forensics, MWSF'2020, Part of IS&T International Symposium on Electronic Imaging, EI'2020*, page 10, Burlingame, California, USA, January 2020.