Impact du redimensionnement sur les images adverses

Benoit BONNET[1], Teddy FURON[1], Patrick BAS[2]

[1]Univ. Rennes, Inria, CNRS, IRISA
Rennes, France

[2]Univ. Lille, CNRS, Centrale Lille,
UMR 9189, CRIStAL Lille, France

benoit.bonnet@inria.fr, teddy.furon@inria.fr, patrick.bas@centralelille.fr

Résumé – Les travaux sur les images adverses utilisent généralement des images dont la petite taille correspond à l'entrée du modèle classifieur. Or l'étape préalable de redimensionnement peut effacer le signal adverse. Cet article explore les attaques sur des grandes images. Plusieurs interpolations sont étudiées pour le redimensionnement. Soit en amont, soit en aval de l'attaque. L'impact des différentes méthodes sur le signal adverse est étudié ainsi que leur transférabilité. Pour augmenter celle-ci, cet article explore finalement l'attaque sur un ensemble de modèles. Ces travaux permettent finalemement de conclure sur les meilleures pratiques à adopter pour se défendre d'une attaque.

Abstract – Most works on adversarial attacks consider that small images whose size already fits the model. Downscaling is however a necessary first step to adapt the size of the image to the model, and it can reform the adversarial signal. This paper explores attacking large images on classifiers with different input sizes. Several interpolations are studied, either behind or ahead of the attack. The distortion of the adversarial signal and the transferability over other downscaling methods are studied. An ensemble model is also proposed to increase the transferability of the attack against a set of downscaling kernels. This yields the best practice to follow for optimal defense.

1 Introduction

Le domaine des images adverses appliqué aux réseaux de neurones est devenu un sujet populaire, ayant plus de 3 000 publications ces quatre dernières années. Dans le cas de la classification d'image, la plupart des articles emploient des jeux de données *jouet* comme MNIST (10 classes, taille 28×28) ou CIFAR (10/100 classes, 32×32). Ces images ont une taille très inférieure aux images modernes communes. Certains travaux utilisent tout de même les données plus réalistes de Imagenet. Cependant les classifieurs sont conçus pour traiter des tailles et formats standardisés. Ces images sont donc redimensionnées avant classification. Généralement 224×224 ou 256×256 . Cela reste de l'ordre de miniatures sur internet. Dans cet article, on explore deux cas d'images adverses :

A. Le classifieur traite des larges images.

B. Le classifieur traite des plus petites images et une étape de redimensionnement préalable est nécessaire.

Dans le scenario A, on étudie l'impact de la taille d'image sur l'énergie d'un signal adverse. D'un côté, plus de pixels mènent à la prédiction (généralement meilleure sur des grandes images) ; mais de l'autre, l'attaque gagne en degrés de liberté.

Dans le scenario B, le redimensionnent fait partie de la classification. L'image de petite taille est donc une représentation intermédiaire.L'attaquant n'y a pas accès. On crée une grande image adverse telle que sa version redimensionnée trompe le classifieur. Les attaques en *boîte-blanche* utilisent le calcul d'un gradient pour créer le signal adverse. Il faut donc ici calculer le gradient à travers le redimensionnement en plus du réseau.

Dans ces travaux on étudie quelle pratique le défenseur devrait observer entre les deux scenarios. On étudie également l'impact de la méthode de redimensionnement employée, et si la connaissance de cette méthode est cruciale pour l'attaquant.

2 Travaux associés

Des travaux théoriques étudient l'impact de la dimension n des entrées sur le signal adverse.

On note $\sigma_X(n)$ l'écart-type des entrées. Si l'entrée $X \in \mathbb{R}^n$ est un vecteur aléatoire centré en 0 on obtient $\sigma_X(n) = \sqrt{E[\|X\|^2]/n}$ où $\|\cdot\|$ est la distance euclidienne. De même, on mesure la *distorsion adverse* $\sigma_A(n)$ comme la valeur type $\sqrt{E[\|P\|^2]/n}$ où P est la perturbation adverse.

Les travaux [1] considèrent un exemple où les données sont uniformément distribuées sur une sphère de rayon R. L'écart-type d'entrée $\sigma_X(n)$ est donc proportionnel à $R/\sqrt{n}$. L'article montre que la norme ℓ_2 de la perturbation adverse évolue en $O(1/\sqrt{n})$. La *distorsion adverse* $\sigma_A(n)$ évolue en $O(1/n)$.

Les articles [2, 3] généralisent ce résultat sur plusieurs distributions de données vérifiant l'inégalité de concentration de Talagrand W_2. Ils avancent que la norme ℓ_2 de la perturbation adverse évolue en fonction du "*bruit intra-classe*" σ_X. Si $X \sim \mathcal{N}(\mu_k, \sigma_X^2 I_n)$ pour la classe k, alors la norme ℓ_2 de l'attaque

est proportionnelle à la puissance de X, indépendante de n. (voir [3, Sec. 2.5.2]). On résume ces travaux par la loi suivante :

$$\sigma_A(n) \propto \sigma_X(n)/\sqrt{n}. \quad (1)$$

Comment ces résultats théoriques s'appliquent-ils à des images ? Un redimensionnement n'a pas d'impact sur l'histogramme des intensités. L'écart-type $\sigma_X(n)$ reste donc constant. En reprenant la loi (1), on trouve alors que $\sigma_A(n) \propto 1/\sqrt{n}$.

Ces travaux restent principalement théoriques. L'article [4] étudie ces questions avec les données *jouet* MNIST et CIFAR. En agrandissant les images, l'article montre que la vulnérabilité du réseau aux attaques n'évolue pas avec la dimension. Ces résultats ne sont pas convaincants puisque l'agrandissement de crée par d'information.

3 Formulation du problème

Cette section propose de rétrécir des images à l'intérieur du réseau. Il s'agit donc du scenario B discuté en Sect. 1. On montre que les attaques adversaires dans le domaine des grandes ou des petites images ne produisent pas les mêmes résultats.

3.1 Inclure le redimensionnement au réseau

Le classifieur étudié est un réseau de neurones de plusieurs couches. Chaque couche applique une transformation linéaire à son entrée (qu'il s'agisse de convolution ou de *fully-connected*). Cette transformation est suivie d'une fonction d'activation non linéaire $\phi(.)$.

$$a_k = \phi(z_k), \quad (2)$$
$$z_k = W_k a_{k-1} + b_k, \quad (3)$$

Où a_k la sortie de la k-e couche $\forall k$ t.q. $1 \leq k \leq K$. Le dernier vecteur a_K correspond aux *logits* des classes. Puisque le redimensionnement est également une fonction linéaire, il est possible d'en faire une première couche. Celle-ci n'a pas d'activation linéaire : $a_0 := x_0 = DX_0$. Où X_0 correspond à l'image d'entrée de taille originale $L \times L$, x_0 l'image de taille réduite $\ell \times \ell$, et D la matrice de redimensionnement $3\ell^2 \times 3L^3$.

Par exemple, si chaque pixel de x_o est interpolé par 4 pixels de X_o, chaque ligne de D est remplie de 0 hormis 4 coefficients $\{\delta_i\}$, $i \in \{0, 1, 2, 3\}$. Ce sont les poids (positifs) du redimensionnement dont la somme vaut 1. Dans le cas de l'interpolation par les plus proches voisins (*Nearest*), il y a seulement un 1 chaque ligne, c'est-à-dire $\delta_1 = 1$, $\delta_i = 0$ for $i \neq 1$.

3.2 Attaque de grandes et de petites images

Dans le cas d'une attaque en *boîte blanche* non ciblée, l'attaquant définit généralement la fonction de perte $\mathcal{L}(X) = a_K(c_o) - \max_{c \neq c_o} a_K(k)$, où c_o le label de vérité terrain (*ground-truth*) de X_o. L'attaque cherche à optimiser la perturbation P telle que $\mathcal{L}(X_o+P)$ est négative et $\|P\|$ est faible. Les attaques en *boîte blanche* utilisent le gradient de cette fonction de perte ou seulement son premier terme :

$$\nabla_X \mathcal{L}(X) := (d\mathcal{L}(X)/dX)^\top = D^\top g \quad \text{avec} \quad (4)$$
$$g := W_1^\top \phi_1' \dots W_{K-1}^\top \phi_{K-1}' W_{K-1}^\top \nabla \mathcal{L}(a_K)$$

Où ϕ_k' est une notation simplifiée de $\phi'(a_k)$.

L'attaque calcule $X_o - \epsilon \nabla_X \mathcal{L}(X_o) = X_o - \epsilon D^\top g$. Ou avec redimensionnement : $x_o - \epsilon D D^\top g$. Dans l'espace des petites images, on a plus simplement $x_o - \epsilon g$.

Avec l'interpolation *Nearest*, $DD^\top = I_\ell$ et les deux sont donc équivalents. Si L est un multiple de ℓ , $DD^\top = \left(\sum_i \delta_i^2\right) I_\ell$, où $\sum_i \delta_i^2 \leq 1$. Il y a une perte de l'énergie, mais la perturbation redimensionnée reste colinéaire avec g.

Si L n'est pas un multiple de ℓ ou si l'interpolation utilisée n'est pas *Nearest*, $DD^\top$ n'est plus proportionnel à la matrice identité I_ℓ. Cet effet est d'autant plus fort que plus de pixels sont considérés par l'interpolation de la petite image. Par exemple lorsque le redimensionnement utilise de l'*anti-aliasing*. Le redimensionnement a donc bien un impact sur la création de l'image adverse.

4 Expériences

4.1 Modèles, données et attaque

On réalise des expériences avec 4 familles de classifieurs : EfficientNet [5] et sa version *lite*, EfficientNet-V2 [6], et NFNet [7]. Chaque famille est composée de plusieurs modèles qui traitent des images couleur de tailles $\ell \times \ell$ différentes. Dans le cas de EfficientNet, de 224 à 600, dans celui de NFNet de 256 à 576 for NFNet (voir Fig. 1). Au sein d'une même famille, les modèle possèdent les mêmes types de couche, avec différentes tailles de noyaux et nombres de filtres. Les entraînements sont également réalisés selon les mêmes procédures. Cela permet de comparer les modèles entre eux.

On a créé pour cette étude un jeu de 1000 images. Chaque image correspond à la première occurrence de chaque classe dans le jeu de validation ImageNet 2012. Chaque image est recentrée et redimensionnée en 600 × 600 par interpolation

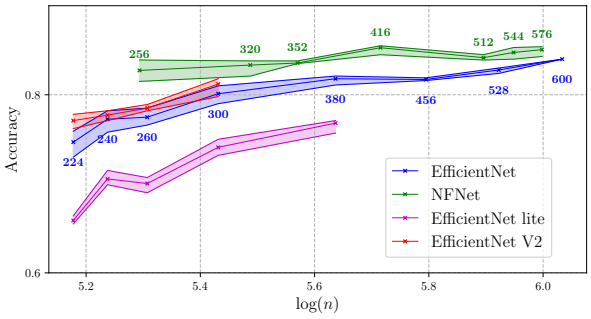

FIGURE 1 – Précision avec redimensionnement. les bandes définissent les max et min des précisions sur les 6 méthodes d'interpolations employées. Les nombres correspondent à la taille ℓ de l'image redimensionnée $n = 3\ell^2$.

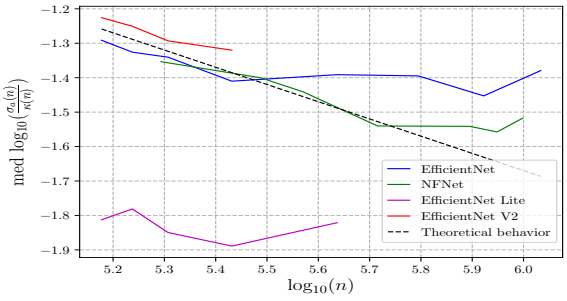

FIGURE 2 – Mesure expérimentale de la distorsion adversaire normalisée $\sigma_A(n)/\kappa(n)$ en fonction de $\log_{10}(n)$.

bilinéaire. Ce sont les dimensions d'entrée de EfficientNet-b7 (plus grand modèle étudié).

Pour le scenario B, on utilise 4 méthodes de redimensionnement : *Nearest* (plus proches voisins), *Bilinear*, *Bicubic* et *Area*. Les 3 premières interpolent les pixels de la petite image par 1, 4 ou 16 (resp.) pixels de la grande. *Area* effectue un *pooling* dont la taille dépend de ℓ. L'article [8] montre l'absence d'*anti-aliasing* dans Pytorch, hormis pour *Area*. On implémente donc deux méthodes d'*anti-aliasing* : moyennage et lissage Gaussien. Les deux utilisent un noyau de taille $k_{size} = \lceil L/\ell \rceil$. Les valeurs gaussiennes sont générées avec un écart-type de $\sigma = 1.6 \times k_{size}$ (inpiré par l'espace d'échelle de SIFT [9]). La figure 1 montre la précision des modèles selon les différentes méthodes de redimensionnement. On observe que la précision augmente avec ℓ; et que l'interpolation de redimensionnement a peu d'impact sur la classification.

On choisit l'attaque en boîte blanche BP [10] dans son implémentation *best-effort* [11]. Cette attaque a un fort taux de succès, pour peu d'itérations et avec des distorsions finales comparables aux attaques les plus optimisées. Pour le scenario B, le redimensionnement est donc implémenté comme couche du réseau pour *rétropropager* le gradient comme vu Sect. 3.

4.2 Scenario A : comparaison avec la théorie

On mesure la distorsion adversaire $\sigma_A(n)$ pour différentes images redimensionnées par interpolation bilinéaire. L'attaque se fait donc sur les images de taille $\ell \times \ell$. Selon section 2 Eq. (1), on mesure la distorsion adversaire normalisée $\sigma_A(n)/\kappa(n)$ comme fonction de $n = 3\ell^2$.

La figure 2 montre cette fonction sur une échelle logarithmique. Ces résultats semblent *partiellement* confirmer la théorie décrite. Pour 3 des familles, on observe entre $\ell \in [224, 416]$ une tendance dont la pente est de $\approx -1/2$. Cela semble confirmer [1, 2, 3] Cependant, à plus grande taille $\ell \in [456, 600]$ la théorie n'est plus du tout observable. Ce n'est également pas le cas pour EfficientNet-Lite.

4.3 Scenario B : attaque avec redimensionnement

La figure 3 montre le taux de succès de l'attaque en fonction de la distorsion créée. Chaque redimensionnement affecte en effet le taux de succès différemment (sect. 3.2). Avec *Nearest*, chaque pixel de la petite image correspond exactement à un pixel de la grande. La perturbation créée porte donc uniquement sur ces pixels et l'intégralité du signal adverse est conservé après redimensionnement. Pour les autres méthodes, le signal adverse est dilué par la *rétro-propagation* du gradient sur les pixels voisins, ce qui affecte l'attaque. *Nearest* est donc le plus facile à attaquer. Inversement, plus le nombre de pixels considérés par l'interpolation est grand, comme pour *Area*, plus le signal adverse est fort.

L'impact du redimensionnement diminue lorsque la taille de l'image augmente. Les différences sont donc moins visibles pour $\ell = 512$ sur NFNet-F4 (Fig. 3).

On observe les mêmes comportement pour les autres familles de réseau. Par exemple pour Efficient-Net, la deuxième colonne de la Table 1 montre que la distorsion augmente avec ℓ pour *Nearest*, mais qu'à l'inverse *Area* diminue. EfficientNet-b0 est donc le plus robuste pour *Area*. EfficientNet-b7 est le plus robuste pour *Nearest*.

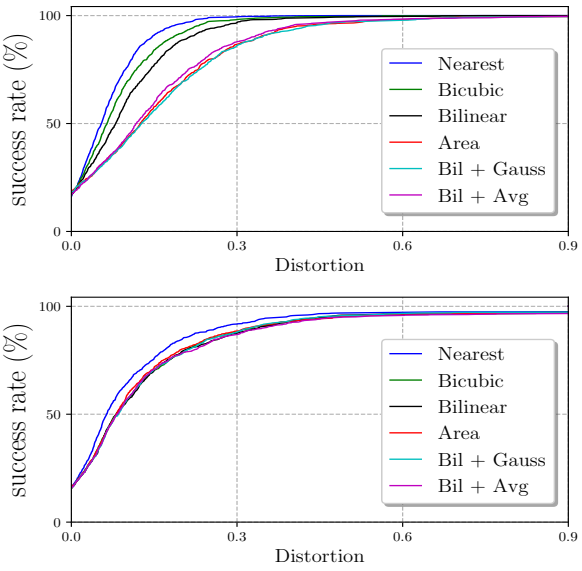

FIGURE 3 – Attaque BP sur NFNet-F0 ($\ell = 256$) et NFNet-F4 ($\ell = 512$) avec toutes les méthodes de redimensionnement .

TABLE 1 – Distorsion obtenue lorsque 90% des images sont attaquées avec succès sur la famille EfficientNet

Modèle	Redim.		Ensemble	
Size ℓ	Nearest	Area	Average	Worst
b0 : 224	0.15	0.39	**0.54**	0.53
b2 : 260	0.17	0.37	0.47	**0.49**
b4 : 380	0.21	0.33	0.40	**0.54**
b7 : 600	**0.37**	**0.37**	**0.37**	**0.37**

TABLE 2 – Précision (%) après attaque en attaquant et testant sur toutes les interpolations (EfficientNet-b0)

	Attaque			
Défense	Bil.	Bic.	Area	Near.
Bil.	0.7	70.7	74.6	75.1
Bic.	5.6	0.9	75.6	75.9
Area	72.8	72.8	0.3	72.8
Near.	75.6	75.6	35.5	0.8

4.4 Scenario B : transférabilité

On suppose ici que l'attaquant ne connaît pas la méthode de redimensionnement employée. L'attaque est basée sur une interpolation, et testée sur une autre (possiblement la même). La table 2 montre ces résultats pour un redimensionnement de $\ell = 600$ à $\ell = 224$.

La transférabilité de ces attaques est globalement faible, hormis pour l'attaque en *Bilinear* qui se transfère sur la défense *Bicubic*. (et légèrement pour l'attaque *Area* sur *Nearest*)

Ce manque de transférabilité vient du fait que BP ajoute un signal faible. L'image est donc classée juste au-delà de la *frontière* pour le classifieur. Pour une autre interpolation, cette frontière de classe est modifiée. Or l'attaque est si fine que même une légère modification parvient à la contrer.

4.5 Scenario B : ensemble de modèles

Pour obtenir une meilleure transférabilité, l'attaquant peut également choisir d'attaquer un ensemble de modèles. Pour un modèle donné (EfficientNet-b0), on compile toutes les interpolations dans un ensemble de modèles. L'attaquant doit finalement parvenir à agréger les gradients de chaque modèle pour faire fonctionner l'attaque. On envisage deux méthodes pour cela :

1. La moyenne des gradients.
2. Une sélection du *pire* gradient, inspiré par Deepfool [12].

Pour un couple classifieur/interpolation donné, la méthode *pire* gradient estime une distance la d_{adv} de l'image xo à la frontière de classe. On utilise pour ça une approximation linéaire [12] : $d_{adv} = \frac{L_{adv}(x)}{\|\nabla L_{adv}(x)\|}$. La classifieur dont la distance est la plus grande est considéré comme le *pire* et son gradient est utilisé pour l'itération en cours.

Une image est jugée adverse si et seulement si elle trompe le classifieur avec *toutes* les interpolations. Pour cette expérience, on utilise un autre sous-jeu de données de 100 images extraites aléatoirement du jeu de validation ImageNet 2012.

La table 1 montre la distorsion pour un taux de réussite de 90 %. On observe qu'il est tout à fait possible de tromper toutes les interpolations. Cela nécessite toutefois plus de distorsion. Les deux méthodes d'agrégation de gradient ont sensiblement les mêmes résultats. Enfin, on peut conclure que dans l'intérêt du défenseur, la meilleure stratégie est d'imposer un redimensionnement aléatoire sur un modèle qui traite des petites images.

5 Conclusion

Nous avons étudié l'impact du redimensionnement en attaquant des grandes images. La meilleure pratique à adopter pour un défenseur se trouve être la suivante : avoir un réseau qui traite des petites images, utilisant redimensionnent avec anti-aliasing (par exemple *Area*). La transférabilité des attaques sur différentes interpolations est presque inexistante. Cela rajoute donc une possibilité de défense par redimensionnement aléatoire. Si l'attaquant connaît toutes les méthodes employés, il reste toutefois possible d'attaquer un ensemble de modèles. Mais c'est au prix d'une plus grande distorsion ($\approx 35\%$)

Références

[1] J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. Goodfellow, "Adversarial spheres," 2018.

[2] A. Fawzi, H. Fawzi, and O. Fawzi, "Adversarial vulnerability for any classifier," in *NeuroIPS 2018*, Montreal, Canada, Dec. 2018.

[3] E. Dohmatob, "Generalized no free lunch theorem for adversarial robustness," in *Proceedings of the 36th ICML*. 09–15 Jun 2019, vol. 97 of *Proceedings of Machine Learning Research*, pp. 1646–1654, PMLR.

[4] N. Chattopadhyay, A. Chattopadhyay, S. Sen Gupta, and M. Kasper, "Curse of dimensionality in adversarial examples," in *IJCNN*, 2019.

[5] Mingxing T. and Quoc V. Le, "Efficientnet : Rethinking model scaling for convolutional neural networks," 2020.

[6] Mingxing T. and Quoc Le, "Efficientnetv2 : Smaller models and faster training," in *Proceedings of 38th ICML*. 18–24 Jul 2021, vol. 139, pp. 10096–10106, PMLR.

[7] A. Brock, S. De, S. L Smith, and K. Simonyan, "High-performance large-scale image recognition without normalization," in *ICML*. PMLR, 2021, pp. 1059–1071.

[8] G. Parmar, R. Zhang, and J-Y. Zhu, "On buggy resizing libraries and surprising subtleties in fid calculation," 2021.

[9] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[10] H. Zhang, Y. Avrithis, T. Furon, and L. Amsaleg, "Walking on the edge : Fast, low-distortion adversarial examples," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 701–713, 2021.

[11] T. Maho, B. Bonnet, T. Furon, and E. Le Merrer, "Robic : A benchmark suite for assessing classifiers robustness," 2021.

[12] S-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool : a simple and accurate method to fool deep neural networks," in *Proceedings of CVPR*, 2016, pp. 2574–2582.