Adversarial images with downscaling

Teddy Furon, Benoit Bonnet, Patrick Bas

▶ **To cite this version:**

Teddy Furon, Benoit Bonnet, Patrick Bas. Adversarial images with downscaling. ICIP 2022 - 29th IEEE International Conference on Image Processing, Oct 2022, Bordeaux, France. pp.1-5. hal-03806425

HAL Id: hal-03806425
https://inria.hal.science/hal-03806425v1

Submitted on 7 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ADVERSARIAL IMAGES WITH DOWNSCALING

*Benoit Bonnet, Teddy Furon**

Univ. Rennes, Inria, CNRS, IRISA
Rennes, France

Patrick Bas

Univ. Lille, CNRS, Centrale Lille,
UMR 9189, CRIStAL Lille, France

ABSTRACT

Most works on adversarial attacks consider small images whose size already fits the model. This paper explores attacking large images on classifiers with different input sizes. Downscaling is a necessary first step to adapt the size of the image to the model that might reform the adversarial signal. This paper studies the possibility of forging adversarial images through different interpolation methods, the distortion of their adversarial signal, and the transferability over other downscaling methods. This paper finally explores attacking an ensemble model which gathers different resizing interpolations to increase the transferability of the attack against a set downscaling kernels.

***Index Terms*—** Adversarial example, Deep Learning

1. INTRODUCTION

The topic of adversarial examples has recently attracted a huge interest with more than 3,000 papers published in the last four years. As far as image classification is concerned, most articles consider toy datasets like MNIST composed of 28×28 images (10 classes) and CIFAR 32×32 images (10 or 100 classes). These are far from nowadays typical image sizes. Few papers deal with the more realistic ImageNet dataset composed of large images (1000 classes). Yet, classifiers usually process these images once downscaled to 224×224 or 256×256. This is still about the size of thumbnails on Internet websites, so not yet a modern image. Indeed, no work uses the *original* ImageNet with larger images.

This paper considers forging realistic, *i.e*. large, adversarial images under a white-box setup for two scenarios:

A. The image classifier can natively process large images. This means that the model underneath is a wide and deep neural network.

B. The image classifier first downscales the images to a smaller size, say 224×224, and then uses a neural network to make a prediction.

Scenario A questions how the energy of the adversarial perturbation evolves with the size of the input data. On one hand, the classifier gets more pixels to make a decision which stems into a higher accuracy on natural images. On the other hand, the attacker has more degrees of freedom to delude the classifier. As far as we know, this paper is the first to show rigorous experimental evidence that a bigger size indeed benefits more to the attacker (under some conditions).

In scenario B, the downscaling is part of the classifier box. The small image is thus an internal data that the attacker cannot have access to: The goal is to forge an adversarial version of the large image and not its downscaled version. White-box attacks rely on computing the gradient of a loss function w.r.t. a neural network using back-propagation. The difficulty is therefore how to propagate further this information through the downscaling back to the space of large images. Another point of interest is the choice of downscaling method for the defender and whether its knowledge is key to forge adversarial examples for the attacker.

In the end, we also compare the distortion of perturbations that delude a classifier according to scenario A or B. This yields the best practice for the defender.

2. RELATED WORKS

Several theoretical works study how the dimension n of the inputs makes the forgery of adversarial examples easier. However, they use different arguments and terminology. It is difficult to verify that they evidence the same phenomenon.

We denote by n the dimension of the input and call by *input deviation* its natural standard deviation $\sigma_X(n)$. If the input $X \in \mathbb{R}^n$ is a centered random vector, it is typically measured by $\sqrt{E[\|X\|^2]/n}$ where $\|\cdot\|$ is the Euclidean distance. In the same way, we measure the *adversarial distortion* $\sigma_A(n)$ by the typical value of $\sqrt{E[\|P\|^2]/n}$ of the adversarial perturbation P for signals of dimension n.

The early attempt [1] considers a synthetic example where data points are uniformly distributed over spheres of constant radius R: the input deviation $\sigma_X(n)$ obviously scales as $R/\sqrt{n}$. Paper [1] then proves that the ℓ_2 norm of the adversarial perturbation in expectation scales as $O(1/\sqrt{n})$ for a fixed classification accuracy. This translates into an adversarial distortion $\sigma_A(n)$ scaling as $O(1/n)$.

Papers [2, 3] generalizes this result to many data distributions verifying the Talagrand W_2 transportation-cost inequal-

*Thanks to ANR/AID for funding chair SAIDA.

ity. They state that the ℓ_2 norm of the perturbation scales as the natural "*intra-class noise level*" σ_X (for a fixed accuracy). For instance, if $X \sim \mathcal{N}(\mu_k, \sigma_X^2 I_n)$ for class k, then the ℓ_2 norm of the attack is proportional to the power of X, which remains constant as n increases (see [3, Sec. 2.5.2]).

We propose to summarize this literature by the following rule of thumb:

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

More precisely, [3] shows that the proportional constant is upper bounded by $\kappa(n) = \sqrt{-2\log(1-\eta(n))} + \sqrt{\pi/2}$, where $\eta(n)$ is the accuracy of the classifier.

How do these theoretical results transfer to images? For square color images of size ℓ, there are $n = 3\ell^2$ pixel values. One argument is that the diameter of the hypercube $[0, 255]^n$ grows as $255\sqrt{n}$, which reflects the typical distance between inputs, hence a constant input deviation. In the same way, up or downscaling does not change the histogram of the pixel values, yielding a constant standard deviation $\sigma_X(n)$. Plugging this hand-waving justification into (1) yields an adversarial distortion scaling as $\kappa(n)/\sqrt{n}$.

On the contrary, paper [4] claims that the expectation of the ℓ_2 norm of the perturbation vanishes as $O(1/\sqrt{n})$ assuming images are $1/f^2$ power spectrum processes. This would make $\sigma_A(d) = O(1/n)$ contradicting the previous paragraph.

No work provide clear experimental evidence. Paper [5] investigates this topic on simple datasets MNIST and CIFAR. By upscaling to higher resolutions, it finds that network accuracy and attack efficiency remain the same whatever the dimension. This is not convincing because upscaling does not create any new information.

In the end, this section shows that there is no clear report about adversarial example as the dimension of the inputs grows as far as realistic image classification is concerned.

3. PROBLEM FORMULATION

This section proposes to include the downscaling inside the neural network, *i.e.* this is scenario B in Sect. 1. It outlines that running an adversarial attack in the large image space or the small image space does not produce the same effect.

3.1. Including the downscaler inside the network

The classifier is a neural network composed of several layers. Each layer applies a linear transformation (be it a convolution or a fully connected layer) followed the non-linear activation function $\sigma(.)$.

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

with a_k the output of the k-th layer $\forall k$ s.t. $1 \leq k \leq K$. The final vector a_K is the predicted logit per class (usually the last layer has no non-linearity). Since the downscaling is also a linear function, it can be incorporated inside the network as the layer 0 without activation: $a_0 := x = DX$, where X is the large $L \times L$ input image, and D the $3\ell^2 \times 3L^3$ matrix downscaling to a smaller $\ell \times \ell$ image x.

For instance, if $\ell = L/2$, one pixel of x_o is interpolated from 4 pixels of X_o, and under a proper flattening we have:

$$D = \begin{bmatrix} \delta_1 & \delta_2 & \delta_3 & \delta_4 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & \delta_1 & \delta_2 & \delta_3 & \delta_4 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{4}$$

where $\{\delta_i\}$ are the positive weights of the downscaling kernel (summing up to 1). Another case is the nearest neighbor interpolation where there is single 1 in each row of D, *e.g.* $\delta_1 = 1$, $\delta_i = 0$ for $i \neq 1$.

3.2. Attacking in the large or small image space

In an untargeted white-box setup, the attacker usually defines the loss $\mathcal{L}(X) = a_K(c_o) - \max_{c \neq c_o} a_K(k)$, where c_o is the ground-truth label of image X_o. The attack aims at finding a perturbation P s.t. $\mathcal{L}(X_o + P)$ is negative and $\|P\|$ is small. White box attacks use the gradient of this loss or its first term:

$$\begin{aligned} \nabla_X \mathcal{L}(X) &:= (d\mathcal{L}(X)/dX)^\top = D^\top g \quad \text{with} \\ g &:= W_1^\top \sigma_1' \dots W_{K-1}^\top \sigma_{K-1}' W_{K-1}^\top \nabla \mathcal{L}(a_K) \end{aligned} \tag{5}$$

With σ_k' a shortcut for $\sigma'(a_k)$. In other words, by back-propagating through the downscaler, the gradient in the large image space is nothing more than the gradient g in the small image space through the matrix $D^\top$.

The attack usually computes $X_o - \epsilon \nabla_X \mathcal{L}(X_o) = X_o - \epsilon D^\top g$. The downscaling maps this to $x_o - \epsilon D D^\top g$. Yet, the same attack in the small input space would give $x_o - \epsilon g$.

With the nearest neighbor interpolation, $DD^\top = I_\ell$ and there is no difference. If L is a multiple of ℓ as in (4), $DD^\top = \left(\sum_i \delta_i^2\right) I_\ell$, with $\sum_i \delta_i^2 \leq 1$. There is a loss of energy but the downscaled perturbation stays colinear with g.

If L is not a multiple of ℓ or if the downscaler is not the nearest neighbor interpolation, there is also a misalignment because $DD^\top$ is not proportional to identity matrix I_ℓ. This is especially true when several pixels of the small image depend on the same pixel of the large image, *e.g.* due to anti-aliasing. This shows that the downscaling has an impact on the forgery of adversarial images.

4. EXPERIMENTAL WORKS

4.1. Models, data, and attacks

We experiment with four families of classifiers: EfficientNet [6] and its Lite implementation, EfficientNet-V2 [7], and NFNet family [8]. One family is composed of several models tackling $\ell \times \ell$ color images with, for instance, ℓ ranging from 224 to 600 for EfficientNet, or from 256 to 576 for NFNet

(see Fig. 1). Family members share the same architecture but with wider activation maps and are trained with the same procedure. This is important for a fair comparison.

We created a subset of 1,000 images from the validation set of Imagenet 2012. Each image is the first occurrence of a class within the original dataset. Each image is first *center-cropped* and resized to $L \times L$ with $L = 600$ and bilinear interpolation (default on OPENCV library). This is the image size natively processed by EfficientNet-b7, the largest size used by the CNN of our experimental work.

For scenario B, we use four different downscaling methods: *Nearest*, *Bilinear*, *Bicubic* and *Area*. The first three methods interpolate one pixel from the 1, 4, or 16 (resp.) closest pixels. *Area* performs an average pooling on the image. Paper [11] highlights the lack of antialiasing in PyTorch implementation except for method *Area*. We implement two antialiasing methods: average and Gaussian. Both use a kernel size $k_{size} = \lceil \ell / L \rceil$. Gaussian values are generated with a standard deviation $\sigma = 1.6 \times k_{size}$ as inspired by SIFT scale space construction [12]. Figure 1 shows the accuracies of all models and over all downscalers (except when $\ell = L$). Our observations are twofold: the accuracy increases with the model size ℓ; and the downscaling method has little impact.

We choose the white-box attack BP [9] in its *best-effort* [10] implementation because of its high probability of success, low distortion, and speed. Distortion is measured as the Root Mean Squared Error: $d(\mathbf{x}_a, \mathbf{x}_o) = \|\mathbf{x}_a - \mathbf{x}_o\| / \sqrt{n}$ where x_o (resp. x_a) is the original (resp. adversarial) image. For scenario B, we implement downscaling as a first layer in order to *back-propagate* the gradient as detailed in Sect. 3.

4.2. Scenario A: comparison with theoretical results

We measure the adversarial distortion for different image sizes by downscaling the baseline images to the input size ℓ of the model with bilinear interpolation. The attacker modifies these $\ell \times \ell$ images in this scenario. According to section 2 Eq. (1), we measure the normalized adversarial distortion

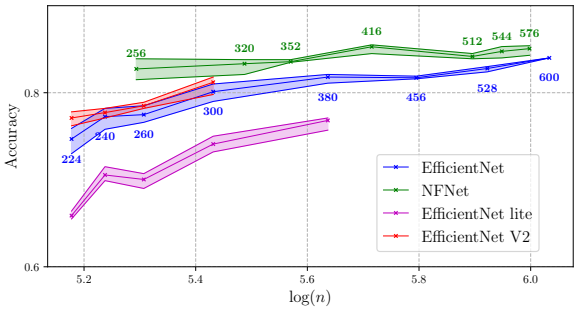

Fig. 1. Accuracies with downscaling. Stripes define max and min accuracies over the 6 downscalers for a given model. Numbers are the sizes ℓ of the downscaled images, $n = 3\ell^2$.

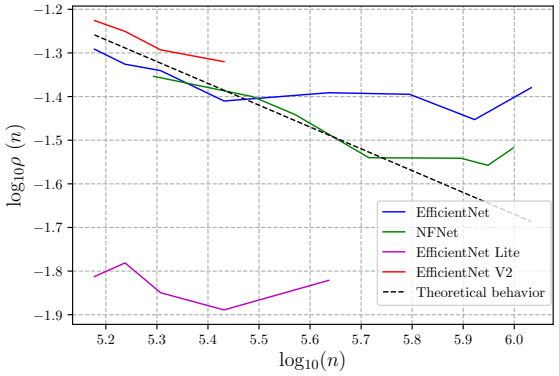

Fig. 2. Experimental measurement of the normalized adversarial distortion $\sigma_A(n)/\kappa(n)$ as a function of $\log_{10}(n)$.

$\sigma_A(n)/\kappa(n)$ as a function of $n = 3\ell^2$.

Figure 2 plots this function in logarithmic scale. These experimental results *partially* confirm the theory found in literature. For three families, in the range $\ell \in [224, 416]$, this gives almost a trend line whose slope is $\approx -1/2$. This gives more credit to papers [1, 2, 3] rather than [4] predicting -1. Yet, in the range $\ell \in [456, 600]$ the adversarial deviation is stable or even increasing, and the family EfficientNet-Lite completely violates the theory.

4.3. Scenario B: attacking through downscaling

Figure 3 shows the success rate of the attack logically increases with distortion. As Sect. 3.2 suspected, there is a noticeable hierarchy in the downscaling methods, especially on smaller sizes like $\ell = 256$ for NFNet-F0. With *Nearest*, any pixel on the smaller image matches exactly a pixel on the bigger one. The perturbation signal is crafted on these pixels and entirely goes through the downscaling. In every other method, the adversarial signal is diluted through the *back-propagation* on neighboring pixels. This creates a counter effect detrimental to the attack. *Nearest* is thus easier to attack because it requires less distortion. Conversely, methods with anti-aliasing such as *Area* are the best options as a defense.

The impact of the downscaling decreases as the network takes a bigger input size. This results in narrower differences in the plotted curves for $\ell = 512$ with NFNet-F4 (Fig. 3).

We observe the same behavior with the other network families. For instance, with Efficient-Net, Table 1 second column shows that the attack distortion goes up as ℓ increases for *Nearest*, whereas *Area* is more robust when downscaling a lot, *e.g.* down to b0 ($\ell = 224$).

4.4. Scenario B: transferability

This section assumes that the attacker does not know which downscaling method is used. The attack targets a specific interpolation and is tested through another one. Table 2 shows these results when downscaling from to $\ell = 224$.

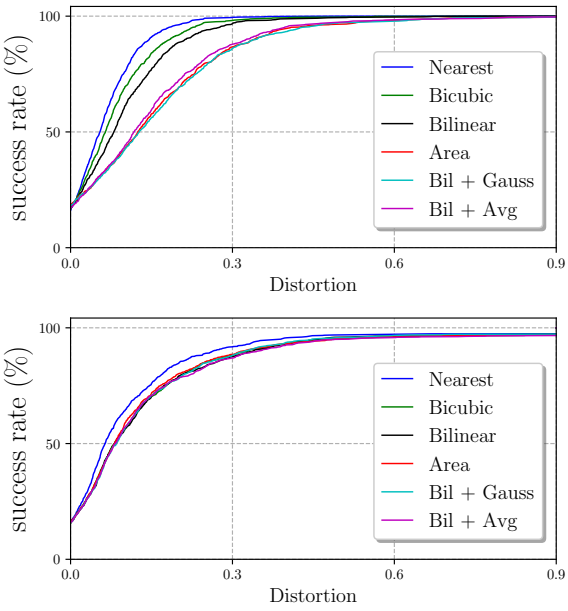

Fig. 3. Attacking NFNet-F0 ($\ell = 256$) and NFNet-F4 ($\ell = 512$) through every downscaling method with the attack BP.

Transferability of the attack is poor except for *Bilinear* (attack) displaying good transferability over *Bicubic* (defense). Our explanation is that BP adds smallest perturbations. Adversarial examples thus lie right behind the frontier of the ground-truth class for a targeted classifier. A change in the downscaling method modifies these frontiers which in return classify the sample accurately again.

4.5. Scenario B: ensemble model

In order to achieve better transferability, the attacker fights against an ensemble of classifiers. For a given model (*e.g.* EfficientNet-b0), we gather all downscaling methods in an ensemble of classifiers. This gives birth to as many gradients, which need to be aggregated into a single one to be exploited by BP attack. We explore two options:

1. A basic averaging of the gradients over all the ensemble, also referred to as *EOT* [13].

2. A worst-case gradient selection, a trick inspired by Deepfool [14].

For a given classifier / downscaler, the second method estimates the distance d_{adv} an image is from the frontier as if the classifier were linear [14]: $d_{adv} = \frac{L_{adv}(x)}{\|\nabla L_{adv}(x)\|}$. The classifier with the biggest distance is deemed as the worst for the attacker, who then targets it in the next iteration of the attack. In other words, the aggregated gradient is simply the gradient of the worst classifier of the ensemble.

For both methods, an image is deemed adversarial if it deludes all the classifiers of the ensemble. For this experiment we use another subset of 100 randomly picked images from the validation set of Imagenet 2012 as a matter of computational time. Table 1 shows the distortion at which 90% of the images are successfully misclassified. Here are the lessons: Attacking an ensemble consumes more distortion than targeting a single classifier ; so guaranteed transferability is possible at the cost of a bigger distortion. There is no clear better option for attacking an ensemble. Distortion increases as the scale goes down. A better defense strategy is to run a model on small scale images and to randomly pick a downscaler over an ensemble at each inference.

Table 1. Distortion for attacking 90% of the images for the EfficientNet family

Model	Downscaling		Ensemble	
Size ℓ	Nearest	Area	Average	Worst
b0: 224	0.15	0.39	**0.54**	0.53
b1: 240	0.16	0.37	**0.56**	0.47
b2: 260	0.17	0.37	0.47	**0.49**
b3: 300	0.17	0.34	**0.46**	0.44
b4: 380	0.21	0.33	0.40	**0.54**
b5: 456	0.26	0.33	0.38	**0.42**
b6: 528	0.25	0.31	0.36	**0.37**
b7: 600	**0.37**	**0.37**	**0.37**	**0.37**

Table 2. Accuracy (%) when downscaling from size $L = 600$ to $\ell = 224$ for EfficientNet-b0

	Attack					
Defense	Bil.	Bic.	Area	Near.	BilG	BilAg
Bil	0.7	70.7	74.6	75.1	73.5	8.2
Bic.	5.6	0.9	75.6	75.9	74.1	9.6
Area	72.8	72.8	0.3	72.8	69.8	71.9
Near.	75.6	75.6	35.5	0.8	69.4	75.6
BilG	73.4	73.5	72.6	73.4	0.50	72.2
BilAg	74.0	73.9	73.2	75.0	72.2	0.7

5. CONCLUSION

We have studied the impact of downscaling when attacking large images. The best practice is scenario B: downscaling to a small size ℓ with a wide downscaling kernel like 'Area' or 'Bil+Gauss'.

The transferability of an adversarial attack to other interpolation kernels is nearly insignificant. This opens the door to a random resizing method as a means of defense. The attacker can however circumvent this issue by attacking an ensemble, if the pool of downscaling methods is disclosed. Nevertheless, this defense increases the distortion by $\approx 35\%$ compared to the best pure strategy without sacrificing accuracy.

6. REFERENCES

[1] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow, "Adversarial spheres," 2018.

[2] Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi, "Adversarial vulnerability for any classifier," in *NeuroIPS 2018*, Montreal, Canada, Dec. 2018.

[3] Elvis Dohmatob, "Generalized no free lunch theorem for adversarial robustness," in *Proceedings of the 36th International Conference on Machine Learning*, Kamalika Chaudhuri and Ruslan Salakhutdinov, Eds. 09–15 Jun 2019, vol. 97 of *Proceedings of Machine Learning Research*, pp. 1646–1654, PMLR.

[4] Simant Dube, "High dimensional spaces, deep learning and adversarial examples," 2018.

[5] Nandish Chattopadhyay, Anupam Chattopadhyay, Sourav Sen Gupta, and Michael Kasper, "Curse of dimensionality in adversarial examples," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

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

[7] Mingxing Tan and Quoc Le, "Efficientnetv2: Smaller models and faster training," in *Proceedings of the 38th International Conference on Machine Learning*, Marina Meila and Tong Zhang, Eds. 18–24 Jul 2021, vol. 139 of *Proceedings of Machine Learning Research*, pp. 10096–10106, PMLR.

[8] Andy Brock, Soham De, Samuel L Smith, and Karen Simonyan, "High-performance large-scale image recognition without normalization," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1059–1071.

[9] Hanwei Zhang, Yannis Avrithis, Teddy Furon, and Laurent Amsaleg, "Walking on the edge: Fast, low-distortion adversarial examples," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 701–713, 2021.

[10] Thibault Maho, Benoît Bonnet, Teddy Furon, and Erwan Le Merrer, "Robic: A benchmark suite for assessing classifiers robustness," 2021.

[11] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu, "On buggy resizing libraries and surprising subtleties in fid calculation," 2021.

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

[13] Xiao Wang, Siyue Wang, Pin-Yu Chen, Yanzhi Wang, Brian Kulis, Xue Lin, and Peter Chin, "Protecting neural networks with hierarchical random switching: Towards better robustness-accuracy trade-off for stochastic defenses," *arXiv preprint arXiv:1908.07116*, 2019.

[14] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.