



Creating High-Resolution Adversarial Images Against Convolutional Neural Networks with the Noise Blowing-Up Method

Franck Leprévost[✉], Ali Osman Topal[✉], and Enea Mancellari[✉]

University of Luxembourg, House of Numbers, 6, avenue de la Fonte,
4364 Esch-sur-Alzette, Grand Duchy of Luxembourg
{Franck.Leprevost, Aliosman.Topal, Enea.Mancellari}@uni.lu

Abstract. Convolutional Neural Networks (CNNs) are widely used for image recognition tasks but are vulnerable to attacks. Most existing attacks create adversarial images of a size equal to the CNN's input size; mainly because creating adversarial images in the high-resolution domain leads to substantial speed, adversity, and visual quality challenges. In a previous work, we developed a method that lifts any existing attack working efficiently in the CNN's input size domain to the high-resolution domain. This method successfully addressed the first two challenges but only partially addressed the third one. The present article provides a crucial refinement of this strategy that, while keeping all its other features, substantially increases the visual quality of the obtained high-resolution adversarial images. The refinement amounts to a *blowing-up* to the high-resolution domain of the adversarial noise created in the low-resolution domain. Adding this blown-up noise to the clean original high-resolution image leads to an almost indistinguishable high-resolution adversarial image. The noise blowing-up strategy is successfully tested on an evolutionary-based black-box targeted attack against VGG-16 trained on ImageNet, with 10 high-resolution clean images.

Keywords: Black-box attack · Convolutional Neural Network · Evolutionary Algorithm · High resolution adversarial image · Noise Blowing-Up

1 Introduction

The profusion of images in today's society and the need to efficiently assess the information they contain for a large series of applications (self-driving cars, face recognition and security controls, satellite images, medical images, etc.) have led to the development of tools to automatically process and sort this type of data. Trained CNNs are among the most powerful and reliable tools available. Nevertheless, specifically designed adversarial images may lead CNNs to erroneous classifications, potentially resulting in catastrophic consequences.

Vice versa, efficient attacks reveal CNNs weaknesses, which in turn may lead to more robust CNNs.

Attacks depend on the scenario considered. For instance, starting with an original image classified by a CNN in a given category, the target scenario essentially consists of choosing a target category, different from the original one, and in creating a variant of the original image that the CNN will classify in the target category, although a human would classify this adversarial image still in the original category, or would be unable to notice any difference between the original and adversarial images. Attacks also depend on the level of knowledge of the CNN at the disposal of the attacker. While White-box attacks (see e.g., [2, 16]) have full knowledge of the architecture of the CNN to attack (number and type of layers, weights, etc.), Black-box attacks [9, 10] have no access to the CNN to attack and are therefore more challenging.

Our objective is to create adversarial images that closely resemble the original ones. Since original digital images in the real world are often in high-resolution, we focus on generating high-resolution adversarial images. Therefore, we aim at creating images, that can replace the original ones without losing visual quality while being able to deceive classification tools. Such achievements have significant potential in the context of privacy preservation, e.g. on social media where images are naturally of high resolution.

1.1 Standard Methodology

CNNs assess images by initially resizing them to fit their input size. In particular, high-resolution images are down-scaled, such as to 224×224 for most ImageNet-trained CNNs. So far, all attacks - black box or otherwise - have involved images of moderate size, or resized to values that CNNs handle natively, what we call here the “low-resolution” \mathcal{R} domain. The construction of adversarial images is then achieved by adding some carefully designed adversarial noise to the potentially resized original image. In particular, the adversarial noise created by all these attacks is in the “low resolution” domain handled natively by the CNNs so that the obtained adversarial images are as large as the CNN’s input size. In particular, these attacks explore a search space of size that does not depend on the size of the original image, but that coincides with the size of the CNN input.

1.2 Three Challenges

Creating adversarial images of large size (with any type of attack) leads to three challenges regarding speed, adversity, and visual quality. Firstly, the complexity of the problem increases quadratically with the size of the images, which of course impacts the speed of the attacks. For instance, we showed in [12] that an EA-based attack, that succeeded in creating adversarial images in the 224×224 domain, did not even indicate any convincing sign of potential success after 40 hours for any of the high-resolution image in Table 1. Secondly, the adversarial noise, introduced in the “high resolution” \mathcal{H} domain, should “survive” the downsizing process from \mathcal{H} to \mathcal{R} to fit the CNN. Thirdly, the noise introduced

in the “high resolution” domain should be indiscernible to the human eye when viewing the images at their original size, not only when they are scaled down to fit in the “low resolution” domain.

1.3 Our Contribution

Our previous works [11,12] provided the design of the first effective strategy that lifts to the high-resolution domain any existing attack working efficiently in the CNN’s input size domain. This was achieved by lifting an adversarial image obtained in the \mathcal{R} domain to an adversarial image in the \mathcal{H} domain. This approach successfully addressed the first two challenges of speed and adversity. However, it only partially addressed the third challenge of visual quality in the \mathcal{H} domain.

Our contributions to the present article are twofold. Firstly, we provide a substantial refinement of the strategy given in [11,12] that, while keeping all its other features – in particular it continues to lift to the high-resolution domain any attack working in the CNN’s input size domain –, substantially increases the visual quality of the high-resolution adversarial images, as well as the speed and efficiency in creating them. The refinement amounts to a “blowing-up” to the high-resolution domain of the adversarial noise – only of the adversarial noise, and not of the full adversarial image—created in the low-resolution domain. Adding this high-resolution noise to the original high-resolution image leads to a tentative high-resolution adversarial image.

Secondly, we apply this adversarial noise blowing-up strategy to one black-box attack for the target scenario against VGG-16 trained on ImageNet. We use the same 10 high-resolution clean images as in [11,12], and run the attack 10 times for each clean image. We then show that the obtained tentative high-resolution adversarial images are indeed adversarial.

To illustrate the visual quality of adversarial images obtained by this refined approach, we consider a challenging example of a high-resolution image. We compare this clean image with the HR adversarial image obtained by the method of [11,12] on the one hand, and with the HR adversarial image obtained by the new method on the other hand. We demonstrate that our new method creates high-resolution adversarial images of enhanced visual quality.

1.4 Organisation of the Paper

Section 2 briefly recalls some standard attack scenarios in \mathcal{R} , clarifies what are their lifted version to \mathcal{H} , and fixes some notations. Section 3 formalizes the noise blowing-up method and provides the scheme of our attack $atk_{\mathcal{H},\mathcal{C}}^{scenario}$ that lifts to \mathcal{H} any attack $atk_{\mathcal{R},\mathcal{C}}^{scenario}$ against a CNN \mathcal{C} that works in the \mathcal{R} domain, and that takes advantage of lifting the adversarial noise only. It sets the main indicators used to assess the quality of the obtained tentative adversarial images.

Section 4 presents a case study. The noise blowing-up strategy is applied for the target scenario to the evolutionary algorithm-based attack presented already

in [3, 5]. To illustrate the gain in visual quality provided by our new approach, one sample is detailed in Sect. 5. Section 6 summarizes our findings and indicates directions for future research.

All algorithms and experiments were implemented using Python 3.8 [18] with NumPy 1.17 [13], TensorFlow 2.4 [1], Keras 2.2 [6], and Scikit 0.24 [19] libraries. Computations were performed on nodes with Nvidia Tesla V100 GPGPUs of the IRIS HPC Cluster at the University of Luxembourg.

2 CNNs and Attack Scenarios

CNNs used for image classification undergo training on a large dataset, denoted as \mathcal{S} , to categorize images into predetermined categories c_1, \dots, c_ℓ . The categories, along with their index number ℓ , are specifically associated with dataset \mathcal{S} and remain consistent across all CNN models trained on \mathcal{S} . One denotes by \mathcal{R} the set of images of size $r_1 \times r_2$ (where r_1 is the height and r_2 is the width of the image) natively adapted to such CNNs.

Once trained, a CNN can be exposed to images (typically) in the same domain \mathcal{R} as those on which it was trained. Given an input image $\mathcal{I} \in \mathcal{R}$, the trained CNN produces a classification output vector

$$\mathbf{o}_{\mathcal{I}} = (\mathbf{o}_{\mathcal{I}}[1], \dots, \mathbf{o}_{\mathcal{I}}[\ell]), \quad (1)$$

where $0 \leq \mathbf{o}_{\mathcal{I}}[i] \leq 1$ for $1 \leq i \leq \ell$, and $\sum_{i=1}^{\ell} \mathbf{o}_{\mathcal{I}}[i] = 1$. Each c_i -label value $\mathbf{o}_{\mathcal{I}}[i]$ measures the plausibility that the image \mathcal{I} belongs to the category c_i .

Consequently, the CNN classifies the image \mathcal{I} as belonging to the category c_k if $k = \arg \max_{1 \leq i \leq \ell} (\mathbf{o}_{\mathcal{I}}[i])$. If there is no ambiguity on the dominating category, one denotes $(c_k, \mathbf{o}_{\mathcal{I}}[k])$ the pair specifying the dominating category and the corresponding label value. In this case, we consider that \mathcal{C} 's classification of \mathcal{I} is

$$\mathcal{C}(\mathcal{I}) \in \mathcal{V} = \{(c_i, v_i), \text{ where } v_i \in]0, 1] \text{ for } 1 \leq i \leq \ell\}. \quad (2)$$

The higher the c_k -label value $\mathbf{o}_{\mathcal{I}}[k]$, the higher the confidence that \mathcal{I} represents an object of the category c_k .

Remark. The dominant category is without ambiguity for most images used in practice. Still, the situation differs when there are different categories for which their corresponding label values while being larger than the remaining ones, are almost equal between themselves. This occurs *a fortiori* when all ℓ label values are almost equi-distributed, like for instance for adversarial images created in the context of the *flat scenario* (see Subsect. 2.2). If \mathcal{I} is such an image, then one considers instead that:

$$\mathcal{C}(\mathcal{I}) \in \mathcal{V} = \{((c_1, v_1), \dots, (c_\ell, v_\ell)), \text{ where } v_i \in]0, 1] \text{ for } 1 \leq i \leq \ell\}. \quad (3)$$

2.1 Assessment of the Human Perception of Distinct Images

Given two images \mathcal{A} and \mathcal{D} of the same size (belonging or not to the \mathcal{R} domain), there are different methods to numerically assess the human perception of the difference between them. In the present study, this assessment is performed mainly

by computing the values of $L_p(\mathcal{A}, \mathcal{D})$ for $p = 1, 2$, or ∞ . In a nutshell, the L_p -distance measures the difference between the pixel values of \mathcal{A} and \mathcal{D} as follows, where $\mathcal{I}(r)$ represents the value of the r^{th} -pixel of the image \mathcal{I} :

$$\begin{cases} L_p(\mathcal{A}, \mathcal{D}) = (\sum_r |\mathcal{A}(r) - \mathcal{D}(r)|^p)^{1/p} & \text{for } p = 1, 2. \\ L_\infty(\mathcal{A}, \mathcal{D}) = \text{Max}_r |\mathcal{A}(r) - \mathcal{D}(r)|. \end{cases} \quad (4)$$

2.2 Attack Scenarios in the \mathcal{R} Domain

Let \mathcal{C} be a trained CNN, c_a be a category among the ℓ possible categories, and \mathcal{A} be a clean image in the \mathcal{R} domain classified by \mathcal{C} as belonging to c_a . Let τ_a be its c_a -label value. Based on these initial conditions, we describe three attack scenarios aiming at creating an adversarial image $\mathcal{D} \in \mathcal{R}$ accordingly.

Whatever the scenario, one requires that \mathcal{D} remains so close to \mathcal{A} , that a human would not notice any difference between \mathcal{A} and \mathcal{D} . This is performed in practice by fixing the value of the parameter ϵ , that controls (or restricts) the global maximum amplitude allowed for the value modifications of each individual pixel of \mathcal{A} to obtain the adversarial image \mathcal{D} . For a given attack scenario, note that the value set to ϵ usually depends on the concrete performed attack. It depends more specifically on the L_p distance used in the attack to assess the human perception between the original image and the adversarial image.

The (c_a, c_t) *target scenario* performed on \mathcal{A} requires first to select a category $c_t \neq c_a$. The attack then aims at constructing an image \mathcal{D} that is either a *good enough adversarial image* or a τ -*strong adversarial image*.

A *good enough adversarial image* is an image that, when subjected to classification by \mathcal{C} , is classified as belonging to the target category c_t , without any strict requirement on the specific label value of c_t , as long as it is dominant compared to all other label values. An adversarial image is considered a τ -strong adversarial image if it is classified by classifier \mathcal{C} as belonging to the target category c_t and its label value for the c_t label, denoted as τ_t , is equal to or greater than a predetermined threshold value τ . Here, τ is a fixed value between 0 and 1 (exclusive) that is determined beforehand.

In the *untarget scenario* performed on \mathcal{A} , the attack aims at constructing an image \mathcal{D} that \mathcal{C} classifies in any category $c \neq c_a$.

In the *flat scenario* performed on \mathcal{A} , the attacks aim at constructing an image \mathcal{D} that \mathcal{C} is unable to classify in any category with sufficient confidence. In other words, for \mathcal{D} , all categories are likely possible. Put otherwise, one has $\mathbf{o}_{\mathcal{D}}[i] \simeq \frac{1}{\ell}$ for all $1 \leq i \leq \ell$.

One writes $atk_{\mathcal{R}, \mathcal{C}}^{\text{scenario}}$ the specific attack performed to deceive \mathcal{C} in the \mathcal{R} domain according to the selected scenario, and $atk_{\mathcal{R}, \mathcal{C}}^{\text{scenario}}(\mathcal{A})$ the adversarial image obtained by running successfully this attack on the clean image \mathcal{A} .

2.3 Attack Scenarios Expressed in the \mathcal{H} Domain

In the context of high resolution (HR) images, let us denote by \mathcal{H} the set of images that are larger than those of \mathcal{R} . In other words, an image of size $h \times w$

belongs to \mathcal{H} if $h \geq r_1$ and $w \geq r_2$. One assumes given a fixed *degradation function*

$$\rho: \mathcal{H} \longrightarrow \mathcal{R}, \tag{5}$$

that transforms any image $\mathcal{I} \in \mathcal{H}$ into a “degraded” image $\rho(\mathcal{I}) \in \mathcal{R}$. Then there is a well-defined composition of maps $\mathcal{C} \circ \rho$.

Given $\mathcal{A}_a^{\text{hr}} \in \mathcal{H}$, one obtains that way the classification of the reduced image $\mathcal{A}_a = \rho(\mathcal{A}_a^{\text{hr}}) \in \mathcal{R}$ as $\mathcal{C}(\mathcal{A}_a) \in \mathcal{V}$. Although not mandatory, we shall assume, for the sake of simplicity, that the dominating category of the reduced image \mathcal{A}_a is without ambiguity. Therefore, let $\mathcal{C}(\mathcal{A}_a) = (c_a, \tau_a) \in \mathcal{V}$ be the outcome of \mathcal{C} 's classification of \mathcal{A}_a .

An adversarial HR image against \mathcal{C} for the (c_a, c_t) *target scenario* performed by an attack $atk_{\mathcal{H},\mathcal{C}}^{\text{target}}$ on $\mathcal{A}_a^{\text{hr}} \in \mathcal{H}$ is an image $\mathcal{D}_t^{\text{hr},\mathcal{C}}(\mathcal{A}_a^{\text{hr}}) = atk_{\mathcal{H},\mathcal{C}}^{\text{target}}(\mathcal{A}_a^{\text{hr}}) \in \mathcal{H}$, that satisfies two conditions.

On the one hand, a human should not be able to notice any visual difference between the original $\mathcal{A}_a^{\text{hr}}$ and the adversarial $\mathcal{D}_t^{\text{hr},\mathcal{C}}(\mathcal{A}_a^{\text{hr}})$ HR images. On the other hand, \mathcal{C} should classify the degraded image $\mathcal{D}_t^{\mathcal{C}}(\mathcal{A}_a^{\text{hr}}) = \rho(\mathcal{D}_t^{\text{hr},\mathcal{C}}(\mathcal{A}_a^{\text{hr}}))$ in the category c_t for a sufficiently convincing c_t -label value. The (c_a, c_t) *target scenario* performed on the HR image $\mathcal{A}_a^{\text{hr}}$ can be visualized by the following scheme.

$$\begin{array}{ccc}
 \mathcal{A}_a^{\text{hr}} \in \mathcal{H} & \xrightarrow{atk_{\mathcal{H},\mathcal{C}}^{\text{target}}} & \mathcal{D}_t^{\text{hr},\mathcal{C}}(\mathcal{A}_a^{\text{hr}}) \in \mathcal{H} \\
 \rho \downarrow & & \downarrow \rho \\
 \mathcal{A}_a \in \mathcal{R} & & \mathcal{D}_t^{\mathcal{C}}(\mathcal{A}_a^{\text{hr}}) \in \mathcal{R} \\
 \mathcal{C} \downarrow & & \downarrow \mathcal{C} \\
 (c_a, \tau_a) \in \mathcal{V} & & (c_t, \tau_t) \in \mathcal{V}
 \end{array}$$

The image $\mathcal{D}_t^{\text{hr},\mathcal{C}}(\mathcal{A}_a^{\text{hr}}) \in \mathcal{H}$ is then a *good enough adversarial image* or a τ -*strong adversarial image* if its reduced version $\mathcal{D}_t^{\mathcal{C}}(\mathcal{A}_a^{\text{hr}}) = \rho(\mathcal{D}_t^{\text{hr},\mathcal{C}}(\mathcal{A}_a^{\text{hr}}))$ is.

Thanks to the degradation function ρ , one can express in a similar way in the \mathcal{H} domain any attack scenario that makes sense in the \mathcal{R} domain. This holds in particular for the *untarget scenario* and for the *flat scenario*. One denotes by $\mathcal{D}_{\text{untarget}}^{\text{hr},\mathcal{C}}(\mathcal{A}_a^{\text{hr}}) = atk_{\mathcal{H},\mathcal{C}}^{\text{untarget}}(\mathcal{A}_a^{\text{hr}})$ the HR adversarial images obtained by an attack $atk_{\mathcal{H},\mathcal{C}}^{\text{untarget}}$ for the untarget scenario performed on $\mathcal{A}_a^{\text{hr}} \in \mathcal{H}$, and by $\mathcal{D}_{\text{untarget}}^{\mathcal{C}}(\mathcal{A}_a^{\text{hr}}) \in \mathcal{R}$ its degraded version. *Mutatis mutandis*, one denotes by $\mathcal{D}_{\text{flat}}^{\text{hr},\mathcal{C}}(\mathcal{A}_a^{\text{hr}}) = atk_{\mathcal{H},\mathcal{C}}^{\text{flat}}(\mathcal{A}_a^{\text{hr}})$ the HR adversarial images obtained by an attack $atk_{\mathcal{H},\mathcal{C}}^{\text{flat}}$ for the flat scenario performed on $\mathcal{A}_a^{\text{hr}} \in \mathcal{H}$, and by $\mathcal{D}_{\text{flat}}^{\mathcal{C}}(\mathcal{A}_a^{\text{hr}}) \in \mathcal{R}$ its degraded version.

3 The Noise Blowing-Up Strategy

We present here a method that attempts to circumvent the speed, adversity, and visual quality challenges, that are encountered when one intends to create HR

adversarial images. While speed and adversity were successfully addressed in [11, 12] *via* a strategy similar to some extent to the present one, the visual quality challenge remained partly unsatisfying. The refinement provided by the noise-blowing up method presented here addresses this issue, simplifies and generalises the attack scheme described in [11, 12], and lifts to the \mathcal{H} domain any attack working in the \mathcal{R} domain.

The design of the noise blowing-up strategy, that aims at creating, in seven steps, an efficient attack in the \mathcal{H} domain once given an efficient attack in the \mathcal{R} domain, is given in Subsect. 3.1. The description of the process is detailed here for the challenging *target scenario* (any other scenario can easily be derived from the presented scheme). Subsection 3.2 gives a series of indicators. The assessment of these indicators depends on the choice of the degrading and enlarging functions used to move from \mathcal{H} to \mathcal{R} , and *vice versa*. These choices are made in the experiments performed in Sect. 4.

3.1 Construction of Adversarial Images in \mathcal{H} for the Target Scenario

Given a CNN \mathcal{C} , the starting point is a large-size clean image $\mathcal{A}_a^{\text{hr}} \in \mathcal{H}$.

In Step 1, one constructs its degraded image $\mathcal{A}_a = \rho(\mathcal{A}_a^{\text{hr}}) \in \mathcal{R}$.

In Step 2, one runs \mathcal{C} on \mathcal{A}_a to get its classification in a category c_a . More precisely, one gets $\mathcal{C}(\mathcal{A}_a) = (c_a, \tau_a)$.

In Step 3, one assumes given an image $\tilde{\mathcal{D}}_{t, \tilde{\tau}_t}^{\mathcal{C}}(\mathcal{A}_a) \in \mathcal{R}$, that is adversarial for the (c_a, c_t) target scenario performed on $\mathcal{A}_a = \rho(\mathcal{A}_a^{\text{hr}})$ for a c_t -label value $\tilde{\tau}_t$ exceeding a threshold $\tilde{\tau}$. As already stated, it does not matter how such an adversarial image is obtained.

Step 4 consists in getting the adversarial noise $\mathcal{N}^{\mathcal{C}}(\mathcal{A}_a) \in \mathcal{R}$ as the difference

$$\mathcal{N}^{\mathcal{C}}(\mathcal{A}_a) = \tilde{\mathcal{D}}_{t, \tilde{\tau}_t}^{\mathcal{C}}(\mathcal{A}_a) - \mathcal{A}_a \in \mathcal{R} \quad (6)$$

of images living in \mathcal{R} , one being the adversarial image of the clean other.

To perform Step 5, one needs a fixed *enlarging function*

$$\lambda: \mathcal{R} \longrightarrow \mathcal{H} \quad (7)$$

that transforms any image of \mathcal{R} into an image in \mathcal{H} (see Sect. 4.1 for the specific used λ function). Anticipating Step 4, it is worth noting that, although the *reduction function* ρ and the *enlarging function* λ have opposite purposes, these functions are not necessarily inverse one from the other. In other words, $\rho \circ \lambda$ and $\lambda \circ \rho$ may differ from the identity maps $id_{\mathcal{R}}$ and $id_{\mathcal{H}}$ respectively (usually they do).

One applies the enlarging function λ to the low-resolution adversarial noise $\mathcal{N}^{\mathcal{C}}(\mathcal{A}_a)$, what leads to the blown-up noise $\mathcal{N}^{\text{hr}, \mathcal{C}}(\mathcal{A}_a^{\text{hr}}) = \lambda(\mathcal{N}^{\mathcal{C}}(\mathcal{A}_a)) \in \mathcal{H}$. Then one creates the HR tentative adversarial image by adding this blown-up noise to the original high-resolution image as follows:

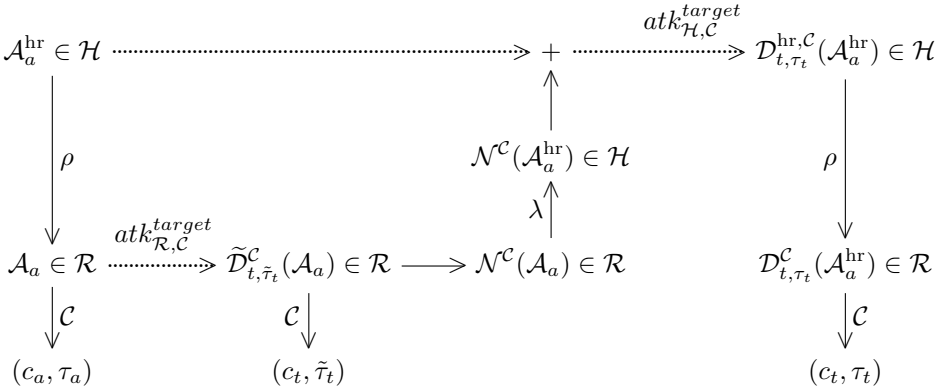
$$\mathcal{D}_{t, \tau_t}^{\text{hr}, \mathcal{C}}(\mathcal{A}_a^{\text{hr}}) = \mathcal{A}_a^{\text{hr}} + \mathcal{N}^{\text{hr}, \mathcal{C}}(\mathcal{A}_a^{\text{hr}}) \in \mathcal{H}. \quad (8)$$

In Step 6, the application of the reduction function ρ to this HD tentative adversarial image creates an image $\mathcal{D}_{t,\tau_t}^{\mathcal{C}}(\mathcal{A}_a^{\text{hr}}) = \rho(\mathcal{D}_{t,\tau_t}^{\text{hr},\mathcal{C}}(\mathcal{A}_a^{\text{hr}}))$ in the \mathcal{R} domain.

In Step 7, one runs \mathcal{C} on $\mathcal{D}_{t,\tau_t}^{\mathcal{C}}(\mathcal{A}_a^{\text{hr}})$ to get its classification.

The attack succeeds if \mathcal{C} classifies this image in c_t , potentially for a c_t -label value τ_t exceeding the threshold value τ fixed in advance, and if a human is unable to notice any difference between the images $\mathcal{A}_a^{\text{hr}}$ and $\mathcal{D}_{t,\tau_t}^{\text{hr},\mathcal{C}}(\mathcal{A}_a^{\text{hr}})$ in the \mathcal{H} domain. The key point is to set the value of $\tilde{\tau}_t$ so that this occurs.

The following scheme, summarizing the seven steps, shows how to create, from a targeted attack $atk_{\mathcal{R},\mathcal{C}}^{\text{target}}$ efficient against \mathcal{C} in the \mathcal{R} domain, the attack $atk_{\mathcal{H},\mathcal{C}}^{\text{target}}$ in the \mathcal{H} domain obtained by the noise blowing-up method:



3.2 Indicators

Although both $\tilde{\mathcal{D}}_{t,\tilde{\tau}_t}^{\mathcal{C}}(\mathcal{A}_a)$ and $\mathcal{D}_{t,\tau_t}^{\mathcal{C}}(\mathcal{A}_a^{\text{hr}})$ stem from $\mathcal{A}_a^{\text{hr}}$, and belong to the same set \mathcal{R} of low-resolution images, these images nevertheless differ in general, since $\rho \circ \lambda \neq id_{\mathcal{R}}$. Therefore, the verification process performed in Step 7 on the HR tentative adversarial image, which checks whether its reduction belongs to c_t , is mandatory. Moreover, should it be the case, $\tilde{\tau}_t$ and τ_t are likely to differ. The real-valued *loss function* \mathcal{L} defined for $\mathcal{A}_a^{\text{hr}} \in \mathcal{H}$ gives the difference:

$$\mathcal{L}_{\mathcal{C}}(\mathcal{A}_a^{\text{hr}}) = \tilde{\tau}_t - \tau_t. \quad (9)$$

Our attack is effective if one can set accurately the value of $\tilde{\tau}_t$ to match the inequality $\tau_t \geq \tau$ for the threshold value τ , or to make sure that $\mathcal{D}_{t,\tau_t}^{\mathcal{C}}(\mathcal{A}_a^{\text{hr}})$ is a good enough adversarial image in the \mathcal{R} domain, while controlling the distance variations between $\mathcal{A}_a^{\text{hr}}$ and the adversarial $\mathcal{D}_{t,\tau_t}^{\text{hr},\mathcal{C}}(\mathcal{A}_a^{\text{hr}})$.

Additionally, the visual proximity between images for a human eye is assessed by L_p distances (see Subsect. 2.1). There are two pairs of images that one wants to compare. On the one hand, there is the pair $(\mathcal{A}_a, \mathcal{D}_{t,\tau_t}^{\mathcal{C}}(\mathcal{A}_a^{\text{hr}}))$ of images in the \mathcal{R} domain, for which one uses the same L_p distance as in the attack $atk_{\mathcal{R},\mathcal{C}}^{\text{target}}$. On the other hand, there is the pair $(\mathcal{A}_a^{\text{hr}}, \mathcal{D}_{t,\tau_t}^{\text{hr},\mathcal{C}}(\mathcal{A}_a^{\text{hr}}))$ of images in the \mathcal{H} domain, for

which one uses the L_2 distance systematically. In this case, the most important of both actually, one writes more simply $L_2^{hr} = L_2(\mathcal{A}_a^{hr}, \mathcal{D}_{t, \tau_t}^{hr, \mathcal{C}}(\mathcal{A}_a^{hr}))$ when there is no ambiguity.

Note that the present approach, unlike the approach introduced in [11, 12], does not require frequently scale up and down via λ, ρ the adversarial images. In particular, if one knows how the loss function behaves (in the worst case, or on average) for a given attack, then one can adjust *a priori* the value of $\tilde{\tau}_t$ accordingly, and be satisfied with one scaling up and down.











4 Case Study

This section provides a (first) proof of concept of our noise blowing-up strategy with one CNN, one scenario, one attack and 10 HR images.

4.1 The CNN, the Scenario, the Images

We consider $\mathcal{C} = \text{VGG-16}$ trained on ImageNet [7], and the 10 clean HR images $\mathcal{A}_1^{hr}, \dots, \mathcal{A}_{10}^{hr}$ pictured in Table 1. These images, including the two images \mathcal{A}_9^{hr} and \mathcal{A}_{10}^{hr} graciously provided by the French artist Speedy Graphito [15], are those considered in [11, 12]. More precisely, Table 1 gives 10 categories c_1, \dots, c_{10} , and, for each c_a , it gives a HR image \mathcal{A}_a^{hr} , whose degraded version is classified by $\mathcal{C} = \text{VGG-16}$ in c_a . Taking advantage of the outcomes of [11, 12] for the choice of most parameters used in the case study, we use $(\rho, \lambda) = (\text{Lanczos}, \text{Lanczos})$ (see [8, 14] for the Lanczos method). Table 1 gives the original size of \mathcal{A}_a^{hr} , the classification (c_a, τ_a) by VGG-16 of $\rho(\mathcal{A}_a^{hr})$, and the category c_t used for the (c_a, c_t) target scenario (identical to those used in [11, 12], picked at random among the categories of ImageNet).

Table 1. For $1 \leq a \leq 10$, the image \mathcal{A}_a^{hr} classified by VGG-16 in the category c_a , and their respective target categories c_t .

a	1	2	3	4	5	6	7	8	9	10
c_a	Cheetah	Eskimo Dog	Koala	Lamp Shade	Toucan	Screen	Comic Book	Sports Car	Binder	Coffee Mug
$h \times w$	604×910	640×960	607×910	2913×2462	607×910	600×641	800×1280	800×1280	2011×1954	1710×1740
\mathcal{A}_a^{hr}										
τ_a	0.9527	0.3434	0.9974	0.5359	0.4553	0.7064	0.4916	0.4802	0.2825	0.0844
c_t	poncho	goblet	Weimaraner	weevil	wombat	swing	altar	beagle	triceratops	hamper

4.2 The Attack

We apply the noise blowing-up strategy with the black-box evolutionary algorithm (EA) based attack developed in [4, 17]. For this EA attack, we keep the same parameters as those of [4, 17]: $\alpha = 1$, $\epsilon = 16$, and $X = 20.000$. The pseudocode of the EA-based attack, expressed in the \mathcal{R} domain, is given as follows:

Algorithm 1. EA attack pseudocode [4, 17]

-
- 1: **Input:** CNN \mathcal{C} , ancestor \mathcal{A} , perturbation magnitude α , maximum perturbation ϵ , ancestor class c_a , ordinal t of target class c_t , g current and X maximum generation;
 - 2: Initialize population as 40 copies of \mathcal{A} , with I_0 as first individual;
 - 3: Compute fitness for each individual;
 - 4: **while** ($\sigma_{I_0}[t] < \tau$) & $x < X$ **do**
 - 5: Rank individuals in descending fitness order and segregate: elite 10, middle class 20, lower class 10;
 - 6: Select random number of pixels to mutate and perturb them with $\pm\alpha$. Clip all mutations to $[-\epsilon, \epsilon]$. The elite is not mutated. The lower class is replaced with mutated individuals from the elite and middle class;
 - 7: Cross-over individuals to form a new population;
 - 8: Evaluate fitness of each individual;
-

We set $\tilde{\tau} = 0.55$ to ensure that the $\tilde{\tau}_t$ -strong adversarial images, obtained by this attack in the \mathcal{R} domain, are clearly in the c_t target category, with a convincing margin ≥ 0.10 with respect to the second best category. Since different seed values for the EA lead to different adversarial images, to ensure the reliability of our results, we performed, for each clean HR image $\mathcal{A}_a^{\text{hr}}$, and each (c_a, c_t) pair, 10 independent runs with random seed values. The EA succeeded in all cases, creating a total of 100 adversarial images, 10 for each clean image $\mathcal{A}_a^{\text{hr}}$.

4.3 Experimental Results

Referring to the steps specified in Subsect. 3.1, for each ancestor image $\mathcal{A}_a^{\text{hr}}$ specified in their 1st column, Table 2 and Table 3 summarize the results of the case study, computed as averages over the 10 independent runs. Note that Step 3, which corresponds to the concrete attack performed in the \mathcal{R} domain, should be considered essentially as “outside” our strategy, in the sense that it is an input on which we have no influence *a priori*. Therefore the computational efforts performed in this Step 3 do not impact the performance of our scheme.

In Table 2, the 2nd column, which corresponds to Step 3 of the noise blowing-up strategy, gives the average number of generations required by the EA to create a 0.55-strong adversarial image for the (c_a, c_t) target scenario (note that the two artistic images are the most challenging of all). The 3rd column gives the average value of $\tilde{\tau}_t$, which of course exceeds $\tilde{\tau} = 0.55$ as expected. The 4th column provides the average c_t -label value for the degraded adversarial images. The 5th column gives the average loss (Eq. 9). This difference between the adversarial images in \mathcal{R} varies between 0.0132 and 0.1950. Still, in all cases, the degraded adversarial image remained classified in the target category c_t . The last column assesses the visual quality difference between the HR clean and adversarial images.

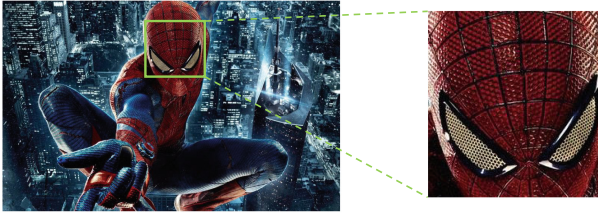
Table 2. Average $\tilde{\tau}_t$ and τ_t with corresponding loss values and average L_2 distances between the ancestor and adversarial images in the HR domain.

	$avgGens^{0.55}$	$avg_{\tilde{\tau}_t}$	avg_{τ_t}	$avg_{\mathcal{L}}$	$avg_{L_2^{hr}}$
\mathcal{A}_1^{hr}	9994	0.5505	0.3929	0.1576	9803
\mathcal{A}_2^{hr}	3985	0.5502	0.5233	0.0270	10476
\mathcal{A}_3^{hr}	3529	0.5510	0.4930	0.0581	10052
\mathcal{A}_4^{hr}	3212	0.5510	0.4815	0.0695	31833
\mathcal{A}_5^{hr}	2845	0.5512	0.4957	0.0556	9532
\mathcal{A}_6^{hr}	5188	0.5505	0.5373	0.0132	8405
\mathcal{A}_7^{hr}	3000	0.5506	0.4177	0.1329	27091
\mathcal{A}_8^{hr}	3377	0.5503	0.4968	0.0535	26237
\mathcal{A}_9^{hr}	15603	0.5504	0.3553	0.1950	12136
\mathcal{A}_{10}^{hr}	11770	0.5501	0.5246	0.0255	12819
Avg	6250	0.5506	0.4718	0.0788	15838

Table 3 lists the average execution time spent on each step of the noise blowing-up method. Out of those, recall again that Step 3 is used in, but is independent from the noise blowing-up strategy. The time overhead required by the noise blowing-up strategy is the sum of the time of all steps except Step 3. Its value, given in the last column, amounts to 0.14571 seconds on average, which is negligible both in absolute terms as well as compared to the circa one hour required by the EA attack referred to in Step 3: the noise blowing-up time overhead amounts to 0.004% for this specific attack.

Table 3. Average time (in seconds) spent on the main steps of the noise blowing-up technique, and noise blowing-up time overhead.

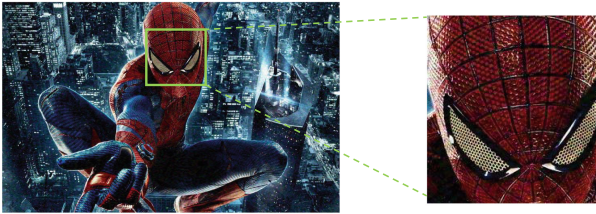
	Step1	Step2	Step 3	Step 4	Step 5	Step 6	Step 7	Overhead
\mathcal{A}_1^{hr}	0.00727	0.03362	5048	0.00018	0.00857	0.00700	0.03569	0.09233
\mathcal{A}_2^{hr}	0.00976	0.03484	2299	0.00019	0.00968	0.00789	0.03679	0.09914
\mathcal{A}_3^{hr}	0.00827	0.03631	1848	0.00020	0.00856	0.00718	0.03689	0.09740
\mathcal{A}_4^{hr}	0.07146	0.03663	2199	0.00020	0.11523	0.06922	0.03831	0.33104
\mathcal{A}_5^{hr}	0.00980	0.03573	1660	0.00020	0.00875	0.00721	0.03716	0.09885
\mathcal{A}_6^{hr}	0.00831	0.03726	2920	0.00019	0.00611	0.00576	0.03764	0.09528
\mathcal{A}_7^{hr}	0.01424	0.03484	1773	0.00021	0.01556	0.01221	0.03736	0.11441
\mathcal{A}_8^{hr}	0.01478	0.03576	1716	0.00020	0.01480	0.01217	0.03627	0.11398
\mathcal{A}_9^{hr}	0.04199	0.03558	9072	0.00024	0.06720	0.04020	0.03731	0.22252
\mathcal{A}_{10}^{hr}	0.03445	0.03637	6564	0.00020	0.05190	0.03186	0.03740	0.19218
Average	0.02203	0.03569	3510	0.00020	0.03064	0.02007	0.03708	0.14571



(a) $\mathcal{A}_7^{\text{hr}}$, classified as "comic book" with confidence ≥ 0.49



(b) Adversarial image obtained by the strategy described in [11,12], classified as "altar" with confidence ≥ 0.52



(c) $\mathcal{D}_{\text{altar},0.41}^{\text{hr},\text{VGG-16}}(\mathcal{A}_7^{\text{hr}})$, classified as "altar" with confidence ≥ 0.41

Fig. 1. Visual comparison of the clean HR image $\mathcal{A}_7^{\text{hr}}$ with the adversarial HR images obtained by $\text{EA}^{\text{target},\mathcal{C}}$ for $\mathcal{C} = \text{VGG-16}$ with $\tilde{\tau}_t \geq 0.55$, and $c_t = \text{altar}$. The clean $\mathcal{A}_7^{\text{hr}}$ (a), the HR adversarial image obtained from [11, 12] (b), and the HR adversarial image obtained from the noise blowing-up strategy (c).

5 One Detailed Example

The “true” visual quality for a human eye is assessed by looking at some representative examples either from some distance or by zooming in on some areas. This section highlights on the clean HR image $\mathcal{A}_7^{\text{hr}}$ the visual quality enhancements that benefit the HR adversarial images obtained by the noise blowing-up strategy, as compared with the HR adversarial images constructed in [11, 12]. Especially, one considers areas that remained visually problematic with the method used in these latter papers.

Figure 1a represents this clean HR image $\mathcal{A}_7^{\text{hr}}$, and a zoom of that picture in some areas. Figure 1b shows the HR adversarial image obtained by the method described in [11, 12]. Figure 1c shows the HR adversarial image obtained by the noise blowing-up method. For both methods, $\tilde{\tau}$ was set to 0.55.

At some distance, both HR adversarial images present a good visual quality. However, the zoomed areas show differences between the HR adversarial images. Details from the HR adversarial image shown in Fig. 1b become blurry for a human eye. Therefore, a human is able to distinguish between the clean image shown in Fig. 1a and the adversarial image shown in Fig. 1b. The situation differs significantly with the adversarial image obtained from the noise blowing-up method. Zooming into the same area does not exhibit any visible blurriness anymore. It becomes much more challenging for a human to distinguish between the clean HR image in Fig. 1a and the HR adversarial image in Fig. 1c.

6 Conclusion

This paper describes the noise blowing-up strategy that constructs high-resolution adversarial images against CNNs at their image recognition task. This strategy applies to any scenario and any effective attack in the low-resolution domain. We presented a convincing proof of concept for this strategy, thanks to one CNN, one scenario, one attack, and a few high-resolution images. This strategy successfully addressed the speed and adversity challenges raised by the construction of HR adversarial images. Foremost, our method substantially enhanced the visual quality of the obtained adversarial images, as compared to previous methods. Finally, our experiments showed that the noise blowing-up strategy overhead is extremely modest compared to the time required by the concrete attack at hand.

This paper will be extended in many ways. Firstly, we intend to apply the strategy to at least 10 diverse and state-of-the-art CNNs, to different attacks (black-box, white-box, GANs) performed on more scenarios, and on many more clean HR images, and explore the deep reasons for the enhanced visual quality provided by our strategy. Secondly, we intend to study variants of this strategy. For instance, instead of blowing up one layer of some strong adversarial noise, one can blow up several layers of lighter adversarial noise. Implementing this variant in parallel may accelerate the overall process. Thirdly, we intend to compare (or combine) this strategy with another one, of a completely different nature, that would involve some pre-processing to select the areas of interest on which to focus the construction of adversarial noise.

Acknowledgements. We thank Bernard Utudjian and Speedy Graphito for providing artistic images used in Sect. 4, and Uli Sorger for fruitful discussions.

References

1. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint [arXiv:1603.04467](https://arxiv.org/abs/1603.04467) (2016)
2. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57. IEEE (2017)

3. Chitic, R., Bernard, N., Leprévost, F.: A proof of concept to deceive humans and machines at image classification with evolutionary algorithms. In: Nguyen, N.T., Jearanaitanakij, K., Selamat, A., Trawiński, B., Chittayasothorn, S. (eds.) ACIIDS 2020. LNCS (LNAI), vol. 12034, pp. 467–480. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-42058-1_39
4. Chitic, R., Leprévost, F., Bernard, N.: Evolutionary algorithms deceive humans and machines at image classification: an extended proof of concept on two scenarios. *J. Inf. Telecommun.*, 1–23 (2020)
5. Chitic, R., Topal, A.O., Leprévost, F.: Evolutionary algorithm-based images, humanly indistinguishable and adversarial against convolutional neural networks: efficiency and filter robustness. *IEEE Access* **9**, 160758–160778 (2021)
6. Chollet, F., et al.: Keras. <https://keras.io> (2015)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: The ImageNet image database (2009). <http://image-net.org>
8. Duchon, C.E.: Lanczos filtering in one and two dimensions. *J. Appl. Meteorol. Climatol.* **18**(8), 1016–1022 (1979)
9. Guo, C., Gardner, J., You, Y., Wilson, A.G., Weinberger, K.: Simple black-box adversarial attacks. In: International Conference on Machine Learning, pp. 2484–2493. PMLR (2019)
10. Hu, W., Tan, Y.: Generating adversarial malware examples for black-box attacks based on GAN. In: Tan, Y., Shi, Y. (eds.) Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, 21–24 November 2022, Proceedings, Part II, pp. 409–423. Springer, Singapore (2023). https://doi.org/10.1007/978-981-19-8991-9_29
11. Leprévost, F., Topal, A.O., Avdusinovic, E., Chitic, R.: Strategy and feasibility study for the construction of high resolution images adversarial against convolutional neural networks. In: Nguyen, N.T., Tran, T.K., Tukayev, U., Hong, TP., Trawiński, B., Szczerbicki, E. (eds.) Intelligent Information and Database Systems. 14th Asian Conference, ACIIDS 2022, Ho-Chi-Minh-City, Vietnam, 28–30 November 2022, pp. 467–480. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-21743-2_23
12. Leprévost, F., Topal, A.O., Avdusinovic, E., Chitic, R.: A strategy creating high-resolution adversarial images against convolutional neural networks and a feasibility study on 10 CNNs. *J. Inf. Telecommun.*, 1–31 (2022)
13. Oliphant, T.E.: A guide to NumPy. Trelgol Publishing USA (2006)
14. Parsania, P.S., Virparia, P.V.: A comparative analysis of image interpolation algorithms. *Int. J. Adv. Res. Comput. Commun. Eng.* **5**(1), 29–34 (2016)
15. SpeedyGraphito: Mes 400 Coups. Panoramart (2020)
16. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint [arXiv:1312.6199](https://arxiv.org/abs/1312.6199) (2013)
17. Topal, A.O., Chitic, R., Leprévost, F.: One evolutionary algorithm deceives humans and ten convolutional neural networks trained on ImageNet at image recognition. *Appl. Soft Comput.* **143**, 110397 (2023). <https://doi.org/10.1016/j.asoc.2023.110397>. <https://www.sciencedirect.com/science/article/pii/S1568494623004155>
18. Van Rossum, G., Drake, F.L.: Python 3 Reference Manual. CreateSpace, Scotts Valley (2009)
19. Van der Walt, S., et al.: The scikit-image contributors: scikit-image: image processing in Python. *PeerJ* **2**, e453 (2014). <https://doi.org/10.7717/peerj.453>