

A strategy creating high-resolution adversarial images against convolutional neural networks and a feasibility study on 10 CNNs

Franck Leprévost, Ali Osman Topal, Elmir Avdusinovic & Raluca Chitic

To cite this article: Franck Leprévost, Ali Osman Topal, Elmir Avdusinovic & Raluca Chitic (2023) A strategy creating high-resolution adversarial images against convolutional neural networks and a feasibility study on 10 CNNs, Journal of Information and Telecommunication, 7:1, 89-119, DOI: [10.1080/24751839.2022.2132586](https://doi.org/10.1080/24751839.2022.2132586)

To link to this article: <https://doi.org/10.1080/24751839.2022.2132586>



© 2022 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 31 Oct 2022.



Submit your article to this journal [↗](#)



Article views: 446



View related articles [↗](#)



View Crossmark data [↗](#)

A strategy creating high-resolution adversarial images against convolutional neural networks and a feasibility study on 10 CNNs

Franck Leprévost , Ali Osman Topal , Elmir Avdusinovic  and Raluca Chitic 

Faculty of Science, Engineering and Medicine and Department of Computer Science, University of Luxembourg, Esch-sur-Alzette, Luxembourg

ABSTRACT

To perform image recognition, Convolutional Neural Networks (CNNs) assess any image by first resizing it to its input size. In particular, high-resolution images are scaled down, say to 224×244 for CNNs trained on ImageNet. So far, existing attacks, aiming at creating an adversarial image that a CNN would misclassify while a human would not notice any difference between the modified and unmodified images, proceed by creating adversarial noise in the 224×244 resized domain and not in the high-resolution domain. The complexity of directly attacking high-resolution images leads to challenges in terms of speed, adversity and visual quality, making these attacks infeasible in practice. We design an indirect attack strategy that lifts to the high-resolution domain any existing attack that works efficiently in the CNN's input size domain. Adversarial noise created via this method is of the same size as the original image. We apply this approach to 10 state-of-the-art CNNs trained on ImageNet, with an evolutionary algorithm-based attack. Our method succeeded in 900 out of 1000 trials to create such adversarial images, that CNNs classify with probability ≥ 0.55 in the adversarial category. Our indirect attack is the first effective method at creating adversarial images in the high-resolution domain.

ARTICLE HISTORY

Received 29 July 2022
Accepted 29 September 2022

KEYWORDS

Black-box attack; convolutional neural network; evolutionary algorithm; high-resolution adversarial image

1. Introduction

The profusion of images in our modern-day society and the need to analyse quickly the information they contain for a large series of applications (self-driving cars, face recognition and security controls, etc) has led to the emergence of tools to automatically process and sort this type of data. Trained Convolutional Neural Networks (CNNs) are among the dominant and most accurate tools for automatic object recognition and classification. Nevertheless, CNNs can be led to erroneous classifications by specifically designed adversarial images. The consequences of such attacks might be catastrophic. For instance for self-driving cars, an attack changing the perception by a CNN of a stop

CONTACT Franck Leprévost  Franck.Leprevost@uni.lu  University of Luxembourg, House of Numbers, 6, avenue de la Fonte, Esch-sur-Alzette, L-4364, Luxembourg

© 2022 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group
This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

sign into a 50 km per hour signal would lead to car accidents, and put the passengers at risk. In a more general context, performing attacks reveals CNN weaknesses, which might lead to more robust CNNs. The present article is in this line of thoughts. Attacks depend on the adversarial scenario considered. For instance, starting with an original image classified by a CNN in a given category, the target scenario essentially consists in 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 the adversarial image.

Attacks, that intend to construct adversarial images, are classified according to the level of knowledge about the CNN at the disposal of the attacker. In this hierarchy, black-box attacks are the most challenging ones, since no knowledge about the architecture of the CNN (number and type of layers, weights, etc.) is assumed. Such attacks already exist (Andriushchenko et al., 2020; Chitic et al., 2021; Guo et al., 2019; Hu & Tan, 2017; Papernot et al., 2017; Topal et al., 2022) (see also (Biggio et al., 2013; Carlini & Wagner, 2017; Szegedy et al., 2013; Tsipras et al., 2018) for gradient-based attacks). For instance, the paper Topal et al. (2022) shows how an evolutionary-based algorithm successfully fooled 10 CNNs trained on ImageNet (Deng et al., 2009) to sort images of size 224×224 into 1000 categories ((Chitic, Bernard et al., 2020; Chitic, Leprévost et al., 2020) provided a first version of this algorithm that fooled VGG-16 (Blier, 2016) trained on CIFAR-10 (Krizhevsky et al., 2009) to sort images of size 32×32 into 10 categories).

1.1. Attacks in the \mathcal{R} domain

So far, all such attacks – black-box or not – addressed images of moderate size, what is called here the \mathcal{R} domain. A moderate size ranges from 32×32 (typically for CNNs trained on CIFAR-10) up to 224×224 (typically for CNNs trained on ImageNet). It also encompasses usually slightly larger sizes that trained CNNs may handle natively. The construction of images, adversarial for the target scenario in this ‘traditional’ context, is achieved by adding some carefully designed adversarial noise to the potentially resized original image in a process illustrated in Figure 1.

In particular, the adversarial noise created by all these attacks is in the \mathcal{R} domain handled natively by the CNNs. Therefore, the obtained adversarial images are as large as the CNN’s input size. Said otherwise, attacks in the ‘traditional’ context create an adversarial

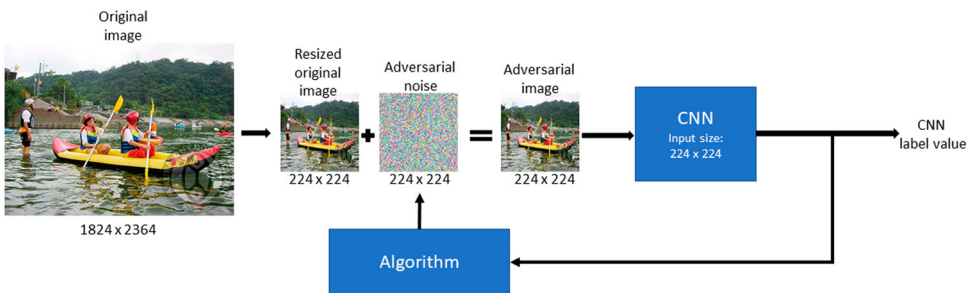


Figure 1. Generating an adversarial image of size 224×224 .

noise of size equal to the size of the CNN input, independently on the size of the original image. This means that the size of the search space of these attacks does not depend on the size of the original image, but coincides with the size of the CNN input. Note *en passant* that the smaller the input size of the CNN, the easier the creation of adversarial noise.

1.2. Three challenges faced by attacks in the \mathcal{H} domain

However, if the adversarial image should preserve almost all the details of an original image of large size, what we call here an image in the \mathcal{H} domain, in particular of a high-resolution (HR) image, the adversarial noise should have the same size as the original image, and consequently the adversarial image should as well have the same size as the original one. A key point is that the adversity character of a modified image is measured only when it is exposed to the CNN, hence when it is resized to fit into the \mathcal{R} domain. The adversarial character of an image should show up when the CNN proceeds to the classification of its resized version, as illustrated in the process given in Figure 2.

Creating adversarial images of large size leads to three challenges in terms of speed, adversity and visual quality. Firstly, the complexity of the problem increases drastically with the size of the images, as the search space for the adversarial noise grows quadratically. For instance, the noise search space provided by the original image represented in Figure 2 is 86 times larger than it is in the 224×224 domain. Secondly, the noise introduced in the \mathcal{H} domain should be assessed as adversarial in the \mathcal{R} domain: it should ‘survive’ the resizing process to fit the CNN. In the example of Figure 2, it would essentially mean that it survives a 86-fold squeezing process. Thirdly, the noise introduced in the \mathcal{H} domain should be imperceptible to a human eye looking at the images at their native size, and not merely once they are reduced to fit the \mathcal{R} domain. For the example in Figure 2, it means that a human should not notice any difference between the first and second images of size 1824×2364 when looked at full size.

Already the first challenge is a very serious one. Indeed, should it even succeed, getting directly such an HR adversarial image can take a very long time, even on a performing HPC. This is probably the reason for which, to the best of our knowledge, so far, no attack – black-box or not – has attempted to address large size images, in particular high-resolution images, by creating convenient adversarial noise in the \mathcal{H} domain, so that the modified image, resized to the size handled natively by the CNN, becomes adversarial. Applying existing methods does not work, at least in reasonable time. Although efficient in the \mathcal{R} domain, their extension to the \mathcal{H} domain is not.

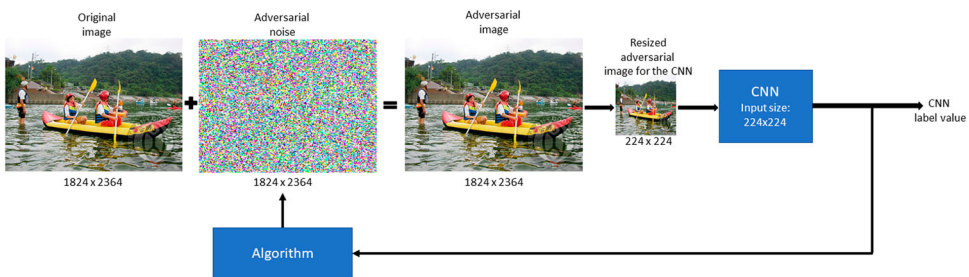


Figure 2. Generating adversarial images in the \mathcal{H} domain.

1.3. Our contribution: an effective strategy supported by an extensive feasibility study

This paper is a first step towards the creation of adversarial noise of size of the original image, whatever this size may be. Our contribution is essentially threefold.

First, we describe an indirect attack strategy that leads to the construction of HR images in the \mathcal{H} domain that are adversarial for the target scenario performed on a trained CNN (Section 3). The conceptual design of the strategy is flexible enough to lift to the \mathcal{H} domain attacks considered as efficient in the \mathcal{R} domain. Furthermore, it lists indicators relevant to the problem, and it describes appropriate tests to assess the behaviour and the efficiency of potential resizing functions.

Second, we perform a feasibility study of this strategy with 10 explicit HR images and on 10 CNNs trained on ImageNet. We lift to the \mathcal{H} domain a black-box attack based on an evolutionary algorithm. We prove experimentally that our strategy is highly efficient in terms of speed and of adversity, and is reasonably efficient in terms of visual quality (Section 4). Concretely, after having briefly described the evolutionary algorithm used, we show that our method succeeds in 900 out of 1000 trials, that the most appropriate resizing function is the Lanczos function, and that the successful attempts require in average between 48' and 119.2' to create 0.55-strong high-resolution adversarial images (and between 35.7' and 98.8' to create good enough high resolution adversarial images).

Third, this study is completed by an attempt to apply the black-box evolutionary algorithm-based attack directly in the \mathcal{H} domain (Section 5). After 48 hours of computation time, our algorithm is unable to create 0.55-strong high-resolution adversarial images for any of the 10 CNNs. Although the learning curve of the algorithm improves, and although it creates images with a c_t -label value increased by a factor in the range [1.71, 5.5] according to the CNN, the attack is not fast enough. These outcomes, that experimentally substantiate the seriousness of already the first challenge, are an additional argument in favour of alternative strategies like ours, to efficiently construct adversarial images in the \mathcal{H} domain.

Two sections and an appendix complete this article. Section 2 fixes some notations about CNNs, formalizes the target scenario in general, and its 'lifted' version in the context of high resolution images. Section 6 wraps up our findings and provides directions for future research. The appendix section contain additional evidence of our findings.

All algorithms and experiments were implemented using Python 3.8 (Van Rossum & Drake, 2009) with NumPy 1.17 (Oliphant, 2006), TensorFlow 2.4 (Abadi et al., 2015), Keras 2.2 (Chollet, 2015) and Scikit 0.24 (Walt et al., 2014) libraries. Computations were performed on nodes with Nvidia Tesla V100 GPGPUs of the IRIS HPC Cluster at the University of Luxembourg.

Our paper Leprévost et al. (2022) dealt with one single CNN only, namely VGG-16. This previous work is substantially extended and enhanced here. First, 10 diverse, state-of-the-art CNNs are considered. Second, we provide the explicit design of a series of tests, study closely the behaviour of indicators according to these tests and perform additional experiments. More specifically, we perform 1000 indirect attacks versus 100 in the previous paper. Third, we carefully study the Loss function (see Section 3.2 for its definition)

according to different resizing functions. Finally, we perform direct attacks on the 10 CNNs and provide their convergence graphs and timings.

2. CNNs and the target scenario

CNNs performing image classification are trained on some large dataset \mathcal{S} to sort images into predefined categories c_1, \dots, c_ℓ . The categories, and their number ℓ , are associated to \mathcal{S} , and are common to all CNNs trained on \mathcal{S} . The training phase of a CNN is essentially made in two steps. During the first step, the CNN is given both a series of training images, and, for each training image, a vector of length ℓ , where each real-value component assesses the probability that the training image represents an object in the corresponding category. During the second step, the CNN is challenged against a validation set of images that assess its ability to sort images accurately.

Once trained, a CNN can be exposed to images (typically) of the same size as those on which it was trained. In practice, given an input image \mathcal{I} , the trained CNN produces a classification output vector

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

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 component $\mathbf{o}_{\mathcal{I}}[i]$ of the output vector defines the c_i -label value measuring the probability 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 $k = \operatorname{argmax}_{1 \leq j \leq \ell} (\mathbf{o}_{\mathcal{I}}[j])$ and one denotes $(c_k, \mathbf{o}_{\mathcal{I}}[k])$ this outcome. The higher the label value $\mathbf{o}_{\mathcal{I}}[k]$, the higher the confidence that \mathcal{I} represents an object of the category c_k .

2.1. The target scenario

Let \mathcal{C} be a trained CNN as above, c_a be a category among the ℓ possible categories, and \mathcal{A} an image classified by \mathcal{C} as belonging to c_a . One denotes by τ_a its c_a -label value. The *target scenario* (c_a, c_t) performed on \mathcal{A} requires first to select a category $c_t \neq c_a$, and then to construct an image \mathcal{D} that is either a *good enough adversarial image* or a *τ -strong adversarial image* in the sense made precise below.

In any case, one requires that \mathcal{D} remains so close to \mathcal{A} that a human cannot notice any difference between \mathcal{A} and \mathcal{D} . The quantities $L_2(\mathcal{A}, \mathcal{D})$ and $\epsilon(\mathcal{A}, \mathcal{D})$ assess numerically this human perception. The L_2 -distance essentially evaluates the difference between the pixel values of \mathcal{A} and \mathcal{D} , and ϵ controls (or restricts) the global maximum amplitude allowed for the value modifications of each individual pixel of \mathcal{A} to obtain \mathcal{D} .

A *good enough adversarial image* is an adversarial image that \mathcal{C} classifies as belonging to the target category c_t , without any requirement on the c_t -label value beyond being strictly dominant among all label values. A *τ -strong adversarial image* is an adversarial image that \mathcal{C} not only classifies as belonging to the target category c_t , but for which its c_t -label value $\tau_t \geq \tau$ for some threshold value $\tau \in]0, 1]$ fixed *a priori*. We write (c_t, τ_t) the outcome of the CNN's classification of \mathcal{D} in this latter case.

2.2. The target scenario lifted to \mathcal{H}

In the experiments of Section 4, we shall consider a CNN \mathcal{C} that handles images of size 224×224 , and that is trained on ImageNet to classify images into 1000 categories. In our context, we ask \mathcal{C} to give the dominating category, and the corresponding label value for that category. Henceforth, \mathcal{C} 's classifications take values in

$$\mathcal{V} = \{(c_i, v_i), \text{ where } v_i \in]0, 1] \text{ for } 1 \leq i \leq 1000\}. \quad (2)$$

To express the target scenario in the context of HR images, let \mathcal{H} denote the set of images of various sizes $h \times w$ and \mathcal{R} denote the set of images of size natively adapted to \mathcal{C} , for instance 224×224 for the specific CNN considered in Section 4. The only assumption on the size of an image $\in \mathcal{H}$ is to be larger than the CNNs input size. One assumes given a fixed *degradation function*

$$\rho : \mathcal{H} \longrightarrow \mathcal{R}, \quad (3)$$

that transforms any image \mathcal{I} of \mathcal{H} into an image $\rho(\mathcal{I})$ of \mathcal{R} . The well-defined composition of maps

$$\begin{array}{ccc} \mathcal{H} & \xrightarrow{\rho} & \mathcal{R} \\ & \searrow \mathcal{C} \circ \rho & \downarrow \mathcal{C} \\ & & \mathcal{V} \end{array} \quad (4)$$

allows \mathcal{C} to classify, in particular, the reduced image $\mathcal{A}_a = \rho(\mathcal{A}_a^{\text{hr}}) \in \mathcal{R}$ in some class c_a , with τ_a being the c_a -label value outputted by \mathcal{C} for \mathcal{A}_a , so that $\mathcal{C}(\mathcal{A}_a) = (c_a, \tau_a)$.

In this context, an adversarial HR image for the (c_a, c_t) target scenario performed on $\mathcal{A}_a^{\text{hr}} \in \mathcal{H}$ is an image $\mathcal{D}_t^{\text{hr}}(\mathcal{A}_a^{\text{hr}}) \in \mathcal{H}$ satisfying the two following 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 adversarial $\mathcal{D}_t^{\text{hr}}(\mathcal{A}_a^{\text{hr}})$ HR images. On the other hand, \mathcal{C} should classify the reduced adversarial image $\mathcal{D}_t(\mathcal{A}_a^{\text{hr}}) = \rho(\mathcal{D}_t^{\text{hr}}(\mathcal{A}_a^{\text{hr}}))$ in the category c_t for a sufficiently convincing c_t -label value. The *target scenario* (c_a, c_t) performed on the HR image $\mathcal{A}_a^{\text{hr}}$ can be visualized by the following scheme:

$$\begin{array}{ccc} \mathcal{A}_a^{\text{hr}} \in \mathcal{H} & \text{-----} \triangleright & \mathcal{D}_t^{\text{hr}}(\mathcal{A}_a^{\text{hr}}) \in \mathcal{H} \\ \downarrow \rho & & \downarrow \rho \\ \mathcal{A}_a \in \mathcal{R} & & \mathcal{D}_t(\mathcal{A}_a^{\text{hr}}) \in \mathcal{R} \\ \downarrow \mathcal{C} & & \downarrow \mathcal{C} \\ (c_a, \tau_a) \in \mathcal{V} & & (c_t, \tau_t) \in \mathcal{V} \end{array} \quad (5)$$

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

3. Attack strategy for the target scenario on HR images

We present here a strategy that attempts to circumvent the three challenges about speed, adversity and visual quality cited in the Introduction.

In a nutshell, the first step consists in getting an image in \mathcal{R} that is adversarial against the image $\mathcal{A}_a \in \mathcal{R}$ reduced from $\mathcal{A}_a^{\text{hr}} \in \mathcal{H}$. Although getting such adversarial images in the \mathcal{R} domain is crucial for obvious reasons, the strategy does not depend on how they are obtained. It applies to all possible attacks that work efficiently in the \mathcal{R} domain. This feature contributes substantially to its flexibility. In a second step, one *lifts* this low-resolution adversarial image up to an HR image, called here the *HR tentative adversarial image*. In the last step, one checks whether this HR tentative adversarial image fulfils the criteria stated in the last paragraph of Section 2.2, namely becomes adversarial once reduced. An HR tentative adversarial image that does so is an HR *good enough adversarial image* or a τ -*strong adversarial image*, depending on the outcome of \mathcal{C} for its reduced version in the \mathcal{R} domain.

3.1. Construction of adversarial images in \mathcal{H}

The starting point is a large size image $\mathcal{A}_a^{\text{hr}} \in \mathcal{H}$, and its reduced image $\mathcal{A}_a = \rho(\mathcal{A}_a^{\text{hr}}) \in \mathcal{R}$, classified by \mathcal{C} as belonging to a category c_a .

For Step 1, one assumes given an image $\tilde{\mathcal{D}}_{t, \tilde{\tau}_t}(\mathcal{A}_a^{\text{hr}}) \in \mathcal{R}$, 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 exceeding a threshold $\tilde{\tau}_t$. As already stated, it does not matter how such an adversarial image is obtained.

To perform Step 2, one needs a fixed *enlarging function*

$$\lambda : \mathcal{R} \longrightarrow \mathcal{H} \quad (6)$$

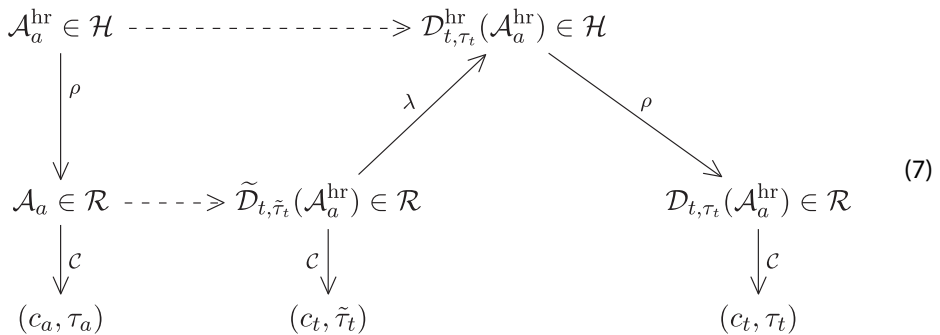
that transforms any image of \mathcal{R} into an image in \mathcal{H} . Anticipating on Step 3, it is worthwhile noting that, although the *reduction function* ρ and the *enlarging function* λ have opposite purposes, these functions are not necessarily inverse one from the other. In other words, $\rho \circ \lambda$ and $\lambda \circ \rho$ may differ from the identity maps $id_{\mathcal{R}}$ and $id_{\mathcal{H}}$ respectively (usually they do differ).

One applies the enlarging function λ to the low-resolution adversarial $\tilde{\mathcal{D}}_{t, \tilde{\tau}_t}(\mathcal{A}_a^{\text{hr}}) \in \mathcal{R}$ to obtain the HR tentative adversarial image $\mathcal{D}_{t, \tau_t}^{\text{hr}}(\mathcal{A}_a^{\text{hr}}) = \lambda(\tilde{\mathcal{D}}_{t, \tilde{\tau}_t}(\mathcal{A}_a^{\text{hr}})) \in \mathcal{H}$.

For Step 3, the application of the reduction function ρ on this HD tentative adversarial image creates an image $\mathcal{D}_{t, \tau_t}(\mathcal{A}_a^{\text{hr}}) = \rho(\mathcal{D}_{t, \tau_t}^{\text{hr}}(\mathcal{A}_a^{\text{hr}}))$ in the \mathcal{R} domain. One runs \mathcal{C} on $\mathcal{D}_{t, \tau_t}(\mathcal{A}_a^{\text{hr}})$ to get its classification, in the hope to obtain a classification in c_t .

The attack succeeds if \mathcal{C} classifies this image in c_t , potentially for a c_t -label value 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 $\mathcal{D}_{t, \tau_t}^{\text{hr}}(\mathcal{A}_a^{\text{hr}})$ in the \mathcal{H} domain.

Scheme 7 essentially summarizes the different steps encountered so far:



3.2. Indicators: the loss function \mathcal{L} and L_2 distances

Although both $\tilde{\mathcal{D}}_{t,\tilde{\tau}_t}(\mathcal{A}_a^{\text{hr}})$ and $\mathcal{D}_{t,\tau_t}(\mathcal{A}_a^{\text{hr}})$ 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 $\rho \circ \lambda \neq id_{\mathcal{R}}$ actually. This fact has two consequences that affect the design of our attack and clarify the adjustment described below.

On the one hand, it justifies the necessity of the verification process performed in Step 3 on the HR tentative adversarial image, namely to check whether its reduction indeed belongs to c_t . On the other hand, should it be the case, it implies as well that $\tilde{\tau}_t$ and τ_t differ. It is then natural to define the real-valued *loss function* \mathcal{L} for a given $\mathcal{A}_a^{\text{hr}} \in \mathcal{H}$ as

$$\mathcal{L}(\mathcal{A}_a^{\text{hr}}) = \tilde{\tau}_t - \tau_t \quad (8)$$

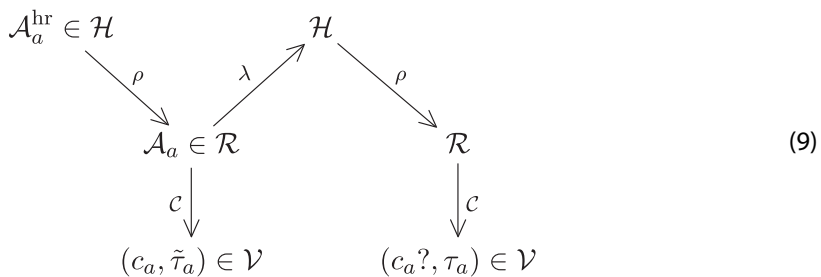
Our attack is effective if one can set accurately the value of $\tilde{\tau}_t$ to match the inequality $\tau_t \geq \tau$ for the threshold value τ , or to make sure that $\mathcal{D}_{t,\tau_t}(\mathcal{A}_a^{\text{hr}})$ is a good enough adversarial image in the \mathcal{R} domain, while controlling the distance variations between $\mathcal{A}_a^{\text{hr}}$ and the adversarial $\mathcal{D}_{t,\tau_t}^{\text{hr}}(\mathcal{A}_a^{\text{hr}})$. For this, one needs to assess the statistical behaviour of the loss function \mathcal{L} on the one hand, and of the L_2 distance of a series of images on the other hand.

Indeed, while the loss function, that measures differences of values coming from images in the \mathcal{R} domain, assesses the objective of getting an image in the \mathcal{H} domain that fools the CNN, other indicators assess the objective of the visual proximity between images for a human eye. Therefore, one computes the L_2 distance of 4 pairs of images. The value of $L_2(\mathcal{A}_a^{\text{hr}}, \mathcal{D}_{t,\tau_t}^{\text{hr}}(\mathcal{A}_a^{\text{hr}}))$, actually the most important one, is between images that live in the \mathcal{H} domain. The values of $L_2(\mathcal{A}_a, \tilde{\mathcal{D}}_{t,\tilde{\tau}_t}(\mathcal{A}_a^{\text{hr}}))$, $L_2(\mathcal{A}_a, \mathcal{D}_{t,\tau_t}(\mathcal{A}_a^{\text{hr}}))$ and $L_2(\tilde{\mathcal{D}}_{t,\tilde{\tau}_t}(\mathcal{A}_a^{\text{hr}}), \mathcal{D}_{t,\tau_t}(\mathcal{A}_a^{\text{hr}}))$ are for images that all live in the \mathcal{R} domain.

The values of these quantities, and therefore the performances and adequacy of the resized adversarials to the addressed problem, clearly depend on the reducing and enlarging functions ρ and λ selected in the scheme.

3.3. Static tests with non-adversarial images natively in \mathcal{H}

To find out which functions ρ and λ are the most appropriate, we designed a series of tests with promising candidates. These *static* tests, called that way since they are performed with non-adversarial images, are convenient to evaluate the candidates. Scheme 9 shows the path of the test performed with an image $\mathcal{A}_a^{\text{hr}} \in \mathcal{H}$ as starting point, knowing that the test is performed with different ancestor images in \mathcal{H} , and the results are averaged among all trials.



First $\mathcal{A}_a^{\text{hr}}$ is reduced to an image $\mathcal{A}_a \in \mathcal{R}$, thanks to the reduction function ρ . One obtains the classification $(c_a, \tilde{\tau}_a) = \mathcal{C} \circ \rho(\mathcal{A}_a^{\text{hr}})$. Then one resizes \mathcal{A}_a first up with λ then down with ρ . One gets the classification of the resulting image $\mathcal{C} \circ \rho \circ \lambda(\mathcal{A}_a) = (c_a?, \tau_a)$, where τ_a is the c_a -label value, whether the resized image is classified to c_a or not. Note that the resized non-adversarial image obtained that way is likely to be classified in c_a . Still, the design of the test cannot make this assumption *a priori*.

One evaluates the value of the loss function $\mathcal{L}(\mathcal{A}_a^{\text{hr}}) = \tilde{\tau}_a - \tau_a$, and of the distance function $L_2(\mathcal{A}_a^{\text{hr}}, \lambda \circ \rho(\mathcal{A}_a^{\text{hr}}))$.

This latter value with images in \mathcal{H} gives a hint at a lower bound on the expected L_2 distance between $\mathcal{A}_a^{\text{hr}}$ and the adversarial image in the \mathcal{H} domain our strategy is aiming at. By construction, it is indeed unlikely that an adversarial in the \mathcal{H} domain could be closer to $\mathcal{A}_a^{\text{hr}}$ than $\lambda \circ \rho(\mathcal{A}_a^{\text{hr}})$ will be. Therefore the L_2 distance of an HR adversarial to $\mathcal{A}_a^{\text{hr}}$ is likely to be $\geq L_2(\mathcal{A}_a^{\text{hr}}, \lambda \circ \rho(\mathcal{A}_a^{\text{hr}}))$, what makes this latter evaluation relevant.

4. Feasibility study

The feasibility study is performed with the 10 CNNs trained on ImageNet shown in [Table 1](#) (this table also gives additional information about the parameters and accuracy of these CNNs), and with the 10 HR images $\mathcal{A}_1^{\text{hr}}, \dots, \mathcal{A}_{10}^{\text{hr}}$ shown in [Table 2](#). Out of them, 8 are taken from the Internet (under Creative Commons Licenses) and 2 are images from the French artist Speedy Graphito (pictured in SpeedyGraphito, [2020](#), the corresponding files were graciously provided by the artist).

[Table 2](#) gives the size of each original HR image, the category c_a and the c_a -label value outputted by VGG-16 for $\mathcal{A}_a^{\text{hr}}$. It also provides the target category c_t , chosen at random among the categories $\neq c_a$ of ImageNet, that is used for the target scenario (c_a, c_t) to perform on each $\mathcal{A}_a^{\text{hr}}$. [Table A1](#) (in Appendix 1) completes [Table 2](#) by providing, for each CNN, the corresponding c_a -categories and label values (all for the Lanczos interpolation method, as explained in [Section 4.1](#)).











One interest of adding the two specific artistic images is that, while a human may have difficulties in classifying them in any category, the CNNs do it, although with relatively small label values (see [Table 2](#) for VGG-16 and [Table A1](#) in Appendix 1 in general).

We run the static tests to select the ρ and λ functions out of 4 candidates ([Section 4.1](#)). Then we briefly describe the evolutionary algorithm $\text{EA}^{\text{target}, \mathcal{C}}$ that we shall use as a black-

Table 1. The 10 CNNs trained on ImageNet, their number of parameters (in millions) and their Top-1 and Top-5 accuracy.

\mathcal{C}_k	Name of the CNN	Parameters	Top-1 accuracy	Top-5 accuracy
\mathcal{C}_1	DenseNet121	8M	0.750	0.923
\mathcal{C}_2	DenseNet169	14M	0.762	0.932
\mathcal{C}_3	DenseNet201	20M	0.773	0.936
\mathcal{C}_4	MobileNet	4M	0.704	0.895
\mathcal{C}_5	NASNetMobile	4M	0.744	0.919
\mathcal{C}_6	ResNet50	26M	0.749	0.921
\mathcal{C}_7	ResNet101	45M	0.764	0.928
\mathcal{C}_8	ResNet152	60M	0.766	0.93
\mathcal{C}_9	VGG16	138M	0.713	0.901
\mathcal{C}_{10}	VGG19	144M	0.713	0.900

Table 2. For $1 \leq a \leq 10$, the image $\mathcal{A}_a^{\text{hr}}$ classified by VGG-16 in the category c_a (interpolation = ‘lanczos’).

a	1	2	3	4	5	6	7	8	9	10
c_a	Cheetah	Eskimo Dog	Koala	Lamp Shade	Toucan	Screen	Comic Book	SportsCar	Binder	Coffee Mug
$w \times h$	910 × 604	960 × 640	910 × 607	2462 × 2913	910 × 607	641 × 600	1280 × 800	1280 × 800	1954 × 2011	1740 × 1710
$\mathcal{A}_a^{\text{hr}}$										
	0.95	0.34	0.99	0.53	0.45	0.70	0.49	0.48	0.28	0.08
c_t	poncho	goblet	Weimaraner	weevil	wombat	swing	altar	beagle	triceratops	hamper

box attack against each of the 10 CNNs (Section 4.2). We apply the strategy with the evolutionary algorithm and get the HR adversarial images that fool CNNs for the target scenario with the threshold value set to $\tau = 0.55$ (Section 4.3). Finally, we discuss the visual quality of the obtained HR adversarial images, especially from a human point of view (Section 4.4).

For $1 \leq a \leq 10$, the HR ancestor image $\mathcal{A}_a^{\text{hr}}$, its resized version $\lambda \circ \rho(\mathcal{A}_a^{\text{hr}}) \in \mathcal{H}$ obtained by the static tests (Section 3.3), and one sample of an adversarial image $\mathcal{D}_{t, \tau}^{\text{hr}}(\mathcal{A}_a^{\text{hr}}) \in \mathcal{H}$ per (c_a, c_t) combination of the target scenario performed on VGG-16, can be retrieved from <https://github.com/aliotopal/HRadversImgs/blob/main/original-advers.md>.

4.1. Selection of ρ and λ

To select the functions ρ and λ , we evaluate four interpolation methods that convert an image from one scale to another. The Nearest Neighbour (Patel & Mistree, 2013), the Bilinear method (Agrafiotis, 2014), the Bicubic method (Keys, 1981) and the Lanczos method (Duchon, 1979; Parsania & Virparia, 2016) are non-adaptive methods among the most common interpolation algorithms, with the additional advantage of being available in python libraries.

The static tests designed in Section 3.3 are performed on the 10 HR images of Table 2 with the 10 CNNs of Table 1 for all 16 possible ρ and λ combinations coming from this selection. Figure 3 summarizes the results in two heatmaps (see Figure A1 in Appendix 1 for individual heatmaps per CNN). They represent the average values (for all CNNs) of the loss function $\mathcal{L}^C(\mathcal{A}_a^{\text{hr}}) = \tilde{\tau}_a - \tau_a$ (Figure 3(a)), and of $L_2(\mathcal{A}_a^{\text{hr}}, \lambda \circ \rho(\mathcal{A}_a^{\text{hr}}))$ (Figure 3(b), the two images being in \mathcal{H}).

Figure 3(a) shows that the best performing loss value, namely 0.039 (which is quite close to the optimal 0 value), is achieved when the images are scaled down with the Bicubic method and up with the Lanczos method (observe that the Nearest Neighbour method is the default upsizing and downsizing method in Keras).

However, Figure 3(b) shows that this combination for (ρ, λ) gives the second best L_2 distance while $(\rho, \lambda) = (\text{Lanczos}, \text{Lanczos})$ gives the best. Additionally, Figure 3(a) shows that the loss achieved by the (Lanczos, Lanczos) combination is the fourth best performing combination and remains very moderate.

Since human visual quality of the adversarials in the \mathcal{H} domain should prevail, especially at a very tolerable cost in terms of the Loss function, we select $(\rho, \lambda) = (\text{Lanczos}, \text{Lanczos})$. This choice is used in all further experiments.

4.2. The evolutionary algorithm $EA^{\text{target}, \mathcal{C}}$

For $\mathcal{C} = \mathcal{C}_1, \dots, \mathcal{C}_{10}$, the adversarial images are obtained with the evolutionary algorithm $EA^{\text{target}, \mathcal{C}}$ detailed in Chitic, Bernard et al. (2020), Chitic, Leprévost et al. (2020), and Topal et al. (2022), whose pseudo code is given in Algorithm 1. Throughout the different generations, the size of the population is constant and is set to 40 as a result of a series of experiments. The algorithm starts with 40 identical copies of the ancestor image. The objective of evolving an individual ind towards an image classified as c_t is encoded in the fitness function

$$fit(ind) = \mathbf{o}_{ind}^{\mathcal{C}}[t]. \quad (10)$$

Throughout the evolution, the individuals are continuously mutated and recombined to create population members with larger fitness.

Algorithm 1: EA attack pseudocode

- 1: **Input:** CNN \mathcal{C} , ancestor \mathcal{A} , perturbation magnitude a , 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 l_0 as first individual;
 - 3: Compute fitness for each individual;
 - 4: While $(o_{i_0}[t] < \tau)$ & $x < X$ 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 new population;
 - 8: Evaluate fitness of each individual;
-

The maximum pixel modification on individuals is limited to a fixed range $\epsilon = [-16, 16]$ throughout the search process to maintain the proximity of the evolved images with the ancestor image. The step size per selected pixel is set to $\alpha = \pm 1$. The individuals compete with each other until one of the EA's stop conditions is satisfied, namely until one individual satisfies $\mathbf{o}_{ind}^{\mathcal{C}}[t] \geq \tau$ (what is called a successful run), or the maximum number of generations $X = 35,000$ is reached.

$\rho \backslash \lambda$	Nearest Neighbor	Bicubic	Bilinear	Lanczos
Nearest Neighbor	0.090	0.093	0.100	0.092
Bicubic	0.051	0.046	0.059	0.039
Bilinear	0.063	0.059	0.076	0.056
Lanczos	0.068	0.063	0.077	0.058

(a)

$\rho \backslash \lambda$	Nearest Neighbor	Bicubic	Bilinear	Lanczos
Nearest Neighbor	49958	42155	39029	43277
Bicubic	42400	37542	39099	36958
Bilinear	42899	39210	40866	38560
Lanczos	42656	37113	38583	36564

(b)

Figure 3. The overall average values of the loss functions $\mathcal{L}^{\mathcal{C}}(\mathcal{A}_a^{\text{hr}}) = \tilde{\tau}_a - \tau_a$ in Table (a) and of $L_2(\mathcal{A}_a^{\text{hr}}), \lambda \circ \rho(\mathcal{A}_a^{\text{hr}})$ in Table (b).

4.3. Running the strategy to get adversarial images with the EA

With the rescaling functions $(\rho, \lambda) = (\text{Lanczos}, \text{Lanczos})$, we deploy the strategy detailed in Section 3.1 with the evolutionary algorithm $\text{EA}^{\text{target}, \mathcal{C}}$ for the 10 CNNs and the 10 ancestor images $\mathcal{A}_a^{\text{hr}}$. With terminology consistent with Section 2.1, the goal is to create 0.55-strong HR adversarial images as well as good enough HR adversarial images for the target scenario (c_a, c_t) specified in Table 2 (see also Table A1).

Since different seed values for the EA may lead to different results, we increased the robustness of the outcomes by performing 10 independent runs with random seeds for each (c_a, c_t) pair and ancestor $\mathcal{A}_a^{\text{hr}}$, leading to altogether 100 trials per CNN, hence to 1000 trials altogether.

90% of the runs terminated successfully in less than 35,000 generations. The detailed success rate for each CNN is shown in Table A2 (Appendix 1).

For each CNN, Table 3 gives the average of four indicators, computed over the successful runs for the specific CNN considered. $\text{avgGen}_C^{0.55}$ is the average number of generations required to obtain the 0.55-strong adversarial images $\mathcal{D}_{t, \tau_t}^{\text{hr}}(\mathcal{A}_a^{\text{hr}}) \in \mathcal{H}$, $\text{avgGen}_C^{\text{ge}}$ is the average number of generations required to obtain *good enough adversarial* HR images $\mathcal{D}_{t, \tau_t}^{\text{hr, ge}}(\mathcal{A}_a^{\text{hr}})$ while being on the way to 0.55-strong adversarial images, and $\text{avg}_{C, \tau_t}^{\text{ge}}$ is their average c_t -label values. The last indicator $\text{AddE}_{C, \text{ge}}^{0.55}$ shows the additional effort to move up from a *good enough* HR adversarial image, to a 0.55-strong HR adversarial image, measured as a percentage assessing the proportion of additional generations required.

The three last columns of Table 3 contain the average computational time per generation (avgTime_C , in second), the average total computational time required to create a good enough adversarial image ($\text{avgTime}_C^{\text{ge}}$, in minutes) and the average total computational time required to create a 0.55-strong adversarial image ($\text{avgTime}_C^{0.55}$, in minutes).

Out of the 900 successful trials from 1000 attempts, Table 3 shows that, on average, *good enough* HR adversarial images are created by our algorithm in 5954 generations and 0.55-strong HR adversarial images in 8314 generations (of course with large variations, depending on the CNN considered). Measured by the number of additional generations required, the effort necessary to move up from a *good enough* HR adversarial image, that has a c_t -label value of 0.163 in average, to a 0.55-strong HR adversarial image is 39.6%.

In terms of the average computational time (on the hardware specified at the beginning of this article), roughly 57 minutes were necessary to create a *good enough*

Table 3. Average performance over the successful runs of $\text{EA}^{\text{target}, \mathcal{C}}$ for each \mathcal{C} trained on ImageNet in creating 0.55-strong and *good enough* HR adversarial images for the target scenario (c_a, c_t) performed on $\mathcal{A}_a^{\text{hr}}$.

	CNNs	$\text{avgGen}_C^{\text{ge}}$	$\text{avg}_{C, \tau_t}^{\text{ge}}$	$\text{avgGen}_C^{0.55}$	$\text{AddE}_{C, \text{ge}}^{0.55}$	avgTime_C	$\text{avgTime}_C^{\text{ge}}$	$\text{avgTime}_C^{0.55}$
\mathcal{C}_1	DenseNet121	4561	0.150	7765	70.2	0.532	40.5	68.9
\mathcal{C}_2	DenseNet169	8112	0.241	11221	38.3	0.608	82.2	113.7
\mathcal{C}_3	DenseNet201	5288	0.166	8077	52.7	0.609	53.7	82.0
\mathcal{C}_4	MobileNet	4201	0.191	5640	34.9	0.510	35.7	48.0
\mathcal{C}_5	NASNetMobile	10765	0.224	12981	20.6	0.550	98.8	119.2
\mathcal{C}_6	ResNet50	4336	0.142	5891	35.9	0.575	41.6	56.5
\mathcal{C}_7	ResNet101	6261	0.151	8656	38.3	0.578	60.4	83.5
\mathcal{C}_8	ResNet152	6268	0.143	8477	35.2	0.649	67.8	91.8
\mathcal{C}_9	VGG16	4069	0.112	6250	53.6	0.567	38.5	59.1
\mathcal{C}_{10}	VGG19	5683	0.109	8180	43.9	0.570	54.0	77.7
Overall Avg.		5954	0.163	8314	39.6	0.575	57.3	80.7

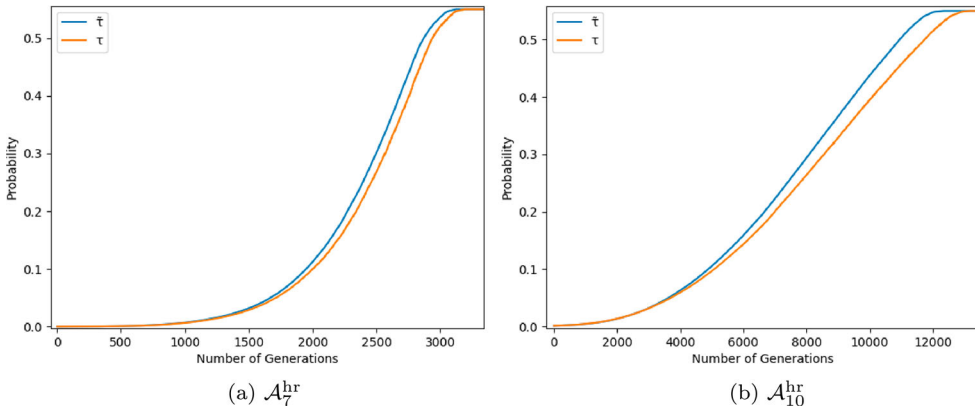


Figure 4. Convergence characteristics for τ_t and $\tilde{\tau}_t$ for $\mathcal{A}_7^{\text{hr}}$ (a) and $\mathcal{A}_{10}^{\text{hr}}$ (b) of $\text{EA}^{\text{target},\mathcal{C}}$ for $\mathcal{C} = \text{VGG16}$.

adversarial image, and 80 minutes for a 0.55-*strong adversarial image*, again with large variations from one CNN to another.

For each ancestor image $\mathcal{A}_a^{\text{hr}}$ for which the algorithm succeeds at least once, one computes the convergence characteristics of the algorithm $\text{EA}^{\text{target},\mathcal{C}}$ for $\tilde{\tau}_t$ and for τ_t on the way to the HR 0.55-*strong adversarial image* $\mathcal{D}_{t,\tau}^{\text{hr}}(\mathcal{A}_a^{\text{hr}})$.

An example, representative of the overall behaviour (see Appendix 1, [Figures A2 and A3](#)), is given for VGG-16 in [Figure 4](#) for $\mathcal{A}_7^{\text{hr}}$, and for $\mathcal{A}_{10}^{\text{hr}}$, where the graphs are capped on the horizontal axis at their respective $\text{avgGens}_{\mathcal{C}_9}^{0.55}$ values.

[Table 4](#) completes the information provided by the convergence graphs. It gives the average, over the successful among the 10 independent runs per ancestor image, of the minimum and maximum values of the loss function $\mathcal{L}(\mathcal{A}_a^{\text{hr}}) = \tilde{\tau}_t - \tau_t$.

A thorough study of the loss function as the algorithm proceeds, generation for generation, towards the construction of the HR 0.55-*strong adversarial image* $\mathcal{D}_{t,\tau}^{\text{hr}}(\mathcal{A}_a^{\text{hr}})$, shows the following outcome, at least for the successful runs performed in this study (see Appendix 2, [Figure A4](#) for one detailed example). During the first generations, the values of the loss function are alternatively positive and negative, and remain very small, typically of order 10^{-4} . Then, at some point, namely from some generation on (that differs from one HR ancestor image to another, and from one CNN to another as well), the loss function becomes ≥ 0 , and remains so until the algorithm terminates.

Table 4. Average of the minimum and maximum values of $\mathcal{L}(\mathcal{A}_a^{\text{hr}}) = \tilde{\tau}_t - \tau_t$.

CNNs		Avg. Loss $_{\mathcal{C}}$ (min)	Avg. Loss $_{\mathcal{C}}$ (max)
\mathcal{C}_1	DenseNet121	-2.09E-04	2.87E-01
\mathcal{C}_2	DenseNet169	-3.96E-05	3.58E-01
\mathcal{C}_3	DenseNet201	-1.28E-05	3.25E-01
\mathcal{C}_4	MobileNet	-4.50E-06	3.32E-01
\mathcal{C}_5	NASNetMobile	-2.89E-06	3.48E-01
\mathcal{C}_6	ResNet50	-2.45E-05	2.18E-01
\mathcal{C}_7	ResNet101	-2.31E-05	2.13E-01
\mathcal{C}_8	ResNet152	-1.53E-05	1.96E-01
\mathcal{C}_9	VGG16	-7.05E-04	3.94E-02
\mathcal{C}_{10}	VGG19	-1.30E-03	4.00E-02
Overall Avg.		-2.34E-04	2.32E-01

Moreover, although some slight fluctuations occur, the asymptotic behaviour of the loss function is to almost strictly grow from there on.

A consequence of the convergence graphs given in Figures A2 and A3 and of the numerical values given in Table 4 is that setting a threshold c_t -label value $\tilde{\tau}_t = \tau_t + \text{Avg.Loss}_{\mathcal{C}}(\max)$ seems a reasonable choice, at least if one aims at getting 0.55-*strong* HR adversarial images by our method. A safer choice would be to add a value exceeding slightly the absolute maximum value of the loss function among all such values for all 10 ancestor images. For VGG-16 for instance, it would mean to set the threshold c_t -label value to $\tilde{\tau}_t = \tau_t + 0.065$ since the largest \mathcal{L}_{\max} value is 0.064 for that CNN. However, for some CNNs, these values vary largely from one ancestor image to another, so that, in a first approach, we would recommend to add the average loss function instead.

4.4. Visual quality

We first assess numerically the quality of the obtained HR adversarial images as compared to the HR ancestors. Table 5 gives the three L_2 differences of images in the \mathcal{R} domain, namely $L_2^1 = L_2(\mathcal{A}_a, \tilde{\mathcal{D}}_{t,\tilde{\tau}_t}(\mathcal{A}_a))$, $L_2^2 = L_2(\mathcal{A}_a, \mathcal{D}_{t,\tau_t}(\mathcal{A}_a))$, and $L_2^3 = L_2(\tilde{\mathcal{D}}_{t,\tilde{\tau}_t}(\mathcal{A}_a), \mathcal{D}_{t,\tau_t}(\mathcal{A}_a))$, and the L_2 difference (in the \mathcal{H} domain) $L_2^4 = L_2(\mathcal{A}_a^{\text{hr}}, \mathcal{D}_{t,\tau_t}^{\text{hr}}(\mathcal{A}_a^{\text{hr}}))$.

The most saying outcome of Table 5 is that the average value of the L_2 distance between the HR ancestor and adversarial images remains comparable, actually even smaller, than the corresponding value (namely for Lanczos–Lanczos) measured for non-adversarial images in the heatmap in Figure 3(b). In other words, at least in average, our attack does not arm the numerical performance of the resizing functions. It even enhances it, what is probably due to some statistical artefact.

Still, the ‘true’ visual quality for a human eye is assessed by looking at some representative examples either from some distance, or by zooming on some areas.

For instance, let us consider the HR ancestor image $\mathcal{A}_7^{\text{hr}}$ represented in Figure 5(a), and a zoom of that picture on some restricted area (taken at random). Figure 5(b) shows the non-adversarial resized image $\lambda \circ \rho(\mathcal{A}_7^{\text{hr}})$ with $(\lambda, \rho) = (\text{Lanczos}, \text{Lanczos})$. Finally, Figure 5(c) shows the HR 0.55-*strong* adversarial image created by $\text{EA}^{\text{target},\mathcal{C}}$ for $\mathcal{C} = \text{VGG-16}$. To further illustrate the phenomenon, we proceed similarly (still for VGG-16) with another ancestor HR image, namely $\mathcal{A}_{10}^{\text{hr}}$ in Figure 6(a–c).

At some distance, both the non-adversarial resized original image and the HR adversarial seem to have a good visual quality as compared to the HR ancestor. However, the zoomed

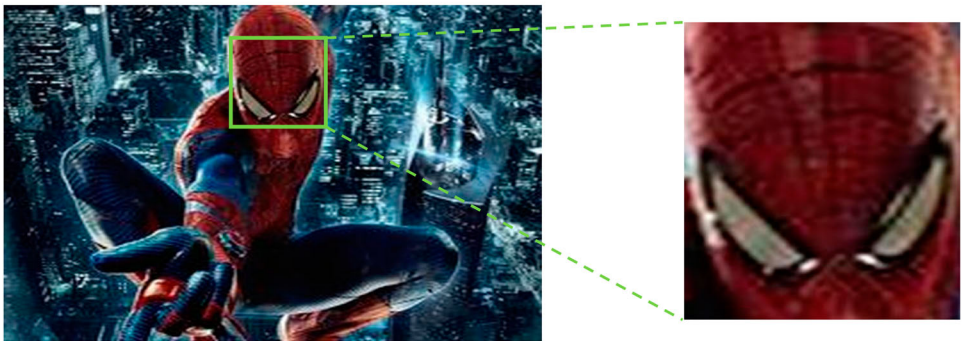
Table 5. The three distances L_2^1 , L_2^2 , and L_2^3 of images in the \mathcal{R} domain, and the distance L_2^4 in the \mathcal{H} domain.

CNNs		L_2^1	L_2^2	L_2^3	L_2^4
\mathcal{C}_1	DenseNet121	2357	2266	4096	28112
\mathcal{C}_2	DenseNet169	2122	2204	1529	33355
\mathcal{C}_3	DenseNet201	2392	2468	1593	35439
\mathcal{C}_4	MobileNet	2182	2255	1463	33437
\mathcal{C}_5	NASNetMobile	2610	2562	1641	28501
\mathcal{C}_6	ResNet50	2631	2485	1426	28040
\mathcal{C}_7	ResNet101	2724	2620	1626	34568
\mathcal{C}_8	ResNet152	2771	2649	1665	34683
\mathcal{C}_9	VGG16	3211	2951	1485	35424
\mathcal{C}_{10}	VGG19	3227	3009	1490	35428
Overall Avg.		2623	2547	1801	32699

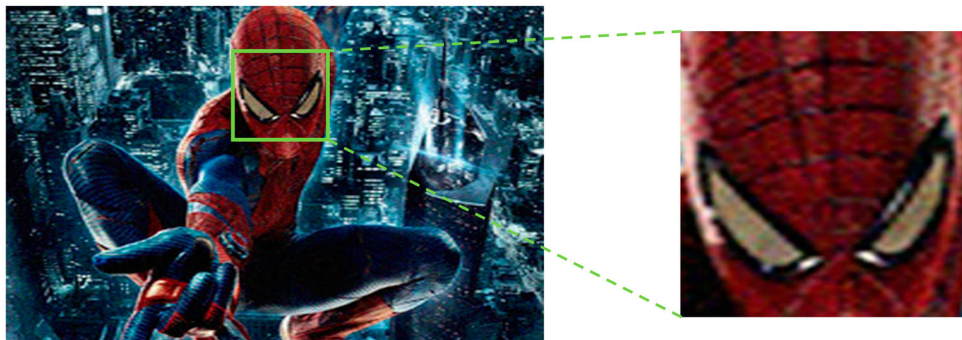
areas show that details from the HR ancestor images become blurry for a human eye, not only in the HR adversarial images (as seen from Figures 5c and 6c) but in the non-adversarial resized images as well (as seen from Figures 5 b and 6b). Moreover, a human eye is not able



(a)

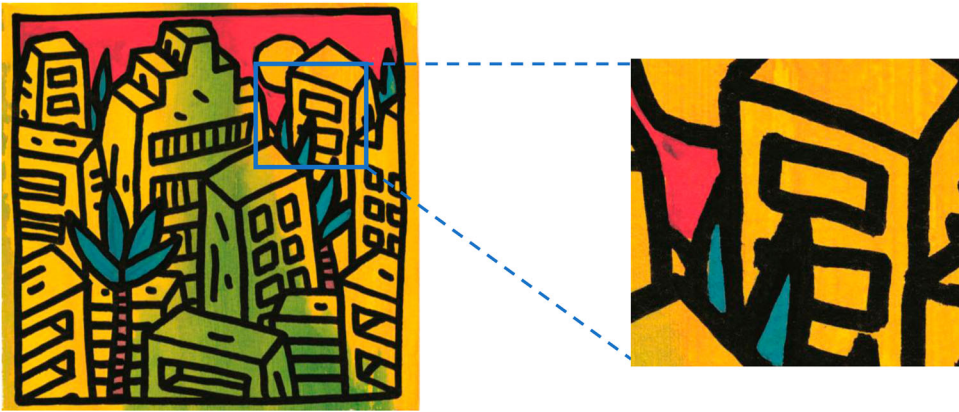


(b)

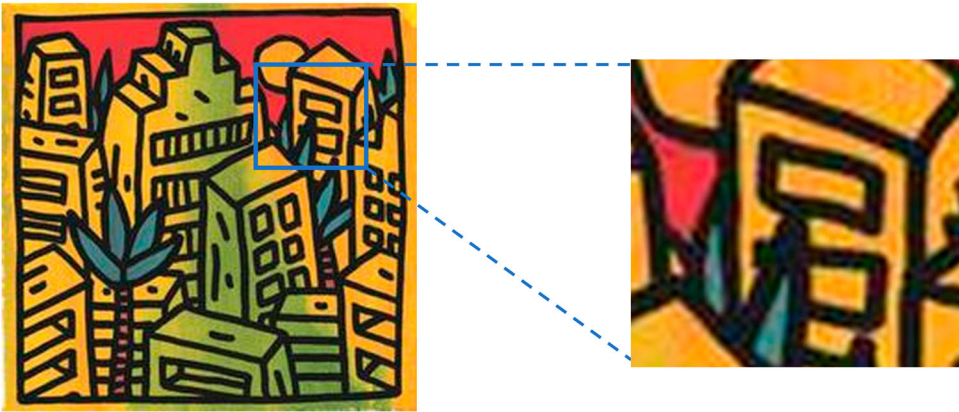


(c)

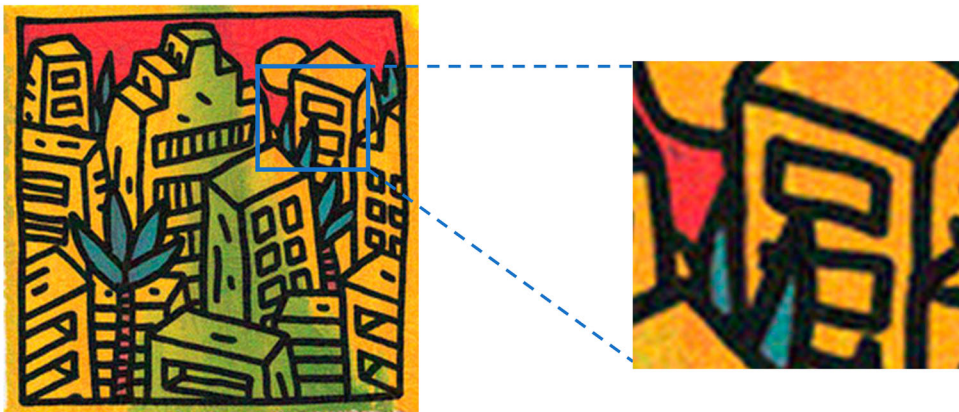
Figure 5. Visual comparison in the \mathcal{H} domain of $\mathcal{A}_7^{\text{hr}}$ (a) with its non-adversarial resized version (b) and its adversarial obtained by $\text{EA}^{\text{target}, \mathcal{C}}$ for $\mathcal{C} = \text{VGG-16}$. (a) $\mathcal{A}_7^{\text{hr}}$, (b) $\lambda \circ \rho(\mathcal{A}_7^{\text{hr}})$, (c) $\mathcal{D}_7^{\text{hr}}(\mathcal{A}_7^{\text{hr}})$.



(a)



(b)



(c)

Figure 6. Visual comparison in the \mathcal{H} domain of $\mathcal{A}_{i_0}^{\text{hr}}$ (a) with its non-adversarial resized version (b) and its adversarial obtained by $\text{EA}^{\text{target}, \mathcal{C}}$ for $\mathcal{C} = \text{VGG-16}$. (a) $\mathcal{A}_{i_0}^{\text{hr}}$, (b) $\lambda \circ \rho(\mathcal{A}_{i_0}^{\text{hr}})$, (c) $\mathcal{D}_{\tau}^{\text{hr}}(\mathcal{A}_{i_0}^{\text{hr}})$.

to distinguish the blurriness that occurs in the non-adversarial resized image from the one that shows up in the HR adversarial: The loss of details looks the same in both cases.

This experiment, representative of the general behaviour over the CNNs, shows that the observed blurry effect is not due to an inefficiency of our strategy, nor of the algorithm $EA^{\text{target},\mathcal{C}}$, at least to a large extent, but is due to the lack of high-quality interpolation methods. Indeed, these experiments show that scaling up to the \mathcal{H} domain images belonging to the \mathcal{R} domain, adversarial or not, results in a loss of high-frequency features on the up-scaled images. Moreover, the very fact that the loss of details looks the same for a resized non-adversarial image as for the adversarial image created by our algorithm in the \mathcal{H} domain speaks in favour of our attack, since it makes our attack harder to detect.

5. Direct attack in the \mathcal{H} domain

In this last part, we show that a direct attack in the \mathcal{H} domain, that would aim at making effective the top arrow of scheme 5 without applying our indirect strategy, is a non-trivial problem in practice.

Concretely, for each $\mathcal{C} = \mathcal{C}_1 \dots, \mathcal{C}_{10}$, we challenge $EA^{\text{target},\mathcal{C}}$ to perform a direct attack in the \mathcal{H} domain for the most promising (ancestor, target) pair and the corresponding ancestor image $\mathcal{A}_a^{\text{hr}}$, in order to create directly a 0.55-strong HR adversarial image. In all cases, the process stops when either a direct attack turns out to be successful, or if the computing time exceeds 48 hours. The most promising pair, and the corresponding ancestor, is defined as the combination for which the *indirect attack* with the algorithm $EA^{\text{target},\mathcal{C}}$ is the fastest in terms of the number of generations required to succeed.

Computation shows that the (toucan, wombat) pair, with the corresponding ancestor image $\mathcal{A}_5^{\text{hr}}$, is the most promising for $\mathcal{C}_4, \mathcal{C}_9, \mathcal{C}_{10}$, and that the (comic book, altar) pair, with the corresponding ancestor image $\mathcal{A}_7^{\text{hr}}$, is the most promising for the 7 remaining CNNs.

Clearly, $EA^{\text{target},\mathcal{C}}$ goes beyond the previous experiments since it now processes a search space of size 910×607 in the case of $\mathcal{A}_5^{\text{hr}}$ and of 1280×800 in the case of $\mathcal{A}_7^{\text{hr}}$, instead of 224×224 for the indirect attack.

Figure 7 illustrates the convergence characteristics of $EA^{\text{target},\mathcal{C}}$ when working directly in the \mathcal{H} domain, at least for the combinations and ancestor images considered (see Figure A5 in Appendix for all 10 CNNs). Figure 7(a) shows the outcome for $\mathcal{C} = \text{VGG-16}$ when one proceeds with the ancestor image $\mathcal{A}_5^{\text{hr}}$, and Figure 7(b) shows the outcome for $\mathcal{C} = \text{ResNet-152}$ when one proceeds with the ancestor image $\mathcal{A}_7^{\text{hr}}$. The horizontal axis of the graph is the number of generations, capped at what one gets after 48 hours, and the vertical axis is the c_t -label value for the fittest individual.

Although the search space increased by around ‘only’ 11 times for $\mathcal{A}_5^{\text{hr}}$ and 20 times for $\mathcal{A}_7^{\text{hr}}$, the EA was nevertheless unable to create high-resolution adversarial images within 48 hours, as shown in Table A3, Appendix 3. The EA stopped at $\approx 50,000$ generations for the 3 CNNs considered in the former case, at $\approx 28,000$ generations for the 7 CNNs considered in the later case, with the fittest individual obtained still classified by the corresponding CNN as belonging to the ancestor category (toucan or comic book).

More precisely, the c_a -label value of the fittest individual takes values in the range $\approx [0.084, 0.748]$, the actual values depending on the CNN considered. Its target category label value remains very small, culminating at $7.0E-04$ in the best case, achieved by \mathcal{C}_1 , one of the 7 CNNs considered for the (comic book, altar) pair.

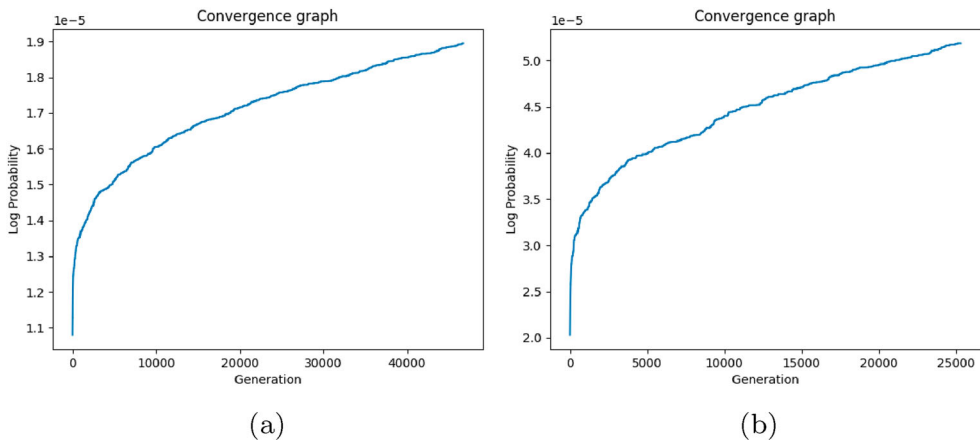


Figure 7. Convergence characteristics of $EA^{\text{target},\mathcal{C}}$ aiming at generating within 48 hours a high-resolution adversarial image by directly evolving (a) $\mathcal{A}_5^{\text{hr}}$ for the (toucan, wombat) pair and $\mathcal{C} = \text{VGG-16}$, and (b) $\mathcal{A}_7^{\text{hr}}$ for the (comic book, altar) and $\mathcal{C} = \text{ResNet-152}$. (a) $\text{VGG-16} - \mathcal{A}_5^{\text{hr}}$. (b) $\text{ResNet-152} - \mathcal{A}_7^{\text{hr}}$.

Although the learning curve of the EA improves (see Figures 7 and A5 in Appendix 3), the c_T -label value of the fittest individual increases by a factor in the range [1.71, 5.5] depending on the considered CNN (see Table A3, Appendix 3), and the critical regions to modify are narrowed down as the EA works, the EA is not fast enough to create images that converge to the target category in reasonable time. Although difficult to assess precisely, our experiments indicate that attacking directly in the \mathcal{H} domain may take weeks or maybe months to succeed. It may also come out that even the threshold c_T -label value of 0.55 may be out of reach in some cases by such a direct attack.

The reasons for this slowness are twofold. On the one hand, a search space of between 11 and 20 times larger than the size 224×224 , for which $EA^{\text{target},\mathcal{C}}$ has proven to be efficient, makes it difficult for the EA to narrow down quickly the regions on which to focus. On the second hand, the average time per generation, that was ≈ 0.575 seconds in the \mathcal{R} domain, is now ≈ 5.74 seconds in the \mathcal{H} domain. Out of the operations purely linked to the EA, Tables A4 and A5 (Appendix 3) show that the most consuming one is the mutation process, and that this operation of the algorithm consumes $3\times$ more time in the \mathcal{H} domain than it used to take in the \mathcal{R} domain. Although with a lesser timing effect, the crossover operation of the algorithm also consumes $3\times$ more time in the \mathcal{H} domain than in the \mathcal{R} domain. This again is due to the size of the images given to the EA.

Therefore, creating high-resolution adversarial images from \mathcal{A}^{hr} directly in the \mathcal{H} domain requires new methods. The results of this section also sustain, in a way, the indirect strategy adopted in this paper to address HR images.

6. Conclusion

Trained CNNs, performing image recognition, convert input images to some fixed and moderate size, say 224×224 for CNNs trained on ImageNet typically. This process transforms the input image into a low-resolution image that the CNN is able to analyse. So far, attacks, aiming at creating adversarial images fooling these CNNs, create some adversarial noise of size equal to the input size of the CNN.

The method presented in this work is the first effective attempt to make the search space for the adversarial noise depend on the size of the original image and not on the CNN's input size. In particular, it is effective for high resolution images in terms of speed, adversity and visual quality.

More specifically, the designed indirect strategy lifts any existing attack, efficient in the low-resolution domain, to an attack that applies in the high resolution domain. We performed an experimental study for 10 CNNs trained on ImageNet, by lifting our evolutionary algorithm-based attack $EA^{\text{target},C}$, with the aim to create HR images adversarial for the target scenario, that these CNNs classify in the target category with confidence ≥ 0.55 . Our algorithm succeeded in 900 cases out of 1000 attempts to create 0.55-strong high resolution adversarial images.

To sustain this indirect strategy, we also showed that attacking directly in the HR domain is not feasible in practice. After 48 computation hours, no HR adversarial image was obtained by the direct attack for any of the 10 CNNs, even for the most promising pairs of target and ancestor categories and corresponding ancestor. *A contrario*, for the 900 successful attempts, our indirect attack succeeded to create 0.55-strong adversarial images within, in average, 48' for the easiest CNN to deceive, and 119' for the hardest CNN to deceive.

While this work successfully addresses the adversity and speed issues, we plan to focus more specifically on the visual quality of the HR adversarial images obtained, by considering alternative scaling functions, such as adaptive interpolation methods ((Hu & Tan, 2017; Hwang & Lee, 2004; Li & Orchard, 2001; Zhang & Wu, 2008)) or ML-base methods ((Schulter et al., 2015; Ye et al., 2020)). Already the mere comparison of these resizing functions for HR non-adversarial images gives a useful benchmark. Additionally, this study could examine whether an HR adversarial, constructed thanks to a specific choice of (ρ, λ) , remains adversarial once reduced via all (or some) different downsizing functions.

Furthermore, we intend to apply our strategy to more CNNs, more HR images (of different nature, e.g. satellite images, medical images, etc), and more attacks, black-box or not. Additionally, we intend to explore how far the very existence of performing our attack can be detected by pre-processing defense mechanisms. Lastly, while the current strategy applies to any existing attack in the \mathcal{R} domain, one can try to take advantage of the outcomes of our study, to design attacks in the \mathcal{H} domain, tailor-made to some existing attacks in the \mathcal{R} domain.

Acknowledgments

The authors express their gratitude to Speedy Graphito and to Bernard Utudjian for the provision of two artistic images used in the feasibility study, and for their interest in this work.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Franck Leprévost is Professor at the University of Luxembourg since 2003. He was its Vice president from 2005 to 2015, in charge of organization, international relations and rankings. He received his

Ph.D. in Mathematics from University Paris 7 in 1992. Before joining the University of Luxembourg, he held academic positions in France (CNRS, Paris, University Joseph Fourier, Grenoble) and Germany (Max Planck-Institut für Mathematik, Bonn, Technische Universität Berlin). He spent a sabbatical year (2016) at Polytech, Saint Petersburg (Russia). His research interests include pure mathematics, security of telecommunication and cryptology, evolutionary algorithms and deep learning, international relations and academic leadership. He is the author of 60 publications and four scientific books.

Ali Osman Topal received his B.S. degree in Electrical and Electronic Engineering from Gaziantep University, Turkey, in 2000, and his M.S. and Ph.D. degrees in Computer Engineering from Epoka University, Tirana, Albania, in 2017. He is currently a researcher, lecturer and post doc in Computer Science at the University of Luxembourg. His research interests lie in the area of artificial intelligence, ranging from theory to implementation. His research on evolutionary algorithms has been published in prestigious journals. He is currently focusing on computer vision, deep learning and XAI.

Elmir Avdusinovic is a second-year bachelor student in Applied Information Technology at the University of Luxembourg. His current focus is on adversarial attacks, deep learning and computer vision.

Raluca Chitic is a third-year Ph.D. student in Computer Science at the University of Luxembourg. She holds a B.Sc. degree in Physics and an M.Sc. degree in Artificial Intelligence. Her current focus is on computer vision, explainable neural networks and evolutionary algorithms.

ORCID

Franck Leprévost  <http://orcid.org/0000-0001-8808-2730>

Ali Osman Topal  <http://orcid.org/0000-0003-0141-4742>

Elmir Avdusinovic  <http://orcid.org/0000-0002-8292-8747>

Raluca Chitic  <http://orcid.org/0000-0003-1113-2343>

References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ...Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*. <https://www.tensorflow.org/> Software available from tensorflow.org.
- Agrafiotis, D. (2014). Chapter 9 – Video error concealment. *Academic Press Library in Signal Processing*, 5(1), 295–321. <https://doi.org/10.1016/B978-0-12-420149-1.00009-0>.
- Andriushchenko, M., Croce, F., Flammarion, N., & Hein, M. (2020). *Square attack: A query-efficient black-box adversarial attack via random search*. *European Conference on Computer Vision*. Springer. doi:10.1007/978-3-030-58592-1_29.
- Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., & Roli, F. (2013). *Evasion Attacks against Machine Learning at Test Time*. In Joint European conference on machine learning and knowledge discovery in databases, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-40994-3_25.
- Blier, L. (2016). *A brief report of the heuritech deep learning meetup 5*. <https://heuritech.wordpress.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/>.
- Carlini, N., & Wagner, D. (2017). *Towards Evaluating the Robustness of Neural Networks*. IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22-26 May 2017, IEEE. doi: 10.1109/SP.2017.49.
- Chitic, R., Bernard, N., & Leprévost, F. (2020). *A proof of concept to deceive humans and machines at image classification with evolutionary algorithms*. In Intelligent information and database systems, 12th Asian conference, ACIIDS 2020, Phuket, Thailand,, March 23–26, 2020. Springer. doi:10.1007/978-3-030-42058-1_39.
- Chitic, R., Leprévost, F., & Bernard, N. (2020). Evolutionary algorithms deceive humans and machines at image classification: An extended proof of concept on two scenarios. *Journal of Information and Telecommunication*, 5(1), 121–143. <https://doi.org/10.1080/24751839.2020.1829388>

- Chitic, R., Topal, A., & Leprévost, F. (2021). Evolutionary algorithm-based images, humanly indistinguishable and adversarial against convolutional neural networks: Efficiency and filter robustness. *IEEE Access*, 9, 160758–160778. <https://doi.org/10.1109/ACCESS.2021.3131255>
- Chollet, F. (2015). *Others Keras*. <https://keras.io>.
- Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Fei-Fei, L. (2009). *The ImageNet image database*. <http://image-net.org>.
- Duchon, C. (1979). Lanczos filtering in one and two dimensions. *Journal of Applied Meteorology and Climatology*, 18(8), 1016–1022. [https://doi.org/10.1175/1520-0450\(1979\)018;1016:LFIQAT;2.0.CO;2](https://doi.org/10.1175/1520-0450(1979)018;1016:LFIQAT;2.0.CO;2)
- Guo, C., Gardner, J., You, Y., Wilson, A., & Weinberger, K. (2019). *Simple black-box adversarial attacks*. International Conference on Machine Learning, Long Beach, California, USA, 9-15 June 2019, PMLR 97:2484-2493.
- Hu, W., & Tan, Y. (2017). *Generating adversarial malware examples for black-box attacks based on GAN*. ArXiv Preprint ArXiv:1702.05983.
- Hwang, J., & Lee, H. (2004). Adaptive image interpolation based on local gradient features. *IEEE Signal Processing Letters*, 11(3), 359–362. <https://doi.org/10.1109/LSP.2003.821718>
- Keys, R. (1981). Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6), 1153–1160. <https://doi.org/10.1109/TASSP.1981.1163711>
- Krizhevsky, A., Nair, V., & Hinton, G. (2009). *CIFAR-10 (Canadian Institute for Advanced Research)*. (0). <http://www.cs.toronto.edu/kriz/cifar.html>.
- Leprévost, F., Topal, A. O., Avdusinovic, E., & Chitic, R. (2022). Strategy and feasibility study for the construction of high resolution images adversarial against convolutional neural networks. In *14th Asian conference, ACIIDS 2022 (Ho Chi Minh City, Vietnam, November 28–30, 2022)* (pp. xx–xx).
- Li, X., & Orchard, M. (2001). New edge-directed interpolation. *IEEE Transactions on Image Processing*, 10(10), 1521–1527. <https://doi.org/10.1109/83.951537>
- Oliphant, T. (2006). *A guide to NumPy*. Trelgol Publishing USA.
- Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z., & Swami, A. (2017). *Practical black-box attacks against machine learning*. In Proceedings of the 2017 ACM on Asia conference on computer and communications security, Abu Dhabi United Arab Emirates, April 2–6, 2017. ACM. <https://doi.org/10.1145/3052973.3053009>.
- Parsania, P., & Virparia, P. (2016). A comparative analysis of image interpolation algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(1), 29–34. <https://doi.org/10.17148/IJARCC>
- Patel, V., & Mistree, K. (2013). A review on different image interpolation techniques for image enhancement. *International Journal of Emerging Technology and Advanced Engineering*, 3(12), 129–133.
- Schulter, S., Leistner, C., & Bischof, H. (2015). *Fast and accurate image upscaling with super-resolution forests*. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 07-12 June 2015. IEEE. doi:10.1109/CVPR.2015.7299003.
- SpeedyGraphito (2020). *Mes 400 coups*. Panoramart.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). *Intriguing properties of neural networks*. ArXiv Preprint ArXiv:1312.6199.
- Topal, A. O., Chitic, R., & Leprévost, F. (2022). One evolutionary algorithm deceives humans and ten convolutional neural networks trained on ImageNet at image recognition. (Under Review).
- Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., & Madry, A. (2018). *Robustness may be at odds with accuracy*. ArXiv Preprint ArXiv:1805.12152.
- Van Rossum, G., & Drake, F. (2009). *Python 3 reference manual*. CreateSpace.
- Walt, S., Schönberger, J., Nunez-Iglesias, J., Boulogne, F., Warner, J., Yager, N., Gouillart, E., & Rajkumar, T. (2014). Contributors Scikit-image image processing in Python. *PeerJ*, 2, e453. <https://doi.org/10.7717/peerj.453>
- Ye, M., Lyu, D., & Chen, G. (2020). Scale-iterative upscaling network for image deblurring. *IEEE Access*, 8, 18316–18325. <https://doi.org/10.1109/Access.6287639>
- Zhang, X., & Wu, X. (2008). Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation. *IEEE Transactions On Image Processing*, 17(6), 887–896. <https://doi.org/10.1109/TIP.2008.924279>

Appendices

Appendix 1

$\rho \backslash \lambda$	Nearest Neighbor	Bicubic	Bilinear	Lanczos
Nearest Neighbor	0.123	0.130	0.118	0.115
Bicubic	0.064	0.061	0.073	0.057
Bilinear	0.024	0.015	0.011	0.023
Lanczos	0.124	0.123	0.132	0.121

(a) \mathcal{C}_1

$\rho \backslash \lambda$	Nearest Neighbor	Bicubic	Bilinear	Lanczos
Nearest Neighbor	0.092	0.084	0.075	0.093
Bicubic	0.074	0.053	0.055	0.061
Bilinear	0.058	0.048	0.058	0.050
Lanczos	0.050	0.041	0.043	0.049

(b) \mathcal{C}_2

$\rho \backslash \lambda$	Nearest Neighbor	Bicubic	Bilinear	Lanczos
Nearest Neighbor	0.095	0.088	0.096	0.086
Bicubic	0.005	-0.001	0.011	-0.009
Bilinear	0.024	0.022	0.039	0.015
Lanczos	0.009	0.002	0.007	-0.003

(c) \mathcal{C}_3

$\rho \backslash \lambda$	Nearest Neighbor	Bicubic	Bilinear	Lanczos
Nearest Neighbor	0.088	0.106	0.123	0.109
Bicubic	0.047	0.064	0.061	0.057
Bilinear	0.052	0.053	0.071	0.055
Lanczos	0.087	0.095	0.084	0.091

(d) \mathcal{C}_4

$\rho \backslash \lambda$	Nearest Neighbor	Bicubic	Bilinear	Lanczos
Nearest Neighbor	0.142	0.138	0.156	0.141
Bicubic	0.024	0.005	0.034	-0.013
Bilinear	0.124	0.112	0.130	0.107
Lanczos	0.017	-0.005	0.034	-0.015

(e) \mathcal{C}_5

$\rho \backslash \lambda$	Nearest Neighbor	Bicubic	Bilinear	Lanczos
Nearest Neighbor	0.089	0.087	0.105	0.078
Bicubic	0.056	0.046	0.055	0.048
Bilinear	0.037	0.033	0.042	0.035
Lanczos	0.113	0.110	0.122	0.102

(f) \mathcal{C}_6

$\rho \backslash \lambda$	Nearest Neighbor	Bicubic	Bilinear	Lanczos
Nearest Neighbor	0.058	0.071	0.081	0.058
Bicubic	0.057	0.038	0.059	0.024
Bilinear	0.087	0.082	0.104	0.076
Lanczos	0.098	0.090	0.115	0.082

(g) \mathcal{C}_7

$\rho \backslash \lambda$	Nearest Neighbor	Bicubic	Bilinear	Lanczos
Nearest Neighbor	0.134	0.131	0.120	0.126
Bicubic	0.057	0.047	0.057	0.043
Bilinear	0.067	0.060	0.076	0.058
Lanczos	0.061	0.050	0.067	0.053

(h) \mathcal{C}_8

$\rho \backslash \lambda$	Nearest Neighbor	Bicubic	Bilinear	Lanczos
Nearest Neighbor	0.058	0.074	0.077	0.084
Bicubic	0.051	0.059	0.076	0.052
Bilinear	0.056	0.066	0.094	0.056
Lanczos	0.048	0.057	0.070	0.051









(i) \mathcal{C}_9

$\rho \backslash \lambda$	Nearest Neighbor	Bicubic	Bilinear	Lanczos
Nearest Neighbor	0.020	0.024	0.048	0.030
Bicubic	0.078	0.082	0.108	0.070
Bilinear	0.099	0.102	0.134	0.090
Lanczos	0.070	0.065	0.092	0.051

(j) \mathcal{C}_{10}

Figure A1. The heat maps of the loss function $\mathcal{L}^{\mathcal{C}}(\mathcal{A}_a^{\text{hr}}) = \tilde{\tau}_a - \tau_a$ for each CNN: (a) \mathcal{C}_1 , (b) \mathcal{C}_2 , (c) \mathcal{C}_3 , (d) \mathcal{C}_4 , (e) \mathcal{C}_5 , (f) \mathcal{C}_6 , (g) \mathcal{C}_7 , (h) \mathcal{C}_8 , (i) \mathcal{C}_9 , (j) \mathcal{C}_{10} .

Table A1. For $1 \leq a \leq 10$, the image \mathcal{A}_a^{hr} classified by each CNN in the category c_a (interpolation = 'lanczos').

a	1	2	3	4	5	6	7	8	9	10
\mathcal{A}_a^{hr}										
$w \times h$	910 × 604	960 × 640	910 × 607	2462 × 2913	910 × 607	641 × 600	1280 × 800	1280 × 800	1954 × 2011	1740 × 1710
c_1	cheetah 0.872	Eskimo_dog 0.691	koala 0.987	lampshade 0.512	white_stork 0.484	screen 0.659	fountain 0.223	sports_car 0.840	book_jacket 0.237	buckle 0.249
c_2	cheetah 0.986	Eskimo_dog 0.822	koala 0.997	lampshade 0.673	Granny_Smith 0.213	screen 0.818	comic_book 0.322	sports_car 0.587	rubber_eraser 0.327	book_jacket 0.168
c_3	cheetah 0.976	Eskimo_dog 0.737	koala 0.997	table_lamp 0.614	toucan 0.194	screen 0.724	comic_book 0.453	sports_car 0.808	handkerchief 0.194	book_jacket 0.237
c_4	cheetah 0.816	Eskimo_dog 0.516	koala 0.999	table_lamp 0.884	flamingo 0.497	screen 0.706	totem_pole 0.161	sports_car 0.740	tray 0.297	book_jacket 0.439
c_5	cheetah 0.923	Eskimo_dog 0.613	koala 0.902	table_lamp 0.488	spoonbill 0.209	screen 0.804	fountain 0.951	sports_car 0.711	book_jacket 0.372	manhole_cover 0.116
c_6	cheetah 0.972	Eskimo_dog 0.704	koala 0.994	lampshade 0.507	macaw 0.433	screen 0.697	gasmask 0.280	sports_car 0.813	pillow 0.163	coffee_mug 0.175
c_7	cheetah 0.948	Eskimo_dog 0.629	koala 0.555	lampshade 0.686	white_stork 0.224	screen 0.904	fountain 0.204	sports_car 0.470	book_jacket 0.378	buckle 0.378
c_8	cheetah 0.899	Eskimo_dog 0.760	koala 0.979	table_lamp 0.641	toucan 0.163	screen 0.699	fountain 0.702	sports_car 0.546	envelope 0.243	matchstick 0.569
c_9	cheetah 0.953	Eskimo_dog 0.343	koala 0.997	lampshade 0.536	toucan 0.455	screen 0.706	comic_book 0.492	sports_car 0.480	binder 0.283	coffee_mug 0.084
c_{10}	cheetah 0.867	Eskimo_dog 0.412	koala 0.964	table_lamp 0.588	lorikeet 0.145	screen 0.665	comic_book 0.553	sports_car 0.649	lighter 0.229	prayer_rug 0.158
c_r	poncho	goblet	weimaraner	weevil	wombat	swing	altar	beagle	triceratops	hamper

Appendix 2

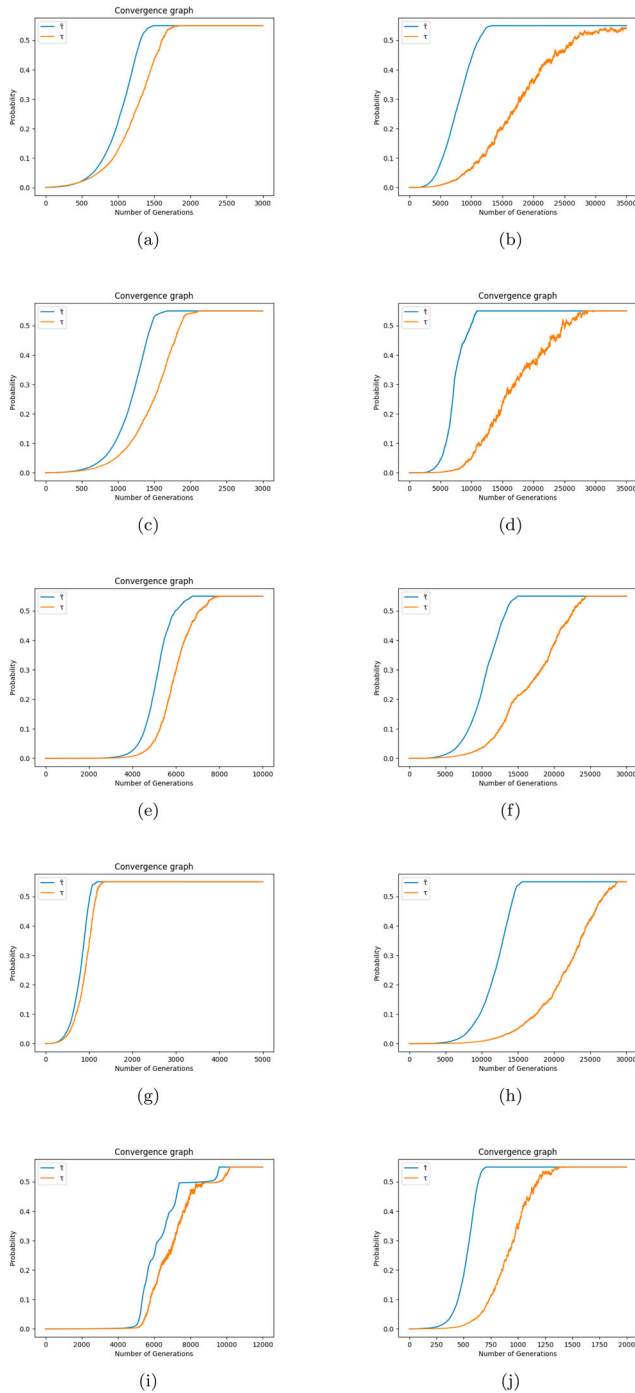


Figure A2. Convergence characteristics of the $EA^{\text{target},C}$ for $C = C_1, \dots, C_5$ based on τ_t and $\tilde{\tau}_t$ for each CNN. Only the pairs with the smallest and largest \mathcal{L}_{max} values are shown in the figures. (a) $C_1 - \min(\mathcal{L}_{max}):A_7$, (b) $C_1 - \max(\mathcal{L}_{max}):A_8$, (c) $C_2 - \min(\mathcal{L}_{max}):A_7$, (d) $C_2 - \max(\mathcal{L}_{max}):A_1$, (e) $C_3 - \min(\mathcal{L}_{max}):A_6$, (f) $C_3 - \max(\mathcal{L}_{max}):A_9$, (g) $C_4 - \min(\mathcal{L}_{max}):A_2$, (h) $C_4 - \max(\mathcal{L}_{max}):A_9$, (i) $C_5 - \min(\mathcal{L}_{max}):A_6$, (j) $C_5 - \max(\mathcal{L}_{max}):A_7$.

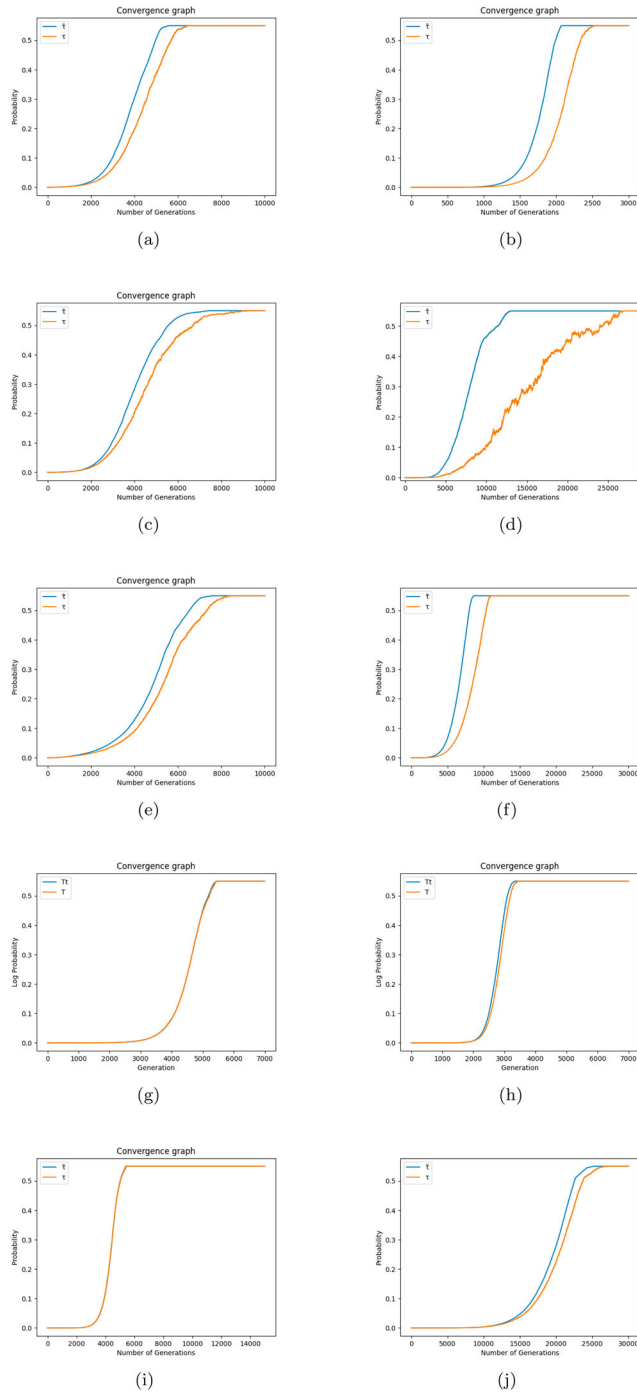


Figure A3. Convergence characteristics of the $EA^{\text{target},C}$ for $C = C_6, \dots, C_{10}$ based on τ_t and $\tilde{\tau}_t$ for each CNN. Only the pairs with the smallest and largest \mathcal{L}_{\max} values are shown in the figures. (a) $C_6 - \min(\mathcal{L}_{\max}):A_2$. (b) $C_6 - \max(\mathcal{L}_{\max}):A_3$. (c) $C_7 - \min(\mathcal{L}_{\max}):A_2$. (d) $C_7 - \max(\mathcal{L}_{\max}):A_1$. (e) $C_8 - \min(\mathcal{L}_{\max}):A_2$. (f) $C_8 - \max(\mathcal{L}_{\max}):A_{10}$. (g) $C_9 - \min(\mathcal{L}_{\max}):A_6$. (h) $C_9 - \max(\mathcal{L}_{\max}):A_4$. (i) $C_{10} - \min(\mathcal{L}_{\max}):A_6$. (j) $C_{10} - \max(\mathcal{L}_{\max}):A_9$.

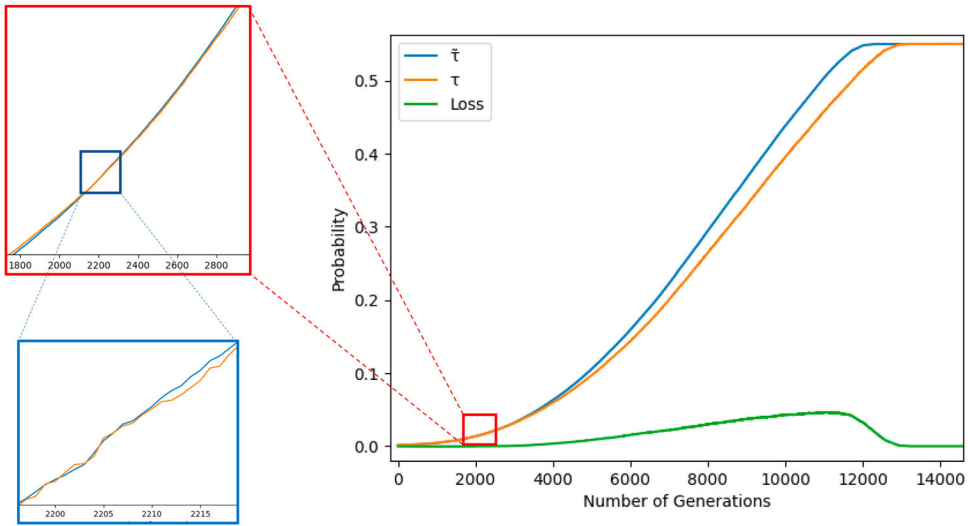


Figure A4. The average convergence characteristics of $EA^{\text{target}, \mathcal{C}}$ for $\mathcal{C} = \text{VGG-16}$ aiming at generating an HR adversarial image by directly evolving $\mathcal{A}_{10}^{\text{hr}}$. The horizontal axis of the graph is the number of generations, and the vertical axis is the target probability τ_t , $\tilde{\tau}_t$ and the loss $\mathcal{L} = \tilde{\tau}_t - \tau_t$. The zoomed-in section of the graph shows when the $\tilde{\tau}_t$ becomes bigger than the τ_t ($\approx 2209^{\text{th}}$ generation). As the loss \mathcal{L} curve shows, the distance between $\tilde{\tau}_t$ and τ_t increases over the generations.

Appendix 3

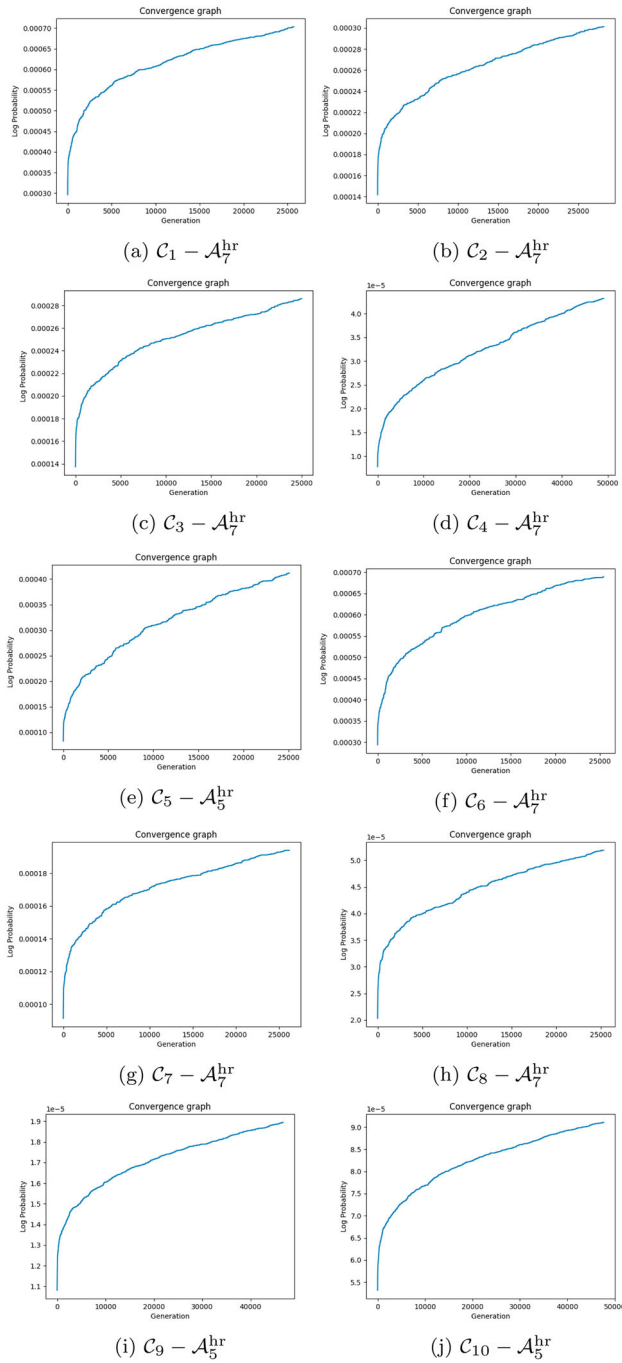


Figure A5. Convergence characteristics of $EA^{\text{target},C}$ aiming at generating within 48 hours a high-resolution adversarial image by directly evolving \mathcal{A}_5^{hr} for the (toucan, wombat) pair and $C = \text{MobileNet}$ (d), VGG-16 (i), VGG-19 (j), and \mathcal{A}_7^{hr} for the (comic book, altar) and $C = \text{DenseNet121}$ (a), DenseNet169 (b), DenseNet201 (c), NasNetMobile (e), ResNet50 (f), ResNet101 (g), ResNet-152 (h). (a) $C_1 - \mathcal{A}_7^{hr}$. (b) $C_2 - \mathcal{A}_7^{hr}$. (c) $C_3 - \mathcal{A}_7^{hr}$. (d) $C_4 - \mathcal{A}_7^{hr}$. (e) $C_5 - \mathcal{A}_5^{hr}$. (f) $C_6 - \mathcal{A}_7^{hr}$. (g) $C_7 - \mathcal{A}_7^{hr}$. (h) $C_8 - \mathcal{A}_7^{hr}$. (i) $C_9 - \mathcal{A}_5^{hr}$. (j) $C_{10} - \mathcal{A}_5^{hr}$.

Table A2. Success rates (SR) of EA^{target,C} for each CNN over 10 independent runs, for $\tau = 0.55$ and $X = 35,000$.











a	1	2	3	4	5	6	7	8	9	10	
A_n^{hr}											
e_t	poncho	goblet	weimaraner	weevil	wombat	swing	altar	beagle	triceratops	hamper	SR(%)
C_1	0	10	10	10	10	10	10	9	0	10	79
C_2	10	10	10	10	10	10	10	9	10	10	99
C_3	0	10	10	10	10	10	10	10	6	10	86
C_4	10	10	10	10	10	10	10	10	5	10	95
C_5	2	2	10	7	10	10	10	2	0	10	63
C_6	10	10	10	10	10	10	10	10	0	10	90
C_7	10	10	10	10	10	10	10	10	2	10	92
C_8	9	10	10	10	10	10	10	10	8	10	97
C_9	10	10	10	10	10	10	10	10	10	10	100
C_{10}	10	10	10	10	10	10	10	10	9	10	99
Avg.	8.9	9.2	10	9.7	10	10	10	9	7.1	10	90

Table A3. Direct attack results of $EA^{\text{target}, C}$ for the easiest (c_a, c_t) pairs after 48 hours of execution of the algorithm. In the last column $c_t_ratio = c_{t_end}/c_{t_start}$.

(c_a, c_t)		(toucan, wombat)					(comic_book, altar)					
		# of gen.	c_a_start	c_a_end	c_t_start	c_t_end	# of gen.	c_a_start	c_a_end	c_t_start	c_t_end	c_t_ratio
C_1	DenseNet121						25695	0.223	0.227	3.0E-04	7.0E-04	2.4
C_2	DenseNet169						28155	0.322	0.294	1.4E-04	3.0E-04	2.1
C_3	DenseNet201						24983	0.453	0.467	1.4E-04	2.9E-04	2.1
C_4	MobileNet	49082	0.497	0.394	7.77E-06	4.31E-05						5.5
C_5	NASNetMobile						25098	0.951	0.748	8.3E-05	4.1E-04	5.0
C_6	ResNet50						25448	0.280	0.270	2.9E-04	6.9E-04	2.3
C_7	ResNet101						26178	0.204	0.084	9.1E-05	1.9E-04	2.1
C_8	ResNet152						25328	0.702	0.575	2.0E-05	5.2E-05	2.6
C_9	VGG16	46721	0.455	0.405	1.08E-05	1.90E-05						1.8
C_{10}	VGG19	47668	0.145	0.132	5.32E-05	9.11E-05						1.7

Table A4. Direct attack results of $EA^{\text{target},C}$ for all (c_a, c_t) pairs after 100 generations (hence less than 48 hours). The results show the time spent by the main operations of $EA^{\text{target},C}$ in one generation.

	Input Image Image size (n)	\mathcal{A}_1 910×604	\mathcal{A}_2 960×640	\mathcal{A}_3 910×607	\mathcal{A}_4 2462×2913	\mathcal{A}_5 910×607	\mathcal{A}_6 641×600	\mathcal{A}_7 1280×800	\mathcal{A}_8 1280×800	\mathcal{A}_9 1954×2011	\mathcal{A}_{10} 1740×1710	%
Avg. of all CNNs	Time per gen	3.528	3.790	3.427	45.570	3.469	2.499	6.239	6.248	24.815	18.871	
	Resize	0.384	0.401	0.371	3.829	0.373	0.294	0.635	0.636	2.209	1.729	9.2
	Prediction	0.155	0.150	0.149	0.156	0.150	0.149	0.150	0.150	0.155	0.153	1.3
	Mutation	2.063	2.215	1.995	29.922	2.026	1.410	3.768	3.778	16.050	12.118	63.6
	Crossover	0.161	0.179	0.161	2.067	0.161	0.112	0.297	0.297	1.133	0.861	4.6
	Time per gen/n	6.42E-06	6.17E-06	6.21E-06	6.35E-06	6.28E-06	6.50E-06	6.09E-06	6.10E-06	6.31E-06	6.34E-06	

Table A5. Indirect attack results of EA^{target,C} for all (c_a, c_t) pairs after 100 generations. The results show the time spent by the main operations of EA^{target,C} in one generation.

Input Image		\mathcal{A}_1	\mathcal{A}_2	\mathcal{A}_3	\mathcal{A}_4	\mathcal{A}_5	\mathcal{A}_6	\mathcal{A}_7	\mathcal{A}_8	\mathcal{A}_9	\mathcal{A}_{10}	%
Image size (n)		910×604	960×640	910×607	2462×2913	910×607	641×600	1280×800	1280×800	1954×2011	1740×1710	
Avg. of all CNNs	Time per gen	0.512	0.516	0.518	0.673	0.517	0.512	0.526	0.530	0.596	0.572	
	Resize	0.020	0.022	0.020	0.174	0.020	0.016	0.032	0.032	0.101	0.079	9.4
	Prediction	0.154	0.155	0.155	0.155	0.156	0.154	0.155	0.155	0.154	0.154	28.3
	Mutation	0.143	0.142	0.145	0.146	0.145	0.147	0.142	0.145	0.144	0.143	26.4
	Crossover	0.009	0.010	0.009	0.010	0.009	0.009	0.009	0.009	0.010	0.010	1.7
	Time per gen/n	9.31E-07	8.39E-07	9.38E-07	9.39E-08	9.36E-07	1.33E-06	5.14E-07	5.17E-07	1.52E-07	1.92E-07	