Evolutionary Algorithm-based images, humanly indistinguishable and adversarial against Convolutional Neural Networks: efficiency and filter robustness

RALUCA CHITIC1, ALI OSMAN TOPAL1, AND FRANCK LEPRÉVOST.1
1University of Luxembourg, House of Numbers, 6, avenue de la Fonte, L-4364 Esch-sur-Alzette, G-D of Luxembourg (e-mail: Raluca.Chitic@uni.lu, Aliosman.Topal@uni.lu, Franck.Leprevost@uni.lu)
Corresponding author: Raluca Chitic (e-mail: Raluca.Chitic@uni.lu).

ABSTRACT Convolutional neural networks (CNNs) have become one of the most important tools for image classification. However, many models are susceptible to adversarial attacks, and CNNs can perform misclassifications. In previous works, we successfully developed an EA-based black-box attack that creates adversarial images for the target scenario that fulfills two criteria. The CNN should classify the adversarial image in the target category with a confidence $\geq 0.95$, and a human should not notice any difference between the adversarial and original images. Thanks to extensive experiments performed with the CNN $C = \text{VGG-16}$ trained on the CIFAR-10 dataset to classify images according to 10 categories, this paper, which substantially enhances most aspects of [1], addresses four issues. (1) From a pure EA point of view, we highlight the conceptual originality of our algorithm $\text{EA}_{\text{target},C}$, versus the classical EA approach. The competitive advantage obtained was assessed experimentally during image classification. (2) We then measured the intrinsic performance of the EA-based attack for an extensive series of ancestor images. (3) We challenged the filter resistance of the adversarial images created by the EA for five well-known filters. (4) We proceed to the creation of natively filter-resistant adversarial images that can fool humans, CNNs, and CNNs composed with filters.

INDEX TERMS Convolutional neural network, evolutionary algorithm, black-box attack, image classification, adversarial perturbation, filter.

I. INTRODUCTION

In 2012, Krizhevsky et al. [2] presented the outstanding performance of convolutional neural networks (CNNs) on a very difficult image classification task [3]. Since then, as computing capacity has increased, CNNs have become the main driver of many computer vision applications, including image classification [4]–[7], facial recognition [8], [9], malware detection [10]–[12], email spam filters [13], speech recognition [14], [15], robotics [16], and self-driving cars [17], [18]. Despite their increasing power and the variety of applications, CNNs are susceptible to deception. In the context of image classification, by analogy with *Trompe-l’œil* that challenges humans’ visual perception, a CNN can be led to misclassification of objects in an image. The generic attack to create such specially crafted adversarial images consists of adding some appropriate noise to a legitimate input, leading the network to label the new input in a different category than expected [13], [19], [20].

White-box and black-box attacks differ according to the level of knowledge about the addressed CNN at the disposal of the attacker. While the attacker has a complete knowledge of the CNN model, its design and its parameters in the former case, the situation is opposite in the latter case, in which the attacker’s knowledge is scarcely limited to the size of the images handled by the CNN, and to classification values outputted by the CNN for ad-hoc queries (but without any information about how these values are obtained). Starting
from an original image labeled by a CNN as representing an object in a specific category, methods that creates adversarial images may adopt different scenarios. For instance, a targeted attack creates an adversarial image that the CNN misclassifies as belonging to an a priori particular predefined class, different from the original one. A different scenario is addressed by untargeted attacks that only require the CNN to misclassify the adversarial image as belonging to any class whatsoever, provided that this class differs from the original one.

Although efficient against a CNN, the perturbations added to an original image to create an adversarial image may be highly noticeable for a human eye, as illustrated in Fig. 1 (a), (b), (c), and (d). Our evolutionary algorithm-based black-box, targeted attack $\text{EA}_{\text{target},\mathcal{C}}$ (introduced in [21], [22], see also [23]) differs from existing techniques in this respect. Not only does our evolutionary algorithm (EA) efficiently produce adversarial images that deceive the targeted CNN model $\mathcal{C}$ with high accuracy, but the perturbations added by our algorithm to the original image are not perceptible to the human eye, as shown in Fig.1e (original image in the first row, our adversarial image in the second row).

![FIGURE 1: The images in the first row represents the original images and in the second row the adversarial images and their respective class labels that are created by (a) One-Pixel attack [24], (b) Few-Pixels attack [25], (c) Fooling Transfer Net (FTN) [26], (d) Scratch that! [27], and (e) our EA-based attack [21], [22].](image)

The purpose of this study, which substantially enhances most aspects of [1], is to address four issues: (1) Conceptual originality and competitive advantage of our algorithm $\text{EA}_{\text{target},\mathcal{C}}$, (2) intrinsic performance of this EA-based attack, (3) filter resistance of the adversarial images created by the EA, and (4) creation of natively filter-resistant adversarial images. Before being more specific, let us point out that all experiments in this paper are performed with distance $d = L_2$ for the CNN $\mathcal{C} = \text{VGG-16}$ [6], [28] trained on the CIFAR-10 [29] dataset to classify images according to 10 categories, and mainly address the target scenario, but also, to a lesser extent, the untargeted scenario (see Section II).

The issue (1) (see Section III) relates to a series of conceptual differences, from a "pure" evolutionary algorithm point of view (hence independent of the task to perform, to some extent), between our adapted version and the classical EA approach [30]. To assess the practical impact of these conceptual differences, we analyzed their respective performances in creating adversarial images for a demanding definition of a successful attack. Indeed, for the $(c_a, c_t)$ target scenario performed on an ancestor $\mathcal{A}$ classified by VGG-16 in $c_a$, we require these algorithms to create in less than 7000 generations an adversarial image $\mathcal{D}$ classified by VGG-16 as belonging to $c_t \neq c_a$ with a $c_t$-label value $\geq 0.95$, while remaining so close to $\mathcal{A}$ that a human would not notice any difference between the adversarial and the ancestor. The first outcome of this study is that our "adapted EA" significantly outperforms the "classic EA" approach at creating such adversarial images for all considered cases, since it requires between 8% and 25% fewer generations to terminate successfully.

We then address issue (2) by a thorough and extended efficiency study of our EA-based attack with the "adapted EA" version (see Section IV). In the first series of experiments with one ancestor per category of CIFAR-10, we performed 10 independent runs per ancestor per target category, leading to a total of 900 attacks. The algorithm $\text{EA}_{\text{adapt},\mathcal{C}}$ achieves a 100% success rate (all ancestor-target categories are achieved for at least one of the 10 runs performed on each ancestor), requiring between 290 and 2793 generations on average, depending on the $(c_a, c_t)$ target scenario. To better assess the importance of the choice of the ancestor in a given category $c_a$, and the impact of the seed value used for a specific run, we extend these experiments. In a second series of experiments, we randomly selected 50 distinct ancestors for each of the 10 categories of CIFAR-10 and altogether ran 4500 attacks for the target scenario. In this case, our algorithm achieved a success rate of 98%, requiring between 461 and 1717 generations on average. Moreover, both series of experiments show that a run of $\text{EA}_{\text{adapt},\mathcal{C}}$ has more than 96% (actually 96.56% for the former, and 98.06% for the latter series) to terminate successfully, and to create images that fool humans and VGG-16 trained on CIFAR-10, despite our demanding requirements for a successful termination.

Issues (3) and (4) (addressed in Sections V and VI, respectively) deserve to be put into the following broader perspective. Let $\mathcal{A}$ be an image classified by a CNN $\mathcal{C}$ in some category $c_a$, and $\mathcal{D}$ be an adversarial image; for the target scenario, $\mathcal{C}$ classifies in a distinct category $c_t$ (at this stage, the type of attack that leads to $\mathcal{D}$ does not matter). One now considers a function $\mathcal{F}$ that acts on such images to create images $\mathcal{F}(\mathcal{A})$ and $\mathcal{F}(\mathcal{D})$ of the size handled by the CNN (which coincides with the same common size of $\mathcal{A}$ and $\mathcal{D}$ in the present case). How does the CNN classify these new images? Does $\mathcal{F}(\mathcal{D})$ remain adversarial, or does the composition $\mathcal{C} \circ \mathcal{F}$ (that consists in putting $\mathcal{F}$ ahead of

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
C) protect C against the attack? If this latter case holds, can one adapt the attack to create images that fool not only C, but also the F-enhanced CNN C ∘ F? If yes, would such images, adversarial for C ∘ F, be adversarial as well for C ∘ G for G ≠ F, and hence have the capability to fool the same CNN C but enhanced by other functions G?

Among the different meaningful functions F one could think of in this context, we undertake the study for filters. Indeed, daily used in image processing, filters substantially impact the visual appearance of images for a human eye on the one hand, and potentially affect the classification process of a trained CNN on the other hand. It is therefore tempting to check whether adding filters may prevent CNNs from misclassification, or reduce this risk to some extent, when facing an adversarial image. Additionally, one may also want to evaluate the quality of adversarial images by their capacity to mimic the ancestor’s image behavior when exposed to filters.

For reasons given in Section V, in which issue (3) is discussed, we proceed to the selection of five filters, namely the inverse filter (F1), the Gaussian blur filter (F2), the median filter (F3), the unsharp mask filter (F4), and the F5 combination of the two last filters. With each of them, we filter the ancestor Aa and the adversarial images Dn,t(An) created by the EA_{L_2,F_5} algorithm in Section IV. VGG-16 was then challenged with the filtered images. The values of a series of specifically designed indicators led to two conclusions. On the one hand, the Inverse, and the Unsharp mask filters are significantly inefficient against our EA, because, for instance, 95% of the adversarial images filtered by F5 remain adversarial for the target scenario, and 95% remain adversarial for the untargeted scenario (in a relaxed sense to be made precise in this Section). A contrario, the other filters, especially the combination F5, render our EA-based attack less effective for both the target and the untargeted scenario.

This led us to address the final issue in (4). For a filter F, we conceived a filter-enhanced F-fitness function (see Section VI), and the corresponding algorithm EA_{L_2,F_5} target,VGG-16 obtained from EA_{L_2,F_5} target,VGG-16 by updating the fitness function accordingly. For the reasons given in Section VI, we select F = F5, and allocate to EA_{L_2,F_5} target,VGG-16 the task to create adversarial images that are moreover natively immune against filter F5. In other words, these adversarial images simultaneously fool C and C ∘ F5 for C = VGG-16 for the target scenario (still with the demanding target label value ≥ 0.95), while remaining so close to the ancestor that no human eye would notice any difference.

We performed similar experiments for issue (2). The first series of 900 attacks (one ancestor per ancestor category, 10 independent runs for each (c_o(A_a), c_t) scenario) shows that EA_{L_2,F_5} target,VGG-16 achieves a success rate of 96.66% (three combinations were not achieved), and that the probability that it terminated successfully for a given run was 95.77%, requiring an average between 798 and 2746 generations for the successful (c_o(A_a), c_t) considered. In a second series of 4500 attacks performed with 50 different ancestors per category, EA_{L_2,F_5} target,VGG-16 showed a success rate of 88%, with between 1250 and 2404 generations on average.

We complete study (4) by exploring whether an adversarial image, constructed by EA_{L_2,F_5} target,VGG-16 to fool both C and C ∘ F5, would also be adversarial against C ∘ Fk for the other filters F1, F2, F3, and F4 for C = VGG-16. Our study shows that it is so for F3 and F4 with (depending on the target or untargeted scenario) between 83% and 89% of the images remaining adversarial against these filters. 56% of these images were also adversarial for F1 for the untargeted scenario, while this percentage dropped to 23% for F2. Therefore, the EA_{L_2,F_5} target,VGG-16 attack, designed to be robust against C and C ∘ F5 for C = VGG-16, is also robust to a significant extent against all individual filters for the untargeted scenario.

Section VII summarizes the conclusions of this case study, and provides a series of research directions.

II. THE TARGET SCENARIO ON VGG-16 TRAINED ON CIFAR-10

Although applicable to any CNN trained at image classification on any dataset, we instantiate our approach on the concrete case of VGG-16 [6] trained on CIFAR-10 [29].

A. VGG-16 TRAINED ON CIFAR-10

The CIFAR-10 dataset encompasses 50,000 training images, and 10,000 test images of size 32 × 32 × 3, meaning that each image has a width and height of 32 pixels, and each pixel has a color resulting from the three RGB values. Once trained, VGG-16 sorts images according to the 10 categories c_i of CIFAR-10 listed in the 2nd row of Table 1, composed of 4 “Object” categories (c_1, c_2, c_9, c_10), and of 6 “Animal” categories (c_i for 3 ≤ i ≤ 8). The 4th row of Table 1 displays the original ancestor images A_a in the categories c_o (and their respective c_o-label values, see below) used throughout this paper, and the 3rd row give their reference number in the test set of CIFAR-10.

In practice, an input image I given to VGG-16 trained on CIFAR-10 is processed through 16 layers to produce a classification output vector:

$$\mathbf{o}_I = (\mathbf{o}_I[1], \cdots, \mathbf{o}_I[10]),$$

where 0 ≤ o_I[i] ≤ 1 for 1 ≤ i ≤ 10, and \(\sum_{i=1}^{10} o_I[i] = 1\). Each c_i-label value o_I[i] measures the probability that image I belongs to category c_i. Consequently, an image I is classified as belonging to category c_k if k = arg max_{1≤i≤10}(o_I[i]). The higher the label value o_I[k], the
TABLE 1: For $1 \leq a \leq 10$, the image $\mathcal{A}_a$ (and its reference number $n^o$ in the test set of CIFAR-10) classified by VGG-16 in the category $c_a$, with its corresponding $c_a$-label values. These images are used as ancestor in most of our experiments.

| $a$ | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
|-----|----|----|----|----|----|----|----|----|----|----|
| $c_a$ | plane | car | bird | cat | deer | dog | frog | horse | ship | truck |
| $n^o$ | 281 | 82 | 67 | 91 | 455 | 16 | 29 | 17 | 1 | 76 |
| $\mathcal{A}_a$ | ![Image](image_url1) | ![Image](image_url2) | ![Image](image_url3) | ![Image](image_url4) | ![Image](image_url5) | ![Image](image_url6) | ![Image](image_url7) | ![Image](image_url8) | ![Image](image_url9) | ![Image](image_url10) |
| $c_a$-label values | 0.6900 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9998 | 0.9996 |

higher the confidence that $I$ represents an object of category $c_k$.

B. TARGETED AND UNTARGETED SCENARIOS

The target scenario consists of first choosing two different categories, $c_t \neq c_a$, among the 10 categories of CIFAR-10. Then, one is given an ancestor image $\mathcal{A}$ labeled by VGG-16 as belonging to $c_a$. Finally, one constructs an adversarial image $D$, classified by VGG-16 as belonging to $c_t$, although $D$ remains so close to $\mathcal{A}$ that a human would likely classify $D$ as belonging to $c_a$ or even be unable to distinguish $D$ from $\mathcal{A}$. The classification threshold value is set at $\tau = 0.95$, meaning that such a $D$ has achieved its purpose if $o_D[i] \geq 0.95$. We shall also encounter in Section V the slightly different untargeted scenario. In this case, an adversarial image $D$ is still required to be similar to $\mathcal{A}$ for a human eye, but one only requires that VGG-16 classifies $D$ as belonging to a category $c \neq c_a$, in the limited sense that the label value of $c$ outputted by VGG-16 for $D$ is the largest among all label values, and is strictly larger than the label value of $c_a$. In particular, an image adversarial for the target scenario is also adversarial for the untargeted scenario, but the inverse may not be true.

III. "ADAPTED_EA" VERSUS "CLASSIC_EA"

Our evolutionary algorithm $E_A^{target,c}$ (see [1], [21]–[23], [31]) is a black-box, targeted attack that constructs adversarial images against a CNN $C$ in the sense sketched in Subsection II-B, where $d$ is a metric assessing the proximity for a human eye between the evolved images and the original image.

In this section, we show, from a "pure" evolutionary algorithm point of view, that $E_A^{target,c}$ presents a series of important and substantial differences compared to the approach classically ([30]) adopted for EAs performing similar tasks, and we prove that these differences lead to a comparative advantage in terms of performance. First, we examine these differences from a conceptual point of view, meaning independently of any specific task. For simplicity, we refer to our version as "adapted_EA" and to its classical version as "classic_EA". We then compared the performances of these algorithms for the task consisting of fooling VGG-16 trained on CIFAR-10 for image recognition in the target scenario. In other words, these algorithms are given the task of evolving an ancestor image $A$ into an adversarial image $D$ fulfilling the conditions described in Subsection II-B. We specify the parameters of the EAs, and run the algorithms for four different ancestor/target combinations.

All experiments in this paper were implemented in Python 3.7 with the NumPy [32] library. For the filter experiments in Sections V and VI, we used the OpenCV implementation library [33]. Keras [34] was used to load and run the VGG-16 [6] model. The experiments were performed on nodes with NVIDIA Tesla V100 GPGPUs of the IRIS HPC Cluster at the University of Luxembourg [35].

A. CONCEPTUAL DIFFERENCES BETWEEN "ADAPTED_EA" AND "CLASSIC_EA"

To illustrate the differences between our version ("adapted_EA") and the classic version ("classic_EA", as described in [30]) of an EA, let us provide their respective algorithmic pseudo-codes. We assume that both have a fixed population size, which remains constant generation for generation. For both, we set the initial population as made of identical copies of the considered ancestor. Based on our experiments, we considered a population size of 160 as the best trade-off in terms of speed and accuracy.

Algorithm 1 "Classic_EA" algorithm pseudo code

1: BEGIN
2: INITIALISE population;
3: EVALUATE each candidate;
4: REPEAT UNTIL (Termination condition is satisfied) DO
5: 1 SELECT top 10-20% (16 to 32 individuals) as parents;
6: 2 RECOMBINE pairs of parents resulting in offsprings;
7: 3 MUTATE the offsprings;
8: 4 EVALUATE new candidates;
9: 5 SELECT individuals for the next generations;
10: END
11: END

The main difference between "classic_EA" (as described in Algorithm 1) and our version (as described in Algorithm 2) is the process of selection, recombination and mutation. In "classic_EA", the best 10-20% of the population are selected as elites (hence between 16 and 32 individuals), and new offsprings are generated with these elites by recombination and mutation. Then the last 10-20% (idem) of the population is eliminated, and only these 10-20% are updated at each
Algorithm 2 "Adapted_EA" algorithm pseudo code

1: BEGIN
2: INITIALISE population;
3: EVALUATE each candidate;
4: REPEAT UNTIL (Termination condition is satisfied) DO
5:   SELECT
6:    split into 3 groups;
7:    Elite: top 10;
8:    "didn’t make it": last 80;
9:    Middle-class: 70;
10: RECOMBINE
11:   elites + middle-class resulting in offsprings;
12:   replace "didn’t make it" with offsprings;
13: MUTATE
14:    middle-class and offsprings;
15: EVALUATE each candidate;
16: END
17: END

generation. However, in our version, the number of elites is set to the first 10 individuals; then, the algorithm starts to modify the whole rest (150 individuals) of the population by eliminating, mutating, and recombining with elites just after the first generation.

B. THE EA PARAMETERS

The task on which we evaluate the performance of both approaches is the construction of adversarial images for CNNs. Although our algorithm $EA_{adapt}$ is efficient for a series of CNNs, here we make our point for the instantiation $EA_{adapt,VGG-16}$ of this algorithm (Algorithm 2) and of its classical EA version (Algorithm 1), for $C = VGG-16$ trained on CIFAR-10, and for the metric $d = L_2$. Starting from a common ancestor image $A$ of size $32 \times 32 \times 3$ labeled by VGG-16 as belonging to $c_a$, and from a target category $c_t \neq c_a$, the specific parameters and choices of the algorithms are as follows:

Population initialization. Both algorithms start the search with the same initial population set, made of 160 identical replicas of the ancestor image $A$.

Evaluation - Fitness Function. This operation is performed on each individual image $ind$ of a given generation $g_p$ via the fitness function $fit_{L_2}(ind, g_p)$ that assesses a dual goal: the evolution of $ind$ towards the target category $c_t$, and its proximity to ancestor $A$, measured by using the $L_2$-norm:

$$fit_{L_2}(ind, g_p) = A(gp, ind) \cdot o_{ind}[c_t] - B(gp, ind) \cdot L_2(gp, ind, A) \geq 0,$$  \hspace{1cm} (2)

where the quantities $A(gp, ind), B(gp, ind) \geq 0$ weight and balance the dual goal. The $L_2$-norm is used to calculate the difference between the pixel values of the ancestor and the considered image $ind$:

$$L_2(ind, A) = \sum_{p_j} |ind[p_j] - A[p_j]|^2,$$  \hspace{1cm} (3)

where $p_j$ is the pixel in the $j^{th}$ position, and $0 \leq ind[p_j], A[p_j] \leq 255$ are the corresponding pixel values of the images $ind$ and $A$. Concretely, for any generation $g_p$, one sets $B(gp, ind) = 10^{-5}$. The value of $A(gp, ind)$ depends on $o_{ind}[c_t]$ (note that $\log_{10} o_{ind}[c_t] \leq 0$).

$$A(gp, ind) = 10^{-\log_{10} o_{ind}[c_t]}$$  \hspace{1cm} (4)

Selection, Recombination, Mutation. The fitness function of each individual in the population is computed (starting with the first generation made of the initial population). Mutations and cross-overs are those described in [21] (they remain similar to some extent to those of [31]).

The first generation of each run EA has a specific seed that controls the randomness of the mutation and crossover operations. The same seed value was applied to the adapted_EA and classic_EA to ensure fair competition. For the mutation, this seed determines the location of the pixels and the magnitude (within a range defined below) of the modifications they undergo. For the cross-over, the seed determines which individuals form pairs, as well as the location and size of the interchanged regions.

Pixel values were modified in the range of $\pm 3$ in both EA versions used here. The algorithms used the same parameters and techniques for the mutation and crossover operations. These operations lead to 160 descendant images composing the individuals of the new generation subject to the next round of evaluation.

Termination condition. For each version of the EA, this loop is repeated until a descendant image is created in less than 7000 generations (this maximum number of iterations is a reasonable trade-off, based on our experiments), which is classified as the target category $c_t$ with a probability $\geq 0.95$, while remaining close to $A$ that a human would not notice any difference between it and $A$ (and \textit{a fortiori} would classify this descendant image still as belonging to the original category $c_a$). This defines a successful termination, in which case one notes $D_{\text{adapt}}(A)$ as the adversarial image resulting from $EA_{adapt,VGG-16}$ (Algorithm 2) run on $A$, and $D_{\text{classic}}(A)$ as the result of the classic (Algorithm 1) version of the EA also run on $A$. Otherwise, the algorithm terminates without success.

Therefore, the algorithms terminate after 7000 generations at the latest, regardless of whether they have succeeded in creating such an adversarial image.

C. EXPERIMENTAL COMPARISON OF "ADAPTIVED_EA" WITH "CLASSIC_EA"

We experimentally compared the efficiency of both versions of the EA for four ancestor/target pairs of categories Animal/Animal, Object/Object, Animal/Object, and
Object/Animal.

Concretely, the Animal ancestor categories are bird and dog, with image \( A_3 \) as ancestor for the bird category \( c_3 \), and \( A_6 \) as ancestor for the dog category \( c_6 \) taken from Table 1. Similarly, the Object ancestor categories are plane and ship, with images \( A_4 \) as the ancestor for the plane category \( c_1 \), and image \( A_3 \) as the ancestor for the ship category \( c_9 \).

With these ancestors, we performed 10 independent runs of the algorithms for each of the following combinations: the bird/cat pair (Animal/Animal), plane/truck pair (Object/Object), dog/car pair (Animal/Object), and ship/horse pair (Object/Animal).

**Performance comparison.** In all cases, the 10 independent runs of each algorithm succeeded in (far) less than 7000 generations. Table 2 lists the minimum number of generations (\( \text{min} \text{gen} \)), maximum number of generations (\( \text{max} \text{gen} \)), and mean generations (\( \text{mean} \text{gen} \)) obtained over the 10 independent runs of each algorithm. The convergence graph, plotted in Figure 2, shows the convergence speed of both algorithms for all cases. The horizontal axis of these graphs is the number of generations, and the vertical axis is the average log probability of the target category obtained for these 10 independent runs.

**TABLE 2: Comparison of classic_EA and adapted_EA in generating adversarial images for the target scenario for 4 different Ancestor/Target combinations (\( A_a \) is the ancestor image in \( c_a \) used in the experiments) to fool VGG-16 trained on CIFAR-10. The results are over the 10 independent runs of each algorithm.**

| Ancestor/Target | Algorithms | \( \text{min} \text{gen} \) | \( \text{max} \text{gen} \) | \( \text{mean} \text{gen} \) |
|----------------|-----------|----------------|----------------|----------------|
| bird (\( A_3 \))/cat | classic_EA | 1726 | 2433 | 2172.9 |
|                  | adapted_EA | 1353 | 2177 | 1629.2 |
| plane (\( A_4 \))/truck | classic_EA | 1311 | 1810 | 1547.5 |
|                  | adapted_EA | 1050 | 1439 | 1194.8 |
| dog (\( A_6 \))/car | classic_EA | 1132 | 1334 | 1199.3 |
|                  | adapted_EA | 811 | 1050 | 907.0 |
| ship (\( A_9 \))/horse | classic_EA | 1972 | 3412 | 2582.1 |
|                  | adapted_EA | 1543 | 3171 | 2377.8 |

**Results and Discussion.** As can be seen in Table 2, "adapted_EA" outperforms "classic_EA" in all cases. The former requires fewer generations than the latter to obtain adversarial images with a confidence of 0.95. Figure 2 confirms that "adapted_EA" converges faster than "classic_EA". The graphs indicate that both algorithms apparently exhaust most of their generations to find the correct regions and/or pixels to modify. Once done, their learning curves accelerate drastically, still with "adapted_EA" leading the race against "classic_EA".

Although both algorithms start the search with the same 160 identical images, their respective performances differ substantially, as a consequence of their distinct updating process of the population. Indeed, "adapted_EA" starts these updates for the whole population, except for the elite individuals passed unchanged to the next generation, and does so right after the 1st generation. However, "classic_EA" only updates 20% of its population in each generation. Changing only 32 individuals, as opposed to changing 150 individuals, makes it much slower for the classic version compared to its adapted competitor. These results not only legitimize the choices made in our earlier work (1, 21–23, 31), but also provide some evidence that for similar exploration problems with a starting point made of the same individuals (hence, not only for the construction of images that are adversarial for a CNN), the generic selection and mutation process adopted in "adapted_EA" (algorithm 2) shortens the learning period of the algorithm and enhances the convergence speed.

We complete this comparative analysis by assessing the potential differences in behavior between the adversarial images created by each version of the EA. To this purpose, we computed the Kullback-Leibler divergence [36] between the probability densities derived from the normalized histograms of the pixel modifications induced by each of them. In all cases, the values (averaged over the ten independent runs) of the Kullback-Leibler divergences were negligible (they vary between 2.24e–04 and 5.17e–03), indicating that the noise created by one version of the EA significantly differs from the noise created by the other. Hence, while both versions of the EA create adversarial images, the modifications introduced by each of them differ strongly, although both these modifications introduced by each EA on the one hand, as well as their differences on the other hand, are not perceptible by a human.

**IV. THE ADVERSARIAL IMAGES OBTAINED BY EA\(_{target}\) VGG-16**

As a result of Section III, from now on we only consider the "adapted_EA" version of our evolutionary algorithm, namely EA\(_{target}\) VGG-16. For an ancestor \( A_a \) in a category \( c_a \), and the target scenario for category \( c_t \), one defines \( D_{a,t}(A_a) = EA\(_{target}\) VGG-16\((A_a, c_t)\)\), provided that the algorithm terminates successfully. One writes more simply \( D_{a,t} \), or even \( D_t \), if there is no ambiguity about the choice of the ancestor \( A_a \) chosen in category \( c_a \) (mutatis mutandis in Sections V and VI).

**A. WITH ONE ANCESTOR PER CATEGORY**

From Table 1, we pick the ancestor image \( A_a \) in category \( c_a \) and perform 10 independent runs (with random seed values) of EA\(_{target}\) VGG-16 for all nine possible target categories \( c_t \neq c_a \).

An example of the quality of the obtained adversarial images is highlighted by the comparison between the dog ancestor \( A_6 \) of Table 1, and its corresponding 9 evolved adversarial images \( D_t \), with \( t \neq 6 \) (obtained after the
first of the 10 independent runs of the EA) as shown in Figure 3. More generally, Figure 11 (Appendix A) contains the adversarial images obtained by the first successful run out of the ten independent runs of EA_{target,VGG-16} for each of the ancestor images in Table 1, and Table 5 (Appendix A) gives their respective label values.

This example illustrates that by slightly changing many pixels instead of heavily changing a few pixels, our approach enhances the indistinguishability between the adversarial image and the ancestor image. In particular, our method differs substantially from [24], [25], [27], where a small fraction of pixels is changed, but at the cost of being noticeable for a human without difficulty (see Figure 1, Section I).

For the ancestor image \(A_a\) (from Table 1) in category \(c_a\) specified in its \(a^{th}\) row, the \(t^{th}\) column of Figure 4 gives the average number of generations required by EA_{target,VGG-16} to terminate, computed over 10 independent runs. In the four ancestor/target combinations, this number is followed by a symbol \((\star x)\) or \((\star x, \star y)\). These symbols indicate that the algorithm did not achieve the \(\tau = 0.95\) threshold value within 7000 generations for \(x\) of the 10 runs, and therefore terminated without success for the corresponding seed values. The \(c_{\star}\)-label values of the corresponding best descendant images remained stuck at a local optimum \(< 0.95\), whose quality is also indicated by the symbol. In the case of symbol \((\star x)\), this local optimum was quite close to 0.95 (not less than 0.9370 actually; we call quasi-adversarial the corresponding images produced by EA_{target,VGG-16}). In the case of symbol \((\star x, \star y)\), the complementary number \(y\) specifies the number of runs among the \(x\) unsuccessful runs for which the local optimum remained very low (between circa \(10^{-4}\) to \(10^{-5}\)).

For each \(1 \leq a \leq 10\), the "Row Average" value, displayed in the rightmost column of the \(a^{th}\) row, indicates the average number of generations required to perform our attack on ancestor \(A_a\) in category \(c_a\) for all \(c_t \neq c_a\) (Mutatis mutandis
FIGURE 3: From the left, comparison of the ancestor $A_6$ in the 6th position with the adversarial images $D_t$ in the $t^{th}$ position ($t \neq 6$). VGG-16 classifies $A_6$ in the dog category with probability 0.9996386, and classifies $D_t$ in the target category $c_t$ with probability $\geq 0.95$.

| plane ($A_0$) | car ($A_1$) | bird ($A_2$) | cat ($A_3$) | deer ($A_4$) | dog ($A_5$) | frog ($A_6$) | horse ($A_7$) | ship ($A_8$) | truck ($A_9$) | Row Average |
|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 1095         | 1080        | 494         | 2700       | 879         | 690         | 454         | 318         | 145         |             | 692         |
| 1059         | 665         | 341         | 6233‡5†8‡9 | 882         | 529         | 221         | 182         | 663         | 494         | 830         |
| 2188         | 1246        | 250         | 460        | 460         | 460         | 309         | 383         | 383         | 383         | 1850        |
| 702          | 768         | 263         | 925        | 129         | 129         | 309         | 834         | 383         | 383         | 2793        |
| 613          | 1545        | 217         | 6921‡9      | 938         | 717         | 303         | 834         | 834         | 834         | 2793        |
| 337          | 543         | 113         | 559        | 397         | 536         | 371         | 834         | 834         | 834         | 2793        |
| 798          | 676         | 411         | 2719       | 280         | 536         | 371         | 834         | 834         | 834         | 2793        |
| 147          | 422         | 526         | 872        | 971         | 536         | 371         | 834         | 834         | 834         | 2793        |
|              |             |             |            |             |             |             |             |             |             |             |
| Column Average | 872         | 1085        | 449         | 931         | 841         | 1415        | 544         | 974         | 1238        | 1373        |

FIGURE 4: EA target, VGG-16's performance on all possible ancestor/target combinations with one ancestor per category. The rows give the ancestor category $c_a$ (and the specific ancestor $A_a$ in $c_a$), the columns indicate the target class $c_t$, and the cell values indicate the average number of generations required by EA target, VGG-16 to terminate, computed on 10 independent runs.

the "Column Average" value displayed in the bottom row of the $t^{th}$ column).

Our EA showed a success rate of 100% since all possible target categories were achieved with at least one of the ten runs for the considered ancestors. Still, some attacks are easier than others. The ancestor image for which $EA_{target, VGG-16}^{L_2}$ requires the least amount of effort in general is the horse ancestor image $A_8$, and bird ($c_3$) is the easiest target category regardless of the ancestor category (with the considered ancestor images at least). At the other end of the scale are the deer ancestor image $A_5$ and the bird ancestor image $A_3$ for which $EA_{target, VGG-16}^{L_2}$ requires the largest amount of effort in general, while dog ($c_6$), truck ($c_9$) and ship ($c_8$) are the hardest target categories. These correspond precisely to the categories (and the ancestors) for which some runs of $EA_{target, VGG-16}^{L_2}$ terminated without having created an appropriate adversarial image within 7000 generations. Indeed, out of the altogether 900 attacks (10 runs for each of the 90 ancestor/target combinations) performed by $EA_{target, VGG-16}^{L_2}$, Figure 4 shows that only 31 did not succeed. It is worth noting the homogeneity and non-diversity of the quality of rare unsuccessful cases. For such an unsuccessful ($c_a, c_t$) combination, either the local optimum is close to the $\tau = 0.95$ value for all failed cases (this occurs for the nine unsuccessful runs of the (bird ($A_4$), dog) combination), or it is very far from this threshold value for all failed cases (this occurs for the 22 unsuccessful runs with the deer ($A_5$) ancestor for the car, ship and truck targets).

Therefore, as a consequence of this study with one ancestor $A_a$ per category $c_a$, our experiments show that the probability that $EA_{target, VGG-16}^{L_2}$ terminates successfully for a given run is 96.56%, and that its termination requires between 290 and 2793 generations on average.

B. WITH 50 DISTINCT ANCESTORS PER CATEGORY

To further evaluate the efficiency of our attack beyond the case of a single ancestor $A_a$ per category $c_a$, as described in Subsection IV-A, and to assess the importance of a specific ancestor chosen in a given category, we considered 50 distinct images taken randomly (from the CIFAR-10 testing set) in each of the 10 categories $c_a$. Unlike the 10 independent runs per ancestor of Subsection IV-A, we considered that running $EA_{target, VGG-16}^{L_2}$ with one single run per ancestor was enough to make our point. So in total, we performed $50 \times 10 \times 9 = 4500$ attacks with $EA_{target, VGG-16}^{L_2}$.

Figure 5, which summarizes the outcome of this experiment, is to be interpreted in a similar way as Figure 4, with the difference that the averages are computed over the 50 ancestors per category.
category \(c_a\). Note also that the \((∗x)\) and \((∗x, ∗y)\) symbols added to some cell values for a given \((c_a, c_t)\) scenario have a different interpretation in Figure 5 compared to Figure 4, since they apply globally to different ancestors here, as opposed to applying to different runs performed on the same ancestor in Figure 4.

Performance differs again from one category to another. The ancestor categories for which \(E^{target}_{L_2}, VGG-16\) requires the least amount of effort in general are the \textit{frog}, \textit{cat}, and \textit{dog} categories. In addition, \(E^{target}_{L_2}, VGG-16\) achieves the target categories \textit{bird} and \textit{deer} fairly fast, regardless of the ancestor categories. Conversely, the ancestor categories \textit{car} and \textit{horse} are those for which \(E^{target}_{L_2}, VGG-16\) requires the largest amount of effort in general, while the \textit{car} and \textit{truck} are the hardest target categories.

In this context, the comparison of these results with those of Figure 4 shows the relevance for \(E^{target}_{L_2}, VGG-16\) of the specific ancestor image chosen in a given category \(c_a\). Indeed, while, for instance, the specific ancestor \(A_8\) in the \textit{horse} category was optimal in a sense (achieving all possible target categories in 290 generations on average), this property did not extend to the \textit{horse} category as a whole as just seen. \textit{A contrario}, for instance, the combination \textit{(deer, truck)} with the ancestor \(A_5\) in the \textit{deer} category was (with 6939 generations on average) the toughest to achieve among all trials of Subsection IV-A. It is reasonably easy to achieve in general (with 1139 generations on average) with the 50 ancestors chosen for our experiment.

Finally, out of the 4500 trials performed by \(E^{target}_{L_2}, VGG-16\), only 87 did not terminate successfully. Therefore, this experiment provides heuristic evidence that one run of \(E^{target}_{L_2}, VGG-16\) has a probability of 98.06% to terminate successfully. To better assess the strength of the failed cases, we ran again the 87 unsuccessful cases 10 times with different seed values: out of them, 28 succeeded in less than 10 runs, while 59 did not. This result, together with the fact that our algorithm required between 461 and 1717 generations on average in this case, and compared to the outcome of the similar experiments performed in the previous Subsection IV-A with other ancestors, further sustains the impact of the specific ancestor \(A_a\) taken in a given category \(c_a\), and of the seed value used to run the EA. It also shows that the success rate of our attack, namely the capacity for \(E^{target}_{L_2}, VGG-16\) to terminate successfully for at least one of ten runs out of a small number of trials, is \(≥ 98.68\%\).

V. ROBUSTNESS OF \(E^{target}_{L_2}, VGG-16\) AGAINST FILTERS

For the reasons given in the introduction to this paper (Section I), the study undertaken in this section essentially amounts to checking whether adding filters may prevent VGG-16 from misclassification, or may reduce this risk to some extent, when facing an adversarial image created by \(E^{target}_{L_2}, VGG-16\).

A. SELECTION OF FILTERS

Although a large list of filters exists, we focus on the following four filters that have a significant impact on images [37, Chapters 7 and 8].

The \textit{inverse filter} \(F_1\) replaces all the colours by their complementary colours. This operation is performed pixel for pixel by subtracting the RGB value (255, 255, 255) of white by the RGB value of that pixel.

The \textit{Gaussian blur filter} \(F_2\) uses a Gaussian distribution to calculate the kernel, \(G(x, y) = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}\), where \(x\) is the distance from the origin on the \(x\)-axis, \(y\) is the distance from the origin on the \(y\)-axis and \(\sigma\) is the standard deviation of the Gaussian distribution. By design, the process gives more priority to the pixels in the center, and blurs around it with a lesser impact as one moves away from the center.

The \textit{median filter} \(F_3\) is used to reduce noise and artifacts in a picture. Although under some conditions it can reduce noise while preserving the edges, this does not really occur for small images such as those considered here. In general, one selects a pixel and computes the median of all the surrounding pixels.

The \textit{unsharp mask filter} \(F_4\) enhances the sharpness and contrast of the images. The unsharp-masked image is obtained by blurring a copy of the image using a Gaussian blur, which is then weighted and subtracted from the original image.

Any filter \(F\), or any combination of filters \(F_{i_1}, F_{i_2}, \ldots, F_{i_k}\), operating successively (in that order) on an image \(\mathcal{I}\), creates a filtered image \(F(\mathcal{I})\) or \(F_{i_k} \circ \cdots \circ F_{i_2} \circ F_{i_1}(\mathcal{I})\).

We make use of these four filters \(F_1, F_2, F_3,\) and \(F_4\) either individually or as the combination \(F_5 = F_3 \circ F_4\). The reason for the choice of the latter \(F_3 \circ F_4\) is that \(F_4\) is used to amplify and highlight detail, while \(F_3\) is used to remove noise from an image without removing detail. Therefore, a combination of these filters can remove the noise created by the EA while maintaining a high level of detail. Moreover, because the computations are performed on images of size 32 \(\times\) 32, we take a filter size \(f = 1\) for \(F_1\) and \(f = 3\) for the others.

For each \(F = F_k, 1 \leq k \leq 5\), we then challenge VGG-16 with these 100 filtered images \(F(A_a)\) and \(F(D_{a,t}(A_a))\).

The complete classification and the corresponding label values output by VGG-16 for \(F(A_a)\) and \(F(D_{a,t}(A_a))\) for the five considered filters and for all \((c_a, c_t)\) combinations are given in Tables 6 to 10 (Appendix A). In these tables, an image is classified as belonging to category \(c\) if \(c\) has the largest label value outputted by VGG-16 among all categories.
The following indicators quantitatively assess the aforementioned issues for each filter $F_k$, with the considered ancestors and adversarial images. These indicators take integer values, and we specify their theoretical bounds (which clearly depend on the number of ancestors, and on the number of target categories considered in this study).

For each $1 \leq a \leq 10$, we first define $\rho_k(A_a)$ as the number of target categories $c_t$ such that VGG-16 classifies $F_k(D_{a,t}(A_a))$ (including potentially $D_{a,a}(A_a) = A_a$) back to the ancestor category $c_a$. One computes $\Sigma_k = \sum_{a=1}^{10} \rho_k(A_a) \in [0, 100]$.

One sets $\delta_k(A_a) = 1$ if $\rho_k(A_a) = 10$, that is, if the filtered ancestor and all filtered adversarial images are classified back to the ancestor category. Otherwise $\delta_k(A_a) = 0$. One computes $\Delta_k = \sum_{a=1}^{10} \delta_k(A_a) \in [0, 10]$.

One sets $\mu_k(A_a) = 0$ if VGG-16 classifies $F_k(A_a)$ back to $c_m$ and $\mu_k(A_a) = 1$ if it does not. One defines $M_k = \sum_{a=1}^{10} \mu_k(A_a) \in [0, 10]$.

Of interest for the target scenario is $\tau_k(A_a)$, the number of $t \neq a$ for which $F_k(D_{a,t}(A_a))$ is classified as belonging to $c_t$ (namely those that "really succeed"), and its sum $T_k = \sum_{a=1}^{10} \tau_k(A_a) \in [0, 90]$.

Finally, we consider $\tilde{T}_k(a)$ to assess the untargeted scenario: $\tilde{T}_k(a)$ counts the number of $t \neq a$ for which $F_k(D_{a,t}(A_a))$ is classified as belonging to $c \neq c_F(A_a)$. One computes its sum $\tilde{T}_k = \sum_{a=1}^{10} \tilde{T}_k(A_a) \in [0, 90]$.

Observe en passant that the inequality $T_k \leq \tilde{T}_k$ may theoretically not hold (as opposed to what happens in the...
absence of any filter, where the corresponding inequality necessarily holds). The reason is that one considers \(c_t \neq c_a\) for the left-hand side of the inequality, and \(c \neq c_{F_t}(A_\mu)\) for the right-hand side. Since the quantities \(c_t\) and \(c_{F_t}(A_\mu)\) may differ, the set whose number of elements is \(\mathcal{T}_k\) may not be included in the set whose number of elements is \(\tilde{\mathcal{T}}_k\).

C. ROBUSTNESS ANALYSIS OF THE ADVERSARIAL \(D_{A,T}(A_\mu)\) AGAINST FILTERS

Let us now proceed to the analysis of Table 3, which provides these quantities resulting from, and summarizing Tables 6 to 10 (Appendix A).

TABLE 3: Indicator values assessing the robustness of adversarial images \(D_{A,T}(A_\mu)\) against filters. For each ancestor \(A_\mu\), computation of \((\mu_k(A_\mu), \delta_k(A_\mu), \mu_k(A_\mu))\) in the 1st row, and of \((\mathcal{T}_k(A_\mu), \tilde{\mathcal{T}}_k(A_\mu))\) in the 2nd row. The last two rows give the sums \(\sum_{k=1}^{10}\) of these quantities.

| \(A_\mu\) | 1 | 2 | 3 | 4 | 5 |
| --- | --- | --- | --- | --- | --- |
| \(A_1\) | (10,1,0) | (0,0,1) | (2,0,0) | (0,0,1) | (7,0,0) |
| \(A_2\) | (1,0,0) | (3,0,0) | (9,0,0) | (3,0,0) | (10,1,0) |
| \(A_3\) | (7,0,0) | (10,1,0) | (10,1,0) | (1,0,0) | (10,1,0) |
| \(A_4\) | (3,0,0) | (10,1,0) | (10,1,0) | (1,0,0) | (10,1,0) |
| \(A_5\) | (1,0,0) | (10,1,0) | (10,1,0) | (1,0,0) | (10,1,0) |
| \(A_6\) | (0,0,1) | (0,0,1) | (1,0,1) | (1,0,0) | (4,0,0) |
| \(A_7\) | (5,0,0) | (10,1,0) | (10,1,0) | (2,0,0) | (10,1,0) |
| \(A_8\) | (0,0,1) | (0,0,1) | (1,0,1) | (1,0,0) | (4,0,0) |
| \(A_9\) | (6,0,0) | (10,1,0) | (10,1,0) | (2,0,0) | (10,1,0) |
| \(A_{10}\) | (0,0,1) | (0,0,1) | (1,0,1) | (1,0,0) | (4,0,0) |

Looking at \(\Sigma_k\) shows that, although all filters \(F_1, \ldots, F_5\) bring some filtered images back to \(c_\alpha\), the unsharp mask \((F_1)\) and the inverse \((F_5)\) filters are less efficient in this regard. In contrast, the three other filters bring back a majority of the filtered images back to \(c_\alpha\). The median \((F_3)\) filter and foremost the combination \((F_5)\) of the Unsharp and Median filters are highly effective, since more than 80% of all filtered images are classified back to \(c_\alpha\). The three filters \(F = F_2, F_3,\) and \(F_5\) are also those that bring all filtered images back to \(c_\alpha\) for 5 (in the case of \(F_5\)), 6 (in the case of \(F_2\)) and 7 (in the case of \(F_3\)) ancestors, including a fortiori the filtered ancestor.

Consistently, the consideration of \(\mathcal{T}_k\) and of \(\tilde{\mathcal{T}}_k\) shows that \(E_{L_2}^{\text{target,VGG-16}}\) resists highly efficiently against the Unsharp mask filter \(F_1\), as 95% (86 out of 90) filtered images remain adversarial for the target scenario (with target label values no less than 0.5505, see Table 9), and 95% (86 out of 90) filtered images are adversarial for the untargted scenario. Our EA is also significantly efficient against the inverse filter \(F_5\), as 17% (16/90) filtered images remain adversarial for the target scenario (with target label values \(\geq 0.4415\), see Table 6), and 53% (48/90) are adversarial for the untargted scenario.

On the other hand, the Gaussian blur \((F_2)\), the median \((F_3)\), and the median and unsharp combined \((F_5)\) filters are effective to a far larger extent against EA \(E_{L_2}^{\text{target,VGG-16}}\), with \(F_3\) and \(F_5\) being particularly efficient at removing the adversarial property of the descendant images. Indeed, only three filtered adversarial images (hence 3% of all filtered) remain adversarial for the target scenario for each of these two filters (with target label values \(\geq 0.4978\) for \(F_3\) and \(\geq 0.8131\) for \(F_5\); see Tables 8 and 10). For the untargted scenario finally, the proportion of filtered images that are adversarial drops to 13% (12/90) for \(F_3\), and to 12% (11/90) for \(F_5\).

This study proves that the inverse \((F_1)\) and the unsharp mask \((F_1)\) filters are largely inefficient against our EA, but that the Gaussian \((F_2)\), and foremost, the median \((F_3)\) and the combination \((F_5 = F_3 \circ F_1)\) of the unsharp mask and the median filters render our EA-based attack significantly less effective for both the target scenario and the untargted scenario, at least with the ancestor images considered.

VI. THE FILTER-ENHANCED \(F\)-FITNESS FUNCTION

The results of the previous section lead to the conception of a new fitness function that naively forces the EA to create adversarial images that remain adversarial (in the sense of Subsection II-B) once filtered. For a filter \(F\), the filter-enhanced \(F\)-fitness function is obtained as the following variant of the fitness function defined in Equation (2):

\[
fit_{L_2}^F(\text{ind}, g_p) = A(g_p, \text{ind})\alpha_{\text{ind}}[c_t] + \alpha_{F(\text{ind})}[c_t] - B(g_p, \text{ind})L_2(\text{ind}, A),
\]

(5)

where the component \(\alpha_{F(\text{ind})}[c_t]\) measures the probability that an individual filtered with \(F\) is classified as the target category. One obtains \(E_{L_2}^{\text{target,VGG-16}}\) from \(E_{L_2}^{\text{target,VGG-16}}\) by updating the fitness function accordingly. The termination and termination with success criteria are the same as in Subsection III-B.

Since \(F_5 = F_3 \circ F_1\) is not only highly efficient against \(E_{L_2}^{\text{target,VGG-16}}\), but is the filter that reverts the largest proportion (89%) of images \(D_{A,T}(A_\mu)\) back to \(c_\alpha\), we limit this study to this case.
A. RUNNING EA\(_{L_2,F_5}\) WITH ONE ANCESTOR PER CATEGORY

For \(1 \leq a \leq 10\), one performs 10 independent runs of EA\(_{L_2,F_5}\) on the ancestor \(A_a\) in category \(c_a\) given in Table 1. If EA\(_\text{target},\text{VGG-16}\) terminates successfully, one writes \(D_{\text{F}_5}(A_a)\) for the first adversarial image obtained by EA\(_\text{target},\text{VGG-16}\) in less than 7000 generations. By construction, this image and its by \(F_5\) filtered version are classified by VGG-16 as belonging to the target category \(c_t\) with probability \(\geq 0.95\), while remaining so close to \(A_a\) for a human eye that no one would notice any difference.

Figure 7 pictures the adversarial images \(D_{\text{F}_5}(A_6)\) obtained that way for the dog ancestor \(A_6\) (all first runs succeeded for the dog ancestor).

For the ancestor image \(A_a\) (taken from Table 1) in category \(c_a\) specified in its \(a^{th}\) row, the \(i^{th}\) row of Figure 8 gives the average number of generations required by EA\(_{L_2,F_5}\) to terminate, computed over 10 independent runs. With a terminology adapted from the one used in Figure 4, this number is followed by a symbol \((x, 1\gamma, \hat{1}z)\) in 5 of the 90 cells. The occurrence of this symbol means that the algorithm did not terminate successfully for \(x\) out of the 10 runs (i.e., the average value = 7000 if \(x = 10\)). Not succeeding means that the \(c_i\)-label value of the most performing descendant images \(D\) or of the filtered image \(F_5(D)\) is stuck at a local optimum < 0.95. The symbols \(1\gamma\) and \(\hat{1}z\) measure the quality of these local optima. \(1\gamma\) (respectively, \(\hat{1}z\)) counts the number of runs among the \(x\) unsuccessful ones for which the local optimum for descendant \(D\) (respectively, \(F_5(D)\)) remained very low (between \(10^{-3}\) and \(10^{-6}\)).

Of the 900 performed runs, 38 did not terminate successfully, and 3 out of the 90 possible ancestor/target scenarios were not achieved, namely the pairs \((plane(A_1), deer), (bird(A_2), car), (horse(A_8), ship)\). Therefore, the experiments show a success rate of EA\(_{L_2,F_5}\) of 96.66\%, and a probability that the algorithm terminates successfully for a given run of 95.77\%.

Comparing Figure 8 to Figure 4, when all 10 runs terminate successfully for both EA\(_\text{target},\text{VGG-16}\) and EA\(_{L_2,F_5}\) for an \((\text{ancestor}(A_a), \text{target})\) pair (83 cases altogether), the latter algorithm usually requires more generations than the former on average (with three notable exceptions, namely the \((ship(A_9), deer), (ship(A_8), dog)\) and the \((truck(A_{10}), cat)\) pairs for which it needs 10\%, 18\% and 13\% fewer generations). The fact that, for the 80 remaining pairs, EA\(_{L_2,F_5}\) requires between 1.12 and 3.87 (depending on the pair considered) times more generations than EA\(_\text{target},\text{VGG-16}\) to terminate successfully is not surprising since there are 3 and no longer 2 criteria to fulfill.

For all 87 combinations \((\text{ancestor}(A_a), \text{target})\) for which EA\(_\text{target},\text{VGG-16}\) terminated successfully in at least one of the 10 independent runs, Figure 12 (Appendix B) displays the first adversarial image \(D_{\text{F}_5}(A_a)\) obtained by EA\(_\text{target},\text{VGG-16}\) (with \(D_{\text{F}_5}(A_a) = A_a\) repeated on the diagonal for the sake of consistency and comparison), and Table 11 (Appendix B) gives the corresponding label values.

B. RUNNING EA\(_\text{target},\text{VGG-16}\) WITH 50 ANCESTORS PER CATEGORY

For the sake of completeness, we performed the same experiments as in Subsection IV-B with the same 500 ancestor images (50 ancestor images per ancestor category), but with EA\(_\text{target},\text{VGG-16}\) instead of EA\(_{L_2,F_5}\). Figure 9 shows the outcome. Of the 4500 attacks, 543 were unsuccessful; hence, the success rate of EA\(_\text{target},\text{VGG-16}\) was 88\%, and required between 1250 and 2404 generations on average.

Comparing Figure 4 with Figure 8 and Figure 5 with Figure 9 shows that EA\(_\text{target},\text{VGG-16}\) usually requires more generations than EA\(_{L_2,F_5}\) to construct adversarial images, which is to be expected since EA\(_\text{target},\text{VGG-16}\) must satisfy not two, but three conditions.

C. ROBUSTNESS OF \(D_{\text{F}_5}(A_A)\) AGAINST VGG-16: \(F_K\) FOR ALL FILTERS

Using again the images of Figure 12 (Appendix B) obtained as described in Subsection VI-A, the ancestor \(A_a\) and the corresponding adversarial images \(D_{\text{F}_5}(A_A)\) were then tested against all five filters of Subsection V-A. Figure 10 shows the outcome of this process for the dog ancestor \(A_6\) and the adversarial images \(D_{\text{F}_5}(A_6)\).

These filtered images are given to VGG-16 for classification (see Appendix B, Table 11 for \(F_5\), and Table 12 for \(F_1, F_2, F_3, \text{ and } F_4\), with \(D_{\text{F}_5}(A_A) = A_a\) to ease the notations).

Mutatis mutandis, Table 4 is obtained in a similar way as in Table 3. Note that the upper bounds of the indicators are impacted by the fact that three combinations \((c_A(A_a), c_t)\) were not achieved. Indeed, one has \(0 \leq \rho_{\text{F}_k}(A_a) \leq 9\) for \(a = 1, 3, 8\), and \(0 \leq \rho_{\text{F}_k}(A_a) \leq 10\) otherwise. One writes \(\delta_{\text{F}_k}(A_a) = 1\) if the filtered ancestor and all filtered adversarial images are classified back to the ancestor category whenever possible. Consistently, one has \(0 \leq \tau_{\text{F}_k}(A_a), \tau_{\text{F}_k}(A_a) \leq 8\) for \(a = 1, 3, 8\), and \(0 \leq \Delta_{\text{F}_k}(A_a), \Delta_{\text{F}_k}(A_a) \leq 9\) otherwise. As a consequence, one has \(0 \leq \Sigma_{\text{F}_k} \leq 97, 0 \leq \Delta_{\text{F}_k}, \Delta_{\text{F}_k} \leq 10\), and \(0 \leq \mathcal{T}_{\text{F}_k}, \mathcal{T}_{\text{F}_k} \leq 87\).

Table 4 clearly shows that the produced images are not only adversarial for \(F_k\) but also for \(F_3\) and \(F_4\) to a large extent for the target scenario (88\% and 84\%, respectively), and for the untargeted scenario (89\% and 88\% respectively). Additionally, 56\% of these images were efficient against \(F_1\)
FIGURE 7: From left to right, comparison of the ancestor $A_0$ in the $6^{th}$ position with the adversarial images $T_{6,1}^{F_6}(A_0)$ in the $t^{th}$ position ($t \neq 6$).

FIGURE 8: $E_{L_2,F_5}^{\text{target, VGG-16}}$'s performance on all possible ancestor/target combinations. The rows give the ancestor categories $c_a$ (and the specific ancestor $A_0$ in $c_a$), the columns indicate the target class $c_t$, and the cell values give the average number of generations required by $E_{L_2,F_5}^{\text{target, VGG-16}}$ to terminate, computed on 10 independent runs.

FIGURE 9: 500 attacked images with 50 samples per ancestor class. Rows correspond to source classes, columns correspond to target classes, and cell values correspond to the average number of generations needed by $E_{L_2,F_5}^{\text{target, VGG-16}}$ to terminate.

for the untargeted scenario, while this percentage dropped to 23% with $F_2$.

This study shows that the $E_{L_2,F_5}^{\text{target, VGG-16}}$ attack, designed to be robust against $F_5$, is also robust to some significant extent against all individual filters considered for the untargeted scenario, and the Gaussian filter ($F_2$) is the most efficient at removing the adversarial character of the constructed images.
designed a variant $EA_{target,VGG-16}^{L_2,F}$ of the EA, specifically designed to fool VGG-16 and VGG-16 composed with a filter $F$, and demonstrated the efficiency of the produced adversarial images not only against the specific chosen filter, but also against other filters.

The results of this paper lead to a series of additional studies. First, from a pure EA point of view, we intend to look for methods to accelerate our algorithm, including early warnings that indicate a high probability of unsuccess of a given run. In this line of thought and more specifically for the construction of adversarial images, other efficiency improvement methods will be studied, such as the restriction of the zones on which the EA should focus its noise creation, or the search for optimized paths between $c_a$ and $c_t$ for a given ancestor via auxiliary categories. Second, since the small $32 \times 32$ images in this study are naturally grainy, we intend to apply our attack to larger images, not only those of ImageNet [3], but foremost high resolution images arising from different horizons (e.g., satellite, medical, or artistic images), which may lead to combinations $C \circ F$ for functions $F$ that are no longer filters. Finally, our EA-based attack can potentially be extended to other domains (natural language processing, speech recognition, etc.) beyond the computer vision applications mentioned in the first paragraph of the Introduction.

VII. CONCLUSION

This study, which substantially complements our previous works [1], [21]–[23], [31], successfully addresses the four issues raised in the introduction. First, we proved that the conceptual originality of our generic evolutionary algorithm leads to a competitive advantage in terms of performance compared to the classical EA approach. Then, an extensive experimental study showed the intrinsic efficiency of our algorithm, $EA_{target,VGG-16}^{L_2}$, at constructing adversarial images for the target scenario performed against VGG-16 with images from CIFAR-10. We then challenged the adversarial images obtained against a series of filters, and finally
FIGURE 11: For $1 \leq a \leq 10$, the image on the diagonal of the $a^{th}$ row is the ancestor $A_a$ (recovered from Table 1) classified by VGG-16 as belonging to the category $c_a$, and the picture in the $t^{th}$ column, with $t \neq a$, is the adversarial picture $D_{a,t}(A_a) = \text{EA}_{L_2}^{\text{target}, \text{VGG-16}}(A_a, c_t)$ classified by VGG-16 as belonging to $c_t$, obtained after the first of the 10 independent runs.

TABLE 5: For $\mathcal{C} = \text{VGG-16}$, each of the cell in $(a, t)^{th}$-position contains a pair (maximum label value, corresponding class) given by $\mathcal{C}$ for $D_{a,t}(A_a)$ (with $D_{a,a}(A_a) = A_a$).

| plane($A_1$) | c1  | c2  | c3  | c4  | c5  | c6  | c7  | c8  | c9  | c10 |
|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| plane($A_1$) | 0.6900 | 0.9506 | 0.9501 | 0.9500 | 0.9501 | 0.9500 | 0.9502 | 0.9504 | 0.9537 | 0.9531 |
| car($A_2$)  | 0.9519 | 0.9999 | 0.9546 | 0.9515 | 0.9534 | 0.9509 | 0.9508 | 0.9508 | 0.9509 | 0.9502 |
| bird($A_3$) | 0.9502 | 0.9999 | 0.9501 | 0.9511 | 0.9509 | 0.9517 | 0.9517 | 0.9523 | 0.9506 | 0.9510 |
| cat($A_4$)  | 0.9514 | 0.9507 | 0.9512 | 0.9998 | 0.9510 | 0.9519 | 0.9543 | 0.9503 | 0.9514 | 0.9552 |
| deer($A_5$) | 0.9524 | 0.9501 | 0.9507 | 0.9514 | 0.9999 | 0.9545 | 0.9520 | 0.9504 | 0.9560 | 0.9510 |
| dog($A_6$)  | 0.9516 | 0.9502 | 0.9529 | 0.9518 | 0.9501 | 0.9996 | 0.9502 | 0.9512 | 0.9508 | 0.9518 |
| frog($A_7$) | 0.9519 | 0.9528 | 0.9501 | 0.9510 | 0.9521 | 0.9527 | 0.9521 | 0.9521 | 0.9552 | 0.9515 |
| horse($A_8$) | 0.9502 | 0.9523 | 0.9503 | 0.9568 | 0.9521 | 0.9510 | 0.9998 | 0.9587 | 0.9514 | 0.9514 |
| ship($A_9$) | 0.9504 | 0.9543 | 0.9584 | 0.9506 | 0.9500 | 0.9516 | 0.9517 | 0.9505 | 0.9996 | 0.9504 |
| truck($A_{10}$) | 0.9525 | 0.9532 | 0.9518 | 0.9517 | 0.9557 | 0.9511 | 0.9516 | 0.9507 | 0.9517 | 0.9984 |
### TABLE 6: For $C = \text{VGG-16}$, the cell in $(a, t)^{\text{th}}$-position gives (top part) the $c_a$-label value and the $c_t$-label value, and (bottom part) the maximum label value and corresponding class of $C \circ F_1$ for $D_{a,t}(A_a)$ (with $D_{a,a}(A_a) = A_a$).

|        | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ | $c_9$ | $c_{10}$ |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| plane  | 0.9923| 0.9987| 0.9888| 0.5912| 0.9845| 0.9857| 0.9436| 0.9007| 0.9806|          |
| car    | 0.9923| 0.9987| 0.9888| 0.5912| 0.9845| 0.9857| 0.9436| 0.9007| 0.9806|          |
| bird   | 0.9963| 0.9877| 0.9878| 0.5912| 0.9845| 0.9857| 0.9436| 0.9007| 0.9806|          |
| cat    | 0.9956| 0.9878| 0.9879| 0.5912| 0.9845| 0.9857| 0.9436| 0.9007| 0.9806|          |
| deer   | 0.9955| 0.9877| 0.9878| 0.5912| 0.9845| 0.9857| 0.9436| 0.9007| 0.9806|          |
| dog    | 0.9952| 0.9877| 0.9878| 0.5912| 0.9845| 0.9857| 0.9436| 0.9007| 0.9806|          |
| frog   | 0.9950| 0.9877| 0.9878| 0.5912| 0.9845| 0.9857| 0.9436| 0.9007| 0.9806|          |
| horse  | 0.9948| 0.9877| 0.9878| 0.5912| 0.9845| 0.9857| 0.9436| 0.9007| 0.9806|          |
| ship   | 0.9946| 0.9877| 0.9878| 0.5912| 0.9845| 0.9857| 0.9436| 0.9007| 0.9806|          |
| truck  | 0.9944| 0.9877| 0.9878| 0.5912| 0.9845| 0.9857| 0.9436| 0.9007| 0.9806|          |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
TABLE 7: For $C = VGG-16$, the cell in $(a, t)^{th}$-position gives (top part) the $c_a$-label value and the $c_t$-label value, and (bottom part) the maximum label value and corresponding class of $C \odot F_2$ for $D_{a,t}(A_A)$ (with $D_{a,a}(A_A) = A_A$).

|          | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ | $c_9$ | $c_{10}$ |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| plane ($A_1$) | 0.2591 | 0.2052 | 0.1978 | 0.1256 | 0.2017 | 0.1385 | 0.1097 | 0.1415 | 9.1e-02 | 3.94e-05 |
| car | 0.2591 | 0.5786 | 0.1627 | 4.95e-03 | 5.82e-04 | 1.89e-02 | 3.08e-04 | 1.51e-04 | 0.7307 | 0.5850 |
| ship | 0.4463 | 0.5786 | 0.3455 | 0.5941 | 0.6175 | 0.4143 | 0.4057 | 0.7010 | 0.7307 | 0.5850 |
| cat | 0.29e-03 | 0.9988 | 9.92e-02 | 5.78e-03 | 0.4091 | 0.1245 | 5.40e-02 | 1.37e-02 | 3.80e-02 | 0.9989 |
| bird | 0.33e-02 | 0.7795 | 0.7865 | 0.4091 | 0.7438 | 0.5196 | 0.9224 | 0.8378 | 0.9989 |
| horse | 0.3924 | 0.7879 | 0.9989 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 |
| truck | 0.9996 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 |
| deer | 0.9876 | 0.9959 | 0.9743 | 0.9992 | 0.9955 | 0.9691 | 0.9983 | 0.9723 | 0.9968 | 0.9917 |
| cat | 0.9876 | 0.9959 | 0.9743 | 0.9992 | 0.9955 | 0.9691 | 0.9983 | 0.9723 | 0.9968 | 0.9917 |
| dog | 0.9997 | 0.9985 | 0.9989 | 0.9988 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 |
| cat | 0.9997 | 0.9985 | 0.9989 | 0.9988 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 |
| frog | 0.9997 | 0.9985 | 0.9989 | 0.9988 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 |
| horse | 0.9994 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 |
| ship | 0.9994 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 |
| truck | 0.9994 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 |

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI: 10.1109/ACCESS.2021.3131255, IEEE Access
| plane (A₁) | 0.7298 | 0.6388 | 0.3414 | 0.1388 | 0.3576 | 0.2870 | 0.3931 | 0.2510 | 0.1163 | 0.3362 |
|------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| truck (A₂) | 0.3643 | 0.9997 | 0.9734 | 0.9666 | 0.9858 | 0.8656 | 0.9075 | 0.9886 | 0.9607 | 0.9997 |
| cat (A₃)   | 0.5191 | 0.9997 | 0.9734 | 0.9666 | 0.9858 | 0.8656 | 0.9075 | 0.9886 | 0.9607 | 0.9997 |
| bird (A₄)  | 0.9998 | 0.9998 | 0.9999 | 0.9998 | 0.9998 | 0.9998 | 0.9999 | 0.9998 | 0.9998 | 0.9998 |
| deer (A₅)  | 0.9994 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 |
| dog (A₆)   | 0.9994 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 |
| cat (A₇)   | 0.7690 | 0.8235 | 0.5443 | 0.9861 | 0.8479 | 0.7329 | 0.9886 | 0.9352 | 0.9401 | 0.9649 |
| frog (A₈)  | 0.9997 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9999 | 0.9999 | 0.9999 |
| horse (A₉) | 0.9965 | 0.9985 | 0.9985 | 0.9985 | 0.9985 | 0.9985 | 0.9985 | 0.9985 | 0.9985 | 0.9985 |
| ship (A₁₀) | 0.5341 | 0.4759 | 0.8869 | 0.7291 | 0.9897 | 0.7917 | 0.6974 | 0.6649 | 0.9924 | 0.9924 |

**TABLE 8:** For $C = \text{VGG-16}$, the cell in $(a, t)^{th}$-position gives (top part) the $c_a$-label value and the $c_t$-label value, and (bottom part) the maximum label value and corresponding class of $C \circ F$ for $D_{a,t}(A_a)$ (with $D_{a,a}(A_a) = A_a$).
TABLE 9: For $C = $ VGG-16, the cell in $(a, t)$th-position gives (top part) the $c_a$-label value and the $c_t$-label value, and (bottom part) the maximum label value and corresponding class of $C \circ F_d$ for $D_{a, t}(A_d)$ (with $D_{a, t}(A_d) = A_d$).

|       | plane | car | bird | cat | deer | dog | frog | horse | ship | truck |
|-------|-------|-----|------|-----|------|-----|------|-------|------|-------|
| $c_a$ | 0.6425| 0.4425| 0.5497| 9.97e-03| 0.9879| 0.9899| 0.9999| 0.9710| 0.9903| 0.9999|
| $c_t$ | 1.10e-02| 1.62e-02| 0.9871| 0.9689| 0.9843| 0.9813| 0.9566| 0.9610| 0.9815| 0.9146|
|       | 5.08e-03| 0.9843| 0.9813| 0.9566| 0.9610| 0.9815| 0.9146| 0.9792| 0.9792|
|       | 6.28e-03| 0.9813| 0.9813| 0.9566| 0.9610| 0.9815| 0.9146| 0.9792| 0.9792|
|       | 1.16e-02| 0.9566| 0.9610| 0.9815| 0.9146| 0.9792| 0.9792|
|       | 2.14e-03| 0.9815| 0.9146| 0.9792|
|       | 4.61e-02| 0.9146| 0.9792|
|       | 4.84e-03| 0.9146| 0.9792|
|       | 10.1109/ACCESS.2021.3131255, IEEE Access

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131255, IEEE Access

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
|    | plane | car   | bird  | cat   | deer  | dog   | frog  | horse  | ship  | truck |
|----|-------|-------|-------|-------|-------|-------|-------|--------|-------|-------|
| c1 | 0.8817| 0.8366| 0.5261| 0.2653| 0.5060| 0.5797| 0.5169| 0.3472 | 0.1666| 0.6224|
| c2 | 0.8817| 4.33e-02| 0.1688| 9.90e-04| 4.08e-04| 3.43e-02| 1.12e-02| 3.96e-04| 0.8131| 3.38e-05|
| c3 | 0.8817| 0.8366| 0.5261| 0.6487| 0.5060| 0.5797| 0.5169| 0.5715| 0.8131| 0.6224|
| c4 |       |       |       |       |       |       |       |        |       |       |
| c5 |       |       |       |       |       |       |       |        |       |       |
| c6 |       |       |       |       |       |       |       |        |       |       |
| c7 |       |       |       |       |       |       |       |        |       |       |
| c8 |       |       |       |       |       |       |       |        |       |       |
| c9 |       |       |       |       |       |       |       |        |       |       |
| c10|       |       |       |       |       |       |       |        |       |       |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
APPENDIX B

FIGURE 12: For $1 \leq a \leq 10$, the image on the diagonal at the $(a, a)^{th}$ position is the ancestor $A_a$ (recovered from Table 1) classified by VGG-16 as belonging to the category $c_a$. The picture in the $(a, t)^{th}$ position, with $t \neq a$, is the adversarial picture $D_{a,t}^{F_5}(A_a) = E_{A_{L_2,F_5}}^{\text{target},VGG-16}(A_a, c_t)$ obtained after the first successful run of the algorithm. Both images $D_{a,t}^{F_5}(A_a)$ and $F_5(D_{a,t}^{F_5}(A_a))$ are classified by VGG-16 as belonging to $c_t$ with a $c_t$-label value $\geq 0.95$. The 3 fully empty pictures correspond to the $(\text{ancestor}(A_a), \text{target})$ combinations for which the algorithm did not terminate successfully for any of the 10 runs.

TABLE 11: For $C = VGG-16$, each of the two parts of the cell in $(a, t)^{th}$-position contains a pair (maximum label value, corresponding class) given by $C$ (top) and by $C \circ F_5$ (bottom) for $D_{a,t}^{F_5}(A_a)$ (with $D_{a,t}^{F_5}(A_a) = A_a$) whenever applicable (3 cells are empty).

|       | plane | car | bird | cat | deer | dog | frog | horse | ship | truck |
|-------|-------|-----|------|-----|------|-----|------|-------|------|-------|
| $c_1$ | plane | 0.69, plane | 0.98, car | 0.95, bird | 0.95, cat | 0.95, dog | 0.95, frog | 0.95, horse | 0.95, ship | 0.95, truck |
| $c_2$ | plane | 0.88, plane | 0.95, car | 0.98, bird | 0.97, cat | 0.98, dog | 0.98, frog | 0.97, horse | 0.98, ship | 0.98, truck |
|       |       |       |       |       |       |       |       |       |       |       |
|       |       |       |       |       |       |       |       |       |       |       |
| $c_3$ | car   | 0.95, plane | 0.99, car | 0.95, bird | 0.95, cat | 0.95, deer | 0.95, dog | 0.95, frog | 0.95, horse | 0.95, ship |
|       |       |       |       |       |       |       |       |       |       |       |
| $c_4$ |       |       |       |       |       |       |       |       |       |       |
|       |       |       |       |       |       |       |       |       |       |       |
| $c_5$ | bird  | 0.99, plane | 0.99, car | 0.99, bird | 0.99, cat | 0.99, deer | 0.99, dog | 0.99, frog | 0.99, horse | 0.99, ship |
| $c_6$ | cat   | 0.99, plane | 0.99, car | 0.99, bird | 0.99, cat | 0.99, deer | 0.99, dog | 0.99, frog | 0.99, horse | 0.99, ship |
|       |       |       |       |       |       |       |       |       |       |       |
| $c_7$ | deer  | 0.99, plane | 0.99, car | 0.99, bird | 0.99, cat | 0.99, deer | 0.99, dog | 0.99, frog | 0.99, horse | 0.99, ship |
| $c_8$ | dog   | 0.99, plane | 0.99, car | 0.99, bird | 0.99, cat | 0.99, deer | 0.99, dog | 0.99, frog | 0.99, horse | 0.99, ship |
| $c_9$ | frog  | 0.99, plane | 0.99, car | 0.99, bird | 0.99, cat | 0.99, deer | 0.99, dog | 0.99, frog | 0.99, horse | 0.99, ship |
| $c_{10}$ |       |       |       |       |       |       |       |       |       |       |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
TABLE 12: For $C = \text{VGG-16}$, each of the 4 parts of the cell in $(a, t)^{th}$-position contains a pair (maximum label value, corresponding class) given, respectively from the top to the bottom, by $C \circ F_1$, $C \circ F_2$, $C \circ F_3$, and $C \circ F_4$ for $D^{\text{Ta}}_{a,t}(A_a)$ (with $D^{\text{Ta}}_{a,t}(A_a) = A_a$) whenever applicable.

|       | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ | $c_9$ | $c_{10}$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|---------|
| plane | 0.99  | 0.95  | 0.97  | 0.54  | 0.65  | 0.95  | 0.92  | 0.87  | 0.97  | 0.97    |
| car   | 0.94  | 0.81  | 0.35  | 0.58  | 0.50  | 0.35  | 0.57  | 0.78  | 0.47  |         |
| truck | 0.72  | 0.82  | 0.98  | 0.64  | 0.87  | 0.90  | 0.40  | 0.98  | 0.48  |         |
| plane | 0.54  | 0.99  | 0.92  | 0.98  | 0.96  | 0.96  | 0.96  | 0.90  | 0.96  |         |
| bird  | 0.97  | 0.76  | 0.92  | 0.99  | 0.55  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99    |
| cat   | 0.91  | 0.97  | 0.80  | 0.99  | 0.91  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99    |
| horse | 0.99  | 0.99  | 0.99  | 0.99  | 0.96  | 0.49  | 0.96  | 0.70  | 0.56  | 0.90    |
| ship  | 0.77  | 0.96  | 0.96  | 0.96  | 0.93  | 0.93  | 0.98  | 0.98  | 0.98  | 0.95    |
| plane | 0.92  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99    |
| bird  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99    |
| cat   | 0.98  | 0.98  | 0.98  | 0.98  | 0.98  | 0.98  | 0.98  | 0.98  | 0.98  | 0.98    |
| deer  | 0.96  | 0.96  | 0.96  | 0.96  | 0.96  | 0.96  | 0.96  | 0.96  | 0.96  | 0.96    |
| dog   | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99    |
| horse | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99    |
| ship  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99    |
| truck | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99  | 0.99    |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
REFERENCES

[1] R. Chitic, N. Deridder, N. Bernard, and F. Leprévost, “Robustness of adversarial images against filters,” in Communications in Computer and Information Science, International Conference on Optimization and Learning (OLA’20), vol. 1443. Springer, 2021.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105, 2012. [Online]. Available: https://papers.nips.cc/paper/2012/file/c399862d39d6b76c8436e92fa4c54b-Paper.pdf

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009, pp. 248–255. [Online]. Available: https://ieeexplore.ieee.org/document/5206848

[4] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016, pp. 2818–2826. [Online]. Available: https://ieeexplore.ieee.org/document/7780677

[5] G. Huang, Z. Liu, Y. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, 2017, pp. 4700–4708. [Online]. Available: https://ieeexplore.ieee.org/document/809726

[6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” CoRR, vol. abs/1409.1556, 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[8] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, “Face recognition: A convolutional neural-network approach,” IEEE Transactions on Neural Networks, vol. 8, no. 1, pp. 98–113, 1997. [Online]. Available: https://ieeexplore.ieee.org/document/554195

[9] S. M. Hizam, W. Ahmed, M. Fahad, H. Akter, I. Sentosa, and J. Ali, “User behavior assessment towards biometric facial recognition system: A semi-neural network approach,” arXiv preprint arXiv:2106.03571, 2021.

[10] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2016, pp. 372–387. [Online]. Available: https://ieeexplore.ieee.org/document/7467366

[11] J. Wu, “Generating adversarial examples in the hash conditions,” CoRR, vol. abs/1908.11332, 2020. [Online]. Available: https://arxiv.org/abs/1908.11332

[12] M. E. J. J. R. Rossi, B. Hitaj, G. Ciocarlie, G. Boracchi, and F. Koushanfar, “Scratch that! An evolution-based adversarial attack against neural networks,” CoRR, vol.abs/1912.02316, 2019. [Online]. Available: https://arxiv.org/abs/1912.02316

[13] Y. Geifman, “cifar-vgg,” 2018. [Online]. Available: https://github.com/geifman/cifar-vgg.

[14] A. Krizhevsky, V. Nair, and G. Hinton, “The CIFAR-10 dataset,” 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html

[15] A. Eiben and J. E. Smith, Introduction to evolutionary computing, Springer, 2003. [Online]. Available: https://www.springer.com/gp/book/9783642072857

[16] R. Chitic, N. Bernard, and F. Leprévost, “Evolutionary algorithms for convolutional neural network visualisation,” in High Performance Computing – 5th Latin American Conference, 2018 (Bucaramanga, Colombia, Sep 23-28, 2018), ser. Communications in Computer and Information Science, vol. 819. Springer, 2019, pp. 138–152. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-17285-9_10

[17] R. Chitic, N. Bernard, and F. Leprévost, “A proof of concept to deceive human and machines at image classification: An extended proof of concept on two scenarios,” Journal of Information and Telecommunication, vol. 5, pp. 121–143, 2021, http://dx.doi.org/10.1080/24751839.2020.1829388.

[18] R. Chitic, N. Bernard, and F. Leprévost, “A proof of concept to deceive human and machines at image classification with evolutionary algorithms,” in Asian Conference on Intelligent Information and Database Systems. Springer, 2020, pp. 467–480.

[19] N. Bernard and F. Leprévost, “How evolutionary algorithms and information hiding deceive machines and humans for image recognition: A research program,” in Proceedings of the OLA’2019 International Conference on Optimization and Learning (Bangkok, Thailand, Jan 29-31, 2019). Springer, Heidelberg, 2019, pp. 12–15.

[20] J. Su, D. V. Vargas, and K. Sakurai, “One pixel attack for fooling deep neural networks,” IEEE Transactions on Evolutionary Computation, vol. 23, no. 5, pp. 828–841, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8601309

[21] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “‘The limitations of deep learning in adversarial settings,’” in 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2016, pp. 372–387. [Online]. Available: https://ieeexplore.ieee.org/document/7467366

[22] J. Su, D. V. Vargas, and K. Sakurai, “One pixel attack for fooling deep neural networks,” IEEE Transactions on Evolutionary Computation, vol. 23, no. 5, pp. 828–841, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8601309

[23] N. Provost and F. Leprévost, “Scratch that! An evolution-based adversarial attack against neural networks,” CoRR, vol.abs/1912.02316, 2019. [Online]. Available: https://arxiv.org/abs/1912.02316

[24] N. Bernard, F. Leprévost, “Adversarial examples for malware detection,” in Proceedings of the OLA 2019 International Conference on Optimization and Learning (Bangkok, Thailand, Jan 29-31, 2019). Springer, Heidelberg, 2019, pp. 12–15.

[25] N. Bernard, F. Leprévost, “A proof of concept to deceive human and machines at image classification with evolutionary algorithms,” in Asian Conference on Intelligent Information and Database Systems. Springer, 2020, pp. 467–480.

[26] N. Bernard and F. Leprévost, “How evolutionary algorithms and information hiding deceive machines and humans for image recognition: A research program,” in Proceedings of the OLA’2019 International Conference on Optimization and Learning (Bangkok, Thailand, Jan 29-31, 2019). Springer, Heidelberg, 2019, pp. 12–15.

[27] R. Chitic, N. Bernard, and F. Leprévost, “Evolutionary algorithms deceive humans and machines at image classification: An extended proof of concept on two scenarios,” Journal of Information and Telecommunication, vol. 5, pp. 121–143, 2021, http://dx.doi.org/10.1080/24751839.2020.1829388.

[28] R. Chitic, N. Bernard, and F. Leprévost, “A proof of concept to deceive human and machines at image classification with evolutionary algorithms,” in Asian Conference on Intelligent Information and Database Systems. Springer, 2020, pp. 467–480.

[29] N. Bernard and F. Leprévost, “How evolutionary algorithms and information hiding deceive machines and humans for image recognition: A research program,” in Proceedings of the OLA’2019 International Conference on Optimization and Learning (Bangkok, Thailand, Jan 29-31, 2019). Springer, Heidelberg, 2019, pp. 12–15.
RALUCA CHITIC is a third-year Ph.D. student in computer science at the University of Luxembourg. She holds a BSc degree in physics and an MSc degree in artificial intelligence. Her current focus is on computer vision, explainable neural networks and evolutionary algorithms.

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.

FRANCK LEPRÈVOST has been a 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 cryptography, evolutionary algorithms and deep learning, international relations and academic leadership. He is the author of 60 publications and three scientific books.