Universal adversarial perturbations

Seyed-Mohsen Moosavi-Dezfooli*†
seyed.moosavi@epfl.ch

Alhussein Fawzi*†
alhussein.fawzi@epfl.ch

Omar Fawzi‡
omar.fawzi@ens-lyon.fr

Pascal Frossard†
pascal.frossard@epfl.ch

Abstract

Given a state-of-the-art deep neural network classifier, we show the existence of a universal (image-agnostic) and very small perturbation vector that causes natural images to be misclassified with high probability. We propose a systematic algorithm for computing universal perturbations, and show that state-of-the-art deep neural networks are highly vulnerable to such perturbations, albeit being quasi-imperceptible to the human eye. We further empirically analyze these universal perturbations and show, in particular, that they generalize very well across neural networks. The surprising existence of universal perturbations reveals important geometric correlations among the high-dimensional decision boundary of classifiers. It further outlines potential security breaches with the existence of single directions in the input space that adversaries can possibly exploit to break a classifier on most natural images.[1]

1. Introduction

Can we find a *single* small image perturbation that fools a state-of-the-art deep neural network classifier on all natural images? We show in this paper the existence of such quasi-imperceptible *universal* perturbation vectors that lead to misclassify natural images with high probability. Specifically, by adding such a *quasi-imperceptible* perturbation to natural images, the label estimated by the deep neural network is changed with high probability (see Fig. 1). Such perturbations are dubbed *universal*, as they are image-agnostic. The existence of these perturbations is problematic when the classifier is deployed in real-world (and possibly hostile) environments, as they can be exploited by ad-

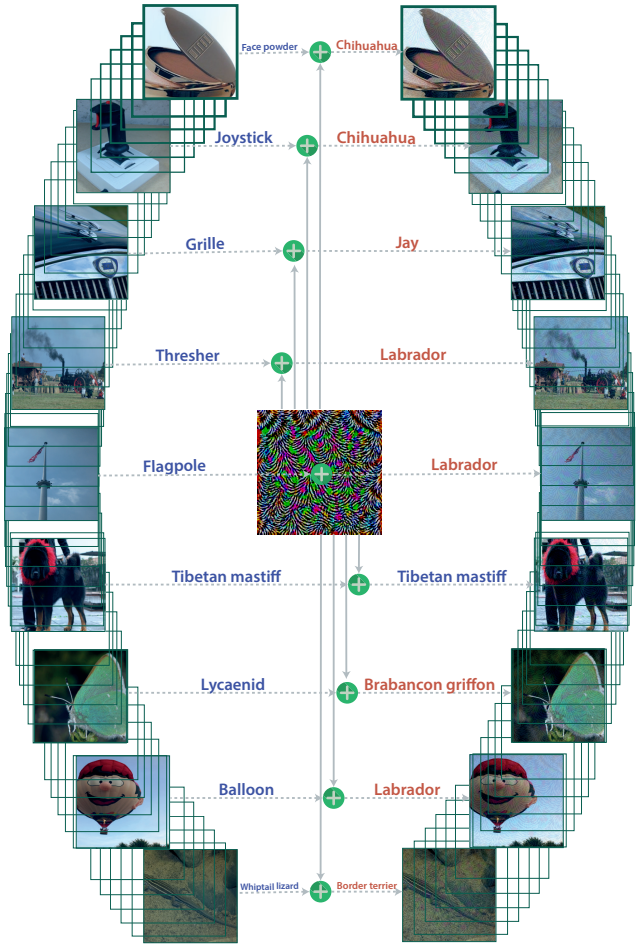


Figure 1: When added to a natural image, a universal perturbation image causes the image to be misclassified by the deep neural network with high probability. *Left images:* Original natural images. The labels are shown on top of each arrow. *Central image:* Universal perturbation. *Right images:* Perturbed images. The estimated labels of the perturbed images are shown on top of each arrow.

*The first two authors contributed equally to this work.
†École Polytechnique Fédérale de Lausanne, Switzerland
‡ENS de Lyon, LIP, UMR 5668 ENS Lyon - CNRS - UCBL - INRIA, Université de Lyon, France
[1]To encourage reproducible research, the code is available at gitHub. Furthermore, a video demonstrating the effect of universal perturbations on a smartphone can be found here.

versaries to break the classifier. Indeed, the perturbation process involves the mere addition of one very small perturbation to all natural images, and can be relatively straightforward to implement by adversaries in real-world environments, while being relatively difficult to detect as such perturbations are very small and thus do not significantly affect data distributions. The surprising existence of universal perturbations further reveals new insights on the topology of the decision boundaries of deep neural networks. We summarize the main contributions of this paper as follows:

- We show the existence of universal image-agnostic perturbations for state-of-the-art deep neural networks.
- We propose an algorithm for finding such perturbations. The algorithm seeks a universal perturbation for a set of training points, and proceeds by aggregating atomic perturbation vectors that send successive datapoints to the decision boundary of the classifier.
- We show that universal perturbations have a remarkable generalization property, as perturbations computed for a rather small set of training points fool new images with high probability.
- We show that such perturbations are not only universal across images, but also generalize well across deep neural networks. Such perturbations are therefore *doubly* universal, both with respect to the data and the network architectures.
- We explain and analyze the high vulnerability of deep neural networks to universal perturbations by examining the geometric correlation between different parts of the decision boundary.

The robustness of image classifiers to structured and unstructured perturbations have recently attracted a lot of attention [19, 16, 20, 3, 4, 12, 13, 14]. Despite the impressive performance of deep neural network architectures on challenging visual classification benchmarks [6, 9, 21, 10], these classifiers were shown to be highly vulnerable to perturbations. In [19], such networks are shown to be unstable to very small and often imperceptible additive *adversarial* perturbations. Such carefully crafted perturbations are either estimated by solving an optimization problem [19, 11, 1] or through one step of gradient ascent [5], and result in a perturbation that fools a specific data point. A fundamental property of these adversarial perturbations is their intrinsic dependence on datapoints: the perturbations are specifically crafted for each data point independently. As a result, the computation of an adversarial perturbation for a new data point requires solving a data-dependent optimization problem from scratch, which uses the full knowledge of the classification model. This is different from the universal perturbation considered in this paper, as we seek a single perturbation vector that fools the network on most natural images. Perturbing a new datapoint then only involves the mere addition of the universal perturbation to the image (and does not require solving an optimization problem/gradient computation). Finally, we emphasize that our notion of universal perturbation differs from the generalization of adversarial perturbations studied in [19], where perturbations computed on the MNIST task were shown to generalize well across different models. Instead, we examine the existence of universal perturbations that are common to most data points belonging to the data distribution.

2. Universal perturbations

We formalize in this section the notion of universal perturbations, and propose a method for estimating such perturbations. Let μ denote a distribution of images in $\mathbb{R}^d$, and $\hat{k}$ define a classification function that outputs for each image $x \in \mathbb{R}^d$ an estimated label $\hat{k}(x)$. The main focus of this paper is to seek perturbation vectors $v \in \mathbb{R}^d$ that fool the classifier $\hat{k}$ on *almost all* datapoints sampled from μ. That is, we seek a vector v such that

$$\hat{k}(x+v) \neq \hat{k}(x) \text{ for "most" } x \sim \mu.$$

We coin such a perturbation *universal*, as it represents a fixed image-agnostic perturbation that causes label change for most images sampled from the data distribution μ. We focus here on the case where the distribution μ represents the set of natural images, hence containing a huge amount of variability. In that context, we examine the existence of small universal perturbations (in terms of the ℓ_p norm with $p \in [1, \infty)$) that misclassify most images. The goal is therefore to find v that satisfies the following two constraints:

1. $\|v\|_p \leq \xi$,
2. $\underset{x \sim \mu}{\mathbb{P}} \left(\hat{k}(x+v) \neq \hat{k}(x) \right) \geq 1 - \delta$.

The parameter ξ controls the magnitude of the perturbation vector v, and δ quantifies the desired fooling rate for all images sampled from the distribution μ.

Algorithm. Let $X = \{x_1, \ldots, x_m\}$ be a set of images sampled from the distribution μ. Our proposed algorithm seeks a universal perturbation v, such that $\|v\|_p \leq \xi$, while fooling most data points in X. The algorithm proceeds iteratively over the data points in X and gradually builds the universal perturbation, as illustrated in Fig. 2. At each iteration, the minimal perturbation Δv_i that sends the current perturbed point, $x_i + v$, to the decision boundary of the classifier is computed, and aggregated to the current instance of the universal perturbation. In more details, provided the current universal perturbation v does not fool data point x_i, we seek the extra perturbation Δv_i with minimal norm that allows to fool data point x_i by solving the following opti-

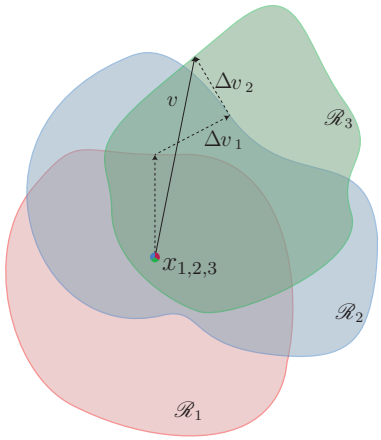


Figure 2: Schematic representation of the proposed algorithm used to compute universal perturbations. In this illustration, data points x_1, x_2 and x_3 are super-imposed, and the classification regions $\mathscr{R}_i$ (i.e., regions of constant estimated label) are shown in different colors. Our algorithm proceeds by aggregating sequentially the minimal perturbations sending the current perturbed points $x_i + v$ outside of the corresponding classification region $\mathscr{R}_i$.

mization problem:

$$\Delta v_i \leftarrow \arg\min_r \|r\|_2 \text{ s.t. } \hat{k}(x_i + v + r) \neq \hat{k}(x_i). \quad (1)$$

To ensure that the constraint $\|v\|_p \leq \xi$ is satisfied, the updated universal perturbation is further projected on the ℓ_p ball of radius ξ and centered at 0. That is, let $\mathcal{P}_{p,\xi}$ be the projection operator defined as follows:

$$\mathcal{P}_{p,\xi}(v) = \arg\min_{v'} \|v - v'\|_2 \text{ subject to } \|v'\|_p \leq \xi.$$

Then, our update rule is given by $v \leftarrow \mathcal{P}_{p,\xi}(v + \Delta v_i)$. Several passes on the data set X are performed to improve the quality of the universal perturbation. The algorithm is terminated when the empirical "fooling rate" on the perturbed data set $X_v := \{x_1 + v, \ldots, x_m + v\}$ exceeds the target threshold $1 - \delta$. That is, we stop the algorithm whenever

$$\text{Err}(X_v) := \frac{1}{m} \sum_{i=1}^{m} 1_{\hat{k}(x_i+v) \neq \hat{k}(x_i)} \geq 1 - \delta.$$

The detailed algorithm is provided in Algorithm 1. Interestingly, in practice, the number of data points m in X need not be large to compute a universal perturbation that is valid for the whole distribution μ. In particular, we can set m to be much smaller than the number of training points (see Section 3).

The proposed algorithm involves solving at most m instances of the optimization problem in Eq. (1) for each pass. While this optimization problem is not convex when $\hat{k}$ is a

Algorithm 1 Computation of universal perturbations.

input: Data points X, classifier $\hat{k}$, desired ℓ_p norm of the perturbation ξ, desired accuracy on perturbed samples δ.
output: Universal perturbation vector v.
Initialize $v \leftarrow 0$.
while $\text{Err}(X_v) \leq 1 - \delta$ **do**
  for each datapoint $x_i \in X$ **do**
    if $\hat{k}(x_i + v) = \hat{k}(x_i)$ **then**
      Compute the *minimal* perturbation that sends $x_i + v$ to the decision boundary:

$$\Delta v_i \leftarrow \arg\min_r \|r\|_2 \text{ s.t. } \hat{k}(x_i + v + r) \neq \hat{k}(x_i).$$

      Update the perturbation:

$$v \leftarrow \mathcal{P}_{p,\xi}(v + \Delta v_i).$$

    end if
  end for
end while

standard classifier (e.g., a deep neural network), several efficient approximate methods have been devised for solving this problem [19, 11, 7]. We use in the following the approach in [11] for its efficency. It should further be noticed that the objective of Algorithm 1 is *not* to find the smallest universal perturbation that fools most data points sampled from the distribution, but rather to find one such perturbation with sufficiently small norm. In particular, different random shufflings of the set X naturally lead to a diverse set of universal perturbations v satisfying the required constraints. The proposed algorithm can therefore be leveraged to generate multiple universal perturbations for a deep neural network (see next section for visual examples).

3. Universal perturbations for deep nets

We now analyze the robustness of state-of-the-art deep neural network classifiers to universal perturbations using Algorithm 1.

In a first experiment, we assess the estimated universal perturbations for different recent deep neural networks on the ILSVRC 2012 [15] validation set (50,000 images), and report the *fooling ratio*, that is the proportion of images that change labels when perturbed by our universal perturbation. Results are reported for $p = 2$ and $p = \infty$, where we respectively set $\xi = 2000$ and $\xi = 10$. These numerical values were chosen in order to obtain a perturbation whose norm is significantly smaller than the image norms, such that the perturbation is quasi-imperceptible when added to

		CaffeNet [8]	VGG-F [2]	VGG-16 [17]	VGG-19 [17]	GoogLeNet [18]	ResNet-152 [6]
ℓ_2	X	85.4%	85.9%	90.7%	86.9%	82.9%	89.7%
	Val.	85.6	87.0%	90.3%	84.5%	82.0%	88.5%
ℓ_∞	X	93.1%	93.8%	78.5%	77.8%	80.8%	85.4%
	Val.	93.3%	93.7%	78.3%	77.8%	78.9%	84.0%

Table 1: Fooling ratios on the set X, and the validation set.

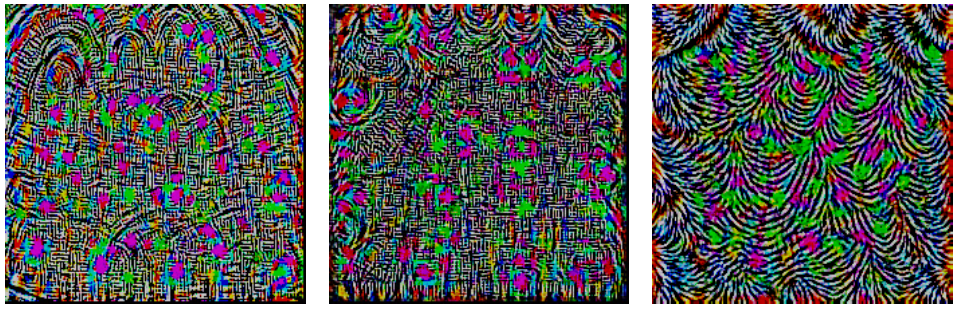
natural images[2]. Results are listed in Table 1. Each result is reported on the set X, which is used to compute the perturbation, as well as on the validation set (that is *not* used in the process of the computation of the universal perturbation). Observe that for all networks, the universal perturbation achieves very high fooling rates on the validation set. Specifically, the universal perturbations computed for CaffeNet and VGG-F fool more than 90% of the validation set (for $p = \infty$). In other words, for any natural image in the validation set, the mere addition of our universal perturbation fools the classifier more than 9 times out of 10. This result is moreover not specific to such architectures, as we can also find universal perturbations that cause VGG, GoogLeNet and ResNet classifiers to be fooled on natural images with probability edging 80%. These results have an element of surprise, as they show the existence of *single* universal perturbation vectors that cause natural images to be misclassified with high probability, albeit being quasi-imperceptible to humans. To verify this latter claim, we show visual examples of perturbed images in Fig. 3, where the GoogLeNet architecture is used. These images are either taken from the ILSVRC 2012 validation set, or captured using a mobile phone camera. Observe that in most cases, the universal perturbation is *quasi-imperceptible*, yet this powerful image-agnostic perturbation is able to misclassify any image with high probability for state-of-the-art classifiers. We refer to the supp. material for the original (unperturbed) images, as well as their ground truth labels. We also refer to the video in the supplementary material for real-world examples on a smartphone. We visualize the universal perturbations corresponding to different networks in Fig. 4. It should be noted that such universal perturbations are not unique, as many different universal perturbations (all satisfying the two required constraints) can be generated for the same network. In Fig. 5, we visualize five different universal perturbations obtained by using different random shufflings in X. Observe that such universal perturbations are different, although they exhibit a similar pattern. This is moreover confirmed by computing the normalized inner products between two pairs of perturbation images, as the normalized inner products do not exceed 0.1, which shows that one can find diverse universal perturbations.

[2]For comparison, the average ℓ_2 and ℓ_∞ norm of an image in the validation set is respectively $\approx 5 \times 10^4$ and ≈ 250.

While the above universal perturbations are computed for a set X of 10,000 images from the training set (i.e., in average 10 images per class), we now examine the influence of the size of X on the quality of the universal perturbation. We show in Fig. 6 the fooling rates obtained on the validation set for different sizes of X for GoogLeNet. Note for example that with a set X containing only 500 images, we can fool more than 30% of the images on the validation set. This result is significant when compared to the number of classes in ImageNet (1000), as it shows that we can fool a large set of unseen images, even when using a set X containing less than one image per class! The universal perturbations computed using Algorithm 1 have therefore a remarkable generalization power over unseen data points, and can be computed on a very small set of training images.

Cross-model universality. While the computed perturbations are universal across unseen data points, we now examine their *cross-model* universality. That is, we study to which extent universal perturbations computed for a specific architecture (e.g., VGG-19) are also valid for another architecture (e.g., GoogLeNet). Table 2 displays a matrix summarizing the universality of such perturbations across six different architectures. For each architecture, we compute a universal perturbation and report the fooling ratios on all other architectures; we report these in the rows of Table 2. Observe that, for some architectures, the universal perturbations generalize very well across other architectures. For example, universal perturbations computed for the VGG-19 network have a fooling ratio above 53% for all other tested architectures. This result shows that our universal perturbations are, to some extent, *doubly-universal* as they generalize well across data points *and* very different architectures. It should be noted that, in [19], adversarial perturbations were previously shown to generalize well, to some extent, across different neural networks on the MNIST problem. Our results are however different, as we show the generalizability of universal perturbations across different architectures on the ImageNet data set. This result shows that such perturbations are of practical relevance, as they generalize well across data points and architectures. In particular, in order to fool a new image on an unknown neural network, a simple addition of a universal perturbation computed on the VGG-19 architecture is likely to misclassify the data point.

Figure 3: Examples of perturbed images and their corresponding labels. The first 8 images belong to the ILSVRC 2012 validation set, and the last 4 are images taken by a mobile phone camera. See supp. material for the original images.



(a) CaffeNet (b) VGG-F (c) VGG-16

(d) VGG-19 (e) GoogLeNet (f) ResNet-152

Figure 4: Universal perturbations computed for different deep neural network architectures. Images generated with $p = \infty$, $\xi = 10$. The pixel values are scaled for visibility.

Visualization of the effect of universal perturbations. To gain insights on the effect of universal perturbations on natural images, we now visualize the distribution of labels on the ImageNet validation set. Specifically, we build a directed graph $G = (V, E)$, whose vertices denote the labels, and directed edges $e = (i \to j)$ indicate that the majority of images of class i are fooled into label j when applying the universal perturbation. The existence of edges $i \to j$ therefore suggests that the preferred fooling label for images of class i is j. We construct this graph for GoogLeNet, and visualize the full graph in the supp. material for space constraints. The visualization of this graph shows a very peculiar topology. In particular, the graph is a union of disjoint components, where all edges in one component mostly connect to one target label. See Fig. 7 for an illustration of two connected components. This visualization clearly shows the existence of several *dominant labels*, and that universal perturbations mostly make natural images classified with such

Figure 5: Diversity of universal perturbations for the GoogLeNet architecture. The five perturbations are generated using different random shufflings of the set X. Note that the normalized inner products for any pair of universal perturbations does not exceed 0.1, which highlights the diversity of such perturbations.

	VGG-F	CaffeNet	GoogLeNet	VGG-16	VGG-19	ResNet-152
VGG-F	**93.7%**	71.8%	48.4%	42.1%	42.1%	47.4 %
CaffeNet	74.0%	**93.3%**	47.7%	39.9%	39.9%	48.0%
GoogLeNet	46.2%	43.8%	**78.9%**	39.2%	39.8%	45.5%
VGG-16	63.4%	55.8%	56.5%	**78.3%**	73.1%	63.4%
VGG-19	64.0%	57.2%	53.6%	73.5%	**77.8%**	58.0%
ResNet-152	46.3%	46.3%	50.5%	47.0%	45.5%	**84.0%**

Table 2: Generalizability of the universal perturbations across different networks. The percentages indicate the fooling rates. The rows indicate the architecture for which the universal perturbations is computed, and the columns indicate the architecture for which the fooling rate is reported.

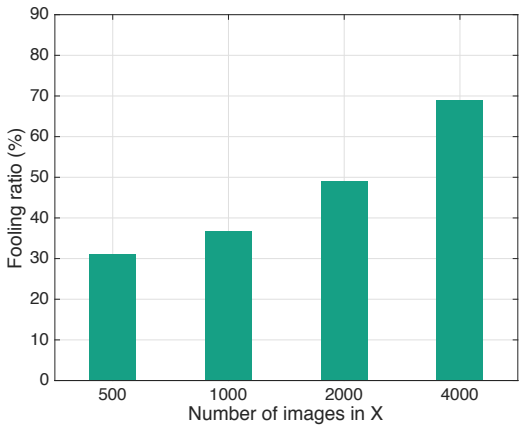


Figure 6: Fooling ratio on the validation set versus the size of X. Note that even when the universal perturbation is computed on a very small set X (compared to training and validation sets), the fooling ratio on validation set is large.

labels. We hypothesize that these dominant labels occupy large regions in the image space, and therefore represent good candidate labels for fooling most natural images. Note that these dominant labels are automatically found by Algorithm 1, and are not imposed a priori in the computation of perturbations.

Fine-tuning with universal perturbations. We now examine the effect of fine-tuning the networks with perturbed images. We use the VGG-F architecture, and fine-tune the network based on a modified training set where universal perturbations are added to a fraction of (clean) training samples: for each training point, a universal perturbation is added with probability 0.5, and the original sample is preserved with probability 0.5.[3] To account for the diversity of universal perturbations, we pre-compute a pool of 10 different universal perturbations and add perturbations to the training samples randomly from this pool. The network is fine-tuned by performing 5 extra epochs of training on the modified training set. To assess the effect of fine-tuning on the robustness of the network, we compute a new universal perturbation for the fine-tuned network (using Algorithm 1, with $p = \infty$ and $\xi = 10$), and report the fooling rate of the network. After 5 extra epochs, the fooling rate on the validation set is 76.2%, which shows an improvement with respect to the original network (93.7%, see Table 1).[4] Despite this improvement, the fine-tuned network remains largely vulnerable to small universal perturbations. We therefore

[3] In this fine-tuning experiment, we use a slightly modified notion of universal perturbations, where the *direction* of the universal vector v is fixed for all data points, while its *magnitude* is adaptive. That is, for each data point x, we consider the perturbed point $x + \alpha v$, where α is the smallest coefficient that fools the classifier. We observed that this feedbacking strategy is less prone to overfitting than the strategy where the universal perturbation is simply added to all training points.

[4] This fine-tuning procedure moreover led to a minor increase in the error rate on the validation set, which might be due to a slight overfitting of the perturbed data.

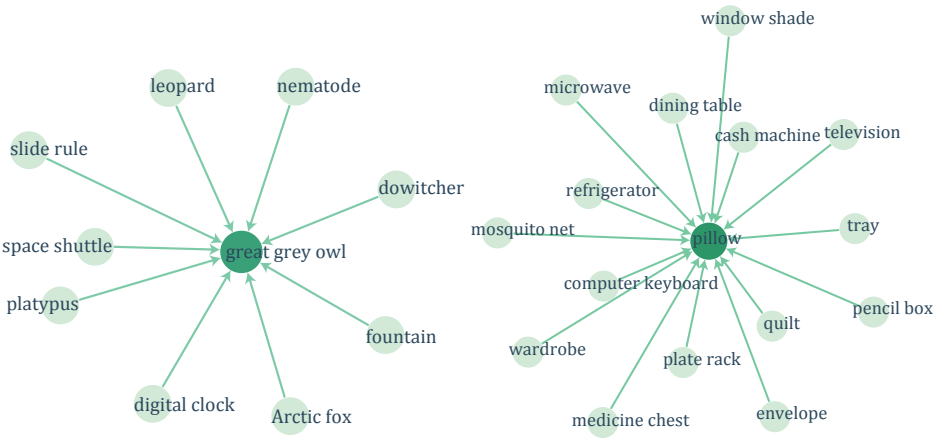


Figure 7: Two connected components of the graph $G = (V, E)$, where the vertices are the set of labels, and directed edges $i \to j$ indicate that most images of class i are fooled into class j.

repeated the above procedure (i.e., computation of a pool of 10 universal perturbations for the fine-tuned network, fine-tuning of the new network based on the modified training set for 5 extra epochs), and we obtained a new fooling ratio of 80.0%. In general, the repetition of this procedure for a fixed number of times did *not* yield any improvement over the 76.2% fooling ratio obtained after one step of fine-tuning. Hence, while fine-tuning the network leads to a mild improvement in the robustness, we observed that this simple solution does not fully immune against small universal perturbations.

4. Explaining the vulnerability to universal perturbations

The goal of this section is to analyze and explain the high vulnerability of deep neural network classifiers to universal perturbations. To understand the unique characteristics of universal perturbations, we first compare such perturbations with other types of perturbations, namely i) *random* perturbation, ii) *adversarial* perturbation computed for a randomly picked sample (computed using the DF and FGS methods respectively in [11] and [5]), iii) *sum* of adversarial perturbations over X, and iv) mean of the images (or *ImageNet bias*). For each perturbation, we depict a phase transition graph in Fig. 8 showing the fooling rate on the validation set with respect to the ℓ_2 norm of the perturbation. Different perturbation norms are achieved by scaling accordingly each perturbation with a multiplicative factor to have the target norm. Note that the universal perturbation is computed for $\xi = 2000$, and also scaled accordingly.

Observe that the proposed universal perturbation quickly reaches very high fooling rates, even when the perturbation is constrained to be of small norm. For example, the universal perturbation computed using Algorithm 1 achieves a fooling rate of 85% when the ℓ_2 norm is constrained to $\xi = 2000$, while other perturbations achieve much smaller ratios for comparable norms. In particular, random vectors sampled uniformly from the sphere of radius of 2000 only fool 10% of the validation set. The large difference between universal and random perturbations suggests that the universal perturbation exploits some *geometric correlations* between different parts of the decision boundary of the classifier. In fact, if the orientations of the decision boundary in the neighborhood of different data points were completely uncorrelated (and independent of the distance to the decision boundary), the norm of the best universal perturbation would be comparable to that of a random perturbation. Note that the latter quantity is well understood (see [4]), as the norm of the random perturbation required to fool a specific data point precisely behaves as $\Theta(\sqrt{d}\|r\|_2)$, where d is the dimension of the input space, and $\|r\|_2$ is the distance between the data point and the decision boundary (or equivalently, the norm of the smallest adversarial perturbation). For the considered ImageNet classification task, this quantity is equal to $\sqrt{d}\|r\|_2 \approx 2 \times 10^4$, for most data points, which is at least one order of magnitude larger than the universal perturbation ($\xi = 2000$). This substantial difference between *random* and *universal* perturbations thereby suggests redundancies in the geometry of the decision boundaries that we now explore.

For each image x in the validation set, we compute the adversarial perturbation vector $r(x) = \arg\min_r \|r\|_2$ s.t. $\hat{k}(x + r) \neq \hat{k}(x)$. It is easy to see that $r(x)$ is *normal* to the decision boundary of the classifier (at $x + r(x)$). The vector $r(x)$ hence captures the local geometry of the decision boundary in the region surrounding the data point x. To quantify the correlation

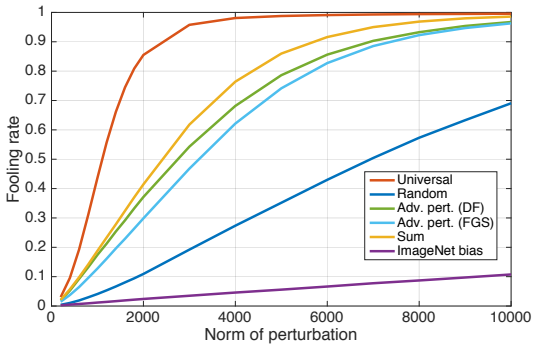


Figure 8: Comparison between fooling rates of different perturbations. Experiments performed on the CaffeNet architecture.

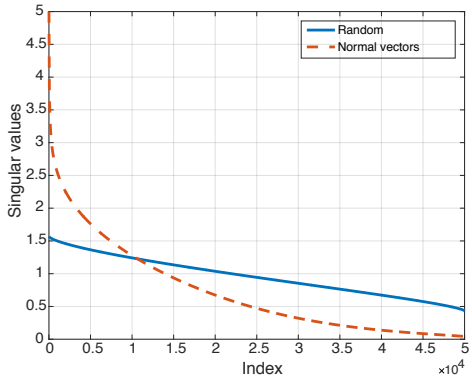


Figure 9: Singular values of matrix N containing normal vectors to the decision decision boundary.

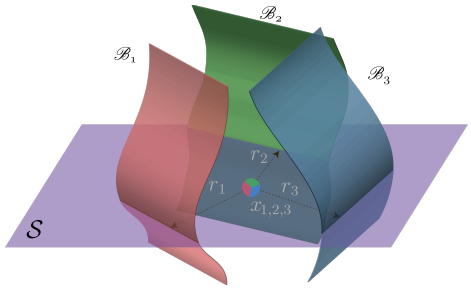


Figure 10: Illustration of the low dimensional subspace $\mathcal{S}$ containing normal vectors to the decision boundary in regions surrounding natural images. For the purpose of this illustration, we super-impose three data-points $\{x_i\}_{i=1}^3$, and the adversarial perturbations $\{r_i\}_{i=1}^3$ that send the respective datapoints to the decision boundary $\{\mathscr{B}_i\}_{i=1}^3$ are shown. Note that $\{r_i\}_{i=1}^3$ all live in the subspace $\mathcal{S}$.

between different regions of the decision boundary of the classifier, we define the matrix

$$N = \left[\frac{r(x_1)}{\|r(x_1)\|_2} \cdots \frac{r(x_n)}{\|r(x_n)\|_2} \right]$$

of normal vectors to the decision boundary in the vicinity of n data points in the validation set. For binary linear classifiers, the decision boundary is a hyperplane, and N is of rank 1, as all normal vectors are collinear. To capture more generally the correlations in the decision boundary of complex classifiers, we compute the singular values of the matrix N. The singular values of the matrix N, computed for the CaffeNet architecture are shown in Fig. 9. We further show in the same figure the singular values obtained when the columns of N are sampled uniformly at random from the unit sphere. Observe that, while the latter singular values have a slow decay, the singular values of N decay quickly, which confirms the existence of large correlations and redundancies in the decision boundary of deep networks. More precisely, this suggests the existence of a subspace $\mathcal{S}$ of low dimension d' (with $d' \ll d$), that contains most normal vectors to the decision boundary in regions surrounding natural images. We hypothesize that the existence of universal perturbations fooling most natural images is partly due to the existence of such a low-dimensional subspace that captures the correlations among different regions of the decision boundary. In fact, this subspace "collects" normals to the decision boundary in different regions, and perturbations belonging to this subspace are therefore likely to fool datapoints. To verify this hypothesis, we choose a *random* vector of norm $\xi = 2000$ belonging to the subspace $\mathcal{S}$ spanned by the first 100 singular vectors, and compute its fooling ratio on a different set of images (i.e., a set of images that have not been used to compute the SVD). Such a perturbation can fool nearly 38% of these images, thereby showing that a *random* direction in this well-sought subspace $\mathcal{S}$ significantly outperforms random perturbations (we recall that such perturbations can only fool 10% of the data). Fig. 10 illustrates the subspace $\mathcal{S}$ that captures the correlations in the decision boundary. It should further be noted that the existence of this low dimensional subspace explains the surprising generalization properties of universal perturbations obtained in Fig. 6, where one can build relatively generalizable universal perturbations with very few images.

Unlike the above experiment, the proposed algorithm does *not* choose a random vector in this subspace, but rather chooses a specific direction in order to maximize the overall fooling rate. This explains the gap between the fooling rates obtained with the random vector strategy in $\mathcal{S}$ and Algorithm 1.

5. Conclusions

We showed the existence of small universal perturbations that can fool state-of-the-art classifiers on natural images. We proposed an iterative algorithm to generate universal perturbations, and highlighted several properties of

such perturbations. In particular, we showed that universal perturbations generalize well across different classification models, resulting in doubly-universal perturbations (image-agnostic, network-agnostic). We further explained the existence of such perturbations with the correlation between different regions of the decision boundary. This provides insights on the geometry of the decision boundaries of deep neural networks, and contributes to a better understanding of such systems. A theoretical analysis of the geometric correlations between different parts of the decision boundary will be the subject of future research.

Acknowledgments

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

[1] O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori, and A. Criminisi. Measuring neural net robustness with constraints. In *Neural Information Processing Systems (NIPS)*, 2016. 2

[2] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014. 4

[3] A. Fawzi, O. Fawzi, and P. Frossard. Analysis of classifiers' robustness to adversarial perturbations. *CoRR*, abs/1502.02590, 2015. 2

[4] A. Fawzi, S. Moosavi-Dezfooli, and P. Frossard. Robustness of classifiers: from adversarial to random noise. In *Neural Information Processing Systems (NIPS)*, 2016. 2, 7

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015. 2, 7

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4

[7] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári. Learning with a strong adversary. *CoRR, abs/1511.03034*, 2015. 3

[8] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia (MM)*, pages 675–678, 2014. 4

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, pages 1097–1105, 2012. 2

[10] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368. IEEE, 2011. 2

[11] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3, 7

[12] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, 2015. 2

[13] E. Rodner, M. Simon, R. Fisher, and J. Denzler. Fine-grained recognition in the noisy wild: Sensitivity analysis of convolutional neural networks approaches. In *British Machine Vision Conference (BMVC)*, 2016. 2

[14] A. Rozsa, E. M. Rudd, and T. E. Boult. Adversarial diversity and hard positive generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2016. 2

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 3

[16] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet. Adversarial manipulation of deep representations. In *International Conference on Learning Representations (ICLR)*, 2016. 2

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2014. 4

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 4

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014. 2, 3, 4

[20] P. Tabacof and E. Valle. Exploring the space of adversarial images. *IEEE International Joint Conference on Neural Networks*, 2016. 2

[21] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, 2014. 2

A. Appendix

Fig. 11 shows the original images corresponding to the experiment in Fig. 3. Fig. 12 visualizes the graph showing relations between original and perturbed labels (see Section 3 for more details).

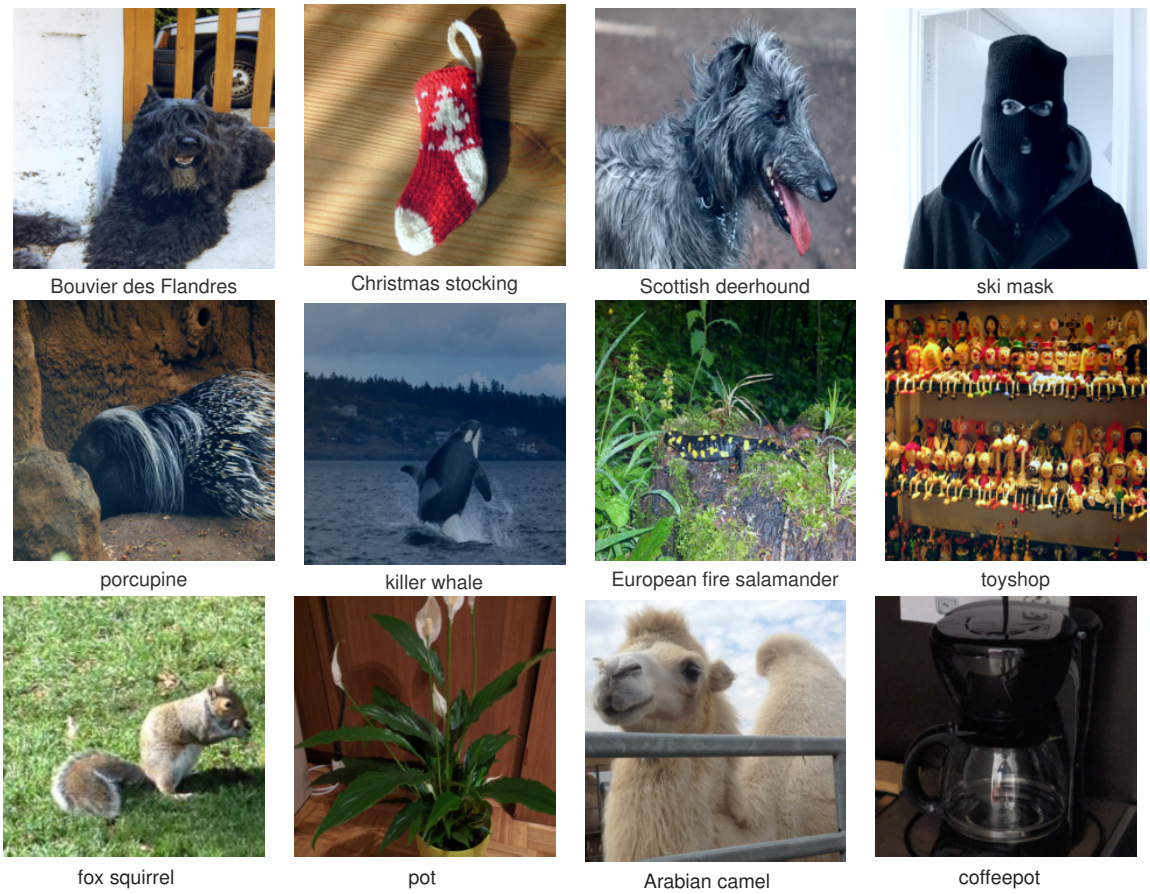


Figure 11: Original images. The first two rows are randomly chosen images from the validation set, and the last row of images are personal images taken from a mobile phone camera.

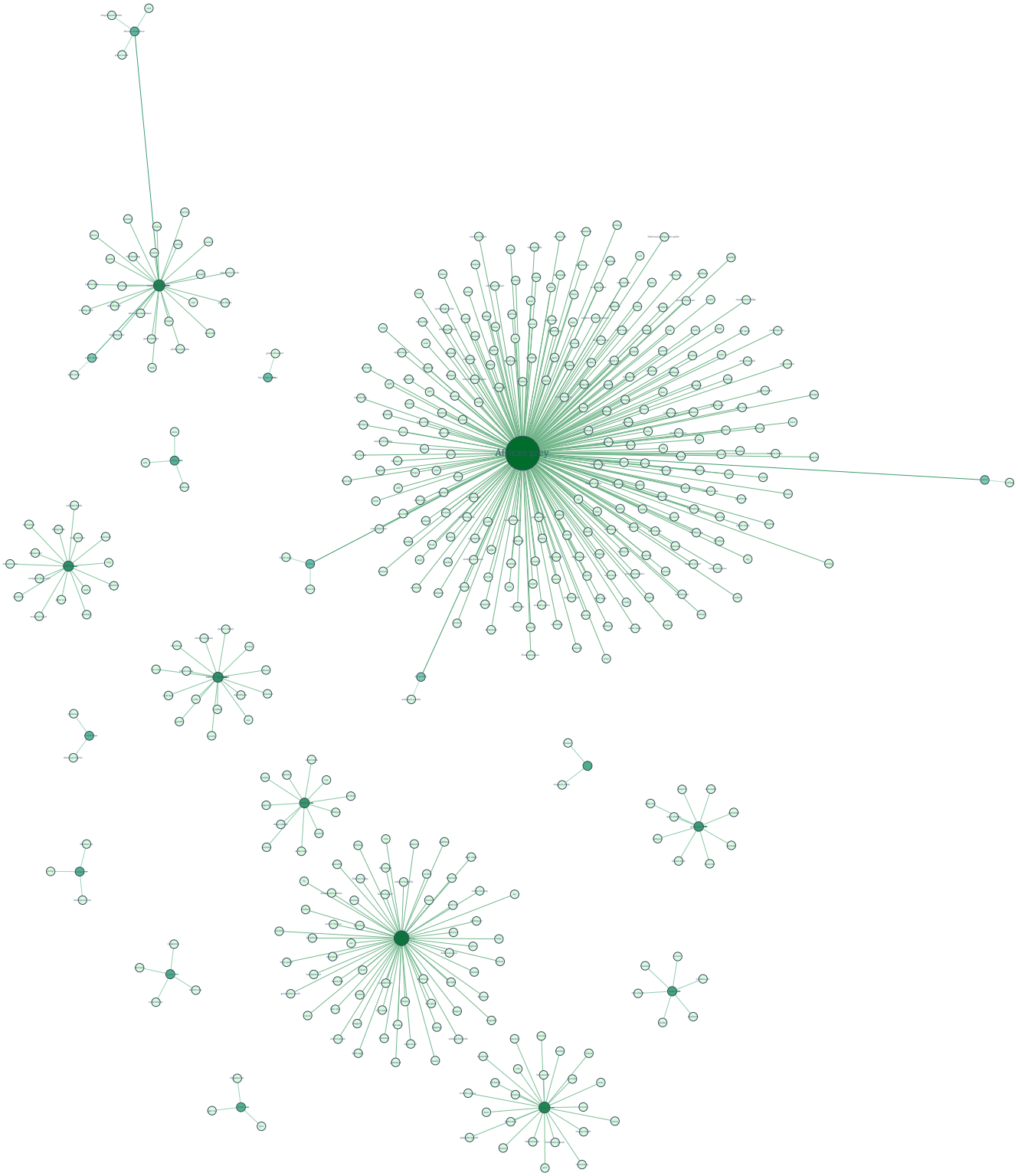


Figure 12: Graph representing the relation between original and perturbed labels. Note that "dominant labels" appear systematically. Please zoom for readability. Isolated nodes are removed from this visualization for readability.