Universal Adversarial Perturbations are Not Bugs, They are Features

Philipp Benz*, Chaoning Zhang*, Tooba Imtiaz, In-So Kweon
* indicates equal contribution

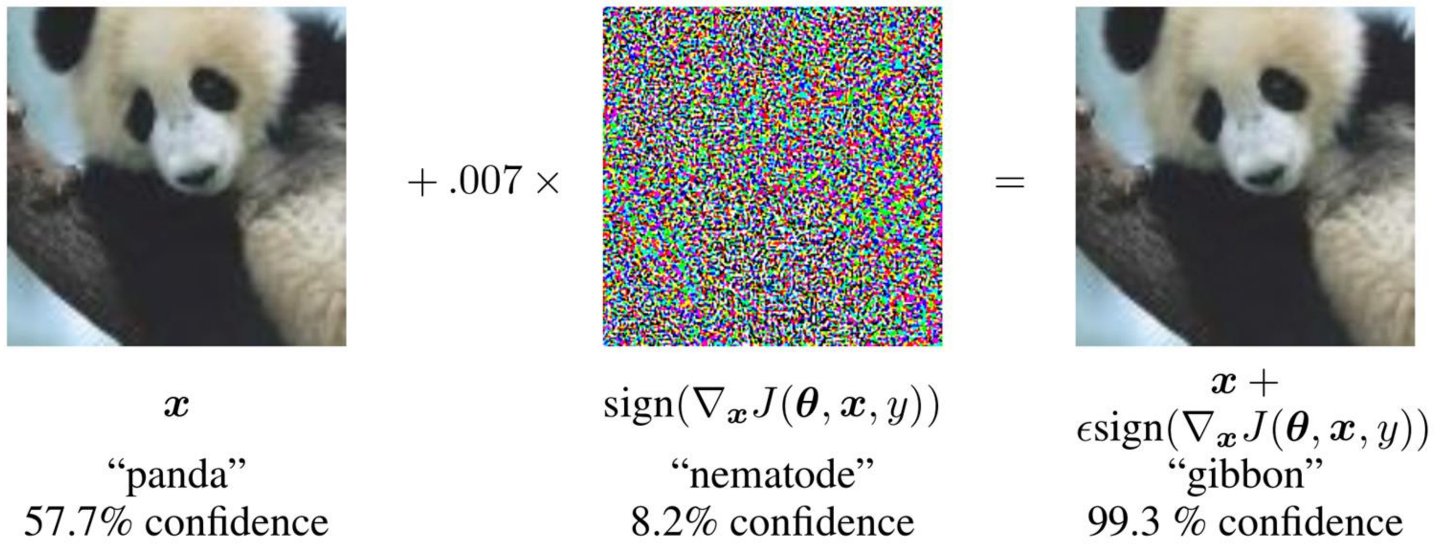
Korea Advanced Institute of Science and Technology (KAIST)

CVPR SEATTLE WASHINGTON JUNE 16-18 2020

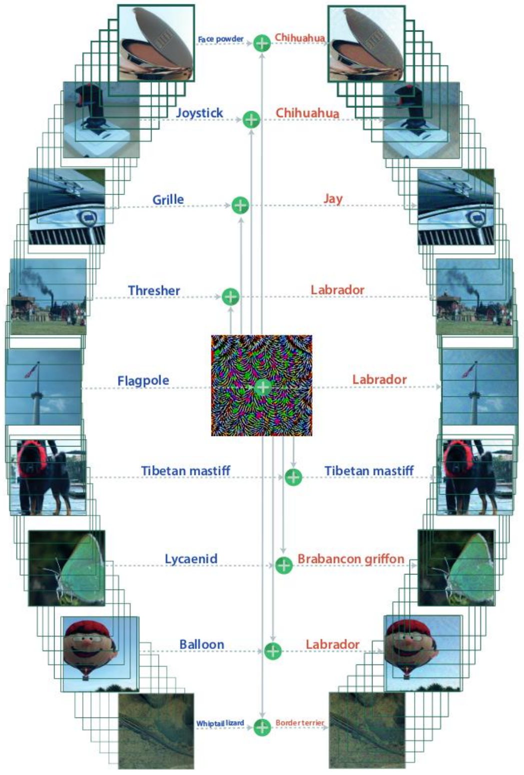
Adversarial Examples

Deep Neural Networks are sensitive to small perturbations in the image, which can lead to misclassifications. These changes are mostly imperceptible for human observers.

Image-dependant Adversarial Perturbations [1,2]



Universal Perturbations [3]



[1] Intriguing properties of neural networks; Szegedy, Zaremba, Sutskever, Bruna, Erhan, Goodfellow, Fergus; ArXiv 2013
[2] Explaining and Harnessing Adversarial Examples; Goodfellow, Shlens, Szegedy; ICLR 2015
[3] Universal adversarial perturbations; Moosavi-Dezfooli, Fawzi, Fawzi, Frossard; CVPR 2017

Universal Adversarial Perturbations

Prior works [1,2] treated the UAP as noise (“bug”) to the samples to be attacked.

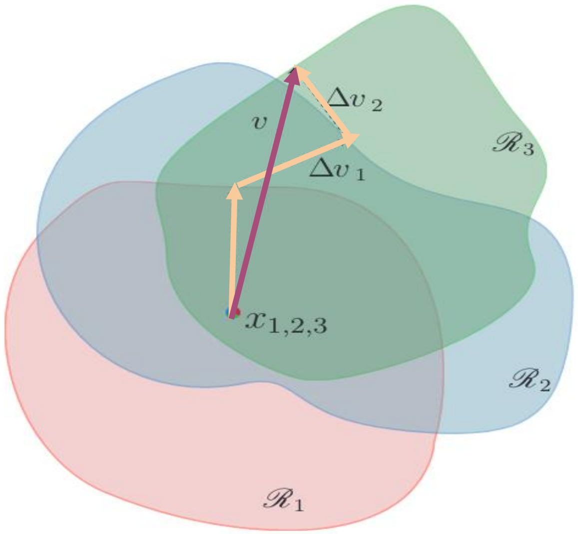


Figure 1: Schematic representation of the algorithm in [1] to compute universal perturbations.

Algorithm:

- Craft single perturbation (via DeepFool [3]) to let one sample cross the decision boundary
- Iterate this process for different samples to aggregate the universal adversarial perturbation.

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 $\mathrm{Err}(X_v) \leq 1 - \delta$ **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:

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

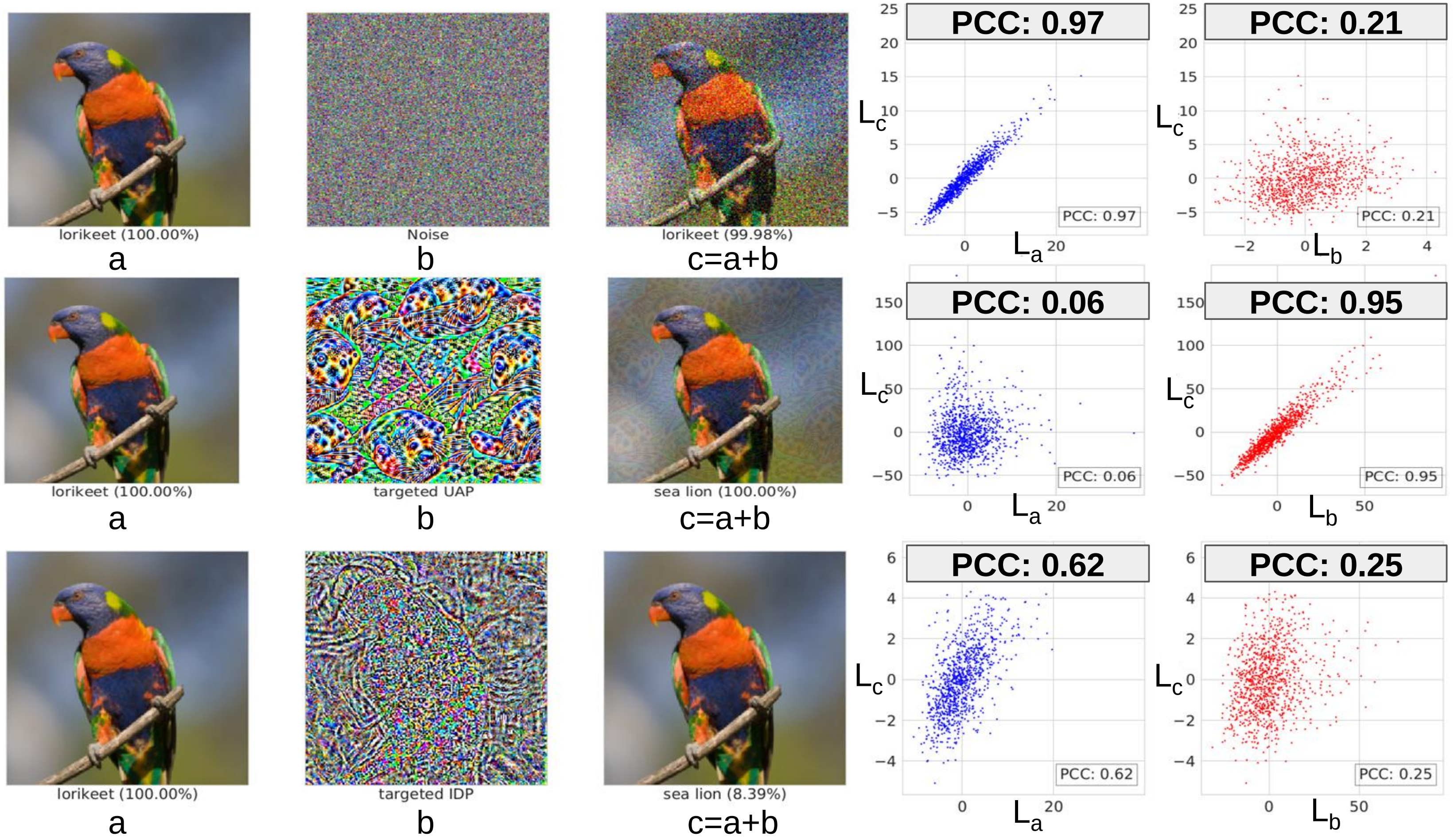
    end if
  end for
end while

[1] Universal adversarial perturbations; Moosavi-Dezfooli, Fawzi, Fawzi, Frossard; CVPR 2017
[2] Analysis of universal adversarial perturbations; Moosavi-Dezfooli, Fawzi, Fawzi, Frossard, Soatto; ArXiv 2017
[3] DeepFool: a simple and accurate method to fool deep neural networks; Moosavi-Dezfooli, Fawzi, Frossard; CVPR 2016

PCC Analysis

Treat the DNN logits as a vector for feature representation and use them to analyze the mutual influence of two independent inputs based on the Pearson correlation coefficient (PCC)

$$\mathrm{PCC}_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$



PCC-Analysis result for one sample image `lorikeet'. Three scenarios of input combinations are considered: 1: image + noise; 2: image + targeted UAP; 3: image + targeted image-dependant AE. The columns show input a, input b, input c=a+b, logit vector analysis of L_c over L_a and vector analysis of L_c over L_b

"Universal perturbations contain dominant features, and images behave like noise to them"

Image-dependant perturbations seem to not contain features by themselves

Noise Perspective vs. Feature Perspective

Noise Perspective (Prior works)

- Treat the targeted UAP as noise (“bug”) to the sample to be attacked
- Requires the samples from the training dataset in the UAP generation process
- Explicitly designed to let individual samples cross the decision boundary
- Assumes that the attack generalizes to unseen samples

Requires the original training dataset
Slow: ~2 hours

Feature Perspective (Ours)

- **UAPs contain features of a certain class**
- Treatment of the **images as noise** to the generated UAP during the optimization process in order to be recognizable by the target network
- **No need for semantic features** as in the original training dataset samples
- **Proxy datasets** as background noise: Downloaded from the Internet, MS-COCO, Pascal VOC, Places365

Requires no original training dataset
Fast: ~2 minutes

Targeted UAP without original training data

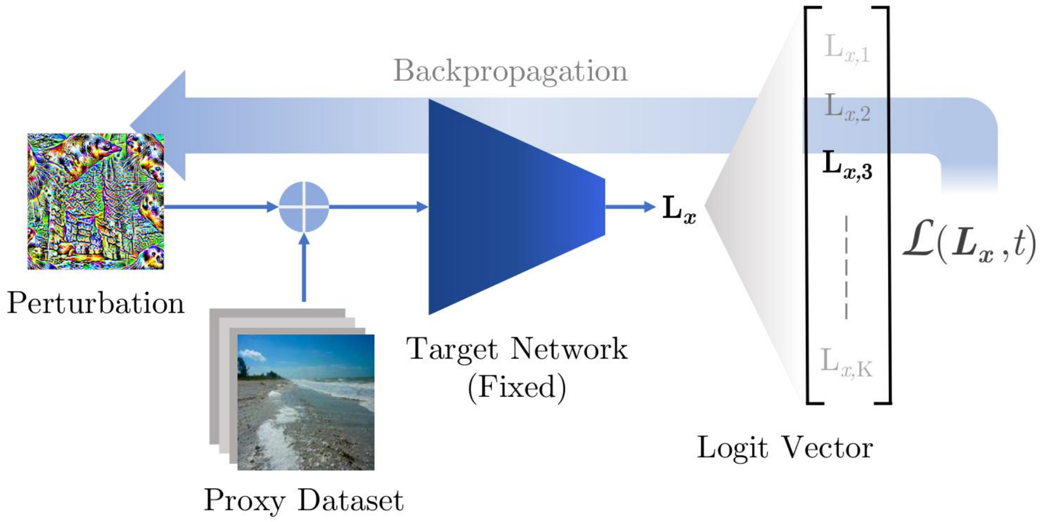


Figure 1: Proposed method of generating targeted universal adversarial perturbations without data, by using a proxy dataset.

Figure 2: Targeted universal perturbations (target class 'sea lion') for different network architectures

Table 1: Results for targeted UAPs trained on four different datasets reported in the targeted fooling ratio (%)

Proxy Data	AlexNet	GoogleNet	VGG16	VGG19	ResNet152
ImageNet [11]	48.6	59.9	75.0	71.6	66.3
COCO [12]	47.2	59.8	75.1	68.8	65.7
VOC [5]	46.9	58.9	74.7	68.8	65.2
Places365 [29]	42.6	60.0	73.4	64.5	62.5

Table 2: Comparison to other methods. The results are divided in universal attacks with access to the original ImageNet training data (upper) and data-free methods (lower). The metric is reported in the non-targeted fooling ratio (%)

Method	AlexNet[1]	GoogleNet	VGG16	VGG19	ResNet152
UAP [14]	93.3	78.9	78.3	77.8	84.0
GAP [19]	-	82.7	83.7	80.1	-
Ours(ImageNet [11])	**96.17**	**88.94**	**94.30**	**94.98**	**90.08**
FFF [18]	80.92	56.44	47.10	43.62	-
AAA [21]	89.04	75.28	71.59	72.84	60.72
GD-UAP [17]	87.02	71.44	63.08	64.67	37.3
Ours (COCO [12])	89.9	**76.8**	**92.2**	**91.6**	**79.9**
Ours (VOC [5])	89.9	76.7	**92.2**	90.5	79.1
Ours (Places365 [29])	**90.0**	76.4	92.1	91.5	78.0

Takeaway

Logit vector based PCC analysis

Universal Adversarial Perturbations are not Bugs  **, They are Features.**

First to achieve data-free targeted universal attack

Understanding Adversarial Examples from the Mutual Influence of Images and Perturbations

Chaoning Zhang* chaoningzhang1990@gmail.com
Philipp Benz* pbenz@kaist.ac.kr
Tooba Imtiaz timtiaz@kaist.ac.kr
In-So Kweon iskweon@kaist.ac.kr
* indicates equal contribution
Robotics and Computer Vision (RCV) Laboratory
Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, Yuseong-gu, Daejeon 34141, Korea

Abstract

A wide variety of works have explored the reason for the existence of adversarial examples, but there is no consensus on the explanation. We propose to treat the DNN logits as a vector for feature representation, and exploit them to analyze the mutual influence of two independent inputs based on the Pearson correlation coefficient (PCC). We utilize this vector representation to understand adversarial examples by disentangling the clean images and adversarial perturbations, and analyze their influence on each other. Our results suggest a new perspective towards the relationship between images and universal perturbations: Universal perturbations contain dominant features, and images behave like noise to them. This feature perspective leads to a new method for generating targeted universal adversarial perturbations using random source images. We are the first to achieve the challenging task of a targeted universal attack without utilizing original training data. Our approach using a proxy dataset achieves comparable performance to the state-of-the-art baselines which utilize the original training dataset.

Figure 1. Based on our observation that *adversarial perturbations contain dominant features and images behave like noise to them*, we design a new method of generating targeted universal adversarial perturbations without data, by using a proxy dataset.

1. Introduction

Deep neural networks (DNNs) have shown impressive performance in numerous applications, ranging from image classification [16, 48] to motion regression [8, 47]. However, DNNs are also known to be vulnerable to adversarial attacks [42, 38]. A wide variety of previous works [14, 43, 44, 21, 34, 3] explore the reason for the existence of adversarial examples, but there is a lack of consensus on the explanation [1]. While the working mechanism of DNNs is not fully understood, one widely accepted interpretation considers DNNs as feature extractors [16], which inspires the recent work [17] to link the existence of adversarial examples to non-robust features in the training dataset.

Contrary to previous works analyzing adversarial examples as a whole (summation of image and perturbation), we instead propose to analyze adversarial examples by disentangling images and perturbations and studying their mutual influence. Specifically, we analyze the influence of two independent inputs on each other in terms of contributing to the obtained feature representation when the inputs are combined. We treat the network logit outputs as a means of feature representation. Traditionally, only the most important logit values, such as the highest logit value for classification tasks, are considered while other values are disregarded. We propose that all logit values contribute to the feature representation and therefore treat them as a

1

Full paper ID:1029