Semantically Adversarial Learnable Filters

Ali Shahin Shamsabadi, Changjae Oh, Andrea Cavallaro

***Abstract*—We present an adversarial framework to craft perturbations that mislead classifiers by accounting for the image content and the semantics of the labels. The proposed framework combines a structure loss and a semantic adversarial loss in a multi-task objective function to train a fully convolutional neural network. The structure loss helps generate perturbations whose type and magnitude are defined by a target image processing filter. The semantic adversarial loss considers groups of (semantic) labels to craft perturbations that prevent the filtered image from being classified with a label in the same group. We validate our framework with three different target filters, namely detail enhancement, log transformation and gamma correction filters; and evaluate the adversarially filtered images against three classifiers, ResNet50, ResNet18 and AlexNet, pre-trained on ImageNet. We show that the proposed framework generates filtered images with a high success rate, robustness, and transferability to unseen classifiers. We also discuss objective and subjective evaluations of the adversarial perturbations.**

I. Introduction

DEEP Neural Networks (DNNs) are vulnerable to image perturbations that are crafted to cause misclassification [1], [2]. These adversarial perturbations can be classified as norm-bounded or content-based. *Norm-bounded* perturbations are characterised by a limited l_p distortion, with the aim of generating imperceptible changes [3], [4], [5], [6], [7], [8], [9], [10]. However, this approach to craft adversarial examples limits their robustness to adversarial defences and hinder their transferability to unseen classifiers [11], [12]. Instead, *content-based* perturbations, which are crafted considering specific image properties, introduce unrestricted intensity changes [13], [14], [15], [16], [17], [18]. Content-based perturbations may modify colours (SemanticAdv [17], ColorFool [13] and ACE [19]) or image structures (EdgeFool [14]). However, these perturbations may cause visible distortions (e.g. unrealistic colours).

To overcome this problem, we propose to adversarially manipulate an image while mimicking the effect of traditional image processing filters (see Figure 1). To this end, we propose an adversarial framework that learns to craft perturbations using a multi-task objective function that combines a structure loss and a semantic adversarial loss to train, end-to-end, a fully convolutional neural network (FCNN). The structure loss supports the learning of the properties of a target image filter to control the structure of the perturbation. The semantic adversarial loss accounts for pre-defined groups of labels and prevents the classifier from predicting a label from the same group (i.e. a semantically similar category). The proposed framework, FilterFool, extends our previous work [14] by learning the residual of filters to generate different types of adversarial enhancement. We also introduce a semantic adversarial loss that improves the effectiveness of adversarial examples. We validate FilterFool on ImageNet [20] with three classifiers, namely ResNet50, ResNet18 [21] and AlexNet [22] using Log transformation, Gamma correction, linear and non-linear detail enhancement as filters. The code implementing FilterFool is available at https://github.com/smartcameras/FilterFool.

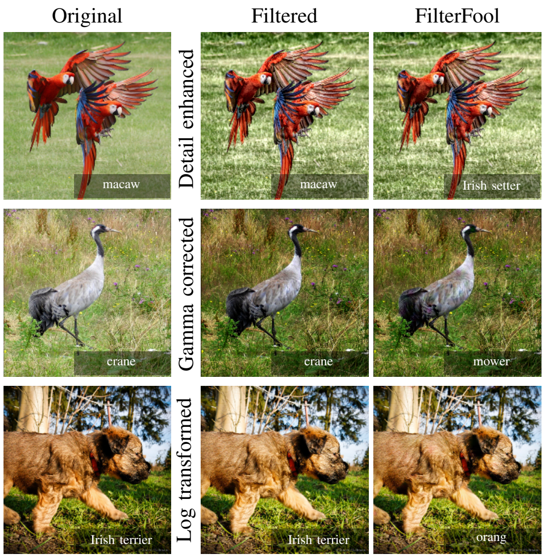

Fig. 1. Original and filtered images modified by three traditional image processing filters and by the proposed FilterFool framework. FilterFool accounts for the image content to produce adversarial examples that mimic the target filter and mislead classifiers. The label shown with each image is the corresponding ResNet50 prediction.

II. Problem Definition

Let $\mathbf{I}$ be an RGB image and $\hat{y} \in \{1, ..., i, ..., D\}$ its ground-truth class. Let a D-class classifier consist of a backbone of layers, a softmax and an argmax operation. The backbone produces logit values for $\mathbf{I}$, $\mathbf{z} = (z_i)_{i=1}^{D}$, where $z_i \in \mathbb{R}$ is the logit value associated with class i. The softmax normalises the logit values to predict the probability for all the classes, $\mathbf{p} = (p_i)_{i=1}^{D}$, where

$$p_i = \frac{e^{z_i}}{\sum_{d=1}^{D} e^{z_d}} \in [0, 1] \tag{1}$$

represents the probability of $\mathbf{I}$ being associated with class i and $\sum_{i=1}^{D} p_i = 1$. The argmax predicts the class $y \in \{1, ..., D\}$ as:

$$y = \arg\max_{i=1,...,D} p_i. \tag{2}$$

Note that $y \neq \hat{y}$ when the prediction of the classifier is incorrect.

Ali Shahin Shamsabadi, Changjae Oh, Andrea Cavallaro are with Centre for Intelligent Sensing, Queen Mary University of London, UK.

An adversarial perturbation, $\boldsymbol{\delta}$, modifies $\mathbf{I}$ to generate an adversarial example, $\dot{\mathbf{I}}$, as $\dot{\mathbf{I}} = \mathbf{I} + \boldsymbol{\delta}$ so that its (adversarial) class, $\dot{y} \in \{1, ..., D\}$, assigned by the classifier, differs from that of $\mathbf{I}$: $\dot{y} \neq y$. Traditional adversarial examples flip the categorical label of an image disregarding the semantics of the labels [23]. However, labels may be synonymous (e.g. screen and television), highly granular (e.g. the ImageNet dataset [20] includes labels for 130 breeds of dogs) [24], or may belong to the same semantic class (e.g. the labels of the Scene365 dataset can be divided into two groups, namely private and non-private scene [25]).

Our goal is to group the D categorical labels into S classes and to prevent an attack from selecting a label within the same class. Let $\mathbf{W} \in \{0,1\}^{D\times S}$ be a matrix that identifies categorical labels belonging to the same class s. In this paper, we focus on the $D = 1000$ ImageNet labels and consider the $S = 11$ classes generated through WordNet [24]. These classes are dogs (containing 130 labels); other mammals (88 labels); birds (59 labels); reptiles, fish, amphibians (60 labels); invertebrates (61 labels); food, plants, fungi (63 labels); devices (172 labels); structures, furnishing (90 labels); clothes, covering (92 labels); implements, containers, misc. objects (117 labels); and vehicles (68 labels). Other scenarios include the $D = 365$ Scene365 labels and the $S = 2$ classes defining private (60 labels) or non-private (305 labels) PixelPrivacy scenes [25]; and the $D = 30$ Cityscape labels and $S = 8$ classes defined for urban scenes: flat (4 labels), construction (6 labels), nature (2 labels), vehicle (8 labels), sky (1 label), object (4 labels), human (2 labels), and void (3 labels) [26].

Adversarial attacks can be untargeted or targeted. The perturbation of an *untargeted* attack is crafted to simply evade the original label, whereas the perturbation of a *targeted* attack is crafted to induce the prediction of a *specific* label, $\dot{y}_t \neq y$. In this paper, we focus on untargeted attacks, which are normally more efficient [7] than targeted attacks, which generally require larger distortions and longer time to craft a perturbation that reaches the desired (adversarial) label [27].

III. Background

In this section, we review norm-bounded attacks (FGSM [3], BIM [4], CW [7], DeepFool [6], SparseFool [8], RP-FGSM [28]) and content-based attacks that consider colour (SemanticAdv [17] , ColorFool [13], ACE [19]) and structure (EdgeFool [14]).

FGSM [3] determines whether to increase or decrease the value of each pixel of $\mathbf{I}$ by defining $\boldsymbol{\delta} = \epsilon \,\text{sign}\left(\nabla_{\mathbf{I}} J\left(\mathbf{I}, y\right)\right)$, with a small $\epsilon \in \mathbb{N}$ based on the sign of $\nabla_{\mathbf{I}} J(\cdot)$, the gradient of the loss function with respect to $\mathbf{I}$:

$$\dot{\mathbf{I}}_{\text{FGSM}} = \mathbf{I} \pm \boldsymbol{\delta} \tag{3}$$

The adversarial image generated by FGSM, $\dot{\mathbf{I}}_{\text{FGSM}}$, within the ϵ-neighbourhood of $\mathbf{I}$, can be the untargeted attack when $\dot{\mathbf{I}}_{\text{FGSM}} = \mathbf{I} + \boldsymbol{\delta}$ or can be the targeted attack when $\dot{\mathbf{I}}_{\text{FGSM}} = \mathbf{I} - \boldsymbol{\delta}$ with selecting y as a specific label. BIM [4] extends FGSM by iteratively generating the adversarial perturbation by aggregating $N = \lfloor \min(\epsilon + 4, 1.25\epsilon) \rceil$, where $\lfloor\cdot\rceil$ is a rounding operation to the nearest integer, perturbations $\boldsymbol{\delta}_n$ as

$$\dot{\mathbf{I}}_{\text{BIM}} = \mathcal{C}_{\mathbf{I},\epsilon}\left(\dot{\mathbf{I}}_N\right), \tag{4}$$

where $\mathcal{C}_{\mathbf{I},\epsilon}(\cdot)$ clips the pixel intensities of the adversarial image, $\dot{\mathbf{I}}_N = \mathbf{I} \pm \sum_{n=1}^{N} \boldsymbol{\delta}_n$, with respect to three constant images, $\mathbf{E}$, $\mathbf{0}$ and $\mathbf{255}$, whose pixel intensities are ϵ, 0 and 255, respectively:

$$\mathcal{C}_{\mathbf{I},\epsilon}(\dot{\mathbf{I}}_N) = \min\{\mathbf{255}, \mathbf{I} + \mathbf{E}, \max\{\mathbf{0}, \mathbf{I} - \mathbf{E}, \dot{\mathbf{I}}_N\}\}, \tag{5}$$

and

$$\boldsymbol{\delta}_n = \text{sign}\left(\nabla_{\mathbf{I}+\boldsymbol{\delta}_{n-1}} J\left(\mathbf{I} + \boldsymbol{\delta}_{n-1}, y\right)\right). \tag{6}$$

Similar to FGSM, BIM can be a untargeted attack by $\dot{\mathbf{I}}_N = \mathbf{I} + \sum_{n=1}^{N} \boldsymbol{\delta}_n$ or an targeted attack by $\dot{\mathbf{I}}_N = \mathbf{I} - \sum_{n=1}^{N} \boldsymbol{\delta}_n$ with selecting y as a specific label. The iterative process helps BIM exploit finer perturbations that improve, compared to FGSM, the ability of the attack to mislead the classifier.

CW [7] minimises the l_0, l_2 or l_∞ norm of the difference between the original image, $\mathbf{I}$, and the adversarial image, $\dot{\mathbf{I}}$, and the difference between the logit value, $\dot{z}_y$, of $\dot{\mathbf{I}}$ belonging to the same class as $\mathbf{I}$ and the maximum logit value among all the other classes:

$$\dot{\mathbf{I}}_{\text{CW}} = \arg\min_{\dot{\mathbf{I}}} \left(\|\dot{\mathbf{I}} - \mathbf{I}\|_p + c_{\mathbf{I}}(\dot{z}_y - \max_{i=1,\dots,D}\{\dot{z}_i; i \neq y\})\right), \tag{7}$$

where $p \in \{0, 2, \infty\}$ and $c_{\mathbf{I}} > 0$ is a constant selected via line search which makes the computations expensive [29], [30].

DeepFool [6] iteratively generates an adversarial perturbation whose l_2 norm is bounded. The adversarial perturbation at each iteration is the orthogonal projection of the adversarial image from the previous iteration onto the closest linearised boundary of class y in the decision space. SparseFool [8] combines the DeepFool adversarial approach and the low mean curvature of DNNs in the neighbourhood of each image to perturb only a few pixels, thus resulting in sparse adversarial perturbations with a small l_1 norm.

Private FGSM (P-FGSM) [10], and its extension to multiple classifier and seen defences RP-FGSM [28], is an iterative norm-bounded attack that uses semantically pre-defined labels ($S = 2$). P-FGSM follows the BIM's (Eq. 4) strategy to produce adversarial images considering private and non-private groups of labels as pre-defined semantics of the categorical labels. P-FGSM chooses the target class randomly from a subset of classes defined based on the prediction probability sorted in a descending order, $\mathbf{p}'$, and a threshold on the cumulative probability, $\tau_p \in [0, 1]$:

$$y_t = R\left(\{y_{j+1} : \sum_{i=1}^{j} {p'}_i > \tau_p, j \in \{1, ..., D-1\}\}\right), \tag{8}$$

where $R(\cdot)$ randomly chooses a class from the set of non-private classes whose cumulative probability exceeds τ_p.

SemanticAdv [17] generates adversarially colourised images in the HSV colour space by adding to the hue and saturation of $\mathbf{I}$ a random perturbation chosen uniformly from the range of valid values. SemanticAdv draws new perturbations until the classifier is misled (up to 1000 attempts). The SemanticAdv adversarial image, $\dot{\mathbf{I}}_{\text{SA}}$, is:

$$\dot{\mathbf{I}}_{\text{SA}} = \beta^{-1}\Big(\left[\mathbf{I}_H + \boldsymbol{\delta}_H, \mathbf{I}_S + \boldsymbol{\delta}_S, \mathbf{I}_V\right]\Big), \tag{9}$$

where $\beta(\cdot)$ converts $\mathbf{I}$ to its hue, $\mathbf{I}_H$, saturation, $\mathbf{I}_S$, and value, $\mathbf{I}_V$ components; $[\cdot,\cdot,\cdot]$ represents the channel-wise concatenation; and $\boldsymbol{\delta}_H$ and $\boldsymbol{\delta}_S$, where each component ranges $[0,1]$, are the final perturbations on the hue and saturation channels, respectively. Because $\mathbf{I}_H$ and $\mathbf{I}_S$ are changed by the same amount, the colours of $\dot{\mathbf{I}}_{\text{SA}}$ may look unnatural.

ColorFool [13] improves the naturalness of adversarial images compared to SemanticAdv [17] by identifying non-sensitive and sensitive regions through semantic segmentation. The perturbations operate on the a and b channels of the Lab colour space: the perturbations of $N_{\bar{S}}$ non-sensitive regions are drawn randomly from the whole range of possible values, whereas the perturbations of N_S sensitive regions are chosen randomly from pre-defined natural-colour ranges, defined based on human perception. The ColorFool adversarial image, $\dot{\mathbf{I}}_{\text{CF}}$, is:

$$\dot{\mathbf{I}}_{\text{CF}} = \rho^{-1}\Big(\sum_{t=1}^{N_S}\left(\rho(\mathbf{S}_t)+\mathbf{N}_t\right)+\sum_{t=1}^{N_{\bar{S}}}\left(\rho(\bar{\mathbf{S}}_t)+\bar{\mathbf{N}}_t\right)\Big), \quad (10)$$

where $\rho(\cdot)$ is the RGB-to-Lab colour-space conversion and $\mathbf{N}_t$ and $\bar{\mathbf{N}}_t$ are the colour perturbations of the t-th sensitive, $\mathbf{S}_t$, and non-sensitive, $\bar{\mathbf{S}}_t$, region, respectively.

Unlike SemanticAdv and ColorFool that randomly change the colour, Adversarial Colour Enhancement (ACE) [19] produces colourised adversarial images using a piecewise-linear colour adjustment filter, $\Gamma_\Omega(\cdot)$, where Ω represents the parameters of the filter, with K pieces. Each pixel of the ACE adversarial image, $\dot{\mathbf{I}}_{\text{ACE}}$, is obtained by filtering the original image, I_x, as follows:

$$\dot{I}_{\text{ACE},x} = \Gamma_\Omega(I_x) = \sum_{i=1}^{\lfloor \frac{I_x}{s} \rfloor} \Omega_i + \frac{I_x(\text{mod} s)}{s}.\Omega_{\lceil \frac{I_x}{s} \rceil}, \quad (11)$$

where s is the size of each piece. ACE learns the parameters of the filter, Ω, by optimising the combination of the CW loss function [7], a colour adjustment constraint on the distance between each parameter Ω_i and its initial value $\frac{1}{K}$, corresponding to the unchanged image:

$$\arg\min_{\Omega}\left((\dot{z}_y - \max_{i=1,\ldots,D}\{\dot{z}_i; i \neq y\}) + \nu\sum_{i=1}^{\kappa}(\Omega_i - \frac{1}{K})^2\right), \quad (12)$$

where ν and κ are a balance factor and size of Ω, respectively.

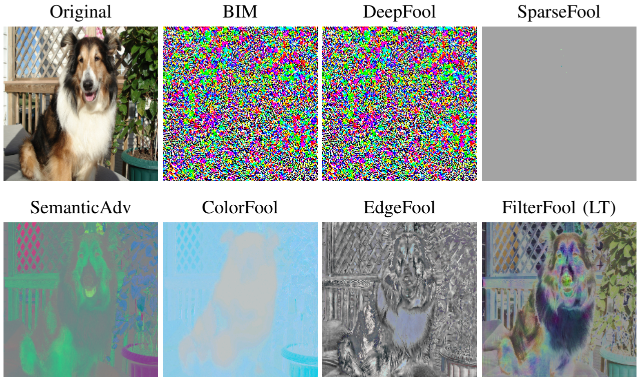

Fig. 2. Comparison of perturbations generated by selected adversarial methods. To facilitate visualisation, the values are scaled between 0 and 255. KEY – BIM: Basic Iterative Method; LT: log transformation.

TABLE I

Comparison of adversarial attacks in terms of information used to craft the perturbations (l_p norm for bounded-perturbations; Col.: colour information; Obj.: information about objects in the scene; Str.: information about structures in the image) and consideration of semantic (Sem.) relationships between classes.

Ref.	Method	l_0	l_1	l_2	l_∞	Col.	Obj.	Str.	Sem.
[3]	FGSM				✓				
[4]	BIM				✓				
[6]	DeepFool			✓					
[8]	SparseFool		✓						
[7]	CW	✓		✓	✓				
[28]	RP-FGSM				✓				✓
[17]	SemAdv					✓			
[13]	ColorFool					✓	✓		
[19]	ACE					✓			
[14]	EdgeFool							✓	
	FilterFool							✓	✓

EdgeFool [14] generates adversarial perturbations that enhance the image details. EdgeFool trains a FCNN, $R_{\theta_{\text{EF}}}(\cdot)$, where θ_{EF} represents all the parameters defining the FCNN, with a multi-task loss function:

$$\theta^*_{\text{EF}} = \arg\min_{\theta_{\text{EF}}}\big(\underbrace{\|R_{\theta_{\text{EF}}}(\mathbf{I})-\mathbf{I}_g\|_2}_{\mathcal{L}_s(\cdot)} + c(\dot{z}_y - \max\{\dot{z}_i; i \neq y\})\big), \quad (13)$$

where $\mathcal{L}_s(\cdot)$, the smoothing loss function, quantifies the difference between $\mathbf{I}_g$, the output of a smoothing filter, and $\mathbf{I}_s = R_{\theta_{\text{EF}}}(\mathbf{I})$, the output of the FCNN. The adversarial perturbation operates on the L channel of the Lab colour space by enhancing the image details, $\mathbf{I}_d = \mathbf{I} - \mathbf{I}_s$, using the sigmoid function, $f(a,b) = \left(1+e^{-ab}\right)^{-1} - 0.5$ [31]:

$$\dot{\mathbf{I}}^L = \left(100 f\left(\frac{\mathbf{I}^L_s - v_1}{100}, v_2\right) + v_1\right) + 100 f\left(\frac{\mathbf{I}^L_d}{100}, v_3\right), \quad (14)$$

where the input to the sigmoid is normalised by the maximum value of the L channel (i.e. 100), and v_1, v_2 and v_3 are constants that adjust the midpoint and slope of the sigmoid. The second term in Eq. 13 guides the FCNN to smooth the image in a way that the EdgeFool adversarial image, $\dot{\mathbf{I}}_{\text{EF}}$:

$$\dot{\mathbf{I}}_{\text{EF}} = \rho^{-1}\Big([\dot{\mathbf{I}}^L, \mathbf{I}^a, \mathbf{I}^b]\Big) \quad (15)$$

causes misclassification.

Figure 2 compares sample perturbations generated by norm-bounded and content-based methods. It is possible to notice that the adversarial perturbations of BIM, DeepFool and SparseFool are unrelated to the content, whereas SemanticAdv modifies the colours of the whole image, ColorFool focuses on specific regions identified through semantic segmentation, and EdgeFool and FilterFool craft perturbations that relate to the structures in the image.

Finally, Table I summarises the adversarial attacks discussed in this section and compares them with the proposed framework, FilterFool, which is detailed in the next section.

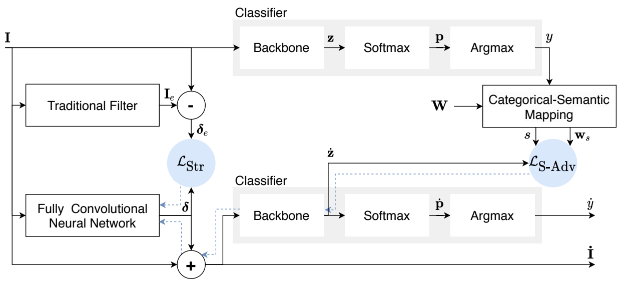

Fig. 3. Block diagram of the proposed framework, FilterFool, which integrates a traditional image processing filter with a Fully Convolutional Neural Network (FCNN) to output adversarial images that mislead a classifier with (adversarial) classes $\dot{y}$ that are semantically different from the original classes (i.e. class predicted by the classifier for the original image, y). During training, the errors measured by $\mathcal{L}_{\text{Str}}$ and $\mathcal{L}_{\text{S-Adv}}$ are backpropagated through the path indicated with ⋯ to optimise the parameters of FCNN. The Backbone represents all parts of the classifier before the Softmax layer. KEY – $\mathbf{I}$ input image; $\mathbf{I}_e$: filtered image; $\boldsymbol{\delta}$: adversarial perturbation; $\mathcal{L}_{\text{Str}}$: structure loss, which is the distance between the enhancement perturbation, $\boldsymbol{\delta}_e = \mathbf{I}_e - \mathbf{I}$, and the learned adversarial perturbation, $\boldsymbol{\delta}$; $\mathbf{W}$ matrix that defines the mapping from categorical to semantic classes; $\mathcal{L}_{\text{S-Adv}}$: semantic adversarial loss; $\dot{\mathbf{I}}$: adversarial image.

IV. FilterFool

A. Learning framework

We aim to generate adversarial perturbations that mimic the output of an image processing filter. To this end, we approximate the target filter with a DNN [32] and, specifically, with a Fully Convolutional Neural Network (FCNN), which learns to produce perturbations that resemble that of an image filter. The parameters of the FCNN are optimised with a multi-objective loss: the *structure loss*, which accounts for the difference between the intensity changes introduced by the filter and the learned perturbation, and an *adversarial loss*, which induces misclassification by operating on pre-defined groups of labels.

Figure 3 shows the block diagram of FilterFool, a general framework to learn to generate adversarial images that resemble those obtained with an image processing filter. Given an image $\mathbf{I}$, the target filter produces $\mathbf{I}_e$. The FCNN learns an adversarial perturbation, $\boldsymbol{\delta}$, by optimising the structure loss, $\mathcal{L}_{\text{Str}}(\cdot,\cdot)$, which represents the error between the intensity changes produced by the target filter, $\boldsymbol{\delta}_e = \mathbf{I}_e - \mathbf{I}$, and the learned adversarial perturbation, $\boldsymbol{\delta}$. The semantic adversarial loss, $\mathcal{L}_{\text{S-Adv}}(\cdot,\cdot)$, causes the adversarial image, $\dot{\mathbf{I}}$, to be misclassified as class $\dot{y}$ that is not only categorically ($y \neq \dot{y}$) but also semantically different from that of the original class, y. The errors measured by $\mathcal{L}_{\text{Str}}(\cdot,\cdot)$ and $\mathcal{L}_{\text{S-Adv}}(\cdot,\cdot)$ are backpropagated to determine the parameters of the FCNN.

In the rest of this section, we describe the details of structure loss $\mathcal{L}_{\text{Str}}$, the semantic adversarial loss $\mathcal{L}_{\text{S-Adv}}$ and the combined multi-task loss used for training the FCNN.

B. Structure loss

We use residual learning to generate intensity changes for the original image, $\mathbf{I}$, to produce the desired filtered image, $\mathbf{I}_e$. We measure the difference between the residual, $\boldsymbol{\delta}_e = \mathbf{I}_e - \mathbf{I}$, and the adversarial perturbation, $\boldsymbol{\delta}$, which is the output of the FCNN. To tailor the perturbation towards the output of the target filter, we define $\mathcal{L}_{l_2}(\cdot,\cdot)$, which penalises the squared error between $\boldsymbol{\delta}$ and $\boldsymbol{\delta}_e$:

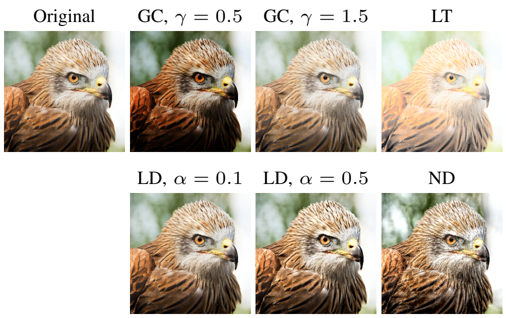

Fig. 4. Sample images filtered by gamma correction (GC, with strength γ), log transformation (LT), linear detail enhancement (LD, with strength α) and nonlinear detail enhancement (ND).

$$\mathcal{L}_{l_2}(\boldsymbol{\delta}, \boldsymbol{\delta}_e) = \|\boldsymbol{\delta} - \boldsymbol{\delta}_e\|^2. \tag{16}$$

As using $\mathcal{L}_{l_2}(\cdot,\cdot)$ alone may cause artefacts in untextured regions [33], we consider another loss that accounts for structures in the image:

$$\mathcal{L}_{\text{SSIM}}(\boldsymbol{\delta}, \boldsymbol{\delta}_e) = 1 - \text{SSIM}(\boldsymbol{\delta}, \boldsymbol{\delta}_e), \tag{17}$$

which is based on the Structural SIMilarity (SSIM) index [34]:

$$\text{SSIM}(\boldsymbol{\delta}, \boldsymbol{\delta}_e) = l(\boldsymbol{\delta}, \boldsymbol{\delta}_e) \cdot c(\boldsymbol{\delta}, \boldsymbol{\delta}_e) \cdot s(\boldsymbol{\delta}, \boldsymbol{\delta}_e), \tag{18}$$

where $l(\cdot,\cdot)$ is a function of the means:

$$l(\boldsymbol{\delta}, \boldsymbol{\delta}_e) = \frac{2\mu_{\boldsymbol{\delta}}\mu_{\boldsymbol{\delta}_e} + c_1}{\mu_{\boldsymbol{\delta}}^2 + \mu_{\boldsymbol{\delta}_e}^2 + c_1}, \tag{19}$$

where $\mu_{\boldsymbol{\delta}} = \frac{1}{M}\sum_{i=1}^{M} \delta_i$ and δ_i is the intensity of i-th element of the image (including the three colour planes) and hence M is three times the resolution of the image; $c(\cdot,\cdot)$ is based on the standard deviations:

$$c(\boldsymbol{\delta}, \boldsymbol{\delta}_e) = \frac{2\sigma_{\boldsymbol{\delta}}\sigma_{\boldsymbol{\delta}_e} + c_2}{\sigma_{\boldsymbol{\delta}}^2 + \sigma_{\boldsymbol{\delta}_e}^2 + c_2}, \tag{20}$$

where $\sigma_{\boldsymbol{\delta}} = \left(\frac{1}{M-1}\sum_{i=1}^{M}(\delta_i - \mu_{\boldsymbol{\delta}})^2\right)^{1/2}$. The structure information, $s(\cdot,\cdot)$, is estimated based on the covariance of the perturbations:

$$s(\boldsymbol{\delta}, \boldsymbol{\delta}_e) = \frac{\sigma_{\boldsymbol{\delta}\boldsymbol{\delta}_e} + c_3}{\sigma_{\boldsymbol{\delta}}\sigma_{\boldsymbol{\delta}_e} + c_3}, \tag{21}$$

where $\sigma_{\boldsymbol{\delta}\boldsymbol{\delta}_e} = \frac{1}{M-1}\sum_{i=1}^{M}(\delta_i - \mu_{\boldsymbol{\delta}})(\delta_i - \mu_{\boldsymbol{\delta}_e})$; and c_1, c_2 and c_3 are small constants that stabilise the division. $\text{SSIM}(\boldsymbol{\delta}, \boldsymbol{\delta}_e) \in [0, 1]$, and the closer to 1, the higher the similarity.

The structure loss, $\mathcal{L}_{\text{Str}}(\cdot,\cdot)$, combines the two losses as

$$\mathcal{L}_{\text{Str}}(\boldsymbol{\delta}, \boldsymbol{\delta}_e) = \mathcal{L}_{l_2}(\boldsymbol{\delta}, \boldsymbol{\delta}_e) + \eta\mathcal{L}_{\text{SSIM}}(\boldsymbol{\delta}, \boldsymbol{\delta}_e), \tag{22}$$

where the hyper-parameter η is determined empirically.

For the specific implementation of this paper, we consider four filters, namely linear and non-linear detail enhancement, log transformation and gamma correction (see Figure 4). For linear detail enhancement, we use an l_0 structure-preserving smoothing filter [35], which linearly scales the image details, $\mathbf{I}_d$, obtained by the difference between the original and the

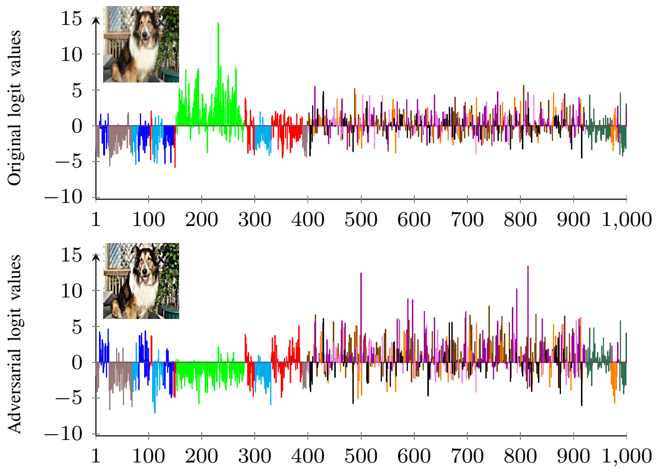

Fig. 5. Logit values generated by ResNet50 for an original image (top) and for adversarial image (bottom) generated with the semantic adversarial loss. This loss minimises the positive logits of the labels that are semantically similar to the original label (e.g. dogs) and increases the logit of a label that belongs to a different class. The horizontal axis represents the labels and colours represent the classes: dogs, other mammals, birds, reptiles,fish, amphibians, invertebrates, food, plants, fungi, devices, structures, furnishing, clothes, covering, implements, containers, misc. objects, vehicles.

output of an l_0 smoothing filter, as $\mathbf{I}_e = \mathbf{I} + \alpha \mathbf{I}_d$, where $\alpha \in (0, 10]$. For non-linear detail enhancement, we enhance image details in the *Lab* colour space with a sigmoid, following [31] and similarly to EdgeFool (see Eq. 15). The log transformation expands darker pixel values and compresses brighter pixel values [36]. Finally, the exponent $1/\gamma$ of gamma correction darkens ($\gamma < 1$) or brightens ($\gamma > 1$) the image [36].

C. Semantic adversarial loss

We aim to devise an adversarial loss that operates on the groups of labels defined by $\mathbf{W}$ (see Sec. II). To this end, we adapt the CW adversarial loss [7] and decrease the logits for the labels that share the same class s with the label of the original image.

Let $\mathbf{w}_s \in \{0, 1\}^D$ be the column of $\mathbf{W}$ that identifies the mapping of the labels to class s. We apply a ReLU to the adversarial logit, $\dot{\mathbf{z}}$, and compute the dot product, $\langle \cdot, \cdot \rangle$, with $\mathbf{w}_s$ to focus the *same-class loss*, $\mathcal{L}_{\text{Sam}}(\cdot, \cdot)$, on the positive logits with the class s:

$$\mathcal{L}_{\text{Sam}}(\dot{\mathbf{I}}, \mathbf{I}) = \langle \text{ReLU}(\dot{\mathbf{z}}), \mathbf{w}_s \rangle. \tag{23}$$

In each training iteration, we select the largest logit of a label not belonging to class s using a *different-class loss* $\mathcal{L}_{\text{Diff}}(\cdot, \cdot)$:

$$\mathcal{L}_{\text{Diff}}(\dot{\mathbf{I}}, \mathbf{I}) = \max(\dot{\mathbf{z}} \odot \hat{\mathbf{w}}_s), \tag{24}$$

where $\hat{\mathbf{w}}_s = \mathbf{1} - \mathbf{w}_s$ represents all the labels whose class differs from that of y, $\mathbf{1} = \{1\}^D$ and $\odot$ is the Hadamard product.

The semantic adversarial loss, $\mathcal{L}_{\text{S-Adv}}(\cdot, \cdot)$, combines the same-class and different-class losses as:

$$\mathcal{L}_{\text{S-Adv}}(\dot{\mathbf{I}}, \mathbf{I}) = \mathcal{L}_{\text{Sam}}(\dot{\mathbf{I}}, \mathbf{I}) - \mathcal{L}_{\text{Diff}}(\dot{\mathbf{I}}, \mathbf{I}). \tag{25}$$

Figure 5 shows an example of logit values for the ImageNet labels before and after using this semantic adversarial loss.

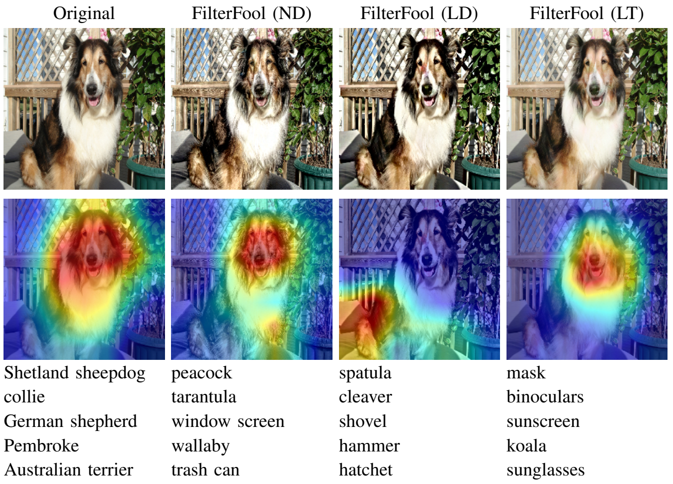

Fig. 6. Comparison between the ResNet50 attention map on an original image and on three FilterFool examples, with the corresponding Top5 predictions. The colour of the heatmap ranging from blue to red indicates the importance of each image region (blue: least important; red: most important) in predicting the Top1 image label.

D. Multi-task loss

We define our objective function, $\mathcal{L}$, as the combination of the structure loss, $\mathcal{L}_{\text{Str}}$, and the semantic adversarial loss, $\mathcal{L}_{\text{S-Adv}}$:

$$\mathcal{L} = \mathcal{L}_{\text{Str}}(\boldsymbol{\delta}, \boldsymbol{\delta}_e) + \mathcal{L}_{\text{S-Adv}}(\dot{\mathbf{I}}, \mathbf{I}). \tag{26}$$

During the iterative process that generates the perturbation, the FCNN learns to craft $\boldsymbol{\delta}$ by backpropagating both $\mathcal{L}_{\text{Str}}$ and $\mathcal{L}_{\text{S-Adv}}$ until $\dot{\mathbf{I}}$ misleads the classifier and $\mathcal{L}_{\text{Str}} < \tau$ (empirically set such that the adversarial image resembles an image enhanced with the target filter) or a maximum number of iterations is reached (3,000 in our case).

Figure 6 shows the Gradient-weighted Class Activation Maps (Grad-CAM) [37] of sample images alongside their Top5 predicted labels by ResNet50. Grad-CAM determines the importance of each neuron in a layer for the predicted Top1 label and the corresponding heatmap of the last convolutional layer is computed as weighted sum of the neuron activations multiplied by their importance, followed by up-sampling to the original image size. It is possible to notice that in the original image ResNet50 focuses on the head, whereas in the FilterFool images the focus is reduced in the face region (ND, LT) or shifted towards another part of the image (LD).

V. Performance measures

We evaluate adversarial attacks based on three main properties, namely effectiveness, robustness and transferability [38].

Effectiveness (or success rate) is the degree to which the adversarial attack succeeds in misleading the classifier. Effectiveness can be measured as the accuracy of the classifier over a dataset. The lower the accuracy, the higher the effectiveness of the attack.

We measure effectiveness as *categorical* (label-level) and *semantic* (class-level) success rates. We also consider the Top$K$ success rate and, because of the study in this paper focuses on ImageNet semantic labels, we also use the so called semantic damage [23]. The semantic damage is the average Wu-Palmar word similarity [39], $\mathrm{Wu}(\cdot,\cdot)$, across the dataset. A damage occurs when the similarity between the original label, λ, and adversarial label, $\dot{\lambda}$ is smaller than a threshold, T_s:

$$\mathrm{Wu}_b = \begin{cases} 1 & \mathrm{Wu}(\lambda, \dot{\lambda}) < T_s \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

where Wu_b is the binarised word similarity based on T_s.

Robustness is the effectiveness of an adversarial attack in the presence of a defence (e.g. median filtering, re-quantization, JPEG compression) that aims to remove the effect of a (potential) perturbation before the image is analysed by the classifier [40]. Robustness can be measured as the difference in accuracy of the classifier over a target dataset when a defence is used with respect to a setting when the defence is not used. The smaller this difference, the higher the robustness of the attack. A special case of defence is adversarial training, when the classifier itself is retrained with adversarial examples to increase its robustness [41].

Finally, *transferability* is the extent to which a perturbation crafted for a classifier is effective in misleading another classifier. Transferability can be measured as the difference in accuracy between the two classifiers over a target dataset. The smaller this difference, the higher the transferability of the attack.

VI. Validation

A. Experimental setup

For the evaluation, we consider the following adversarial attacks: Basic Iterative Method (BIM) [4], DeepFool [6], SparseFool [8], SemanticAdv [17], ColorFool [13], EdgeFool [14], least-likely targeted [4] and private targeted [10] FGSM and BIM. As classifiers we use ResNet with 50 layers (ResNet50) and 18 layers (ResNet18) [21], and AlexNet [22]) trained for the image classification task on ImageNet [20], which includes $D = 1,000$ classes. We use the same 3,000 images (3 images per class) that are randomly selected from the validation set by [13], [14]. We consider $S = 11$ WordNet semantic classes [24].

The dimensions of $\mathbf{I}$ are $224 \times 224 \times 3$. We instantiate FCNN with the architecture of [42], which consists of 7 convolution layers with 24 intermediate feature maps and 3×3 kernels. The last convolution layer applies a 1×1 convolution that generates $\mathbf{I}_s$. The dilation factor of each layer is set to 1, 2, 4, 8, 16, 32, 1, and 1, respectively. A leaky rectified linear unit is applied after padding and normalising each intermediate convolutional layer, except the last one. The hyper-parameter $\eta = 0.01$ is chosen empirically to balance the errors of $\mathcal{L}_{l_2}$ and $\mathcal{L}_{\text{SSIM}}$ on the structure loss as $\mathcal{L}_{\text{SSIM}}$ is bigger than $\mathcal{L}_{l_2}$. Note that the hyper-parameter can affect the number of iterations needed to craft the perturbations. The structure loss and the semantic adversarial loss have equal contribution in the training. Important components of our optimisation process are the choice of the stopping criteria applied to the structure loss and the semantic adversarial loss. The stopping threshold τ is set 0.04, 0.003 and 0.0005 for detail enhancement, log transformation and gamma correction, respectively. For linear detail enhancement and gamma correction, α and γ range from 0.1 to 10. The parameters of the sigmoid in nonlinear detail enhancement are [31], [14]: $v_1 = 56$, $v_2 = 1$, and $v_3 = 15$. The parameters for bit reduction and median smoothing are [40]: 1 to 7 bits per colour channel in steps of 1; kernel of 2×2, 3×3 and 5×5 (the median values in odd and even kernels are the middle ones and the mean of two middle ones, respectively). The quality parameters we use for JPEG are 25, 50, 75 and 100. As an adversarially trained classifier, we use ResNet50 re-trained on adversarial images crafted with BIM [41].

TABLE II
Similarity of the FilterFool adversarial images and perturbations and their corresponding traditionally filtered images and pixel changes.

Attack	Model	SSIM	
		Adv. images	Adv. perturbation
FF (ND)	R50	0.98 ± 0.02	0.97 ± 0.02
	R18	0.98 ± 0.01	0.97 ± 0.01
	A	0.97 ± 0.02	0.96 ± 0.02
FF (Log)	R50	0.99 ± 0.01	0.99 ± 0.00
	R18	0.99 ± 0.01	0.99 ± 0.00
	A	0.98 ± 0.02	0.98 ± 0.02
FF (LD1)	R50	0.99 ± 0.01	0.99 ± 0.00
	R18	0.99 ± 0.01	0.99 ± 0.00
	A	0.98 ± 0.02	0.99 ± 0.01
FF (GC.5)	R50	0.99 ± 0.02	1.00 ± 0.01
	R18	0.99 ± 0.02	1.00 ± 0.01
	A	0.98 ± 0.02	0.99 ± 0.01

B. FilterFool and traditional filters

Figure 7 shows the categorical and semantic success rates of FilterFool and the target filters. As the filter effect becomes stronger, the success rates of Gamma correction and linear detail enhancement in traditional filters increase. However, the traditional filters have lower success rates than FilterFool, as expected. For example, the success rates of Log transformation, non-linear detail enhancement, Gamma correction ($\gamma = 5$) and linear detail enhancement ($\alpha = 5$) in misleading ResNet50 are 6%, 29%, 39% and 51%, respectively. FilterFool is successful in misleading ResNet50, ResNet18 and AlexNet with any strength of filters.

We use both objective and subjective evaluations to measure the differences between the FilterFool adversarial images and the target filtered images. As objective evaluation, we measure the SSIM between FilterFool and traditionally filtered images (see Table II). Because the adversarial perturbation of FilterFool is produced based on the structure loss function in Eq. 22, the SSIM values show that the outputs of FilterFool and the corresponding target filter are highly similar. The structure-aware, small and adversarially guided differences between the FilterFool adversarial images and the target filtered images enable FilterFool to mislead classifiers with high success rates.

We assessed the similarity of the FilterFool images with a panel of observers (Queen Mary Ethics of Research Committee reference number: QMERC20.452). We considered all four filters: linear detail enhancement ($\alpha = 1$), non-linear detail enhancement, Log transformation and Gamma

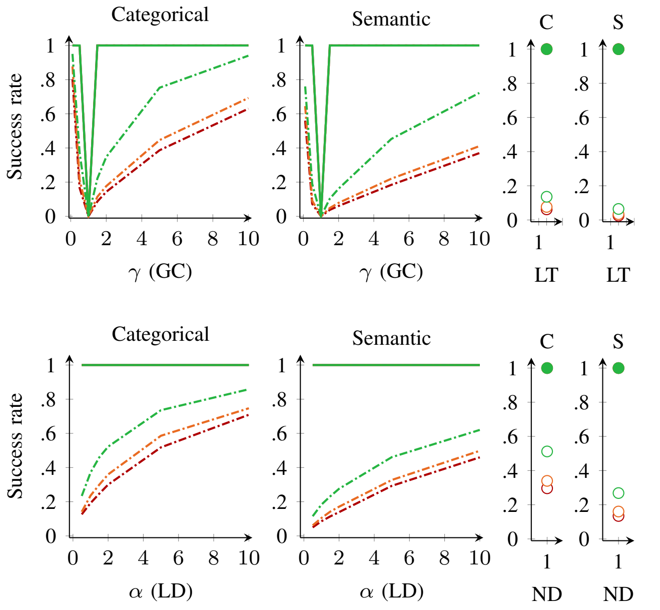

Fig. 7. Comparing the categorical (C) and semantic (S) success rates of FilterFool (——— or ●) with its corresponding traditional filters (-·-·- or ○): Nonlinear Detail enhancement (ND), Log Transformation (LT), Gamma Correction (GC), Linear Detail enhancement (LD) with strengths varying from 0.1 to 10 on the ImageNet dataset against ResNet50, ResNet18 and AlexNet.

correction ($\gamma = 0.5$). We first ranked the 3,000 output images from FilterFool based on their SSIM values with respect to the filtered images and then divided the images into three groups, namely low, middle, and high SSIM values (the average and standard deviation of SSIM values are reported in Table II). Next, we randomly chose 5 images from each group to obtain a total of 15 images for each filter. Next 33 human subjects were shown a pair of images from FilterFool and the corresponding target filter with an original image, and asked: *Are there any differences between these two filtered images?* Each pair of images was shown for 7 seconds, after which the answer (*Yes* or *No*) was provided. We randomised the display order of the image pairs and used the Mean Opinion Score (MOS) to quantify the percentage of answers that perceive no differences between the FilterFool adversarial images and the target filtered images. The resulting MOS is $83.41 \pm 9.83\%$, which suggests a high fidelity by FilterFool in mimicking the target filters.

Figure 8 shows the effect of different filters used in FilterFool and the influence of their strengths on the categorical success rate and *transferability*. In each plot, the two top lines (or points) are the categorical success rate against the classifier that was used to generate the adversarial images. Four bottom lines (or points) show the categorical transferability to unseen classifiers. FilterFool achieves a high categorical success rate for all three classifiers. As expected, the transferability increases with the strength of the filtering effect [17], [13]. We also consider here FilterFool-c, the categorical version of FilterFool, for which each categorical label is considered as a class on its own, i.e. $S = D$ and hence $\mathcal{L}_{\text{S-Adv}}(\cdot,\cdot)$ becomes the adversarial CW loss. The use of the semantic adversarial loss improves the categorical transferability by 8%, as unseen classifiers may misclassify an image to a categorically different label which is, however, semantically similar.

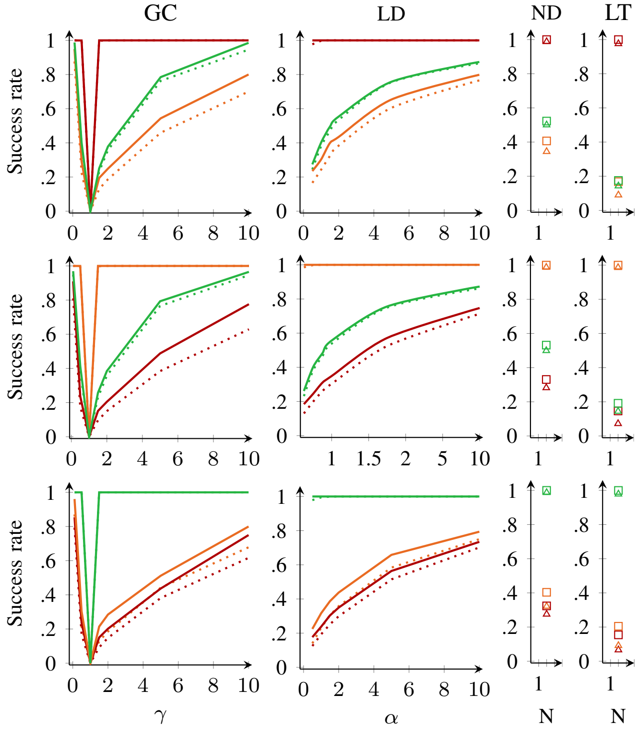

Fig. 8. The categorical success rate and transferability of FilterFool (——— or □) and FilterFool-c (······ or △) to evaluate the effect of type and strength of selected filters, namely Nonlinear Detail enhancement (ND), Log Transformation (LT), Gamma Correction (GC) with strength γ and Linear Detail enhancement (LD) with strength α, as well as the proposed semantic adversarial loss function. Adversarial images are generated against ResNet50, ResNet18 and AlexNet in the first, second and third rows, respectively. Hence, in each plot, the top two overlapped lines (or points) show the categorical success rate when the classifier is seen, while other four bottom lines (or points) show the transferability against unseen classifiers. Note that $\gamma = 1$ in the Gamma correction plots corresponds to the original images.

Figure 9 shows the impact of the filters in FilterFool on the *robustness* against defences. Coloured regions represent the range of categorical success rate in the presence of various parameters of bit reduction, median filtering and JPEG compression, which are defined in Sec. VI-A. As for transferability, from the coloured region covered by linear detail enhancement and gamma correction, the robustness improves with the strength of filters. FilterFool is more robust than FilterFool-c as the semantic adversarial loss reduces the group of logits that are sematically related to the predicted label.

C. Comparisons

The tables in this section show the success rate on *on-diagonal* elements of each sub-table and the transferability on *off-diagonal* elements of each sub-table. The second column of each table shows the classifiers used to craft the adversarial images, whereas the first row shows the classifiers (with →) used for testing.

Table III reports the categorical success rate and the transferability of several state-of-the-art adversarial attacks. Limiting the l_p norm of adversarial perturbations also limits the

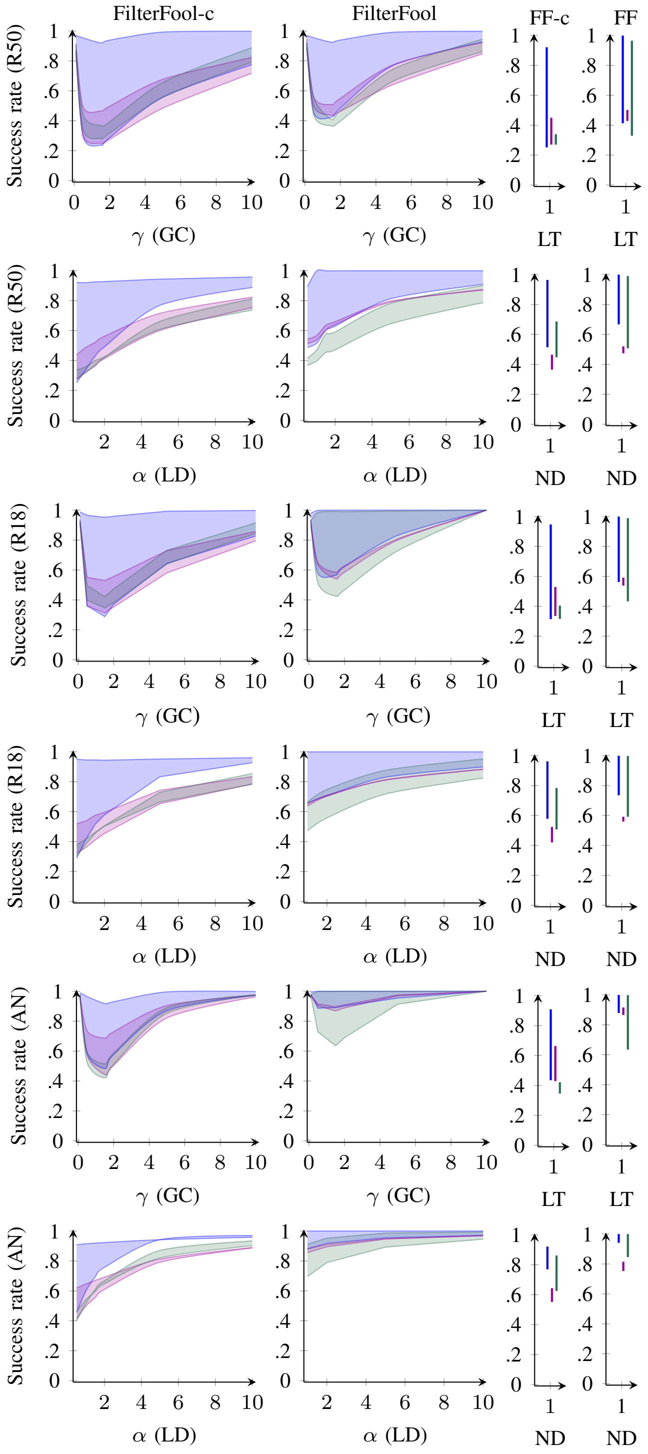

Fig. 9. Robustness of FilterFool-c and FilterFool to defence frameworks. The coloured areas are the categorical success rate of FilterFool-c and FilterFool in the presence of the defence frameworks that use (1-7) bit reduction, (2,3,5) median smoothing and (25,50,75,100) JPEG compression against ResNet50 (R50), ResNet18 (R18) and AlexNet (AN) for all four filters; Nonlinear Detail enhancement (ND), Log Transformation (LT), Gamma Correction (GC), Linear Detail enhancement (LD) with different strengths. γ and α are the strengths of GC and LD that range from 0.1 to 10.

TABLE III
CATEGORICAL SUCCESS RATE AND TRANSFERABILITY ON IMAGENET.

Attack	Classifier	→ResNet50	→ResNet18	→AlexNet
BIM [4]	ResNet50	1.000	0.127	0.037
	ResNet18	0.109	1.000	0.041
	AlexNet	0.038	0.050	1.000
LL-FGSM [4]	ResNet50	0.850	0.483	0.414
	ResNet18	0.418	0.966	0.434
	AlexNet	0.266	0.362	0.993
LL-BIM [4]	ResNet50	1.000	0.173	0.097
	ResNet18	0.129	1.000	0.100
	AlexNet	0.117	0.153	0.998
P-FGSM [10]	ResNet50	1.000	0.233	0.172
	ResNet18	0.203	1.000	0.183
	AlexNet	0.151	0.189	1.000
DeepFool [6]	ResNet50	0.983	0.071	0.018
	ResNet18	0.055	0.991	0.017
	AlexNet	0.019	0.031	0.967
SparseFool [8]	ResNet50	0.990	0.167	0.176
	ResNet18	0.086	0.997	0.134
	AlexNet	0.062	0.079	1.000
SemanticAdv [17]	ResNet50	0.890	0.540	0.770
	ResNet18	0.422	0.931	0.757
	AlexNet	0.359	0.431	0.994
ColorFool [13]	ResNet50	0.917	0.346	0.592
	ResNet18	0.223	0.934	0.541
	AlexNet	0.114	0.147	0.995
EdgeFool [14]	ResNet50	0.981	0.357	0.512
	ResNet18	0.278	0.989	0.510
	AlexNet	0.272	0.333	0.995
FilterFool (LT)	ResNet50	1.000	0.168	0.175
	ResNet18	0.146	1.000	0.191
	AlexNet	0.154	0.205	1.000
FilterFool (ND)	ResNet50	1.000	0.407	0.523
	ResNet18	0.331	1.000	0.532
	AlexNet	0.324	0.403	1.000
FilterFool (GC)	ResNet50	0.999	0.292	0.402
	ResNet18	0.237	1.000	0.414
0.5	AlexNet	0.237	0.303	1.000
FilterFool (LD)	ResNet50	1.000	0.303	0.402
	ResNet18	0.245	1.000	0.402
1.0	AlexNet	0.237	0.316	1.000

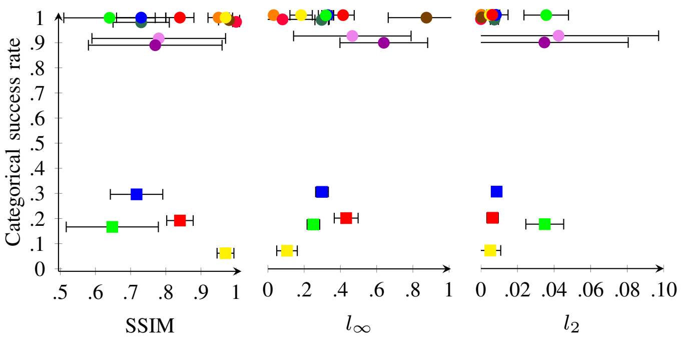

Fig. 10. SSIM, l_2 and l_∞ values and categorical success rate of BIM●, DeepFool●, SparseFool●, SemanticAdv●, ColorFool●, EdgeFool●, FilterFool (ND)●, FilterFool (LD1)●, FilterFool (Log)●, FilterFool (GC.5)●, Traditional filter (GC.5)■, Traditional filter (ND)■, Traditional filter (LD1)■ and Traditional filter (Log)■.

categorical transferability of adversarial images. For example, although DeepFool adversarial images on ResNet50 achieve 98.3% categorical success rate, only 7.1% and 1.8% of them are transferable to ResNet18 and AlexNet, respectively.

Next, we relate the magnitude of the perturbations to their categorical success rate. Figure 10 shows the categorical success rate of adversarial images with respect to their SSIM, l_2 and l_∞. The l_∞ of SparseFool is the biggest as SparseFool perturbs a few pixels but with a large magnitude. BIM changes

TABLE IV
SEMANTIC SUCCESS RATE AND TRANSFERABILITY ON IMAGENET.

Attack	Classifier	→ResNet50	→ResNet18	→AlexNet
BIM [4]	ResNet50	0.405	0.052	0.015
	ResNet18	0.045	0.425	0.019
	AlexNet	0.015	0.021	0.480
LL-FGSM [4]	ResNet50	0.526	0.244	0.211
	ResNet18	0.217	0.727	0.225
	AlexNet	0.122	0.179	0.823
LL-BIM [4]	ResNet50	0.969	0.112	0.078
	ResNet18	0.095	0.975	0.088
	AlexNet	0.074	0.085	0.973
P-FGSM [10]	ResNet50	0.898	0.103	0.083
	ResNet18	0.095	0.890	0.084
	AlexNet	0.070	0.086	0.897
DeepFool [6]	ResNet50	0.358	0.030	0.009
	ResNet18	0.023	0.384	0.008
	AlexNet	0.006	0.012	0.413
SparseFool [8]	ResNet50	0.390	0.073	0.087
	ResNet18	0.034	0.408	0.065
	AlexNet	0.024	0.030	0.441
SemanticAdv [17]	ResNet50	0.406	0.274	0.496
	ResNet18	0.198	0.442	0.472
	AlexNet	0.169	0.209	0.563
ColorFool [13]	ResNet50	0.400	0.170	0.380
	ResNet18	0.100	0.426	0.331
	AlexNet	0.048	0.062	0.476
EdgeFool [14]	ResNet50	0.374	0.165	0.273
	ResNet18	0.126	0.407	0.262
	AlexNet	0.119	0.149	0.483

TABLE V
SEMANTIC SUCCESS RATE AND TRANSFERABILITY OF FILTERFOOL ON IMAGENET.

Attack	Classifier	→ResNet50	→ResNet18	→AlexNet
FilterFool (LT)	ResNet50	1.000	0.085	0.082
	ResNet18	0.069	1.000	0.093
	AlexNet	0.065	0.094	1.000
FilterFool (ND)	ResNet50	1.000	0.221	0.287
	ResNet18	0.173	1.000	0.299
	AlexNet	0.148	0.204	1.000
FilterFool (GC) 0.5	ResNet50	0.999	0.149	0.206
	ResNet18	0.118	1.000	0.211
	AlexNet	0.111	0.145	1.000
FilterFool (LD) 1.0	ResNet50	1.000	0.153	0.209
	ResNet18	0.121	1.000	0.212
	AlexNet	0.103	0.151	1.000

all the pixels controls the maximum change of each pixel, thus resulting in the smallest l_∞. DeepFool has the smallest l_2 norm, which indeed it minimises. DeepFool, SparseFool, FilterFool (Log) and BIM achieve the highest SSIM values of 1, 0.98, 0.97 and 0.95, respectively.

The magnitude of content-based perturbations are larger than thos of norm-bounded perturbations. The standard deviation of the magnitudes of the perturbations generated by ColorFool and SemanticAdv are bigger than those of content-based perturbations (5× FilterFool's), as their perturbations are chosen randomly from a range that gradually increases until the misleading property is satisfied. FilterFool and the corresponding target filter achieve similar SSIM and l_2 values with respect to the original images. In addition to mimicking the effect of filters, FilterFool can achieve high categorical success rate. For example, the SSIM values of both FilterFool (Log) and Traditional filter (Log) are .97, while the success rate of the former is 100% and the latter is only 6%. In general, l_2 and l_∞ of FilterFool and traditional filters are larger than l_p norm-bounded attacks and smaller than other content-based

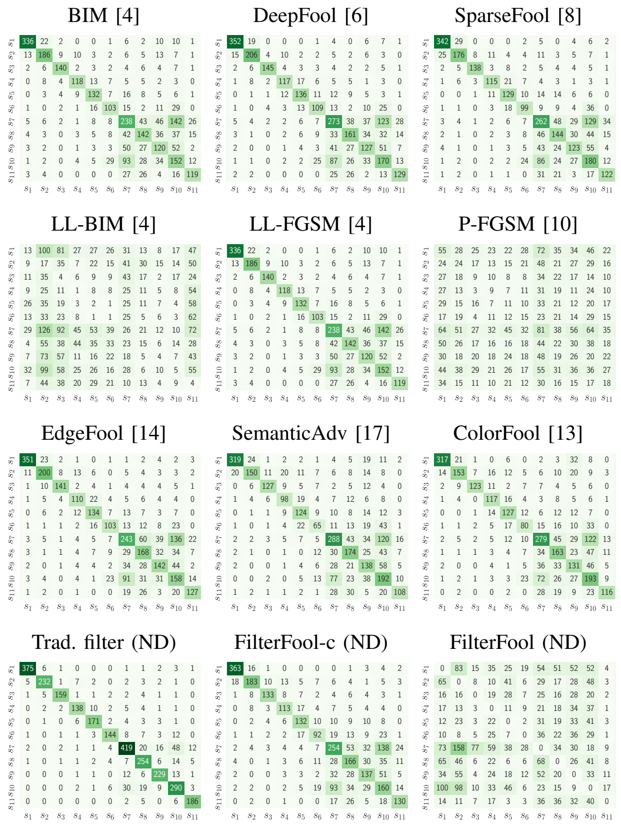

Fig. 11. Confusion matrices showing the success rate of adversarial attacks per semantic class against ResNet50. Rows and columns show the semantic class of the original and adversarial images, namely Dogs (s_1), other mammals (s_2), Bird (s_3), Reptiles, fish, amphibians (s_4), Invertebrates (s_5), Food, plants, fungi (s_6), Devices (s_7), Structures, furnishing (s_8), Clothes, covering (s_9), Implements, containers, misc. objects (s_{10}), Vehicles (s_{11}) generated by BIM, LL-BIM, LL-FGSM, P-FGSM, DeepFool, SparseFool, EdgeFool, SemanticAdv, ColorFool, Nonlinear Detail enhancement (ND) of traditional Trad. filter, ND of FilterFool-c, and ND of FilterFool. The on-diagonal elements of the matrices show that ResNet50 is still able to classify adversarial images of state-of-the-art adversarial attacks, Filter and FilterFool-c with semantically similar classes to the classes of the original images.

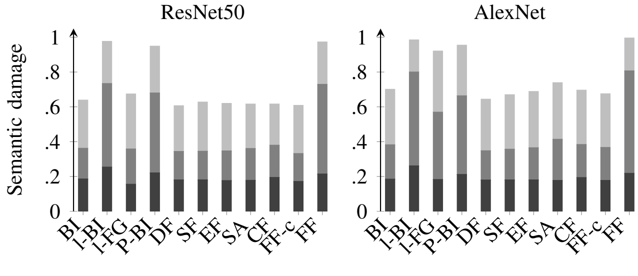

Fig. 12. Semantic damage incurred by Basic Iterative method (BI), least-likely BI (l-BI), least-likely Fast Gradient Sign Method (l-FG), Private BI (P-BI), DeepFool (DF), SparseFool (SF), EdgeFool (EF), SemanticAdv (SA), ColorFool (CF), Nonlinear detail enhancement of FilterFool-c (FF-c) and FilterFool (FF) using three values for T_s: 0.3 ■, 0.5 ▨ and 0.7 □ against ResNet50 and AlexNet trained on the ImageNet dataset. Note that ResNet18 results are similar to ResNet50 results. For each pair of original and adversarial image, the semantic damage is 1 only when the semantic similarity, measured by the word similarity metric (Eq. 27), between the adversarial class and original class is less than the chosen T_s.

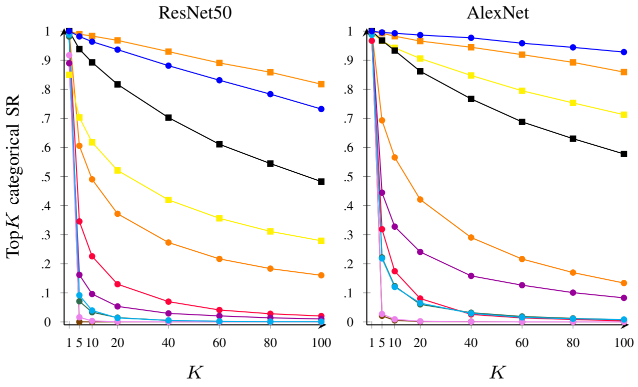

Fig. 13. Top$K$ categorical success rate (SR) of BIM —●—, LL-BIM —■—, LL-FGSM —■—, P-FGSM —■—, DeepFool —●—, SparseFool —●—, EdgeFool —●—, SemanticAdv —●—, ColorFool —●—, Nonlinear detail enhancement of FilterFool-c —●— and FilterFool —●— against ResNet50 and AlexNet trained on ImageNet. ResNet18 results are similar to ResNet50 results.

attacks, e.g. ColorFool and SemanticAdv.

Table IV shows the semantic success rate and transferability of state-of-the-art attacks. The predicted classes of more than 50% of untargeted adversarial images generated by each state-of-the-art attack are semantically similar to the predicted classes of their corresponding original images. While the semantic success rate of LL-FGSM is low, LL-BIM is highly effective in pushing adversarial images to reach the least-likely class, thus resulting in more than 97% semantic success rate, as not all the least-likely classes are semantically different than the original class.

Table V compares the semantic success rate and transferability of FilterFool. In general, the semantic transferability of adversarial attacks is lower than the categorical transferability. Hence, improving the semantic transferability of adversarial attacks is an important direction for future research.

Figure 11 visualises the confusion matrix of the semantic success rate for original and adversarial classes. It is possible to note, for example, that 336, 352, 342, 351, 319, 317 of the adversarial images generated by BIM, DeepFool, SparseFool, EdgeFool, SemanticAdv and ColorFool, respectively, for (original) 390 dog images are misclassified as another breed of dogs. Note also that the 50% semantic success rates of the state-of-the-art attacks reported in Table III are due to confusion between devices and containers. Moreover, we observe the effect of the adversarial loss and the semantic adversarial loss by comparing the confusion matrix of the traditional filter, FilterFool-c, FilterFool with nonlinear detail enhancement. The results of traditional filter mostly falls on the diagonal because of the low success rate.

Figure 12 compares the *semantic damage* for three values of the similarity threshold. FilterFool, least-likely and private attacks inflict greater semantic damage on classifiers than other untargeted attacks as their adversarial labels are semantically different from the original labels.

Figure 13 shows the Top$K$ categorical success rate of the attacks. The categorical success rate of untargeted attacks substantially decreases as K increases, since untargeted attacks mostly shift the original label from the most probable one to another one with high probability. This results in, for instance, the decreased success rate of DeepFool from 98% with $K = 1$ to 35% with $K = 5$. The least-likely targeted attack and FilterFool maintain high success rate even for $K = 100$ as a consequence of targeting semantically different classes and the proposed semantic adversarial loss function, respectively. For example, the Top$K$ categorical success rate of FilterFool only drops to 73%, 80% and 93% for ResNet50, ResNet18 and AlexNet, respectively, when the K changes from 1 to 100. Furthermore, Figure 14 compares the Top$K$ semantic success rate of FilterFool and targeted attacks. The Top$K$ semantic success rate of FilterFool and the least-likely attacks decreases as K increases. However, this drop is higher in least-likely targeted attacks than FilterFool, as FilterFool performs on a group of labels as opposed to one label considered in targeted attacks. For example, Top1, Top5 and Top10 semantic success rates of LL-BIM is 95%, 75% and 58%, while 100%, 97% and 95% for FilterFool with detail enhancement ($\alpha = 1$).

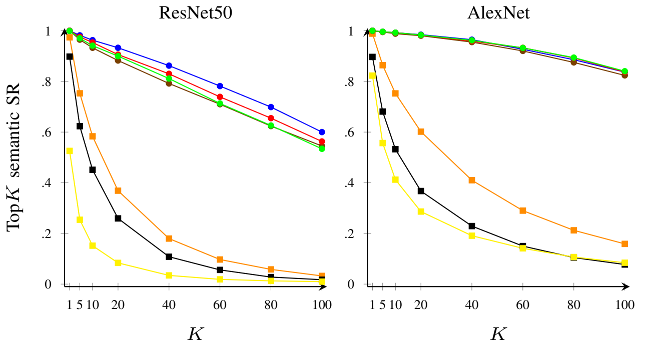

Fig. 14. Top$K$ semantic success rate (SR) of least-likely FGSM —■—, least-likely BIM —■— and P-FGSM —■— with Log transformation —●—, Gamma correction ($\gamma = 0.5$ —●—), nonlinear detail enhancement —●— and linear detail enhancement ($\alpha = 1$ —●—) of FilterFool using ResNet50 and AlexNet trained on ImageNet. ResNet18 results are similar to ResNet50 results.

TABLE VI
TOP1 AND TOP5 CLASSIFICATION ACCURACY (↓) WITH RESPECT TO THE CATEGORICAL LABELS. RESNET50, RESNET18 AND ALEXNET ARE EVALUATED ON THE ORIGINAL AND ADVERSARIAL IMAGES.

Images	ResNet50		ResNet18		AlexNet	
	Top1	Top5	Top1	Top5	Top1	Top5
Original	0.726	0.906	0.650	0.868	0.517	0.753
BIM [4]	0.082	0.482	0.091	0.388	0.094	0.431
LL-FGSM [4]	0.149	0.297	0.035	0.090	0.010	0.035
LL-BIM [4]	0.001	0.010	0.001	0.006	0.003	0.006
P-FGSM [10]	0.000	0.062	0.000	0.044	0.001	0.027
DeepFool [6]	0.115	0.693	0.109	0.618	0.140	0.641
SparseFool [8]	0.097	0.899	0.097	0.843	0.106	0.716
SemanticAdv [17]	0.156	0.760	0.117	0.705	0.057	0.452
ColorFool [13]	0.160	0.872	0.141	0.827	0.106	0.707
EdgeFool [14]	0.085	0.840	0.079	0.784	0.073	0.613
FilterFool-c (ND)	0.085	0.814	0.086	0.767	0.084	0.607
FilterFool (ND)	0.006	0.026	0.004	0.024	0.004	0.012

Table VI shows the Top1 and Top5 accuracy of ResNet50, ResNet18, AlexNet on the original and adversarial images. Most of the attacks achieve low Top1 accuracy, but those of untargeted attacks with the categorical adversarial loss show high Top5 accuracy. The targeted attacks with the least-likey label, LL-FGSM and LL-BIM, can avoid this performance decrease

TABLE VII

ROBUSTNESS (↑) OF IMAGES GENERATED WITH VARIOUS ATTACKS: BIM, LL-BIM, LL-FGSM, P-FGSM, DEEPFOOL (DF), SPARSEFOOL (SF), SEMANTICADV (SA), COLORFOOL (CF) AND VARIOUS VERSION OF FILTERFOOL (FF) – LINEAR WITH $\alpha = 1$ AND NONLINEAR DETAIL ENHANCEMENT (LD) AND (ND), LOG TRANSFORMATION (LT) AND GAMMA CORRECTION WITH $\gamma = 0.5$ (GC) – ON IMAGENET FOR RESNET50 (R50), RESNET18 (R18) AND ALEXNET (A).

Method	Model	Bit reduction							Median smoothing			JPEG compression			
		1-bit	2-bit	3-bit	4-bit	5-bit	6-bit	7-bit	2×2	3×3	5×5	q-25	q-50	q-75	q-100
BIM [4]	R50	0.933	0.622	0.527	0.687	0.809	0.859	0.870	0.509	0.465	0.462	0.381	0.402	0.504	0.837
	R18	0.951	0.702	0.697	0.832	0.911	0.935	0.945	0.663	0.591	0.553	0.466	0.533	0.686	0.936
	A	0.927	0.780	0.784	0.880	0.924	0.939	0.943	0.649	0.656	0.663	0.569	0.653	0.771	0.927
LL-BIM [4]	R50	0.939	0.648	0.613	0.952	0.999	1.000	1.000	0.568	0.468	0.477	0.370	0.375	0.453	0.997
	R18	0.952	0.760	0.798	0.991	1.000	1.000	1.000	0.687	0.579	0.564	0.459	0.473	0.667	1.000
	A	0.946	0.905	0.962	0.995	0.998	0.998	0.998	0.806	0.830	0.751	0.598	0.747	0.918	0.995
P-FGSM [10]	R50	0.938	0.636	0.601	0.972	1.000	1.000	1.000	0.559	0.443	0.462	0.368	0.342	0.429	0.998
	R18	0.953	0.731	0.818	0.995	1.000	1.000	1.000	0.706	0.563	0.529	0.432	0.434	0.694	1.000
	A	0.942	0.873	0.973	0.999	1.000	1.000	1.000	0.814	0.801	0.722	0.573	0.771	0.960	1.000
LL-FGSM [4]	R50	0.947	0.760	0.790	0.830	0.846	0.850	0.850	0.719	0.688	0.650	0.573	0.682	0.758	0.839
	R18	0.965	0.897	0.930	0.961	0.965	0.965	0.966	0.864	0.846	0.781	0.754	0.867	0.927	0.961
	A	0.960	0.959	0.982	0.988	0.992	0.992	0.992	0.943	0.943	0.896	0.912	0.959	0.979	0.989
DF [6]	R50	0.937	0.625	0.371	0.329	0.345	0.357	0.362	0.304	0.342	0.412	0.348	0.377	0.362	0.364
	R18	0.953	0.683	0.465	0.455	0.481	0.495	0.499	0.388	0.399	0.481	0.426	0.466	0.488	0.499
	A	0.922	0.733	0.553	0.584	0.605	0.614	0.614	0.431	0.506	0.602	0.472	0.527	0.600	0.600
SF [8]	R50	0.941	0.619	0.386	0.346	0.422	0.548	0.714	0.180	0.226	0.372	0.363	0.312	0.293	0.328
	R18	0.956	0.690	0.455	0.403	0.497	0.638	0.786	0.210	0.280	0.432	0.423	0.358	0.333	0.397
	A	0.929	0.783	0.650	0.719	0.831	0.911	0.959	0.382	0.462	0.605	0.502	0.477	0.521	0.698
SA [17]	R50	0.958	0.809	0.701	0.716	0.768	0.830	0.866	0.652	0.697	0.783	0.676	0.662	0.667	0.737
	R18	0.975	0.873	0.783	0.787	0.835	0.875	0.909	0.715	0.754	0.813	0.730	0.712	0.712	0.782
	A	0.971	0.923	0.903	0.919	0.949	0.977	0.987	0.869	0.888	0.922	0.840	0.846	0.869	0.914
CF [13]	R50	0.946	0.785	0.654	0.618	0.675	0.763	0.831	0.546	0.622	0.740	0.664	0.607	0.606	0.614
	R18	0.967	0.824	0.686	0.642	0.683	0.768	0.843	0.609	0.667	0.771	0.684	0.637	0.613	0.617
	A	0.951	0.861	0.761	0.743	0.808	0.888	0.929	0.684	0.740	0.828	0.732	0.693	0.703	0.735
EF [14]	R50	0.950	0.610	0.560	0.640	0.806	0.917	0.966	0.397	0.433	0.490	0.516	0.512	0.519	0.691
	R18	0.963	0.677	0.633	0.723	0.861	0.942	0.975	0.461	0.483	0.545	0.581	0.563	0.586	0.756
	A	0.929	0.819	0.820	0.904	0.959	0.981	0.990	0.585	0.598	0.628	0.714	0.749	0.795	0.888
FF (ND)	R50	0.953	0.642	0.667	0.880	0.986	0.999	1.000	0.474	0.485	0.520	0.507	0.541	0.621	0.990
	R18	0.967	0.735	0.802	0.954	0.996	0.999	0.999	0.567	0.560	0.591	0.592	0.631	0.774	0.998
	A	0.942	0.942	0.991	0.999	1.000	1.000	1.000	0.803	0.815	0.753	0.846	0.946	0.989	1.000
FF (LD)	R50	0.925	0.569	0.515	0.782	0.960	0.998	1.000	0.539	0.537	0.567	0.396	0.413	0.470	0.977
	R18	0.949	0.657	0.668	0.890	0.987	0.999	1.000	0.633	0.637	0.654	0.472	0.536	0.663	0.994
	A	0.935	0.881	0.973	0.998	1.000	1.000	1.000	0.855	0.880	0.826	0.695	0.799	0.912	1.000
FF (LT)	R50	0.926	0.592	0.413	0.616	0.902	0.992	0.999	0.437	0.428	0.501	0.357	0.330	0.363	0.965
	R18	0.951	0.701	0.562	0.802	0.971	0.998	1.000	0.567	0.538	0.591	0.433	0.450	0.538	0.989
	A	0.937	0.879	0.952	0.994	1.000	0.999	0.999	0.885	0.916	0.866	0.635	0.746	0.884	0.999
FF (GC)	R50	0.955	0.658	0.454	0.561	0.840	0.986	0.997	0.514	0.497	0.558	0.457	0.418	0.436	0.960
	R18	0.968	0.728	0.579	0.729	0.946	0.995	0.998	0.656	0.637	0.651	0.516	0.514	0.572	0.985
	A	0.969	0.887	0.928	0.994	0.999	1.000	1.000	0.902	0.915	0.875	0.728	0.794	0.898	0.999

as the least-likely class selected for the perturbations can be semantically different from the original label. FilterFool, even with the untargeted attack, can still mislead Top5 classes.

Table VII reports the robustness of attacks to bit reduction, median filtering and JPEG compression. Content-based attacks are more robust than norm-bounded attacks, as high-frequency norm-bounded perturbations can be easily removed by these input-based transformations. For example, the most effective parameter of bit reduction, median smoothing and JPEG compression drop the 89% success rate of SemanticAdv against ResNet50 to 70.1%, 65.2% and 66.2%, respectively. The reason is that SemanticAdv, similarly to other content-based attacks such as ColorFool, generates large, low-frequency perturbations as opposed to the high-frequency perturbations of BIM, in which the categorical success rates drop to 52.7%, 50.9% and 38.1%, respectively. Some variants of the FilterFool adversarial perturbations even improve the robustness of existing content-based attacks. Reducing the number of bits from 8 to 3 decreases the categorical success rate of adversarial images. However, the categorical success rate increases again for 2 bits and 1 bit because of the quality of the resulting images.

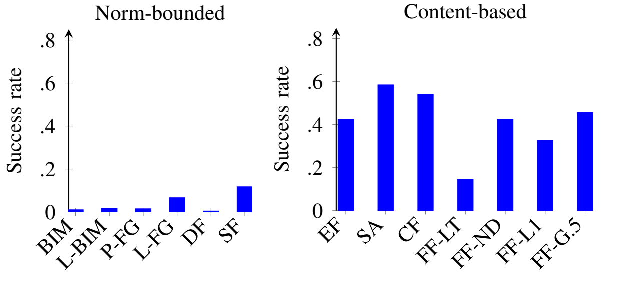

Fig. 15. Robustness of norm-bounded attacks – BIM, least-likely BIM (L-BIM), least-likely FGSM (L-FG), Private FGSM (P-FG), DeepFool (DF), SparseFool (SF) – and content-based attacks – EdgeFool (EF), SemanticAdv (SA), ColorFool (CF), FilterFool with Log Transformation (FF-LT), Nonlinear detail enhancement (FF-ND), $\alpha = 1$ Linear Detail Enhancement (FF-LD1), $\gamma = .5$ Gamma correction (FF-GC.5) – whose adversarial images were generated on ResNet50 against ResNet50 adversarially re-trained with BIM images.

Finally, Figure 15 shows the categorical success rate of adversarial images against *adversarial training*, when each training mini-batch is augmented with adversarial examples to improve the robustness of the classifier. We use the adversarially re-trained ResNet50 with BIM images [43], which reduces the success rate of all adversarial attacks. The success of content-based adversarial images is higher than that of norm-bounded ones as content-based perturbations generally have larger magnitudes. For example, the success rate of ColorFool, SemanticAdv and FilterFool (Gamma Correction) is above 50%. However, the success rate of norm-bounded adversarial perturbations generated by DeepFool and FGSM variants substantially decreases to below 20%.

VII. Conclusion

We proposed FilterFool, an adversarial framework that crafts adversarial perturbations based on the content of an image and on the pre-defined semantics of its label. FilterFool is flexible and can incorporate different image filters to generate various types of adversarial perturbations. These perturbations are larger than those produced by norm-bounded methods, thereby improving the transferability of the attacks to unseen classifiers and their robustness against defences. While in this work we considered the grouping of the labels according to the semantic relationship defined by WordNet [24], the framework is general and other groupings can be considered. As future work, we will extend FilterFool to cope with multi-label images and further image filters.

Acknowledgments

We thank Dmitrii Mukhutdinov and Ashish Alex for their help in developing the online subjective evaluation tool. We also thank the Alan Turing Institute (EP/N510129/1), which is funded by the EPSRC, for its support through the project PRIMULA.

References

[1] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Prague, Czech Republic, September 2013.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Banff, Canada, April 2014.

[3] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, USA, June 2015.

[4] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proceedings of the International Conference on Learning Representations (ICLR) workshop track*, Toulon, France, April 2017.

[5] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, California, USA, June 2019.

[6] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, Nevada, USA, June 2016.

[7] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of the Symposium on Security and Privacy (S&P)*, San Jose, California, USA, May 2017.

[8] A. Modas, S. Moosavi-Dezfooli, and P. Frossard, "SparseFool: a few pixels make a big difference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, California, USA, June 2019.

[9] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*, Saarbrücken, Germany, March 2016.

[10] C.Y. Li, A.S. Shamsabadi, R. Sanchez-Matilla, R. Mazzon, and A. Cavallaro, "Scene privacy protection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, May 2019.

[11] M. Sharif, L. Bauer, and M.K. Reiter, "On the suitability of lp-norms for creating and preventing adversarial examples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1605–1613.

[12] A. Sen, X. Zhu, L. Marshall, and R. Nowak, "Should adversarial attacks use pixel p-norm?," *arXiv preprint arXiv:1906.02439*, 2019.

[13] A.S. Shamsabadi, R. Sanchez-Matilla, and A. Cavallaro, "ColorFool: Semantic adversarial colorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, Washington, USA, 2020.

[14] A.S. Shamsabadi, C. Oh, and A. Cavallaro, "EdgeFool: An adversarial image enhancement filter," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, May 2019.

[15] C. Laidlaw and S. Feizi, "Functional adversarial attacks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2019.

[16] Z. Zhao, Z. Liu, and M. Larson, "Towards large yet imperceptible adversarial image perturbations with perceptual color distance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, Washington, USA, 2020.

[17] H. Hosseini and R. Poovendran, "Semantic adversarial examples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) workshop track*, Salt Lake City, Utah, USA, June 2018.

[18] A. Bhattad, M.J. Chong, K. Liang, B. Li, and D.A. Forsyth, "Unrestricted adversarial examples via semantic manipulation," in *International Conference on Learning Representations (ICLR)*, 2019.

[19] Z. Zhao, Z. Liu, and M. Larson, "Adversarial color enhancement: Generating unrestricted adversarial images by optimizing a color filter," in *Proceedings of the Brisith Machine Vision Conference (BMVC)*, 2020.

[20] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami Beach, Florida, USA, June 2009.

[21] 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 (CVPR)*, Las Vegas, Nevada, USA, June 2016.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, Lake Tahoe, Nevada, USA, December 2012.

[23] K.R. Mopuri, V. Shaj, and R.V. Babu, "Adversarial fooling beyond flipping the label," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) workshop track*, Seattle, Washington, USA, June 2020.

[24] D. Tsipras, S. Santurkar, L. Engstrom, A. Ilyas, and A. Madry, "From ImageNet to image classification: Contextualizing progress on benchmarks," in *ArXiv preprint arXiv:2005.11295*, 2020.

[25] "Pixel privacy task, mediaeval 2018," http://www.multimediaeval.org/mediaeval2018/, [Last accessed October 2019].

[26] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, Nevada, USA, June 2016.

[27] H. Kwon, Y. Kim, H. Yoon, and D. Choi, "Classification score approach for detecting adversarial example in deep neural network," *Multimedia Tools and Applications*, pp. 1–22, 2020.

[28] R. Sanchez-Matilla, C.Y. Li, A.S. Shamsabadi, R. Mazzon, and A. Cavallaro, "Exploiting vulnerabilities of deep neural networks for privacy protection," *IEEE Transactions on Multimedia*, vol. 22, no. 7, pp. 1862–1873, 2020.

[29] J. Rony, L.G. Hafemann, L.S. Oliveira, I.B. Ayed, R. Sabourin, and E. Granger, “Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4322–4330.

[30] Z. Yao, A. Gholami, P. Xu, K. Keutzer, and M.W. Mahoney, “Trust region based adversarial attack on neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11350–11359.

[31] Q. Fan, J. Yang, D. Wipf, B. Chen, and X. Tong, “Image smoothing via unsupervised learning,” *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 259:1–259:14, 2018.

[32] Q. Chen, J. Xu, and V. Koltun, “Fast image processing with fully-convolutional networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2497–2506.

[33] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, “Loss functions for image restoration with neural networks,” *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.

[34] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing (TIP)*, vol. 13, no. 4, pp. 600–612, 2004.

[35] L. Xu, C. Lu, Y. Xu, and J. Jia, “Image smoothing via l0 gradient minimization,” *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6, pp. 174:1–174:12, 2011.

[36] R.C. Gonzalez and R.E. Woods, *Digital Image Processing (3rd Edition)*, Prentice-Hall, Inc., USA, 2006.

[37] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.

[38] C. Oh, A. Xompero, and A. Cavallaro, “Visual adversarial attacks and defenses,” in *Advanced Methods and Deep Learning in Computer Vision*, E.R. Davies, O. Camps, and M. Turk, Eds. Elsevier/Academic Press, 2021.

[39] Z. Wu and M. Palmer, “Verb semantics and lexical selection,” in *Proceedings of the Annual Meeting on Association for Computational Linguistics*, Las Cruces, New Mexico, USA, June 1994.

[40] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” in *Proceedings of the Network and Distributed Systems Security Symposium (NDSS)*, San Diego, California, USA, February 2018.

[41] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in *International Conference on Learning Representations*, 2018.

[42] H. Wu, S. Zheng, J. Zhang, and K. Huang, “Fast end-to-end trainable guided filter,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, Utah, USA, June 2018.

[43] L. Engstrom, A. Ilyas, H. Salman, S. Santurkar, and D. Tsipras, “Robustness (python library),” 2019.