# EDGEFOOL: AN ADVERSARIAL IMAGE ENHANCEMENT FILTER

*Ali Shahin Shamsabadi, Changjae Oh, Andrea Cavallaro*

Centre for Intelligent Sensing, Queen Mary University of London, UK

## ABSTRACT

Adversarial examples are intentionally perturbed images that mislead classifiers. These images can, however, be easily detected using denoising algorithms, when high-frequency spatial perturbations are used, or can be noticed by humans, when perturbations are large. In this paper, we propose EdgeFool, an adversarial image enhancement filter that learns structure-aware adversarial perturbations. EdgeFool generates adversarial images with perturbations that enhance image details via training a fully convolutional neural network end-to-end with a multi-task loss function. This loss function accounts for both image detail enhancement and class misleading objectives. We evaluate EdgeFool on three classifiers (ResNet-50, ResNet-18 and AlexNet) using two datasets (ImageNet and Private-Places365) and compare it with six adversarial methods (DeepFool, SparseFool, Carlini-Wagner, SemanticAdv, Non-targeted and Private Fast Gradient Sign Methods). Code is available at `https://github.com/smartcameras/EdgeFool.git`.

***Index Terms*—** Adversarial images, detail enhancement

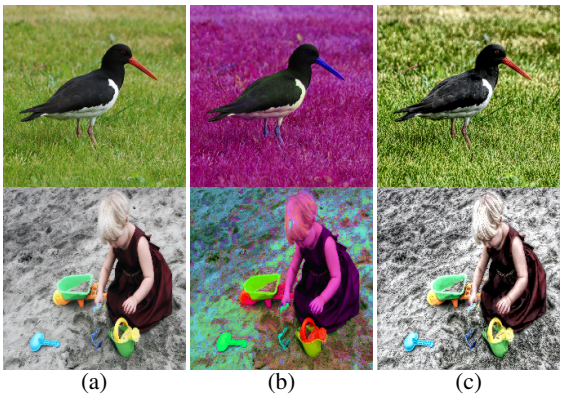
## 1. INTRODUCTION

An adversarial image is generated by perturbing the pixel values of an original image to induce a Deep Neural Network (DNNs) to fail in its classification task. However, most adversarial images are *detectable* [1, 2, 3] by defence mechanisms that use denoising algorithms, such as median filtering or bit-depth reduction [4], or, when large distortions are generated, the adversarial images are *noticeable* to humans [5, 6]. Moreover, most adversarial methods operate under a white-box attack assumption and need to access the parameters of a specific classifier. Therefore, the resulting perturbations do not generally *transfer* across classifiers [7].

Adversarial methods can be classified as constrained or unconstrained, depending on whether they generate bounded perturbations. *Constrained* adversarial methods [1, 2, 3, 6, 8] iteratively minimise an $\ell_p$ norm of the perturbations. Non-targeted Fast Gradient Sign Method (N-FGSM) [1] and Private Fast Gradient Sign Method (P-FGSM) [2] generate an adversarial perturbation whose $\ell_\infty$ norm is constrained. DeepFool [8] and Carlini-Wagner (CW) [3] constrain the $\ell_2$ difference between the original and adversarial image, while SparseFool [6] generates sparse adversarial perturbations based on the $\ell_1$ norm. However, these adversarial images are easily detectable [4] and are not transferable [7].

More recent adversarial methods introduce *unconstrained* perturbations, for example by randomly shifting hue and saturation values [5], by transferring new textures to input images [9] or by colourisation [9, 10]. These adversarial methods are more transferable and less detectable than constrained adversarial methods, but unconstrained adversarial images may be severely degraded by unnatural textures or colours (see Figure 1(b)).

Andrea Cavallaro wishes to thank the Alan Turing Institute (EP/N510129/1), which is funded by the EPSRC, for its support through the project PRIMULA.



(a) (b) (c)

**Fig. 1**. Adversarial examples produced with *unconstrained* perturbations. (a) Original images. Adversarial images generated with (b) SemanticAdv [5] and (c) EdgeFool, the proposed method. Note the unnatural colours produced by SemanticAdv, and the enhanced details and natural colours produced by EdgeFool.

In this paper, we propose employing an image enhancement filter to generate adversarial images, with the objective of reducing detectability and noticeability, and improving transferability. We achieve these contrasting objectives with a structure-aware adversarial perturbation that enhances image details. The proposed adversarial image enhancement filter, EdgeFool, learns the adversarial perturbation by training a Fully Convolutional Neural Network (FCNN) end-to-end with a multi-task loss, which includes detail enhancement and class misleading objectives. Using image smoothing [11], the network learns to decompose the image, isolating the details from smoother image structures. Then, the multi-task loss function guides the network to enhance details in a way that causes an incorrect classification. The block diagram of the proposed method is shown in Figure 2.

We validate EdgeFool with object and scene classifiers on ImageNet [12] and Private-Places365 [13] using deep residual neural networks with 50 layers (ResNet-50) and 18 layers (ResNet-18) [14], and AlexNet [15]. Experiments show that EdgeFool generates detail-enhanced adversarial images that are more transferable and less detectable than constrained adversarial methods.

## 2. PROPOSED METHOD

### 2.1. Preliminaries

Adversarial methods aim to mislead a $D$-class classifier, $C(\cdot)$, by modifying the intensity values of an RGB image, $\mathbf{I}$, and generating an adversarial image, $\dot{\mathbf{I}}$, whose classification result, $C : \dot{\mathbf{I}} \rightarrow \dot{y}$,

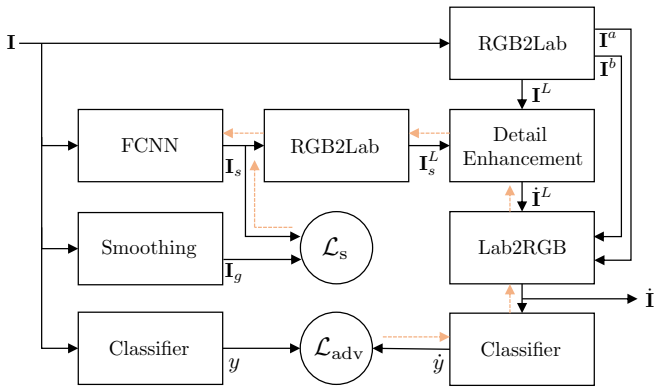



**Fig. 2**. Block diagram of EdgeFool, which generates a detail-enhanced adversarial image, $\dot{\mathbf{I}}$, by training a Fully Convolutional Neural Network (FCNN) using a smoothing loss, $\mathcal{L}_s$, and an adversarial loss, $\mathcal{L}_{\text{adv}}$ (— forward pass; - - backward pass). The FCNN learns a smooth image, $\mathbf{I}_s$, based on the desired output, $\mathbf{I}_g$, of the image smoothing filter and the objective of predicting a class of $\dot{\mathbf{I}}$, $\dot{y}$, that is different from $y$, the class of the input image $\mathbf{I}$.

differs from that of the original image:

$$\dot{y} = C(\dot{\mathbf{I}}) \neq y = C(\mathbf{I}), \tag{1}$$

where $y$ is the predicted class of the original image.

For each of the $D$ classes, the classifier $C(\cdot)$ predicts logit scores $\mathbf{z} = (z_1, ..., z_i, ..., z_D)$ for $\mathbf{I}$, where $z_i \in (-\infty, +\infty)$ is the logit score for class $i$. The logit scores are then normalised and the softmax is used to predict the probability $p_i \in [0, 1]$ of each class $i$:

$$p_i = \frac{e^{z_i}}{\Sigma_{d=1}^{D} e^{z_d}}, \tag{2}$$

where $\mathbf{p} = (p_1, ..., p_i, ..., p_D)$ represents the probability of all the classes.

The predicted class, $y$, is decided by a winner-take-all approach:

$$y = \arg\max_{d=1,...,D} p_d. \tag{3}$$

### 2.2. Adversarial image enhancement

We propose an adversarial image enhancement filter, EdgeFool, whose perturbations enhance details, preserve structure and maintain the original colours. EdgeFool decomposes $\mathbf{I}$ into its structural component, $\mathbf{I}_s$ (Figure 3(a)), containing smooth regions, and a residual component, $\mathbf{I}_d$ (Figure 3(b)), corresponding to image details:

$$\mathbf{I} = \mathbf{I}_s + \mathbf{I}_d. \tag{4}$$

We learn this image decomposition by training a FCNN [16] with a multi-task loss, $\mathcal{L}$, which includes an image smoothing loss, $\mathcal{L}_{\text{s}}$, and the adversarial loss, $\mathcal{L}_{\text{adv}}$:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{s}}(\mathbf{I}_s, \mathbf{I}_g) + \mathcal{L}_{\text{adv}}(\dot{\mathbf{I}}, \mathbf{I}), \tag{5}$$

where the hyper-parameter $\alpha$ controls the relative importance of $\mathcal{L}_{\text{s}}$ and $\mathcal{L}_{\text{adv}}$. The *smoothing loss*, $\mathcal{L}_{\text{s}}$, measures the difference between the learned structure and an image representation that excludes high spatial frequency components, $\mathbf{I}_g$, output by an $\ell_0$ structure-preserving smoothing filter [11]:

$$\mathcal{L}_{\text{s}}(\mathbf{I}_s, \mathbf{I}_g) = \|\mathbf{I}_s - \mathbf{I}_g\|^2. \tag{6}$$

We then avoid changing the colours of the input image and enhance the image details of the $L$ image channel only, after conversion to the $Lab$ colour space. The detail-enhanced adversarial image, $\dot{\mathbf{I}}$ (Figure 3(d)), is finally generated by combining the channel with enhanced details, $\dot{\mathbf{I}}^L$, with the $a$ and $b$ colour channels of the original image, and then transforming the resulting image back to the RGB space:

$$\dot{\mathbf{I}} = \begin{cases} \dot{\mathbf{I}}^L = \left(f\left(\frac{\mathbf{I}_s^L - v_1}{100}, v_2\right) \cdot 100 + v_1\right) + f\left(\frac{\mathbf{I}_d^L}{100}, v_3\right) \cdot 100, \\ \dot{\mathbf{I}}^a = \mathbf{I}^a, \\ \dot{\mathbf{I}}^b = \mathbf{I}^b, \end{cases} \tag{7}$$

where $f(a, b) = \left(1 + e^{-ab}\right)^{-1} - 0.5$ [17], $v_1$ is a constant value that adjusts the midpoint of the sigmoid curve, $v_2$ and $v_3$ control the slope of the sigmoid curves in $\mathbf{I}_s^L$ and $\mathbf{I}_d^L$, respectively. The input to the sigmoid function is normalised by 100, the range of the $L$ channel.

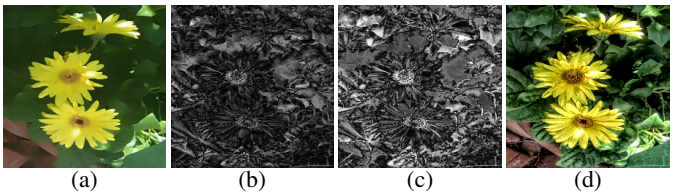
The *adversarial loss*, $\mathcal{L}_{\text{adv}}$, quantifies the difference between the score of the adversarial image $\dot{\mathbf{I}}$ belonging to the same class as $\mathbf{I}$, $\dot{z}_y$, and the maximum score among other classes [3]:

$$\mathcal{L}_{\text{adv}}(\dot{\mathbf{I}}, \mathbf{I}) = \dot{z}_y - \max\{\dot{z}_i : i = 1, ..., D; i \neq y\}, \tag{8}$$

where $\dot{z}_i \in \dot{\mathbf{z}}$ is the logit score of $\dot{\mathbf{I}}$ for class $i$.

We iteratively train the whole pipeline end-to-end by minimising $\mathcal{L}$ in Eq. 5 using the Adam optimiser [18], until the generated $\dot{\mathbf{I}}$ misleads the target classifier and $\mathcal{L}_{\text{s}} < \tau$. We empirically set the threshold $\tau = 5\text{e}^{-4}$ to make $\mathbf{I}_s$ close to $\mathbf{I}_g$, enabling the separation of the details for enhancement.

Figure 4 shows an example of a perturbation generated by EdgeFool and compares it with the perturbations generated by state-of-the-art adversarial methods applied to the same image.



(a) (b) (c) (d)

**Fig. 3**. Intermediate results of the EdgeFool adversarial image generation process. (a) Image structure, $\mathbf{I}_s$. (b) Image details, $\mathbf{I}_d^L$; (c) Enhanced details, $\dot{\mathbf{I}}^L - \mathbf{I}_s^L$. (d) Adversarial image, $\dot{\mathbf{I}}$. Note that (b) and (c) are scaled for visualisation.

## 3. VALIDATION

We compare the proposed method, EdgeFool, with six state-of-the-art adversarial methods: N-FGSM [1], P-FGSM [2], DeepFool [8], SparseFool [6], CW [3] and SemanticAdv [5]. We apply these adversarial methods to three state-of-the-art classifiers: ResNet with 50 layers (ResNet-50) and 18 layers (ResNet-18) [14], and AlexNet [15] on Private-Places365 [13, 19] and ImageNet [12] as scene and object datasets. The Private-Places365 validation set includes 3K images of 60 privacy-sensitive scene classes. The ImageNet validation set includes 50 images of different size for 1K

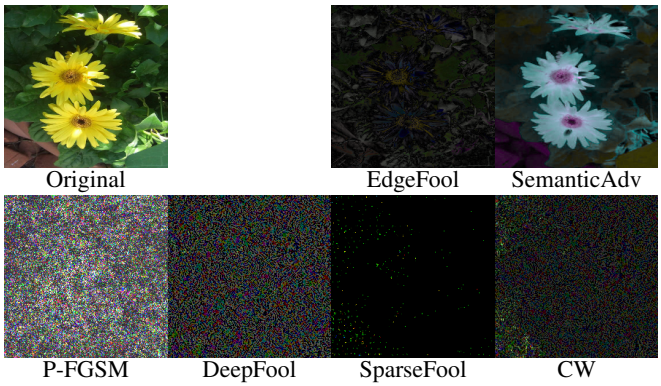



**Fig. 4**. Comparison of adversarial perturbations generated by different methods, namely EdgeFool, SemanticAdv [5], P-FGSM [2], DeepFool [8], SparseFool [6] and CW [3]. Note that the constrained perturbations (second row) are scaled for visualisation.

classes. In order to perform the evaluation on all 1K classes of ImageNet and reduce the computational cost, we consider 3K images by randomly selecting 3 images per class.

We used the PyTorch implementations provided by the authors for P-FGSM[1], DeepFool[2] and SparseFool[3] and we implemented N-FGSM. We also implemented SemanticAdv in PyTorch based on their Keras version[4], while we used Foolbox[5] for CW. We instantiate FCNN from the architecture implemented in [16]. The input size is $224 \times 224 \times 3$. The FCNN architecture consists of 7 convolution layers with 24 intermediate feature maps and kernels of size $3 \times 3$. The last convolution layer applies a $1 \times 1$ convolution that generates $\mathbf{I}_s$. The dilation factors of each convolution layer are set to 1, 2, 4, 8, 16, 32, 1, and 1, respectively. Leaky Rectified Linear Unit (L-ReLU) is applied after padding and normalising each intermediate convolutional layer, except the last convolution layer. For the detail enhancement, we follow the parameters in [17], $v_1 = 56$, $v_2 = 1$, and $v_3 = 15$, which magnify the details by providing a steeper slope to the sigmoid curve for $\mathbf{I}_d^L$. Also, we choose $\alpha = 10$ for the loss function.

As performance measures we consider the misleading rate, transferability and detectability. The *misleading rate* is the ratio between the number of adversarial images that successfully mislead a classifier and the total number of images. *Transferability* is the misleading rate of adversarial images on a classifier that is different from the one that was used to generate the perturbations. Finally, *detectability* is the ratio between the number of detected adversarial images and the total number of images. To detect an image as adversarial we use the so-called feature squeezing framework [4], which consists of applying median filtering and bit-depth reduction. An image is considered adversarial if the $\ell_1$ difference between the $D$-dimentional class predictions of the image, $\dot{\mathbf{p}}$, and its *squeezed* version exceed the threshold, which is defined based on the $\ell_1$ difference of the original images and their corresponding squeezed images by accepting a $5\%$ false positive rate.

Figure 5 shows the misleading rate and transferability of N-FGSM, P-FGSM, DeepFool, SparseFool, CW, SemanticAdv, and EdgeFool against ResNet-50, ResNet-18, and Alexnet. The top circle and bottom circles indicate the misleading rate against the classifier that adversarial images are generated for and transferability to other classifiers, respectively. While most adversarial methods successfully mislead a particular classifier, EdgeFool and SemanticAdv have higher transferability to other classifiers. For example, CW generates adversarial images that mislead ResNet-18 above $99\%$, while only $2\%$ and $4\%$ are transferable to AlexNet and ResNet-50. EdgeFool adversarial images generated for ResNet-50 have above $99\%$ success rate on ResNet-50 and the SemanticAdv misleading rate is $93\%$; however, EdgeFool and SemanticAdv are $52\%$ and $71\%$ transferable to AlexNet. The higher transferability of SemanticAdv is due to the large colour distortions of the generated adversarial images.

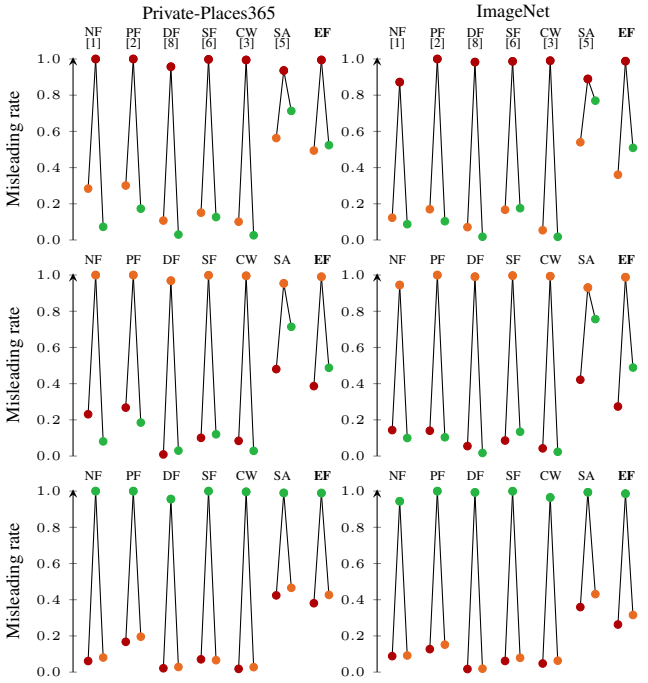



**Fig. 5**. Misleading rate and transferability of N-FGSM (NF), P-FGSM (PF), DeepFool (DF), SparseFool (SF), CW, SemanticAdv (SA) and EdgeFool (EF) for ResNet-50 (●), ResNet-18 (●) and AlexNet (●) on Private-Places365 and ImageNet. EdgeFool is more transferable than other adversarial methods, except SemanticAdv that however severely distorts the colours (see Figure 6).

Table 1 reports the detectability rate of adversarial images for ResNet-50, ResNet-18 and AlexNet on ImageNet and Private-Places365. Median filtering and bit-depth reduction help detect the small but high spatial-frequency perturbations of constrained adversarial methods (especially P-FGSM and CW), while SemanticAdv generates the most undetectable adversarial images to median filtering, as it perturbs hue and saturation of all pixels by the same randomly-generated amount. EdgeFool is less detectable than constrained adversarial methods, but more detectable than SemanticAdv, as some of the enhanced (adversarial) details are smoothed out by the median filtering. Although EdgeFool and SemanticAdv are comparable in terms of the misleading rate and transferability, SemanticAdv perturbs each pixel more than EdgeFool and severely distorts the adversarial images as the colour changes are random (see Figure 6 and Figure 7). SparseFool perturbs only a few pixels with large and prominent changes, whereas the adversarial images of EdgeFool contain adversarial perturbations, which enhance details.

[1] https://github.com/smartcameras/P-FGSM
[2] https://github.com/LTS4/DeepFool
[3] https://github.com/LTS4/SparseFool
[4] https://github.com/HosseinHosseini/Semantic-Adversarial-Examples
[5] https://foolbox.readthedocs.io/en/stable/

**Table 1**. Detectability rate (↓) of N-FGSM, P-FGSM, DeepFool, SparseFool, CW, SemanticAdv, and EdgeFool on Private (P)-Places365 and ImageNet for ResNet-50 (R-50), ResNet-18 (R-18) and AlexNet (A) classifiers. For the Quantization (Q), we reduce the number of bits from 8 to 4-bit (4b) to 7-bit (7b). Smoothing (S) is a median filter with $2 \times 2$ (2m) and $3 \times 3$ (3m) on the adversarial images. Although SemanticAdv is less detectable than EdgeFool, its adversarial images suffer from unnatural colour changes (see Figure 6 and Figure 7).

| | | | N-FGSM [1] | | | P-FGSM [2] | | | DeepFool [8] | | | SparseFool [6] | | | CW [3] | | | SemanticAdv [5] | | | EdgeFool | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | R-50 | R-18 | A | R-50 | R-18 | A | R-50 | R-18 | A | R-50 | R-18 | A | R-50 | R-18 | A | R-50 | R-18 | A | R-50 | R-18 | A |
| **P-Places365** | **Q** | **4b** | .08 | .07 | .06 | .05 | .02 | .01 | .26 | .23 | .14 | .06 | .05 | .06 | .31 | .24 | .14 | .06 | .06 | .06 | .01 | .00 | .00 |
| | | **5b** | .05 | .06 | .06 | .01 | .01 | .01 | .27 | .23 | .13 | .07 | .07 | .06 | .33 | .28 | .14 | .09 | .09 | .07 | .01 | .00 | .00 |
| | | **6b** | .04 | .04 | .06 | .01 | .01 | .01 | .23 | .18 | .15 | .07 | .06 | .06 | .28 | .23 | .13 | .11 | .07 | .07 | .01 | .00 | .00 |
| | | **7b** | .06 | .05 | .08 | .01 | .01 | .01 | .17 | .15 | .15 | .08 | .06 | .07 | .19 | .15 | .11 | .12 | .09 | .07 | .02 | .02 | .03 |
| | **S** | **2m** | .39 | .36 | .43 | .86 | .82 | .93 | .49 | .49 | .40 | .35 | .34 | .22 | .48 | .45 | .36 | .14 | .11 | .08 | .28 | .27 | .21 |
| | | **3m** | .39 | .41 | .24 | .96 | .94 | .85 | .35 | .41 | .19 | .17 | .22 | .10 | .33 | .38 | .15 | .12 | .13 | .05 | .19 | .25 | .14 |
| **ImageNet** | **Q** | **4b** | .24 | .18 | .11 | .45 | .24 | .02 | .48 | .37 | .22 | .13 | .08 | .07 | .99 | .96 | .68 | .11 | .08 | .08 | .04 | .01 | .01 |
| | | **5b** | .18 | .16 | .09 | .01 | .01 | .01 | .32 | .25 | .20 | .14 | .11 | .08 | .98 | .97 | .71 | .12 | .11 | .09 | .02 | .01 | .01 |
| | | **6b** | .12 | .11 | .07 | .01 | .01 | .01 | .20 | .15 | .19 | .14 | .09 | .09 | .97 | .94 | .67 | .14 | .11 | .09 | .03 | .01 | .01 |
| | | **7b** | .10 | .09 | .09 | .01 | .01 | .01 | .11 | .09 | .17 | .12 | .11 | .10 | .88 | .85 | .56 | .14 | .12 | .10 | .03 | .03 | .06 |
| | **S** | **2m** | .40 | .35 | .29 | .95 | .91 | .93 | .62 | .57 | .48 | .42 | .38 | .24 | .99 | .98 | .85 | .14 | .09 | .07 | .32 | .27 | .21 |
| | | **3m** | .31 | .29 | .17 | .98 | .97 | .93 | .55 | .53 | .24 | .09 | .10 | .11 | .92 | .88 | .51 | .09 | .07 | .05 | .13 | .13 | .12 |

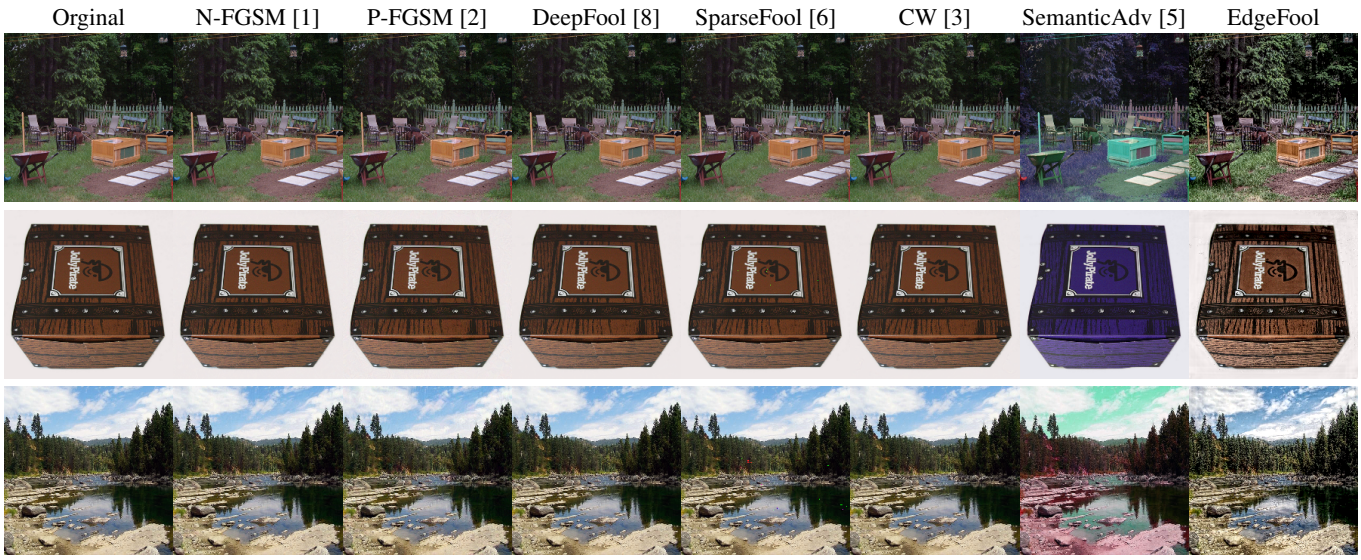



**Fig. 6**. Adversarial images generated by EdgeFool and other state-of-the-art adversarial methods for ResNet-50 in ImageNet and Private-Places365. N-FGSM, P-FGSM, DeepFool and CW adversarial images are similar to the original images to human eyes, while they are detectable by defence frameworks and not transferable (see Table 1 and Figure 5). EdgeFool takes advantage of image processing filtering in order to generate structure-aware adversarial perturbations that enhance the details of the image.

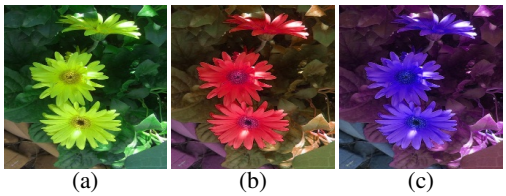



**Fig. 7**. Examples of adversarial images generated by SemanticAdv [5] against (a) AlexNet, (b) ResNet18 and (c) ResNet50 for the original image in Figure 3(a).

## 4. CONCLUSION

We presented EdgeFool, an adversarial image enhancement filter that trains a multi-task fully convolutional neural network to generate adversarial images whose details are enhanced. We compared EdgeFool with six state-of-the-art adversarial methods on ResNet-50, ResNet-18 and AlexNet classifiers using ImageNet and Private-Places365 datasets and showed that EdgeFool satisfies misleading, transferability and undetectability objectives. As future work, we will validate EdgeFool on other classifiers and with other datasets, as well as perform a formal subjective evaluation of the adversarial image quality.

## 5. REFERENCES


[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] C. Xie, Z. Zhang, J. Wang, Y. Zhou, Z. Ren, and A. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, California, USA, June 2019.

[8] 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.

[9] A. Bhattad, M. J. Chong, K. Liang, B. Li, and D. A Forsyth, "Big but imperceptible adversarial perturbations via semantic manipulation," *arXiv preprint arXiv:1904.06347*, 2019.

[10] Z. Zhu, Y. Lu, and C. Chiang, "Generating adversarial examples by makeup attacks on face recognition," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, September 2019.

[11] 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.

[12] 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.

[13] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 40, no. 6, pp. 1452–1464, June 2018.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, California, USA, December 2015.

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