A Real-world reflections in natural images

Reflections exist in natural images can also deteriorate classification performance. Fig. 7 shows three such examples in the *ImageNet-a* [23] dataset, where all the three images were misclassified by a DNN classifier. For instance, the black bear in the first image was misclassified to be rock chair with 82% confidence.

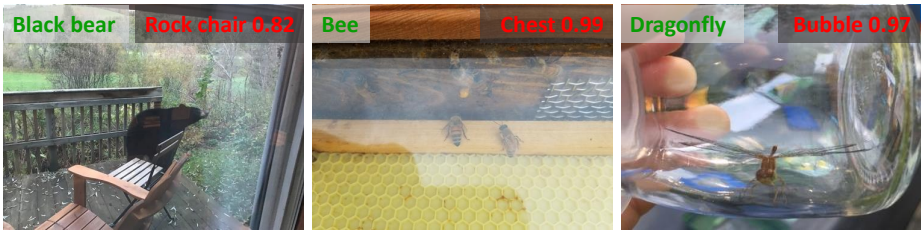

Fig. 7. Real-world reflections (from *ImageNet-a* [23]) influence the performance of a DNN classifier. Labels in green and red colors are ground-truth and predicted labels, respectively.

B Adversarial reflection image selection algorithm

This section describes the algorithm of adversarial reflection image selection, as shown in algorithm 1.

Algorithm 1: Adversarial reflection image selection

Input: Training set D_{train}, a candidate reflection set R_{cand}, validation set D_{val}, a DNN model f, target class y_{adv}, number of injected samples m, number of selection iterations T

Output: Adversarial reflection set R_{adv}

$i \leftarrow 0$; $W \leftarrow \{1\}_{\text{size}(R_{cand})}$ ▷ a list of 1 with the size of R_{cand}

$R_{adv} \leftarrow random\text{-}m(R_{cand})$ ▷ random selection

while $i \leq T$ **do**

  $D_{inject} \leftarrow$ randomly select m samples from D_{train}

  $D^{adv}_{train} \leftarrow$ inject R_{adv} into D_{inject} using Eqn. (1)

  $f_{adv}(\mathbf{x}, \boldsymbol{\theta}) \leftarrow$ train model on D^{adv}_{train}

  $W_i \leftarrow$ update effectiveness by Eqn. 2 for $\mathbf{x}^i_R \in R_{adv}, \mathbf{x} \in D_{val}$

  $W_j \leftarrow \text{median}(W)$ for $\mathbf{x}^j_R \in R_{cand} \backslash R_{adv}$

  $R_{adv} \leftarrow top\text{-}m(R_{cand}, W)$ ▷ top m selection

end

return R_{adv}

C More implementation details

The statistics of the datasets and DNN models used in our experiments are summarized in Table 5.

Table 5. Statistics of image datasets and DNN models used in our experiments.

Task	Dataset	# Labels	# Input Size	# Training Images	DNN model
Traffic Sign Recognition	GTSRB	13	224×224	4772	ResNet-34
	BelgiumTSC	11	224×224	3556	ResNet-34
	CTSRD	22	224×224	2028	ResNet-34
Face Recognition	PubFig	60	300×300	5181	ResNet-34
Object Classification	ImageNet subset	12	300×300	12406	ResNet-34 DenseNet-121

Detailed implementation of baselines. There are two baselines for our experiments. For clean-label attack (CL) *et al.* [53], we use the same settings as reported in their paper. Specifically, we use Projected Gradient Descent (PGD) adversarial perturbation bounded to L_∞ maximum perturbation ϵ=16. For SIG [3], Backdoored image are generated with horizontal sinusoidal signal defined by

$$v(i,j) = \Delta sin(2\pi jf/m), 1 \leq j \leq m, 1 \leq i \leq l, \tag{4}$$

where f is a certain frequency, we follow [3] and set $\Delta = 20$ and $f = 6$.

D Original test accuracy on different datasets.

As reported in Table 6, we show the test accuracy of the same model but trained on the original clean data

E Results on more target classes

We run more experiments with different target classes (*e.g.*class indexes 1, 2, 3, 4) on GTSRB dataset. The test accuracy and attack success rate are reported in

Table 6. The "original test accuray" is the test accuracy of the same model but trained on the original clean data. † denotes the model is replaced by a DenseNet.

Dataset	GTSRB	BelgiumTSC	CTSRD	PubFig	ImageNet	ImageNet†
Original test acc.	87.40	99.89	97.11	91.31	91.78	93.01

Table 7. While there are some variations, the overall results of our *Refool* attack are consistent over different target classes.

Table 7. Attack success rate and test accuracy (on clean test samples) of our *Refool* attack on different target classes of the GTSRB dataset.

Class ID	Test accuracy	Attack success rate
0	86.30%	91.67%
1	81.75%	87.98%
2	85.48%	89.74%
3	85.75%	90.83%
4	81.29%	91.81%

F More quantitative results for stealthiness comparison

By randomly selecting 500 images from CTSRD, we conduct a quantitative comparison of the stealthiness between our *Refool* and the baselines CL [53] and SIG [3]. The average L2, L1 distances and Mean Square Error (MSE) between the original images and their backdoored versions are reported in Table 8. The distortions of our *Refool* are much lower than either CL or SIG, indicating higher stealthiness. This is further verified by more visual inspections on some randomly selected examples in Fig. 8.

Table 8. The average distortions (measured by L2, L1 and MSE distances) made by different backdoor attacks on 500 randomly selected clean training images.

	CL [53]	SIG [3]	*Refool*
L2 norm	145.15	147.13	**113.67**
L1 norm	119.65	125.50	**72.06**
MSE	273.73	201.55	**75.30**

G More results against state-of-the-art backdoor defenses

White-box trigger removal. For Fine-Pruning [35], we replicate the Fine-pruning via PyTorch [1] and prune the last convolutional layer (*i.e.*, `layer4.2.conv2`) of the DNNs. In terms of white-box trigger removal, for our *Refool*, we adopt a state-of-the-art reflection removal method [61]. For Badnets [20], we simply

Fig. 8. More visual inspections for the stealthiness of CL [53], SIG [3] and our *Refool*.

replace the value of the trigger by the mean pixel value of their three adjacent patches. For CL *et al.* [53], we use the non-Local means denoising technique [6]. For SIG [3], we add the $-v(i,j)$ defined in Eqn. (4) on backdoored image back to the backdoor image to remove the trigger pattern. We apply trigger removal on the poisoned training data, then retrain the model under the same condition for all the other four datasets: BelgiumTSC, CTSRD, PubFig, and ImageNet. As shown in Table 9, our *Refool* maintains a much higher success rate after trigger removal than either CL or SIG across all datasets. We notice that *Refool* also exhibits an obvious success rate drop on ImageNet datasets. We suspect this is caused by the large amount of natural noise exists in ImageNet images. These natural noise tends to affect the effectiveness of all backdoor patterns, and also increase the possibility for them to be removed. We believe that, for our attack,

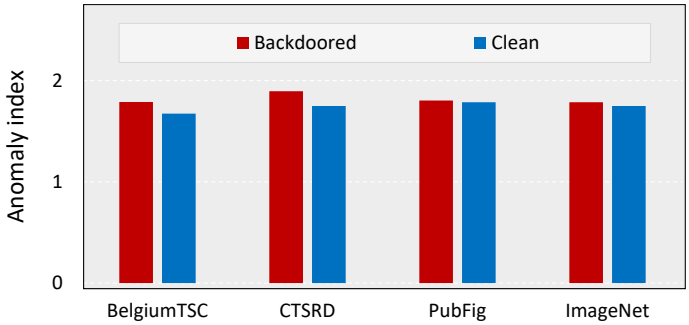

Fig. 9. More results of Neural Cleanse on five datasets.

Table 9. The attack success rate (%) of different backdoor attacks before or after white-box trigger removal.

Dataset	Badnets [20]		CL [53]		SIG [3]		Ours	
	before	after	before	after	before	after	before	after
BelgiumTSC	11.40	0.75	46.25	8.33	51.86	0.88	85.70	77.78
CTSRD	25.24	7.23	63.63	11.52	57.39	6.10	91.70	83.09
PubFig	42.86	13.33	78.67	31.74	69.01	8.34	81.30	68.42
ImageNet+ResNet	15.77	8.98	55.38	17.69	63.84	8.45	82.11	36.93
ImageNet+DenseNet	20.14	7.32	67.43	12.93	68.00	7.37	75.16	28.07

this can be addressed by simply increasing the intensity of the reflection. A more adaptive reflection backdoor to this situation is an interesting future work.

Neural Cleanse detection. Fig. 9 illustrates more results of *Refool* backdoored models against Neural Cleanse detection on datasets BelgiumTSC, CTSRD, PubFig and ImageNet. None of the four backdoored models by our *Refool* can be detected by Neural Cleanse. Note that only an anomaly index > 2 indicates a successful detection.

Input denoising or data augmentation based defenses. We further evaluated the resistance of our *Refool* attack to input denoising methods on CTSRD dataset. Specifically, we consider denoising techniques from Guo *et al.* [21]: image quilting, Total Variation denoising (TV denoise), JPEG compression, and Pixel quantization. We also include the data augmentation based mixup defense in [60]. These denoising or augmentation defenses are mostly proposed for adversarial attacks, but can be directly applied to backdoor attacks. We apply the denoising methods on all test samples (both backdoored and non-backdoored), and report the model's performance on denoised samples. For mixup, we retrain the network on the backdoored training set with its default setting. As shown in Table 10, these denoising or augmentation methods indeed can decrease the attack success rate for 4%. However, they are less effective than defenses like fine-tuning or trigger (*e.g.* reflection) removal. And image quilting seems great-

Table 10. The resistance of our *Refool* attack to input denoising or data augmentation defenses on CTSRD dataset

Methods	Test accuracy (%)	Attack success accuracy (%)
Refool (proposed)	86.30	91.67
Quilting	11.35	89.09
TV denoise	85.43	89.84
JPEG compression	86.57	90.98
Pixel quantization	86.30	91.01
Mixup	87.79	87.08
Reflection removal	86.41	85.01

ly decrease the model's performance on clean samples, *i.e.*, test accuracy drops from 86.30% to 11.35%.