Perceptual-Sensitive GAN for Generating Adversarial Patches

Supplementary Material

GAN Training Skills

We use several tricks to make the training process of our GAN-based model more stable. They are listed as follows:

- Input images are normalized between -1 and +1.
- WGAN-GP technique (Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017) is further introduced in order to make the training process more stable.
- Tanh is utilized as the activation function at the last layer of the generator.
- Batch Normalization (Ioffe and Szegedy 2015) and Layer Normalization (Ba, Kiros, and Hinton 2016) skills are used for better results.
- Initial learning rate is set to be 0.0002 with a decrease by 10% every 900 steps.
- LeaklyRelu is employed instead of Relu or Sigmoid to avoid sparse gradients.
- We use Adam optimizer for the generator while SGD optimizer for the discriminator.
- We alter the training of the generator and the discriminator. More specifically, we train the generator once also once for the discriminator.

Generator Architecture

The architecture of the generator in our model is listed as follows:

Layer	Num of Channels	Filter Size	Activation
conv0	16	4×4	LReLU
conv1	32	4×4	LReLU
conv2	64	4×4	LReLU
conv3	128	4×4	ReLU
deconv0	64	4×4	ReLU
deconv1	32	4×4	ReLU
deconv2	16	4×4	ReLU
deconv3	3	4×4	Tanh

Discriminator Architecture

The architecture of the discriminator in our model is shown below:

Layer	Num of Channels	Filter Size	Activation
conv0	64	5×5	LReLU
conv1	128	5×5	LReLU
conv2	256	5×5	LReLU
conv3	512	5×5	LReLU
FC	64		Sigmoid

Adversarial Example Definition

An adversarial example is defined as a kind of modified image that is extremely similar to the original one. The perturbations added are visible to human beings while misleading to deep learning models. The noise is confined to a very small norm. So, pixel-level difference unawareness to humans is the key feature of adversarial example when this concept was proposed for the first time. Nevertheless, no high-level modification has been analyzed, e.g., perceptual modification to original images. Just like the method presented in this paper, we added perceptually correlative adversarial patches. The noise has high perceptual correlations with the image context, in which case people are less likely to feel unnatural or get confused. The result is, people can recognize the attacked image with no doubt while deep learning models fail to do so. Indeed, it is an adversarial example but quite different from the previous definition. Thus, we suggest that the definition of adversarial example should be reconsidered and re-discussed by taking perceptual similarity and correlation into consideration in the future.

Target Model Architecture

We will present the detailed architecture of the target models we used in our experiments. Specifically, we present the architecture of basic models, i.e., VY and $VGG16$, the variations of them will not be repeated here.

VY We also employ VY as the classifier on GTSRB. Compared to $VGG16$, it is a smaller network. The architecture is shown below:

Layer	Num of Channels	Filter Size	Activation
conv0	3	1×1	ReLU
conv1	32	5×5	ReLU
conv2	32	5×5	ReLU
maxpool	32	2×2	
conv3	64	5×5	ReLU
conv4	64	5×5	ReLU
maxpool	64	2×2	
conv5	128	5×5	ReLU
conv6	128	5×5	ReLU
maxpool	128	2×2	
FC	1024		ReLU
FC	1024		ReLU
FC	43		Softmax

VGG16 (modified) $VGG16$ is utilized as the classifier for both GTSRB and ImageNet. The only difference between the $VGG16$ models of two datasets lies on the channel number of the last fully connected layer. It is shown as follows:

Layer	Num of Channels	Filter Size	Activation
conv0	64	3×3	ReLU
conv1	64	3×3	ReLU
maxpool	64	2×2	
conv2	128	3×3	ReLU
conv3	128	3×3	ReLU
maxpool	128	2×2	
conv4	256	3×3	ReLU
conv5	256	3×3	ReLU
conv6	256	3×3	ReLU
maxpool	256	2×2	
conv7	512	3×3	ReLU
conv8	512	3×3	ReLU
conv9	512	3×3	ReLU
maxpool	512	2×2	
conv10	512	3×3	ReLU
conv11	512	3×3	ReLU
conv12	512	3×3	ReLU
maxpool	512	2×2	
FC	4096		ReLU
FC	4096		ReLU
FC	1000		softmax
FC	4096		ReLU
FC	43		softmax

More Experiment Results

In this section, more experiment results will be presented. Figure 1 illustrates the adversarial patches in physical world and the attacking results. Figure 2 shows the different generated adversarial patches and their corresponding seed patches.

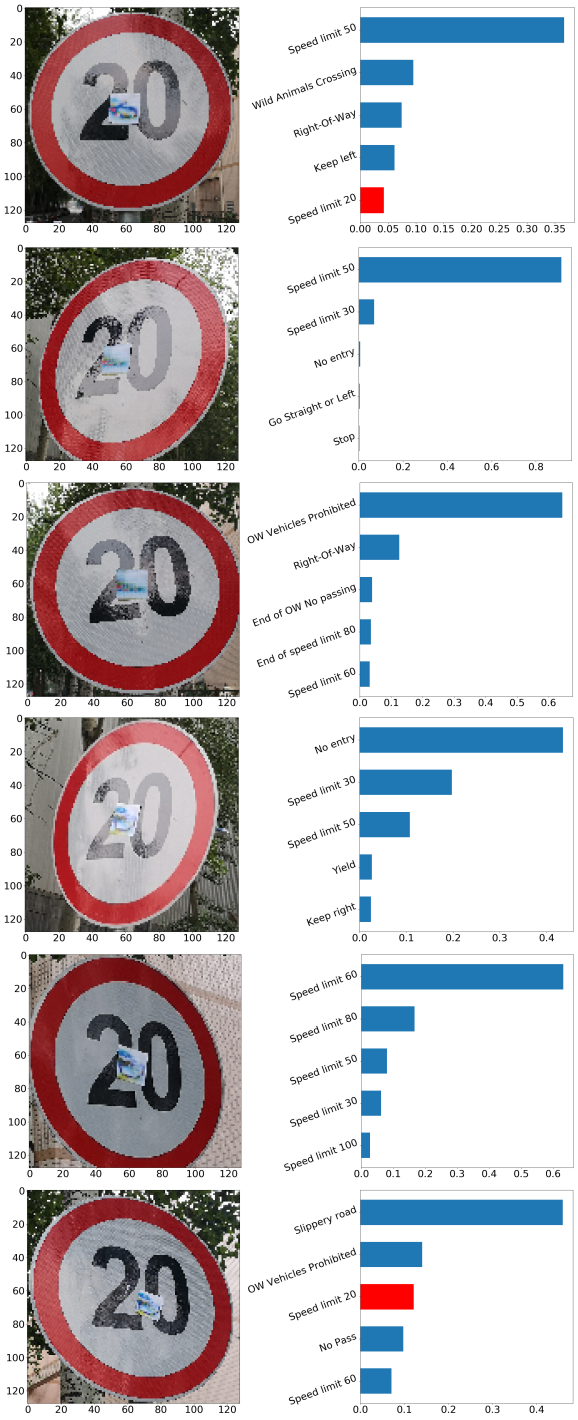
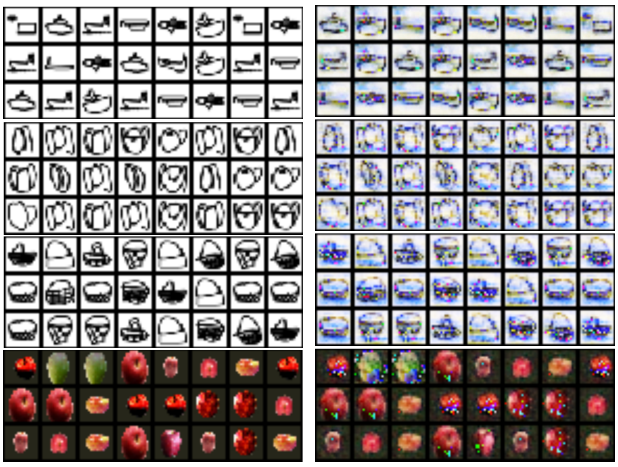


Figure 1: On the left, we show example photos of real-world traffic signs with adversarial patches generated by our PS-GAN on them. On the right, each histogram demonstrates the top-5 classification results of the corresponding attacked traffic sign on the left, where the red bar in the histogram indicates the true class of the traffic sign.



(a) Seed Patches (b) Adversarial Patches

Figure 2: The seed patches and the corresponding adversarial patches generated by PS-GAN.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, 214–223.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, 5767–5777.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.