Translation Invariance Experiment

Adam Spindler

March 11, 2021

1 Methodology

The standard approach for creating adversarial images is to iteratively update an image so that the loss of the target model increases. The image is updated by computing the derivative of the loss function with respect to the input image, which gives you a tensor that points in the direction of increasing the loss function. This gradient is multiplied by a small constant (like 0.01) and added to the input image to slightly move it in the direction of increasing loss. This process is repeated iteratively until the image is modified enough to cause a mis-detection. For my experiments, I go several iterations beyond that to improve transferability.

In another attempt to increase the transferability of my adversarial examples I implemented the translation invariance technique as described in [1]. The idea behind this approach is to shift the adversarial image during its creation so that the adversarial image could be shifted after creation and still retain its adversarialness. This was also shown to increase transferability between models in their paper.

The way that this works specifically is that each iteration of the adversarial image creation, several copies of the current version of the image are made where each one is translated by a small amount. Then each image is passed through the model and its gradient is computed for the loss function. Each of these gradients are then translated back to where they would be on the original input image and averaged together. This combined gradient is then added to the input image. A graphical representation is illustrated in figure 1, and this idea is expressed mathematically below:

Original:

$$X := X + \epsilon \nabla_X f_{\text{ensemble}}(X) \tag{1}$$

With Translation Invariance:

$$X := X + \epsilon \frac{1}{N} \sum_{i=1}^{N} T_i^{-1}(\nabla_X f_{\text{ensemble}}(T_i(X))) \tag{2}$$

Where:

X	Current version of input image
ϵ	Learning Rate
f_{ensemble}	Ensemble of models which outputs a scalar loss, which is basically the sum of confidence scores
N	Number of translations
T	Translation function
T^{-1}	Inverse of translation function

For this experiment the following set of translations were used each iteration:

```
[(0, 0), (2, 2), (4, 0), (0, 4), (-4, -4), (-8, 0), (0, -8), (0, 8), (0, 8), (8, 8), (-8, -8)]
```

1.1 Evaluation on AWS Rekognition

20 images of stop signs were used to evaluate the effectiveness of this technique. These images are available in my GitHub repository in the "/test_images/test_set20/" directory. 3 sets of images were sent to AWS Rekognition for evaluation: the unmodified images, the regular ensemble adversarial images, and the translation invariant ensemble adversarial images. The results of this are listed in figure 3.

The confidence score of each of the classes did go down with this technique and so did the number of bounding boxes detected, at least for the "Road Sign" class. However, when creating the adversarial images in general it increased the number of images in which the stop signs and regular signs were detected which is unexpected.

Overall, this result is interesting though because there is a non-negligible decrease in the confidence score despite the translation invariant images looking very similar to the regular adversarial images I created. It also shows that translation invariance is effective at increasing transferability of adversarial images.

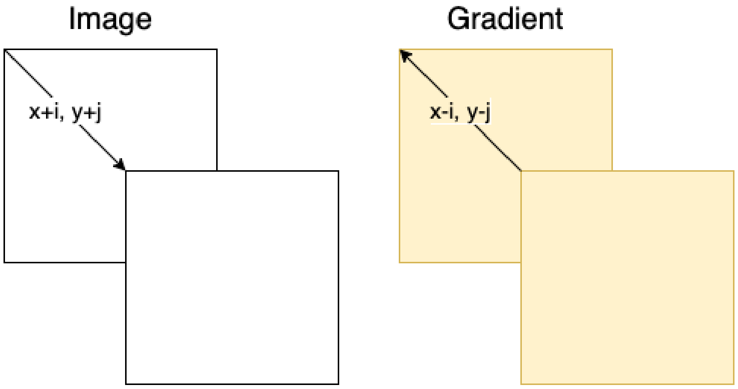

Figure 1: The image is translated in the X and Y directions, shifting it before being passed into the network. Then the gradient is computed and shifted back to align its values with the original image.

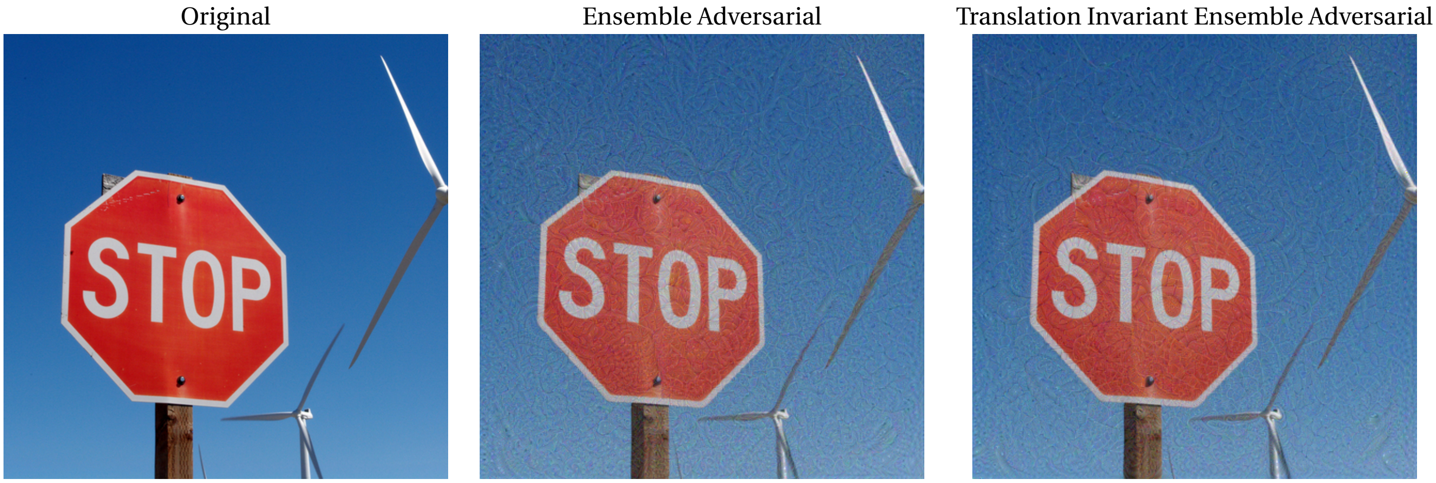

Figure 2: Visual Comparison of Adversarial Image Techniques

	Original	**Ensemble Adversarial**	**Translation Invariant Ensemble Adversarial**
Stop Sign	9 images 78.497 avg confidence 0 bboxes	15 images 74.891 avg confidence 0 bboxes	15 images 71.905 avg confidence 0 bboxes
Road Sign	18 images 95.671 avg confidence 24 bboxes	18 images 91.348 avg confidence 9 bboxes	18 images 87.795 avg confidence 5 bboxes
Sign	16 images 96.117 avg confidence 0 bboxes	19 images 90.422 avg confidence 0 bboxes	19 images 88.508 avg confidence 0 bboxes

Figure 3: The evaluation of adversarial images on AWS Rekognition, listing the number of images each class appeared in, the average confidence score of the detections, and the number of bounding boxes detected across all 20 images.

2 Future Work

The adversarial images created for this experiment have a moderate level of distortion applied. The images have clearly been altered, but they are easily detectable as a stop sign when looking at the images as a human. Each of these images were run for 30 iterations of adversarial updates which may not be enough to fool AWS. More significant alterations may be required to create transferable adversarial images. The computation time is the real limitation here. It takes about 8 minutes to create a single adversarial image using translation invariance with 30 total iterations even on a machine with a decent GPU. Which makes quick experimentation difficult.

In short, the things I can try next are: increase the number of iterations, increase the number of translations, and randomize which translations are used each iteration. I can also try limiting the image modifications to the inside of the bounding boxes of the stop signs.

References

[1] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks, 2019.