Rotational Adversarial Patch

Roei Zaady
318747946
roeiza@post.bgu.ac.il

Tom Segal
208945519
tomsega@post.bgu.ac.il

Gil Shenderovitz
204810295
gilshend@post.bgu.ac.il

March 13, 2024

Colab Notebook: [1]

1 Introduction

An Adversarial Patch is a small figure designed to trick a computer vision model to classify an object incorrectly [2]. Based on the idea of Adversarial Examples, which are inputs which cause the model to behave unexpectedly, machine learning models have been found to be quite vulnerable to these types of attacks [3]. These small figures can be printed and placed in the physical world, to affect models which receive inputs in which these patches are observable [2, 3]. The patches are created by irritably running a Deep Neural Network (DNN) that introduces perturbations in the space of the patch, across many images, to find an optimal pixel-arraignment, which is the representation of some label for the target model. Therefore, when this patch will be visible, it will shift the model to focus on the patch, causing it to mislabel other objects in the image.

Our work in this paper is based on [2], in which the patches are always displayed at the same orientation they were created and always cause the model to classify objects with the same incorrect label. In our work, we take this attack a step forward by creating a patch which can cause the model to mislabel objects to two different labels, based on the patch's orientation.

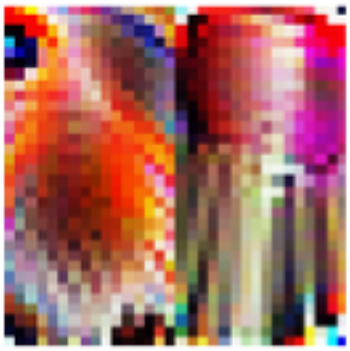
2 Methodology

We first loaded a pretrained ResNet34 model using the torchvision python library and froze its layers as we do not perform any training of the model. For a dataset, we used a variation of Tiny-ImageNet which contains 5,000 images of 1000 classes, downsized to 224×224 colored images [4]. The images were normalized according to the dataset's mean and standard deviation prior to usage.

We then approached the issue from three directions:

2.1 Concatenation

Based on the framework from [2], two Adversarial Patches were created, each for a different class, each half the area of a standard patch. A vertically-aligned patch with half the width of a standard patch and a horizontal patch with half the height compared of a standard patch. To form a single square patch, the vertical patch is concatenated with a the horizontal patch after rotating it 90 degrees counter-clockwise.



(a) Goldfish, Lipstick - size 32

(b) Toaster, Goldfish - size 48

(c) Schoolbus, Pineapple - size 64

Figure 1: Concatenated Patches

2.2 Simultaneous Training

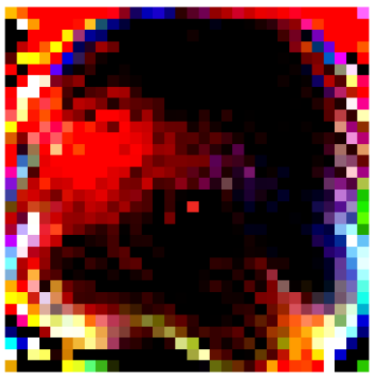
Here, a single patch is created for both classes, according to its orientation. To create this patch, a different cross-entropy loss is calculated for each patch with respect to its label and orientation and then aggregated to guide subsequent training iterations.

The loss function can be represented as follows:

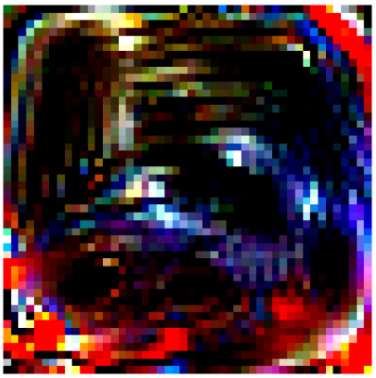
$$CombinedLoss = \sum_{i=1}^{N} (CE(pred_i, \text{target\_class}) + CE(pred_{\text{rot}_i}, \text{target\_class\_rot})) \tag{1}$$

Where:

- N is the number of images in the dataset
- CE represents the cross-entropy loss function
- $pred_i$ and $pred_{\text{rot}_i}$ are the predictions of the model on the original image and the image with the rotated patch, respectively.
- target_class and target_class_rot are the target classes for the original image and the rotated image, respectively.



(a) Goldfish, Lipstick - size 32



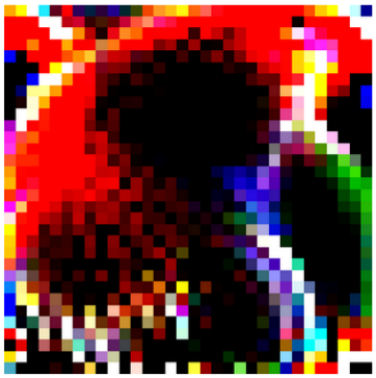
(b) Toaster, Goldfish - size 48

(c) Schoolbus, Pineapple - size 64

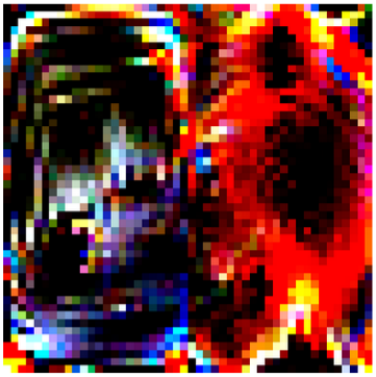
Figure 2: Simultaneous Training Patches

2.3 Fine-Tuned Concatenated Patches

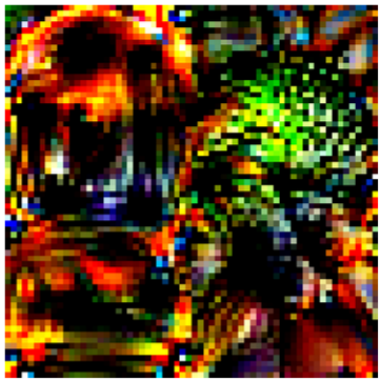
In an attempt to enhance the attack's success, we combined both previous methods. A concatenated patch is created and fine-tuned, each patch according to its label and orientation, using the combined loss as in Simultaneous Training.



(a) Goldfish, Lipstick - size 32



(b) Toaster, Goldfish - size 48



(c) Schoolbus, Pineapple - size 64

Figure 3: Simultaneous Training Patches

3 Experiments

For each method, we tested patches from five classes, which were combined to create three sets of twenty-five patches overall. The effectiveness of the adversarial patches was evaluated based on their ability to deceive the pretrained ResNet34 model towards a certain label, on the Tiny-ImageNet dataset. In the following table, we present the top-1 and top-5 accuracy results attained in our experiments. We also include the performance of non-rotated Standard-Patch and of Half-Sized Patches, to examine the effects of the transformation we have done.

Attack Type	**Size 32**		**Size 48**		**Size 64**	
Accuracy	**Top-1**	**Top-5**	**Top-1**	**Top-5**	**Top-1**	**Top-5**
Concatenation	7.90%	19.85%	34.85%	52.90%	81.90%	93.30%
Simultaneous Training	13.75%	23.95%	18.70%	37.75%	34.35%	67.70%
Fine-Tuned Concatenated	14.25%	28.80%	33.70%	57.10%	58.45%	92.85%
Standard Patch	64.04%	82.57%	91.83%	98.51%	98.12%	99.89%
Half-Sized Patch*	20.46%	42.90%	57.38%	71.86%	91.19%	98.14%

*Half-Sized Patches sizes are: 32 - (32,16)/ (16,32), 48 - (24,48)/(48,24), 64 - (32,64)/(64,32).

Table 1: Top-1 & Top-5 Accuracy of Attacks

4 Discussion

According to experiments' result, it is evident that the effectiveness of the adversarial patches varies depending on the method employed and the size of the patch. The larger the patch, the more it can capture in detail the model's representation of the targeted label. In addition, a larger patch takes a larger portion of the image, further pulling the model's attention to it. These shortcomings, are visible in the sharp drop of performance of half-sized patches.

The standard patch, which is not rotatable to elicit a different target label, demonstrates higher accuracy compared to the other methods across all patch sizes. However, the concatenated and fine-tuned concatenated patches, manage to introduce an entirely new capability of dual-label according to orientation, with a relatively small impact on performance, when at size of 64x64 pixels. Overall, the concatenated and fine-tuned concatenated patches, show significant improvement over the simultaneous training method, especially at larger patch sizes.

Interestingly, the concatenated patches perform worse than half-sized patches despite being a combination of two half-sized patches.

The Concatenated Patches show the best performance out of all the methods we examined, at 81.9% Top-1 accuracy and 93.3% Top-5 accuracy. Fine-Tuning the concatenated patches has only decreased performance, presumably due to over-fitting, since each label in the dataset has a mere five samples. It is also possible that during gradient computation, in the fine-tuning stage, the rotation may cause gradients to align in different directions, potentially weakening their combined strength.

This may also be the reason as to why do patches created using Simultaneous Training, show lesser performance.

References

[1] G. Shenderovitz, T. Segal, and R. Zaady, "Rotational adverserial patch," `Colab`, 2024, retrieved from https://drive.google.com/file/d/1FFHp3EsG3Oj1iSOeUKRY7iaD3jg4aXPk/view?usp=sharing.

[2] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *CoRR*, vol. abs/1712.09665, 2017. [Online]. Available: http://arxiv.org/abs/1712.09665

[3] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *CoRR*, vol. abs/1607.02533, 2016. [Online]. Available: http://arxiv.org/abs/1607.02533

[4] P. Lippe, "Tiny imagenet variation," 2020. [Online]. Available: https://github.com/phlippe/saved_models/blob/main/tutorial10/TinyImageNet.zip