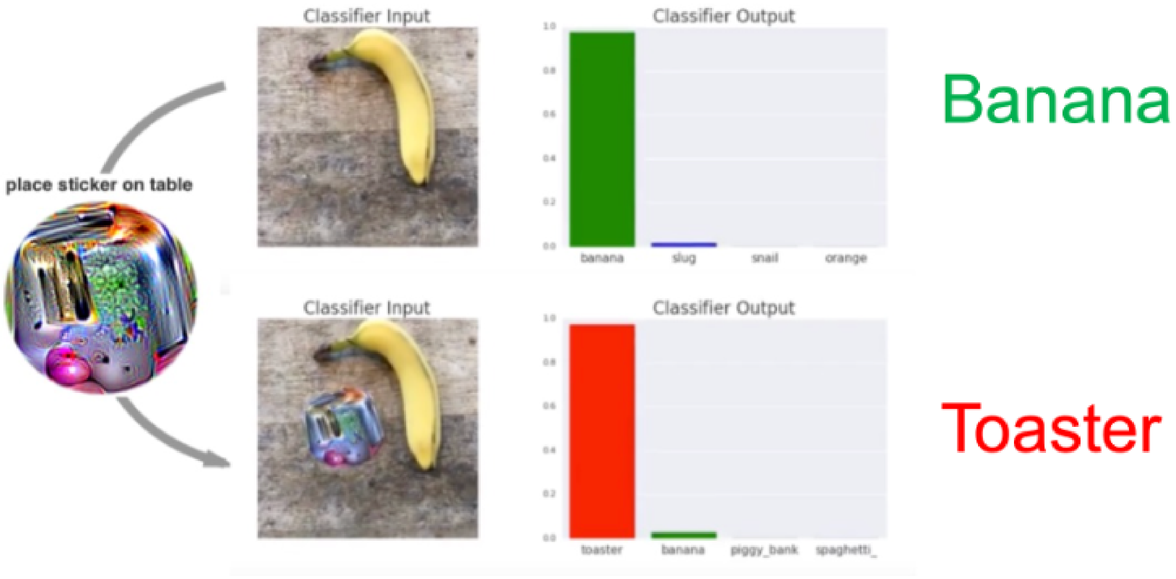
Introduction

The last two decades have witnessed a tremendous increase in the applicability and performance of deep neural networks. Specifically, the advent of convolution-based neural networks (CNNs) has led to a giant leap in progress for vision-based tasks such as object recognition and face recognition, with accuracy rates matching or even out-performing human perception. At the same time, a myriad of adversarial attacks has demonstrated the vulnerability of these deep learning systems by exposing the threats associated with implementing them in the real world. Take the case of autonomous driving, which uses image recognition models to classify items on the road such that the car can respond appropriately to them. If an adversarial attack were able to misclassify pedestrians as empty roads, it could have severe consequences for people walking near that car. The same is true for models used to identify a patient's problems in medical research (brain imaging for instance). Hence given the wide applicability of neural networks in our daily lives, these attacks can have severe consequences if unchecked.

Adversarial attacks commonly work by modifying each pixel in an image by a small amount, using optimization strategies such as L-BFGS, Fast Gradient Sign Method (FGSM), DeepFool, Projected Gradient Descent (PGD), and so on. By adding salient but carefully constructed noise to the images, the networks start classifying classes incorrectly. Often this noise is imperceptible to the human eye, which makes it harder to know when the model has been attacked.

However, despite their efficiency, these attacks can be infeasible in their applicability. By focusing on small or imperceptible changes to the input, these attacks are heavily image-dependent such that generalizing them to the real world requires prior knowledge of the lighting conditions, camera angle, type of classifier being attacked, or even the other items within the scene. In response, we now explore developing "physical" attacks that can simply be printed and applied in the physical world without prior knowledge of the scene. An example is shown in the image below, wherein adding a specially designed patch on or near the object of interest proves to be a successful attack against the neural network.



Related Work

Many researchers have studied the generalizability of adversarial attacks to the real world. Kurakin et al.[1] demonstrated that when printed out, an adversarially constructed image will continue to be adversarial to classifiers even under different lighting and orientations. Athalye et al.[2] demonstrated adversarial objects which can be 3d printed and misclassified by networks at different orientations and scales. Evtimov et al.[3] demonstrated various methods for constructing stop signs that are misclassified by models, either by printing out a large poster that looks like a stop sign, or by placing various stickers on a stop sign.

Adversarial patches specifically were first introduced by Brown et al.[4] for image classifiers. Their goal was to produce localized, robust, and universal perturbations that are applied to an image by masking instead of adding pixels. The patch found by Brown et al. was able to fool

[1] *A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533, 2016.*

[2] *A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. arXiv preprint arXiv:1707.07397, 2017.*

[3] *I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song. Robust physical-world attacks on deep learning models. arXiv preprint arXiv:1707.08945, 2017.*

[4] *Brown, T. B., Mane, D., Roy, A., Abadi, M., and Gilmer, J. ´ Adversarial patch. CoRR, abs/1712.09665, 2017. URL http://arxiv.org/abs/1712.09665.*

multiple ImageNet models into predicting “toaster” whenever the patch is in view, even in physical space as a printed sticker. However, because classification systems only classify each image as a single class, to some extent this attack relies on the fact that it can simply place a high-confidence “deep net toaster” into an image (even if it does not look like a toaster to humans) and override other classes in the image.

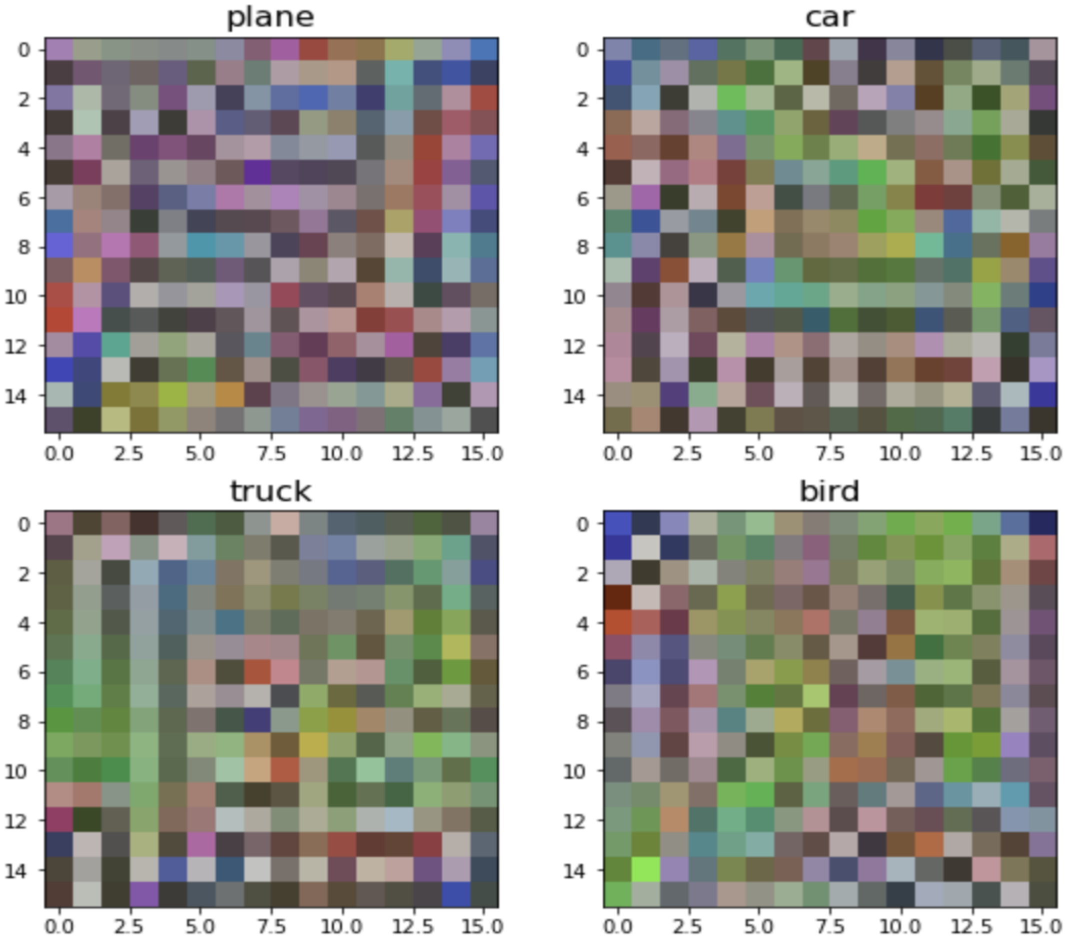
We now try to emulate their work for the CIFAR-10 dataset.

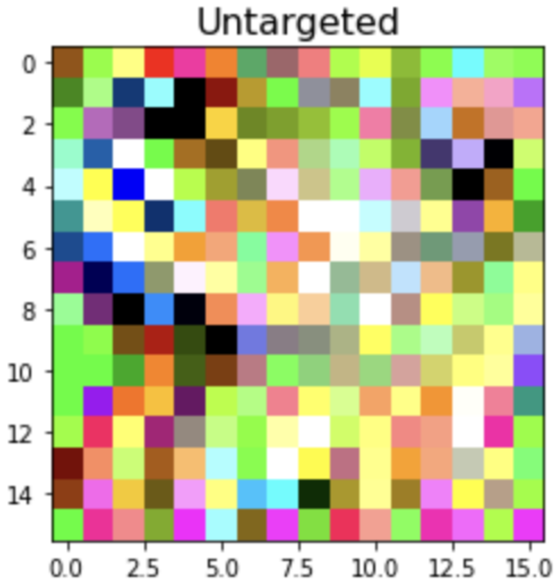
Methodology

The CIFAR-10 data set consists of 60,000 colored images with 10 label classes. Each image is of the size 32 by 32 pixels in terms of height and width. The dataset was split into training and testing data with 50,000 images in the training data and 10,000 images in the test data. Both the training and the testing datasets were normalized using the parameters calculated from the training dataset.

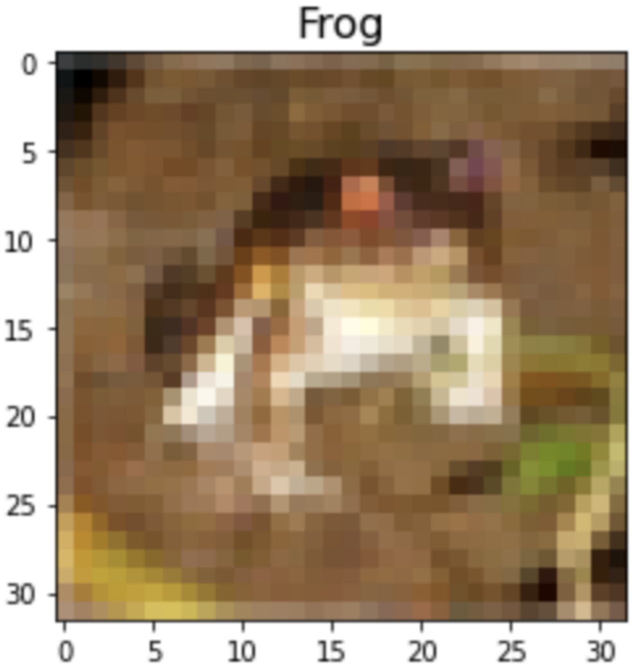
A convolution neural network following the Resnet 20 architecture was trained on this data. The model was further tuned to achieve an accuracy of 90% on the test dataset. This trained model was then used to generate targeted and untargeted patches. Due to a lack of computing resources, we were only able to generate a limited number of patches. The targeted patches were generated for the car, bird, truck, and the plane classes.

A rectangular patch of the required size was initialized and placed at a random location in the training images. The patch was then optimized over 10 epochs with a learning rate of 0.1 using our trained Resnet 20 model. We used Adam optimizer for this purpose as it led to the best results. For untargeted patches, the patch was optimized to maximize our cross-entropy loss function on the training data set. For targeted patches, the patch was optimized by minimizing the loss function for our target class. The 16x16 patches are shown below.


plane
car
truck
bird
0
2
4
6
8
10
12
14
0.0
2.5
5.0
7.5
10.0
12.5
15.0


Untargeted
0
2
4
6
8
10
12
14
0.0
2.5
5.0
7.5
10.0
12.5
15.0

We do not notice any visual patterns in our patches for several reasons. The Cifar-10 dataset images are of small size containing only 32x32 pixels. Hence, when these images are viewed, they appear blurred. A sample image of a frog taken from our training dataset is shown below.

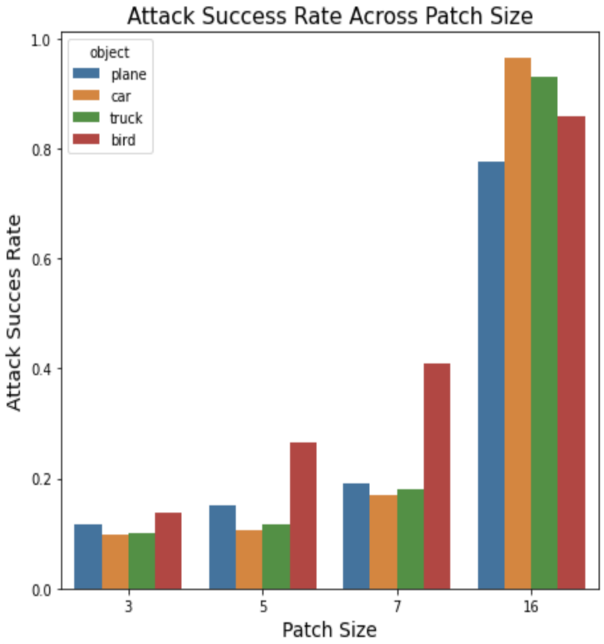


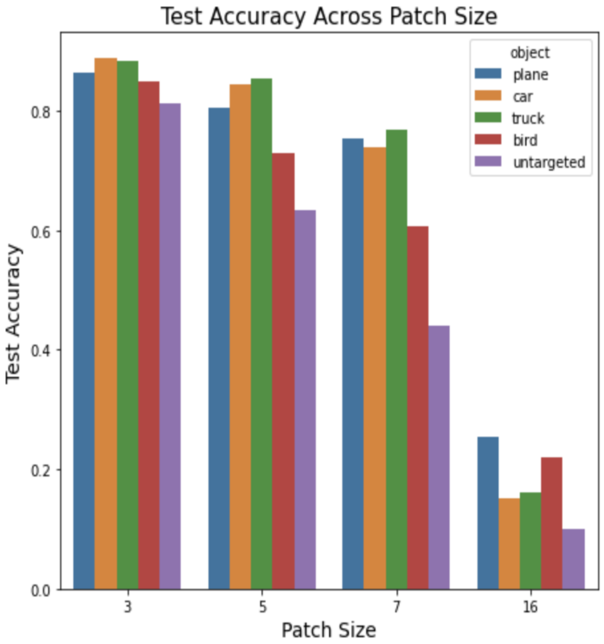
Secondly, due to computational constraints, each of the patches were optimized over 10 epochs only. We believe that if further optimized, we will start seeing noticeable visual patterns.

The generated patches were then placed at a random location inside the test images. These patches were evaluated by measuring the performance of white box and black box models on the modified test data. The untargeted patches were evaluated using test accuracy and the targeted patches were evaluated using both test accuracy and attack success rate (ASR). ASR was defined as the proportion of images that were predicted as the target class.

Results of our Experiments

For each of the previously mentioned class labels, we generated patches of patch size 3x3, 5x5, 7x7, and 16x16. The same-sized untargeted patches were also generated. The success of these patches was then evaluated on our test data using our Resnet 20 CNN, which served as our white box model. We noticed that the larger patches were more successful in fooling our white box model as compared to smaller patches. As the patch size increased, our test accuracy dropped, and the attack success rate increased. This was found to be true for all targeted and untargeted attacks. The results of our white box model are shown below.





To test whether these patches would generalize well across other models, we used a VGG 16 model architecture as our black box model. This model was pretrained on the Image Net dataset, and we used transfer learning to optimize the model on our Cifar-10 training data, which resulted in an accuracy of 92% of our unperturbed test data. On adding our patches to the data, we notice that validation accuracy fell for patches of all sizes, while the ASR increased as well. Overall, we notice similar trends as the whitebox model, wherein larger patch sizes result in worse accuracies.

Validation Accuracy for various patch sizes in targeted attacks				
Class name	***3x3***	***5x5***	***7x7***	***16x16***
Bird	26.08%	26.06%	23.76%	14.88%
Car	25.22%	24.27%	22.53%	12.64%
Plane	26.67%	24.51%	24.14%	13.04%%
Truck	25.98%	25.42%	24.96%	12.95%

# of times target was predicted for various patch sizes in targeted attacks				
Class name	***3x3***	***5x5***	***7x7***	***16x16***
Bird	1391	1684	2759	325
Car	462	690	1309	378
Plane	1424	1270	1409	26
Truck	100	107	112	175

Conclusion

Overall, we can say that adversarial patches were successful in fooling the neural network model. Although it's hard to discern trends in the patches because of the nature of the Cifar 10 dataset; we notice that larger the patch size, lower the accuracy of our white box or black box model. The patches generalize to the location and orientation in which they are applied, and are transferable across models. For future work in this project, we would like to see how these patches generalize to the real world if they were printed out and applied to settings with different conditions.