Abstract

In this paper, we implement a method of robust 3D adversarial attacks which considers different viewpoints where the victim camera can be placed. In particular, we find a method to create 3D adversarial examples that can achieve 100% attack success rate from all viewpoints with any integer spherical coordinates. Our method is simple as we only perturb the texture space. We create 3D models with realistic textures using 3D reconstruction from multiple uncalibrated images. With the help of a differentiable renderer, we then apply gradient based optimization to compute texture perturbations based on a set of rendered images, i.e., training dataset. Our extensive experiments show that even only including 1% of all possible rendered images in training, we can still achieve 99.9% attack success rate with the trained texture perturbations. Furthermore, our thorough experiments show high transferability of the multiview robustness of our 3D adversarial attacks across various deep neural network models.

1. Introduction

Despite the fact that adversarial examples are well explored in the 2D realm, physical adversarial attack is still a few steps away from realization [21, 13, 18, 15, 3, 10, 23, 12, 16, 14]. One of the main reasons is the lack of robust 3D adversarial examples that can consistently fool deep neural networks in multi-view settings. To overcome the challenges, there are some prior works on generating 3D adversarial examples and have made significant progress in obtaining improved attack success rates [1, 2, 27, 24]. It was found that the attack success rate depends on the range of viewpoints where the victim camera can be placed [24]. Given the same number of victim image instances used in optimization (training), when the range of viewpoints increases, the attack success rate of the adversarial 3D model drops. However, it is not known yet whether a 3D adversarial model with 100% attack success rate from all possible viewpoints could ever be generated against current popular deep neural network models. The next question to ask is if such 3D adversarial models exist, how many training images are at least needed in the process of optimization.

In this paper, we investigate the above two questions and provide insights into multiview attack robustness of 3D adversarial examples. In particular, we propose a method to create 3D adversarial models that can achieve 100% attack success rate from viewpoints with any integer spherical coordinates. Those integer spherical coordinates constitute a dense sampling of the viewing sphere around an object, which ensures a statistically high confidence level in the success rate achieved by the proposed method. We apply the method and generate 3D adversarial examples for 5 different realistic 3D objects. One challenge is to ensure the victim camera can be fooled from any viewpoint and at the same time make the 3D adversarial example realistic. Realistic models are important because their existence is less conspicuous, matching real-world objects and the environment around them in detail and thus less noticeable by humans. To tackle this challenge, our method only perturbs the texture, and the original 3D models with realistic textures are created using 3D reconstruction from multiple uncalibrated images. Fast Gradient Sign Method based training is applied to compute the texture perturbations that maximize the loss between the prediction of the rendered images and the correct class.

We further investigate the minimum number of training images required to obtain such a robust 3D adversarial example. We find that for victim images uniformly distributed at different perspectives, our method only needs 1% of the total in the process of optimization to achieve 99.9% attack success rate. This result is encouraging because it means with less computation resource and time restrictions, robust 3D adversarial examples can be generated and studied. We also perform black-box attacks on 12 popular deep neural networks. Results show that there is a high transferability of perturbations of our method.

*This work was supported in part by NSF grants CNS-1758017 and MRI-1828644.
2. Approach

2.1. Multiview Robust 3D Adversarial Example Training

Our method to generate 3D adversarial models includes the following three stages: 1) create a 3D model with realistic textures using 3D reconstruction from numerous un-calibrated images; 2) render 2D victim images from the 3D model to form the training dataset; 3) compute the texture perturbation by applying gradient based optimization on the training dataset.

To achieve the multiview robustness of 3D adversarial examples, in the second stage, we render 2D images at different viewpoints which are spherically uniformly distributed. We define a renderer’s viewpoint location using spherical coordinates \((\rho, \theta, \phi)\) representing distance, altitude, and azimuth. We vary the altitude and azimuth with different integer values, but keep the distance fixed for sake of simplicity. When we test the attack success rate of our 3D adversarial models, the victim camera is allowed to be placed at viewpoints with any integer value of \(\theta\) ranging from \(-90^\circ\) to \(90^\circ\), and the azimuth \(\phi\) ranging from \(1^\circ\) to \(360^\circ\).

Given a 3D textured model \(X(T)\) with a texture \(T\) and a differentiable renderer \(r(\bullet)\), a 2D image \(Y\) rendered from the camera \((\rho, \theta, \phi)\) can be expressed as

\[
Y = r(X(T), \rho, \theta, \phi),
\]

where \(\psi\) denotes other rendering parameters such as light and shading. Denote the output classification of a deep neural network \(f(\bullet)\) by \(Z\) such that \(Z = f(Y)\), for the rendered 2D image \(Y\). Let \(Z_{\text{Correct}}\) be the actual ImageNet label for the 3D model that we are using. If \(Z \neq Z_{\text{Correct}}\), then we disregard \(Y\) and proceed to the next image.

Optimization Objective. For correctly classified images, we compute the loss between the image’s output classification and 3D object’s correct class. We use the cross entropy loss function, defined as \(-\log p_{\hat{Y},c}\) where \(p_{\hat{Y},c}\) is the predicted probability that the input \(Y\) is of the correct class. The loss is accumulated across the entire training dataset, becoming

\[
L(T) = -\sum_{Y \in R} \mathbb{1}(Y) \ast \log p_{\hat{Y},c} \quad (1)
\]

where \(T\) represents the texture of the 3D model and \(R\) the training dataset. By following the FGSM-based optimization, the texture is updated in the direction of the gradient \(\nabla_T L(T)\) such that

\[
T = T + \epsilon \ast \text{sign}(\nabla_T L(T)). \quad (2)
\]

The noise magnitude \(\epsilon\) is assigned a small value like 0.001 each iteration in order to find a minimum perturbation required. With the proposed optimization, we can obtain the trained texture \(T_{\text{perturbed}}\) such that all rendered 2D images in the training dataset are misclassified by the target deep learning model. Our pseudocode summarizing the entire procedure is shown in Algorithm 1.

**Algorithm 1** Multiview Robust 3D Adversarial Example Training

1: procedure ADVTRAIN\((X(T), f, Z_{\text{Correct}}, \rho)\)
2: \(\epsilon = 0.001\)
3: altitude_range = range\((-90, 90)\)
4: azimuths_range = range\((0, 360)\)
5: \(\rho = 2.732\)
6: while true do
7:     for \(\theta\) in altitude_range do
8:         for \(\phi\) in azimuths_range do
9:             \(Y = r(X(T), \rho, \theta, \phi)\)
10:            \(Z = f(Y)\)
11:            if \(Z = Z_{\text{Correct}}\) then
12:                \(T = T + \epsilon \ast \text{sign}(\nabla_T L(T))\)
13:            end if
14:         end for
15:     end for
16:     if all \(Z \neq Z_{\text{Correct}}\) then
17:         Break
18:     end if
19: end while
20: Return the perturbed texture \(T\)
21: end procedure

2.2. Training Image Dataset Size

In our approach the training dataset size directly affects the training time, and may affect the attack success rate of 3D adversarial examples. Our goal is a 100% attack success percentage from any viewpoints with integer altitude and azimuths coordinates. In order to determine the minimum number of training images needed to achieve our goal, we conduct a search for this training dataset size by starting with the largest training dataset possible and then shrinking it at a quadratic rate. We find a tight range in which the model remains completely adversarial from any viewpoint, as shown in Section 4.3. The images in our datasets are all evenly spaced, but if appropriate, one can also choose to include more rendered images from some particular angles of a 3D model than others.

In Algorithm 1, we define a sampling step size \(\rho\) which represents the number of integer degrees in both the azimuth and altitude direction per image sample. For example, when \(\rho_{\text{train}} = 10\), for every 10 degree change in the azimuth and for every 10 degree change in the altitude, one rendered image is included into this training dataset, totaling \(18 \times 36 = 648\) images.
3. Experiments

3.1. Experiment Setup

We first created realistic 3D models using photogrammetry, and then used a differentiable Neural Renderer [8] on the 3D models to obtain 2D victim images for the training dataset. The testing dataset’s sampling step is fixed at $p_{test} = 1$, i.e., we test the 3D adversarial model from all viewpoints with integer coordinates. To reduce the number of hyperparameters, we fixed the perturbation per iteration in Algorithm 1, $\epsilon$, at $10^{-3}$ for all experiments.

3.2. Results on Attacking in Texture Space

In this experiment, we test the efficacy of only attacking the texture space of a 3D model of a grey running shoe. For the experiments in this subsection we set our sampling step $p_{train}$ at 3, and we use the Inception v3 model. As the computing time and resources are an important factor of attack feasibility, we investigate how increasing the number of iterations of Algorithm 1 on rendered 2D images will affect the false negative rate on the training dataset. The false negative rate of the 2D image classifier on the training images reflects the percentage of training images that can fool the classifier. Figure 1 shows that after only 6 iterations, more than 90% of our training images are misclassified, and after 15 iterations all training images become adversarial. 100% of the testing dataset becomes false negatives once the perturbations are finished training.

After we finish training, i.e., all 2D images in the training dataset are misclassified, we reconstruct the 3D adversarial model using the perturbed images. Figure 2 shows renderings of the model without the texture perturbations, with the perturbations, and the perturbations themselves once noise training is finished. As we can see in the figure, the difference between (c) Model with perturbation and (a) Model without perturbations is not noticeable by humans’ eyes.

3.3. Results on Different Sampling Ratios

Continuing with the grey running shoe and Inception v3 model, in this subsection, we investigate the effect of different sampling ratios in training dataset on the multiview attack success rate. In this experiment, we render 2D images from the 3D adversarial model at all viewpoints with integer altitude and azimuth coordinates and calculate the percentage of rendered images that are misclassified by the classifier. This percentage is denoted as the attack success rate. The attack success rates reported in Section 4.4 and 4.5 are calculated in the same way.

As shown in Figure 3, setting the sampling step size $p_{train} = 1$ results in a 100% attack success rate, which is expected because the training dataset is iteratively trained until reaching a 100% false negative rate and all 64800 images in the testing dataset are included in the training dataset. More interestingly, setting $p_{train} = 2$ or 3 preserves a 100% attack success rate. In other words, even if the training procedure only utilizes a small subset of all possible rendered images, the entire testing dataset can still be misclassified on Inception v3. Furthermore, if a 100% multiview attack success rate is not needed, we can greatly reduce the amount of computation time by choosing a very small training dataset, e.g. setting $p_{train} = 10$. This implies a training dataset of only 648 images (1% of rendered images are used in training), but it still yields an extremely high attack success rate of more than 99%, allowing users to quickly generate the 3D adversarial examples. Note that we still ensure the training dataset is fully adversarial in training.

3.4. Results on Other Models

Our approach is generalizable to a diverse set of 3D models. In our experiments, in total we have five lifelike models, corresponding to the following 4 ImageNet labels: running shoes (grey and black respectively), a pineapple, a power drill, and a teddy bear (Figures 2 and 4.) We perform the same experiments in Section 4.3 on the other four models, and results are shown in Figure 4. With a small $p_{train} = 3$ corresponding to a larger training dataset, all four models reach an attack success rate greater than 99%, and with a smaller training dataset $p_{train} = 10$, three models (black running shoe, pineapple, power drill) retain a
We select 12 popular deep learning models with dissimilar architectures, and conduct experiments on the gray running shoe model. We first collect the initial false negative rates (misclassification rates) of different classifiers on the original gray running shoe model. Then for every deep learning model, we generate a 3D adversarial example and found that no model requires more than 19 iterations in Algorithm 1 to obtain a fully adversarial training dataset when \( p_{\text{train}} = 10 \). Using each 3D adversarial example created based on one particular learning model, we launch attacks on the other remaining 11 models, and measure the attack success rates. Table 1 shows that there is a high transferability of perturbations, agreeing with previous research [11]. Specifically, our multiview robust 3D model created based on Inception v3 preserves attack success rate at above 93% on all the other deep learning models. The attack success rates on the other models show similar results. Therefore, our 3D adversarial attacks remains effective in the black-box setting.

4. Conclusion

In this paper we propose an approach to generate 3D adversarial models that can achieve 100% attack success rate from any viewpoints with integer spherical coordinates. Our approach is simple and realistic, as we perturb only the texture space. We find that even with only a small portion of 2D images in the training process, we can still achieve close to 100% attack success rates. Our extensive experiments including black-box tests have shown the effectiveness of our approach and the perturbation has very good transferability.

Table 1. Attack success rates of multiview robust 3D adversarial examples on different deep learning models. Each row indicates the deep model based on which the 3D adversarial example is generated. The column names indicate different target deep learning models. The data unit is %.

|--------|------------------|--------------|--------|-----------|----------|-----------|------------|-----------|--------|------------|---------|

References


