On Multiview Robustness of 3D Adversarial Attacks

Philip Yao
University of Michigan
Ann Arbor, MI
philiyao@umich.edu

Andrew So
California State Polytechnic University, Pomona
Pomona, CA
acso@cpp.edu

Tingting Chen
California State Polytechnic University, Pomona
Pomona, CA
tingtingchen@cpp.edu

Hao Ji
California State Polytechnic University, Pomona
Pomona, CA
hji@cpp.edu

ABSTRACT

Nowadays deep neural networks have been applied widely in many applications of computer vision including medical diagnosis and self-driving cars. However, deep neural networks are threatened by adversarial examples usually in which image pixels were perturbed unnoticeable to humans but enough to fool the deep networks. Compared to 2D image adversarial examples, 3D adversarial models are less invasive in the process of attacks, and thus more realistic. There have been many research works on generating 3D adversarial examples. In this paper, we study the robustness of 3D adversarial attacks when the victim camera is placed at different viewpoints. 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 adversraial attacks across various state-of-the-art deep neural network models.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Security and privacy** → **Software and application security**.

KEYWORDS

deep neural networks; 3D adversarial examples; multiview robustness

PEARC '20, July 26–30, 2020, Portland, OR, USA

ACM ISBN 978-1-4503-6689-2/20/07…$15.00
https://doi.org/10.1145/3311790.3396652

ACM Reference Format:
Philip Yao, Andrew So, Tingting Chen, and Hao Ji. 2020. On Multiview Robustness of 3D Adversarial Attacks. In *Practice and Experience in Advanced Research Computing (PEARC '20), July 26–30, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3311790.3396652

1 INTRODUCTION

As more applications become reliant on computer vision systems in today's world, robustness in machine learning algorithms is also becoming more critical. For example, convolutional neural network (CNN) based image recognition [15] is foretold to be used in life-threatening contexts, ranging from self-driving cars [12] to disease detection through X-rays [3, 5]. However, deep neural networks are facing the threat by adversarial examples which have human-vision-unnoticeable perturbations to original images, and lead to incorrect predictions [31].

Adversarial examples are well explored in the 2D realm, but with secured cameras, it is unrealistic for attackers to manipulate the images before being sent to the deep neural network models locally. On the other hand, realistic physical attacks with 3D adversarial examples are posing a bigger challenge to the robustness of deep neural network models. In general, a change in the physical parameters of an object does not necessarily correspond to a local change in the rendered images. Consequently, it is difficult to calculate the physical parameters that will achieve high attack success rate especially when an adversarial object could be viewed by the victim camera from many different viewpoints. There are some prior works on generating 3D adversarial examples and have made significant progress in obtaining good attack success rates[2, 6, 34, 37].

A successful approach to perturb the physical parameters in generating 3D adversarial examples is to use gradient based optimization [2, 6, 34, 37]. It applies loss on the network output and propagates the gradient from network prediction to the physical properties, e.g., shape or texture of the mesh, with the help of a differentiable renderer. It was found that the attack success rate depends on the range of viewpoints where the victim camera can be placed[34]. 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 in 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.

Our contributions in this paper are summarized as follows. 1) We generate 3D adversarial models that can achieve 100% attack success rate from viewpoints all-around with any integer spherical coordinates. We find that even when we only use 1% of all possible victim images from different viewpoints in training, we can still achieve 99.9% attack success rate. 2) We create realistic high definition adversarial models that can fool state-of-the-art deep neural networks, by only manipulating the texture. 3) We perform extensive tests on the robustness of our 3D adversarial attacks. We render at all integer angles and our comprehensive test for each model is comprised of 64800 images. It demonstrates that our attack is robust and adversarial from most, if not all, viewpoints.

2 RELATED WORK

There has been some significant research in adversarial machine learning in the 2D realm [18, 22, 25, 28, 31]. Goodfellow et al. developed the popular Fast Gradient Sign Method (FGSM) attack, which alters an image in a fixed step size towards the direction of most likely misclassification [8]. Due to its speed and effectiveness, it is now a popular choice in adversarial machine learning, and in our research we bring its iterative version into the 3D realm. Kurakin et al. [16] compared the FGSM attack against iterative and targeted iterative variations. Xiao et al. developed adversarial examples using generative adversarial networks [33]. Black-box attacks are more difficult to perform since they loosen the assumption that adversaries have knowledge of the target neural network architecture. Our experiments investigate black-box attacks, agreeing with other research which has shown that classification error due to perturbations are transferable to other models separate from those used in training [16, 21, 26]. Kurakin et al. [16] focused on the transferability of noise resultant from the FGSM attack. Papernot et al. [24] approached the black-box attack by learning an explicit substitute of their target deep neural network.

Regarding the 3D realm, Eykholt et al. [6] and Brown et al. [4] were able to develop robust physical attacks using stickers and patches. They did not directly modify their subject's texture, which is the focus of our research. Athalye et al. [2] has 3D printed adversarial models using a method called "Expectation Over Transformation," which maximizes the likelihood of a target class for the expectation of various transformations on an image. Zeng et al. [37] has shown that 3D adversarial models can be constructed using both differentiable and non-differentiable renderers. Their method using the differentiable renderer alters multiple physical parameters, thus requiring more degrees of freedom. Furthermore, their objective function included only one rendering and hence only one perspective which remained adversarial. Liu et al. [20] has developed physical adversaries using differentiable renderers, but their attack approach focuses on altering lighting. Xiao et al. [34] were also able to develop adversarial examples using a differentiable renderer and focused on altering the mesh. Their approach was effective on models with "rich shape features but minimal texture variation" [34]. Their novel approach encompasses a new smoothing loss function, and showed promising results on black-box attacks. Their untargeted attacks performed well on black-box attacks under controlled rendering parameters.

Differentiable renderers are a critical component of our pipeline. We use Neural Renderer by Kato et. al., who achieved differentiability by developing an approximate gradient for the discrete rasterization operation [14]. Other variations of developing differentiable rasterization exist as well [7]. RenderNet is a deep neural network which was trained using pixel-space loss for rendering [23], and Li et al. developed a differentiable ray tracer [17].

3 APPROACH

3.1 Perturbation in Texture Space

The texture and mesh are the defining characteristics for 3D models. Although there are other physical parameters like surface shading and light scattering, our method focuses on perturbing the textures in order to show that adversaries need to control only this one degree of freedom, to make the 3D adversarial examples less noticeable by human eyes.

3.2 3D Model Multiview Description

In a differentiable renderer, virtual cameras can be placed from multiple angles to render pictures of 3D objects. To describe the multiview of 3D models, we use spherical coordinates (ρ, θ, ϕ). The altitude θ ranges from -90° to 90°, and the azimuth ϕ ranges from 1° to 360°; hence all faces of the target model are visible to the deep learning classifier. The distance ρ from the renderer's viewpoint to the object is fixed such that no area is truncated for all rendered

orientations for every rendered model. In our approach of 3D adversarial attack, we measure the multiview robustness by rendering 2D images from the 3D adversarial example at all possible integer θ and ϕ, and calculate the misclassification rate as the attack success rate.

3.3 Gradient Based Optimization

To achieve the multiview robustness of 3D adversarial examples, we propose to use gradient based optimization to obtain effective texture perturbations and include rendered images at a uniformly distributed array of angles in the training dataset. In particular, we collect a set of rendered images from the original 3D model by varying the altitude and azimuth with different integer values in the defined range. We call this set of images the training dataset. This set of images is used to calculate the gradients for the perturbation. The testing dataset is the set of images rendered on the adversarial model used to evaluate attack efficacy.

Given a 3D textured model $X(T)$ with a texture T and a differentiable renderer $\mathbf{r}(\bullet)$, a 2D image Y rendered from the camera (ρ, θ, ϕ) can be expressed as

$$Y = \mathbf{r}(X(T), \rho, \theta, \phi, \psi),$$

where ψ denotes other rendering parameters such as light and shading.

Denoting the output classification Z of a deep neural network $f(\bullet)$ such that $Z = f(Y)$, for the rendered 2D image Y. If the image was misclassified, then we disregard this rendering location and proceed to the next coordinate location. To formalize this, let $Z_{Correct}$ be the actual ImageNet label for the 3D model that we are using, and create this indicator function:

$$\mathbb{I}(Y) = \begin{cases} 1, & \text{If } f(Y) = Z_{Correct} \\ 0, & \text{Otherwise} \end{cases}$$

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 $-\log p_{Y,c}$, where $p_{Y,c}$ is the predicted probability that the input Y is of the correct class. The loss is accumulated across the entire training dataset, becoming

$$L(T) = -\sum_{Y \in R} \mathbb{I}(Y) \times \log p_{Y,c} \tag{1}$$

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

$$T = T + \epsilon \times sign(\nabla_T L(T)). \tag{2}$$

The noise magnitude ϵ 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 $T_{perturbed}$ such that all rendered 2D images in the training dataset are misclassfied by the target deep learning model.

3.4 Training Image Dataset Size

In our approach the training dataset size is one of the most important factors because it directly affects the training time, and may effect 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.

To succinctly describe how many images and which images are included in a dataset, we define a sampling step size p which represents the number of integer degrees in both the azimuth and altitude direction per image sample. For example, when $p_{train} = 10$, 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. Likewise, when $p_{train} = 1$, a total of $180 \times 360 = 64800$ images are included in the training dataset.

We first include 64800 images in the training dataset, then 16200 images, then 7200 images, etc. and 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 this is not restricted by our method. If appropriate, one can also choose to include more rendered images from some particular angles of a 3D model than others.

3.5 Multiview Robust 3D Adversarial Example Training

Our pseudocode summarizing the entire procedure is shown in Algorithm 1. For an original 3D model X, our algorithm generates an adversarial 3D example based on a deep learning model f. The other two inputs to the function are object class $Z_{Correct}$ as the ground truth, and the sampling step size p. Based on p we select a set of uniformly distributed integer θ and ρ, and render the 2D images at corresponding viewpoints. We apply FGSM-based attacks on the textures of these 2D images until all images in the training dataset become adversarial. After the perturbed texture T is returned in the last step, the 3D adversarial model could be generated by mapping perturbed texture directly to the mesh faces originally from multiple uncalibrated images.

4 EXPERIMENTS

In this section, our experiments first demonstrate the efficacy of only attacking in texture space to achieve the multiview robustness of 3D adversrial models. We then investigate how many images are needed in the training dataset to achieve 100% attack success rate, by sweeping across different p_{train} values. We show that even with larger p_{train} values (i.e. smaller training dataset) the constructed 3D adversarial model can still be robust. We extend our experiments on multiple realistic 3D models, and finally we perform black-box attacks and examine the transferability of noise generated by our method among various deep learning models.

4.1 Experiment Setup

In our experiments, from multiple photographs of physical objects, we create realistic original 3D models using photogrammetry software such as Agisoft PhotoScan[1]. Fig. 1 shows an example of the reconstruction process to obtain the 3D model of a running shoe.

Algorithm 1 Multiview Robust 3D Adversarial Example Training

procedure AdvTrain($X(T), f, Z_{Correct}, p$):
  $\epsilon = 0.001$
  altitude_range = range($-90, 90, p$)
  azimuth_range = range($0, 360, p$)
  $\rho = 2.732$
  while *true* **do**
    for θ *in altitude_range* **do**
      for ϕ *in azimuth_range* **do**
        $Y = r(X(T), \rho, \theta, \phi)$
        $Z = f(Y)$
        if $Z == Z_{Correct}$ **then**
          $T = T + \epsilon \times sign(\nabla_T L(T))$
        end
      end
    end
    if *all* $Z != Z_{correct}$ **then**
      Break
    end
  end
  return the perturbed texture T;
end

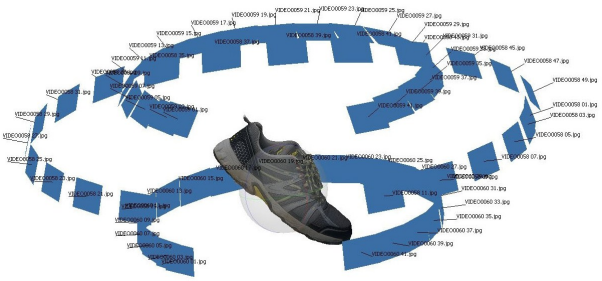


Figure 1: A couple of images taken from various viewpoints are used to reconstruct the 3D model.

We used a differentiable Neural Renderer [14] on the 3D models to obtain 2D victim images and form the training dataset. The testing dataset's sampling step is fixed at $p_{test} = 1$. It means that 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, ϵ, at 10^{-3} for all experiments. This proves to be a sensible choice because within certain number of iterations for all of our models, all images in all training datasets become adversarial.

4.2 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. Before any texture perturbation is produced, the Inception v3 model classifies 65.06% of the testing dataset correctly, i.e. there is a 34.94% false negative rate for the 64800 rendered images. Fig. 2 shows a map where the blue areas indicate coordinates of correctly classified images in the testing dataset.

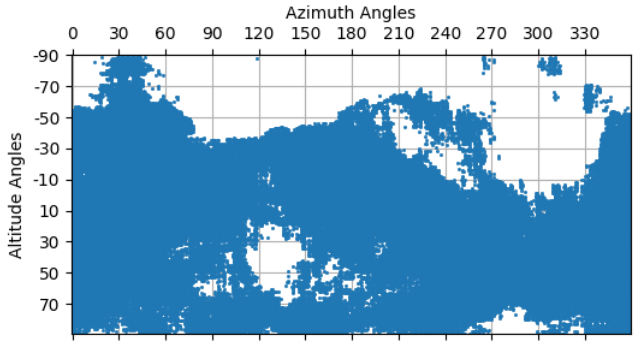


Figure 2: The blue areas cover coordinates of viewpoints from which the rendered images are classified correctly by Inception v3.

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 in the training dataset will affect the false negative rate on the training dataset. The false negative rate of the 2D image classifier on the training images reflects how many percent of the training images can fool the classifier. Fig. 3 shows how the false negative rate on training dataset grows with the increasing number of iterations. 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.

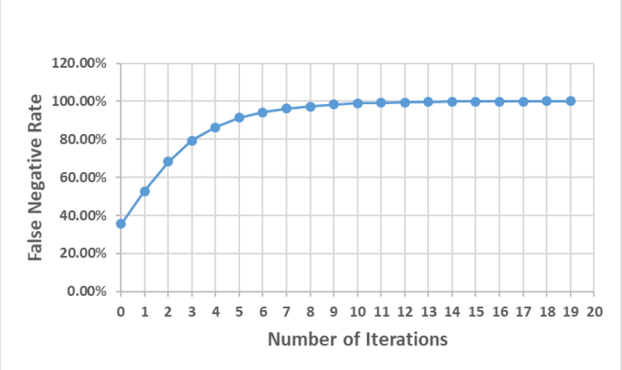
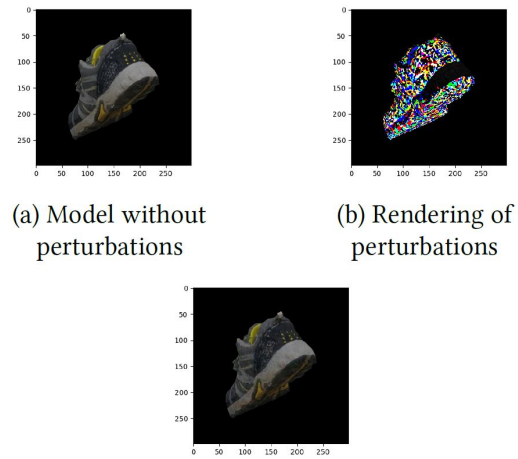


Figure 3: The effect on the number of iterations of the I-FGSM on the percentage of false negatives for the training dataset

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. Fig. 4 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.



(a) Model without perturbations

(b) Rendering of perturbations

(c) Model with perturbations

Figure 4: Renderings of a running shoe at $(\phi, \theta) = (30°, 270°)$

4.3 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 mis-classified 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.

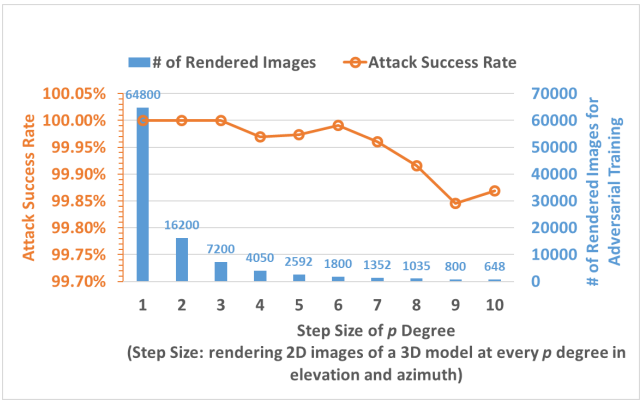


Figure 5: Plot of attack success rate versus p_{train}. The number of images used for adversarial training is completely determined by p_{train} but is also shown for convenience.

As shown in Fig. 5, 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. This result loosens the amount of computational power required to develop robust 3D adversarial examples. 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

(A) Black Running Shoe

(B) Pineapple

(C) Power Drill

(D) Teddy Bear

Models	Label in ImageNet	Initial False Negative Rate	Training with $P_{train} = 3$	Training with $P_{train} = 10$
A	770	34.7886	100	99.0401
B	953	8.7515	99.9877	99.8889
C	740	22.9614	99.9923	99.7716
D	850	10.1698	99.9352	81.8981

Figure 6: Four models are listed with their ImageNet labels. The right of each model lists the testing results on training with $p = 3$ and $p = 10$ respectively. The Sampling Step Size $P_{test} = 1$, and the data unit is %.

Table 1: Initial false negative rates of 12 deep classifiers on the gray running shoe model. Each false negative rate equals the misclassification rate of the deep model on 64800 rendered images from all viewpoints. The Sampling Step Size $p = 1$, and the data unit is %.

Target Deep Models	Initial False Negative Rates
Inception	34.94
AlexNet	35.69
VGG	30.20
ResNet	47.33
SqueezeNet	46.11
DenseNet	25.20
GoogLeNet	46.90
ShuffleNet	38.31
MobileNet	39.98
ResNeXt	34.16
Wide ResNet	23.37
MNASNet	55.62

Table 2: Attack success rates of multi-view 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 %.

	No. of Iter	Inception	AlexNet	VGG	ResNet	SqueezeNet	DenseNet	GoogLeNet	ShuffleNet	MobileNet	RetNeXt	Wide_ResNet	MNASNet
Inception [30]	19	100	93.1481	99.9429	99.8349	99.8194	98.7145	97.7485	99.3148	99.983	99.9012	98.4614	100
AlexNet [15]	15	92.4352	99.9738	94.7083	96.1821	97.821	78.9907	91.3536	97.8565	96.6821	87.966	84.3241	99.2901
VGG [27]	8	83.3951	50.8272	99.9923	83.1235	82.5525	68.4784	79.0864	69.6265	95.0818	81.0031	63.8735	96.2716
ResNet [9]	10	92.6373	70.091	97.9429	99.9969	98.7515	94.1034	91.216	96.321	98.7392	97.213	90.7886	99.9506
SqueezeNet [13]	9	78.1651	60.9244	88.8426	87.6281	99.9969	63.179	80.8194	83.2901	88.1836	76.8812	57.1059	93.0864
DenseNet [11]	11	94.8287	66.1728	97.3951	98.3704	93.3843	99.9923	90.5694	95.2793	98.179	98.8843	94.608	99.9151
GoogLeNet [29]	8	98	85.8071	99.6806	99.2284	98.9522	98.3796	99.9969	98.8735	99.6852	99.3843	97.6003	99.9985
ShuffleNet [38]	10	87.1559	66.5633	87.6744	93.9336	93.2623	78.9954	86.3009	99.9923	95.4892	90.1204	78.5123	99.4182
MobileNet [10]	10	91.6698	62.7438	98.7454	94.0972	91.5802	84.6636	89.0664	89.6358	99.9985	93.1852	86.8333	99.9846
ResNeXt [35]	11	95.2901	66.608	98.4321	95.6713	93.3225	92.9043	90.5216	93.3287	98.0633	99.9815	96.0849	99.983
Wide ResNet [36]	14	98.1543	79.3858	98.6713	99.517	97.1698	97.9444	95.5617	97.8071	99.3395	99.3596	99.9691	99.9907
MNASNet [32]	6	74.9429	46.1358	84.125	79.6512	70.5787	62.9599	77.625	68.6682	92.3904	71.966	52.3997	99.9738

training dataset of only 648 images (only 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.

4.4 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 (Fig. 4 and 6.)

We perform the same experiments in Section 4.3 on the other four models, and results are shown in Fig. 6. 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 false negative rate of more than 99%. The adversarial Teddy Bear model obtains 81.90% attack success rate.

4.5 Results on Black-Box Attacks

All the experiments so far are conducted against the Inception v3 model. In this section, we perform a set of black-box attacks on various deep learning models, in order to test the transferability of the perturbation effectiveness of our 3D adversarial attacks.

We select 12 popular deep learning models with dissimilar architectures, and conduct experiments on the gray running shoe model. We first collect the false negative rates (misclassification rates) of different classifiers on the original gray running shoe model. As shown in Table 1, the initial false negative rates range between 20% to 60%, depending on the models used.

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 $p_{train} = 3$. 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 2 shows that there is a high transferability of perturbations, agreeing with previous research [20]. 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. For example, noise developed from GoogLeNet provoked more than 98% attack success rates on most of the other models, except for AlexNet which retains a false negative rate of 85.8%. Therefore, our 3D adversarial attacks remains effective in the black-box setting.

5 DISCUSSION

In our proposed method, we do not change the radius of the viewpoint due to the increase in computational complexity. Consequently, our adversarial noise is not completely scale invariant. For one of our models, we find that much of it does remain adversarial for a reasonable range, with a proportion of viewpoints reverting to a correct classification beyond that limit. We surmise that this issue can be resolved using an alternative attack method which learns scale invariant noise. One specific method is described by Lin et al. [19], which can be investigated in a future work. Also, although it seems unlikely that there will be significant classification difference in the non-integer angles and the rendered integer set, an additional future work should investigate random sampling of testing dataset viewpoints to ensure complete adversarial robustness with a determined p-value.

Additionally, our experiments have confirmed that the I-FGSM performs better than the single iteration FGSM at creating deceiving models. Kurakin et al. [16] showed evidence that the I-FGSM performs poorer than the FGSM on developing transferable noise. Our work on black-boxes shows that, despite their results, we can still achieve very high transferability across a wide variety of deep neural networks.

6 CONCLUSION

In this paper we study the multiview robustness of 3D adversarial examples. 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, and 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.

ACKNOWLEDGMENTS

This work was supported in part by NSF grant CNS-1758017. This work used the High-Performance Computing Cluster with the Mass Storage System at California State Polytechnic University, Pomona, which was supported in part by NSF grant MRI-1828644.

REFERENCES

[1] LLC Agisoft. 2018. Agisoft PhotoScan User Manual: Professional Edition.
[2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. 2017. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397* (2017).
[3] Yaniv Bar, Idit Diamant, Lior Wolf, Sivan Lieberman, Eli Konen, and Hayit Greenspan. 2015. Chest pathology detection using deep learning with non-medical training. In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 294–297.
[4] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. 2017. Adversarial patch. *arXiv preprint arXiv:1712.09665* (2017).
[5] Yuxi Dong, Yuchao Pan, Jun Zhang, and Wei Xu. 2017. Learning to read chest X-ray images from 16000+ examples using CNN. In *Proceedings of the Second IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies*. IEEE Press, 51–57.
[6] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2018. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1625–1634.
[7] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. 2018. Unsupervised training for 3d morphable model regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8377–8386.
[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).
[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[10] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).
[11] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.
[12] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, et al. 2015. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716* (2015).
[13] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360* (2016).
[14] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3D Mesh Renderer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.
[16] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236* (2016).
[17] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–11.
[18] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. 2019. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. *arXiv preprint arXiv:1905.00441* (2019).
[19] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. 2019. Nesterov Accelerated Gradient and Scale Invariance for Adversarial Attacks. arXiv:1908.06281 [cs.LG]
[20] Hsueh-Ti Derek Liu, Michael Tao, Chun-Liang Li, Derek Nowrouzezahrai, and Alec Jacobson. 2018. Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. (2018).
[21] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2016. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770* (2016).
[22] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2574–2582.
[23] Thu H Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yongliang Yang. 2018. Rendernet: A deep convolutional network for differentiable rendering from 3d shapes. In *Advances in Neural Information Processing Systems*. 7891–7901.
[24] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. ACM, 506–519.
[25] Eitan Rothberg, Tingting Chen, and Hao Ji. 2019. Towards Better Accuracy and Robustness with Localized Adversarial Training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 10017–10018.
[26] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. 2016. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*. 1528–1540.
[27] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[28] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation* (2019).
[29] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.
[30] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.
[31] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).
[32] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. 2019. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2820–2828.
[33] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610* (2018).
[34] Chaowei Xiao, Dawei Yang, Bo Li, Jia Deng, and Mingyan Liu. 2019. Meshadv: Adversarial meshes for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6898–6907.
[35] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1492–1500.
[36] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* (2016).
[37] Xiaohui Zeng, Chenxi Liu, Yu-Siang Wang, Weichao Qiu, Lingxi Xie, Yu-Wing Tai, Chi-Keung Tang, and Alan L Yuille. 2019. Adversarial attacks beyond the image space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4302–4311.
[38] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6848–6856.