Multiview-Robust 3D Adversarial Examples of Real-world Objects*

Philip Yao
University of Michigan
Ann Arbor, MI 48109
philiyao@umich.ed

Andrew So, Tingting Chen and Hao Ji
California State Polytechnic University
Pomona, CA, 91768
{acso,tingtingchen,hji}@cpp.edu

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 uncalibrated 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 (ρ, θ, ϕ) 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 θ ranging from $-90°$ to $90°$, and the azimuth ϕ ranging from $1°$ to $360°$.

Given a 3D textured model $X(T)$ with a texture T and a differentiable renderer **r**(•), 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. 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 $Z_{Correct}$ be the actual ImageNet label for the 3D model that we are using. If $Z \neq Z_{Correct}$, 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 $-\log p_{Y,c}$, 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

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

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

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

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. Our pseudocode summarizing the entire procedure is shown in Algorithm 1.

Algorithm 1 Multiview Robust 3D Adversarial Example Training

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

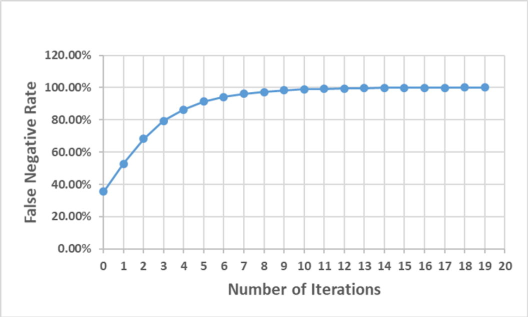


Figure 1. The effect on the number of iterations of the I-FGSM on the percentage of false negatives for the training dataset

3. Experiments

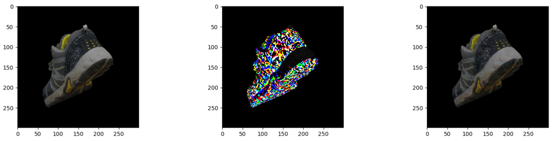
3.1. Experiment Setup

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, ϵ, 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.



(a) Model without Noises

(b) Rendering of Noises

(c) Model with Noises

Figure 2. Renderings of a running shoe at $(\phi, \theta) = (30°, 270°)$

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 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.

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

	InitialFNR	Iter#	Inception	AlexNet	VGG	ResNet	SqueezeNet	DenseNet	GoogLeNet	ShuffleNet	MobileNet	RetNeXt	Wide ResNet	MNASNet
Inception [20]	34.94	19	100	93.1481	99.9429	99.8349	99.8194	98.7145	97.7485	99.3148	99.983	99.9012	98.4614	100
AlexNet [9]	35.69	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 [17]	30.20	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 [4]	47.33	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 [7]	46.11	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 [6]	25.20	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 [19]	46.90	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 [28]	38.31	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 [5]	39.98	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 [25]	34.16	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 [26]	23.37	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 [22]	55.62	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

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 %.

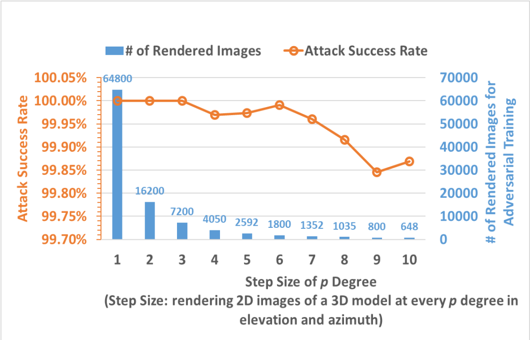
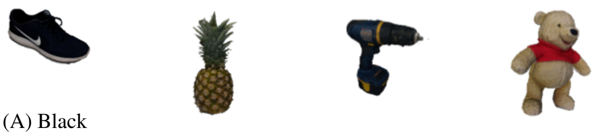


Figure 3. Plot of attack success rate versus p_{train}. The number of images used for adversarial training is completely determined by p_{train} but is also shown for convenience.



(A) Black Running Shoe (B) Pineapple (C) Power Drill (D) Teddy Bear

Models	Label in ImageNet	Attack Success Rate (Sampling Ratio: $P_{test} = 1, Unit: \%$) 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 4. 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.

false negative rate of more than 99%. The adversarial Teddy Bear model obtains 81.90% attack success rate.

3.5. 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 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 $p_{train} = 10$. 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.

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017. 1

[2] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on

deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. 1

[3] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[5] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 4

[6] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 4

[7] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 4

[8] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 4

[10] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 1

[11] Hsueh-Ti Derek Liu, Michael Tao, Chun-Liang Li, Derek Nowrouzezahrai, and Alec Jacobson. Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. 2018. 4

[12] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016. 1

[13] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 1

[14] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519. ACM, 2017. 1

[15] Eitan Rothberg, Tingting Chen, and Hao Ji. Towards better accuracy and robustness with localized adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 10017–10018, 2019. 1

[16] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. 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*, pages 1528–1540, 2016. 1

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[18] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019. 1

[19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 4

[20] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 4

[21] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1

[22] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019. 4

[23] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018. 1

[24] Chaowei Xiao, Dawei Yang, Bo Li, Jia Deng, and Mingyan Liu. Meshadv: Adversarial meshes for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6898–6907, 2019. 1

[25] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 4

[26] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 4

[27] Xiaohui Zeng, Chenxi Liu, Yu-Siang Wang, Weichao Qiu, Lingxi Xie, Yu-Wing Tai, Chi-Keung Tang, and Alan L Yuille. Adversarial attacks beyond the image space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4302–4311, 2019. 1

[28] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 4