Towards Interpreting Vulnerability of Object Detection Models via Adversarial Distillation

Yaoyuan Zhang[1], Yu-an Tan[2], Mingfeng Lu[3], Lu Liu[2], Quanxing Zhang[1], Yuanzhang Li[1], and Dianxin Wang[1(✉)]

[1] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
{yaoyuan,zhangqx,popular,dianxinw}@bit.edu.cn

[2] School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China
{tan2008,liulu}@bit.edu.cn

[3] School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China
lumingfeng@bit.edu.cn

Abstract. Recent works have shown that deep learning models are highly vulnerable to adversarial examples, limiting the application of deep learning in security-critical systems. This paper aims to interpret the vulnerability of deep learning models to adversarial examples. We propose adversarial distillation to illustrate that adversarial examples are generalizable data features. Deep learning models are vulnerable to adversarial examples because models do not learn this data distribution. More specifically, we obtain adversarial features by introducing a generation and extraction mechanism. The generation mechanism generates adversarial examples, which mislead the source model trained on the original clean samples. The extraction term removes the original features and selects valid and generalizable adversarial features. Valuable adversarial features guide the model to learn the data distribution of adversarial examples and realize the model's generalization on the adversarial dataset. Extensive experimental evaluations have proved the excellent generalization performance of the adversarial distillation model. Compared with the normally trained model, the mAP has increased by 2.17% on their respective test sets, while the mAP on the opponent's test set is very low. The experimental results further prove that adversarial examples are also generalizable data features, which obeys a different data distribution from the clean data. Understanding why deep learning models are not robust to adversarial samples is helpful to attain interpretable and robust deep learning models. Robust models are essential for users to trust models and interact with the models, which can promote the application of deep learning in security-sensitive systems.

Keywords: Adversarial examples · Interpretability · Object detection · Deep learning

J. Zhou et al. (Eds.): ACNS 2022 Workshops, LNCS 13285, pp. 53–65, 2022.
https://doi.org/10.1007/978-3-031-16815-4_4

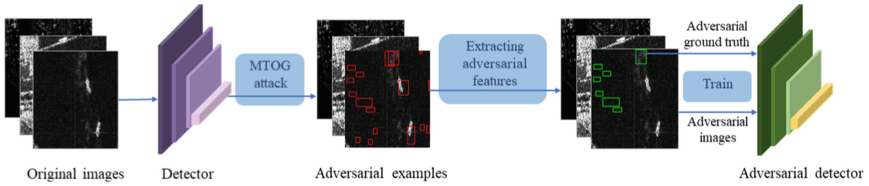

Fig. 1. A conceptual diagram of our framework. The paper adopts the MTOG attack to generate adversarial examples to make an object detector fabricate many bounding boxes. Adversarial features are extracted from these bounding boxes to construct a new dataset, called the adversarial dataset. By training an object detection model on the adversarial dataset, we obtain an adversarial distillation model.

1 Introduction

Recent works have shown that deep learning models are vulnerable to adversarial examples [1,27], which imperceptibly perturbed natural inputs to induce DNN models to make erroneous predictions. Previous work tried to explain this phenomenon from multiple perspectives, [2,24] interpret the existence of adversarial examples from the standpoint of theoretical models, and [8,18,25] focus on the demonstration based on high-dimensions quantities. However, these theories often fail to capture the behavior we observe in practice fully. More broadly, previous work in the field tends to treat adversarial examples as aberrations caused by the high dimensional nature of the input space or statistical fluctuations in the training data [8,10,27]. [13] propose a new perspective on adversarial examples. They demonstrate that adversarial examples are not bugs but features in image classification. Still, there are no explanations for adversarial examples in more complex computer vision tasks, such as object detection.

In this paper, we commit to interpreting the vulnerability of deep learning object detection models to adversarial examples, inspired by [13]. We illustrate that adversarial examples are classification features and localization features. Object detectors are vulnerable to adversarial examples because they do not learn the data distribution of adversarial examples. Object detectors tend to exploit any available features to localize the position of objects and classify them to a specific class, even those features that seem inexplicable to humans. We demonstrate that object detection models can learn valuable features on adversarial examples and be generalized to the whole data distribution, just like benign examples.

To corroborate our hypothesis, we propose adversarial distillation. Given an object detection model trained on the benign training set, we improve the TOG attack [6] to generate adversarial examples and design an extracting adversarial features module to construct an adversarial dataset. The inputs of this dataset are nearly identical to the originals, but all appear incorrectly localized and labeled. They are associated with their new ground truth (not the originals) only through small adversarial perturbations (and hence utilize only adversarial features). We train the adversarial distilled model on this adversarial dataset

and evaluate the performance of this model both on the adversarial test set and the original test set. Experimental results have shown that the adversarial distilled model yields well generalization despite the lack of predictive human-visible information, which indicates that adversarial examples are features satisfying a specific data distribution but different from the distribution of benign data. We consider one class object detection dataset because this type of dataset has a simple category, and the model trained on the dataset can better focus on localization features. We further choose the SAR ship detection dataset [29] as the original dataset and implement adversarial distillation on this dataset. In summary, we make the following contributions:

- We train an object detection model on the SAR ship dataset, which obtains a great mAP. Simultaneously, We craft adversarial examples to attack this model and effectively decrease the mAP.
- We establish experiments to illustrate that adversarial examples are not vulnerabilities but well-generalizable features satisfying a specific data distribution.

The rest of this paper is organized as follows. In Sect. 2, we briefly review the related backgrounds. We present the detail of the framework in Sect. 3. Section 4 reports all experimental results. Finally, we summarize the conclusion in Sect. 5.

2 Related Works

2.1 Interpretable Adversarial Examples

[13] propose a novel explanation for the existence of adversarial examples. The standard training method can learn both useful robust and non-robust features in their work. The non-robust features are beneficial to generalization but very sensitive, which makes classifiers vulnerable to adversarial examples. The sensitivity of non-robust features should be understood as their small changes will significantly change the model's predictions. The useful, robust feature is the common feature with certain interpretability, such as cat ears and cat tail in cat classification. When performing formal training based on robust features and non-robust features, respectively, classifiers can obtain good accuracy on the standard test set. Classifiers with different structures trained on different datasets of the same distribution may learn similar non-robust features, which makes the adversarial examples transferable. [13] validates the hypothesis by extracting the image classification dataset into roust features and non-robust features and use the Gaussian distribution as an example.

2.2 Object Detection

Object detection is one of significant computer vision tasks, which detects the class and location of objects in digital images [9,15,21]. In object detection, we take YOLOv3 [22] as an example. Given an input image x, the model first

generates a great number of S candidate bounding boxes $\hat{B}(x) = \{\hat{o}_1, ..., \hat{o}_S\}$ where $\hat{o}_i = (\hat{b}_i^x, \hat{b}_i^y, \hat{b}_i^w, \hat{b}_i^h, \hat{C}_i, \hat{p}_i)$ represents a candidate centered at coordinates $(\hat{b}_i^x, \hat{b}_i^y)$, and $(\hat{b}_i^w, \hat{b}_i^h)$ is width and height of the candidate. The objectness score $\hat{C}_i \in [0, 1]$ denotes whether the candidate contains an object, and a K-class probability vector $\hat{p}_i = (\hat{p}_i^1, \hat{p}_i^2, ..., \hat{p}_i^K)$ estimates the class of the corresponding candidate. The detection process usually divides the input x into grids with different scales, and each grid cell generates plenty of candidate bounding boxes based on the anchors and localizes the object centered at the cell. The candidates with low prediction confidence are excluded via applying confidence threshold, and those with high overlapping are removed by non-maximum suppression. The remaining candidates constitute the final detection result $\hat{O}(x)$.

For training an object detector, each object o_i in a training sample (x, O) is allocated to one of the S bounding boxes according to the center coordinates and the amount of overlapping with the anchors. $O = \{o_i | 1_i = 1, 1 \leq i \leq S\}$ is a set of objects in ground truth where $1_i = 1$ if the i-th bounding box is responsible for an object and 0 otherwise, $o_i = (b_i^x, b_i^y, b_i^w, b_i^h, p_i)$ with $p_i = (p_i^1, p_i^2, ..., p_i^K)$ and $p_i^c = 1$ if the class of o_i is c. Training a DNN model often begins with initializing the parameters of the model randomly and updating parameters slowly via taking the derivative of the loss function L concerning parameters θ on a mini-batch of input-output pairs $\{(x, O)\}$ with the following equation until convergence:

$$\theta_{t+1} = \theta_t - \alpha \nabla_{\theta_t} L(x, O; \theta), \tag{1}$$

where α is the learning rate. The loss function of a deep object detection network is divided into three parts, each part corresponds to describing the existence, locality, and category of a detected object. The objectness score $\hat{C}_i$ can be learned by minimizing the binary cross-entropy l_{BCE}:

$$\begin{aligned} L_{obj}(x, O; \theta) &= \sum_{S}^{i=1} [1_i l_{BCE}(1, \hat{C}_i)] \\ L_{noobj}(x, O; \theta) &= \sum_{S}^{i=1} [1 - 1_i l_{BCE}(0, \hat{C}_i)], \end{aligned} \tag{2}$$

The spatial locality is learned by minimizing the squared error l_{SE}:

$$\begin{aligned} L_{loc}(x, O; \theta) = \sum_{S}^{i=1} 1_i [&l_{SE}(x_i, \hat{x_i}) + l_{SE}(y_i, \hat{y_i}) \\ &+ l_{SE}(\sqrt{W_i}, \sqrt{\hat{W}_i}) + l_{SE}(\sqrt{W_i}, \sqrt{\hat{W}_i})] \end{aligned} \tag{3}$$

The K-class probabilities $\hat{p}_i$ is optimized by minimizing the binary cross-entropy:

$$L_{prob}(x, O; \theta) = \sum_{S}^{i=1} 1_i \sum_{c \in classes} l_{BCE}(p_i^c, \hat{p}_i^c) \tag{4}$$

Therefore, the deep object detection network can be optimized by the linear combination of the above loss functions:

$$\begin{aligned} L(x, O; \theta) = & L_{obj}(x, O; \theta) + \lambda_{noobj} L_{noobj}(x, O; \theta) \\ & + \lambda_{loc} L_{loc}(x, O; \theta) + L_{prob}(x, O; \theta), \end{aligned} \tag{5}$$

Synthetic Aperture Radar (SAR) [12,14] is a high-resolution imaging radar that can generate high-resolution two-dimensional images of range and azimuth via reflecting the emitted electromagnetic wave onto the target. For SAR can provide high-resolution images in all weather conditions, SAR images have been widely used for complex object detection and recognition tasks, such as ship object detection. With the widespread application of SAR in ship detection [28], some large-scale datasets have emerged, such as SSDD [14], OpenSARShip [12] and SAR ship dataset [29].

2.3 Adversarial Examples

Adversarial examples are first found in image classification task [23], an adversarial example x' is crafted by adding imperceptible perturbations to a clean input x, making the target model output incorrect predictions [4,10,16,17,19]. The process of generating an adversarial example can be defined as

$$min \parallel x' - x \parallel_p \quad s.t. \hat{O}(x') \neq \hat{O}(x), \tag{6}$$

where p represents the distance metric, which can be the L_0, L_2 and L_∞ norm.

Adversarial examples also exist in object detection task [3,26,30]. TOG attack is a family of adversarial attacks on object detection, including object-vanishing attack, object-fabrication attack, object-mislabeling attack and untargeted attack [6,7]. We take the untargeted attack as an example to introduce the TOG attack. TOG attack fixes the model parameters and initializes with a clean image (i.e., $x'_0 = x$), iteratively updating the adversarial example with the following equation:

$$\begin{aligned} L(x'_t, \hat{O}(x); \theta) = & L_{obj}(x'_t, \hat{O}(x); \theta) + L_{noobj}(x'_t, \hat{O}(x); \theta) \\ & + L_{loc}(x'_t, \hat{O}(x); \theta) + L_{prob}(x'_t, \hat{O}(x); \theta), \end{aligned} \tag{7}$$

$$x'_{t+1} = \Pi_{x,\epsilon}[x'_t + \alpha \Gamma(\nabla_{x'_t} L(x'_t, \hat{O}(x); \theta))] \tag{8}$$

where $\Pi_{x,\epsilon}[\cdot]$ is the projection onto a hypersphere with a radius ϵ centered at x in L_p norm, Γ is a sign function.

2.4 Distillation

[11] initially propose a distillation method to reduce a large model (the teacher) to a smaller distillation model (the student), thereby improving accuracy on the test set and speeding up the rate of the student predicting hard labels (ground truth). At a high level, the working principle of distillation can be summarized

into three steps: one is to train the teacher on the training set in a standard way. The second is to use the teacher to label each instance on the training set with soft labels (the output vector of the teacher). For example, the hard label on an image of a dog indicates that it is classified as a dog. At the same time, the soft label describes that it is a dog with 76% probability, a cat with 22% probability, and a cow with 0.2% probability. The third is to train the distillation model on the soft labels from the teacher instead of the hard labels from the training set. Distillation is exploited in multiple domains [5,20].

3 Methodology

3.1 Definitions

We consider object detection with one class ($K = 1$ in Sect. 2.2), where input-output pairs $(x, O) \in X \times \{(b_i^x, b_i^y, b_i^w, b_i^h, p_i^c)\}$ are sampled from a data distribution D. Following the definition of [13], we define a function f to represents an object detector. Additionally, we define f_l as a localization function and f_c as a classification function.

- $\gamma-$valuable localization features: For an input x, we call a localization feature f_l $\gamma-$valuable ($\gamma > 0$) if it is correlated with the ground-truth bounding boxes in expectation, that is if

$$E_{(x,O)\sim D}[B(x) \cdot f_l(x)] \geq \gamma, \tag{9}$$

  where $B(x) = \{(b_i^x, b_i^y, b_i^w, b_i^h)\}$.
- $\rho-$valuable classification features: For an input x, we call a classification feature f_c $\rho-$valuable ($\rho > 0$) if it is correlated with the ground-truth label in expectation, that is if

$$E_{(x,O)\sim D}[c \cdot f_c(x)] \geq \rho. \tag{10}$$

- Valuable adversarial features: When the input is x', we define $\gamma-$valuable adversarial localization feature and $\rho-$valuable adversarial classification feature satisfying Eq. 9 and Eq. 10, respectively.

3.2 Framework

In this work, we elaborate on adversarial distillation for interpreting that adversarial examples are the features satisfying a specific data distribution. A conceptual description of these experiments can be found in Fig. 1. We construct an adversarial dataset where the input-output association is based on valuable adversarial features. We show that this dataset suffices to train an object detector with good performance on the adversarial test set. Still, poor performance on the original test set results from the gap between original and adversarial distribution.

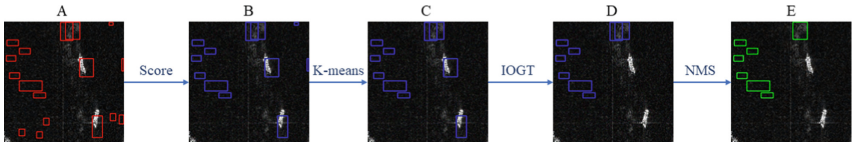

Fig. 2. An example in Sect. 3.3. A is the adversarial example generated by MTOG; B, C and D are the intermediate images selected after Score, K-means and IOGT, respectively; E is the final result selected after NMS.

3.3 Extracting Adversarial Features

We construct a dataset where the input-output association is based on valuable adversarial features, including localization and classification. To accomplish this, we modify each input-output pair (x, O) as follows. We integrate momentum into TOG attack [6] (MTOG) to generate the corresponding adversarial examples on original datasets so that the original object detector f can detect many objects which do not exist in human eyes. We then extract adversarial features via selecting these forged bounding boxes according to the following steps.

Given an adversarial example x', (1) Score: we discard bounding boxes with scores below the threshold to ensure adversarial classification features $\rho-$valuable; (2) K-means: we analyze the range of original ground-truth bounding boxes by k-means clustering algorithm and remove the bounding boxes that exceed this range to a certain threshold; (3) IOGT: we design the IOGT method to discard those bounding boxes intersecting with the original ground truth, which ensures that the selected bounding boxes do not include original localization features. IOGT can be formulated as

$$IOGT(B) = \frac{B \cap GT}{GT}, \tag{11}$$

where GT represents all original ground-truth bounding boxes $\{(b_i^x, b_i^y, b_i^w, b_i^h)\}$, B represents the candidate bounding box; (4) NMS: we exploit non-maximum suppression to ensure that the forged bounding boxes do not intersect, making the generated localization features not duplicated. The remaining bounding boxes are aligned as O' to form the new input-output pair (x', O'). Finally, the resulting input-output pairs make up the new dataset, named adversarial dataset. The whole process is described in Algorithm 1, and Fig. 2 shows an example of processing by the extracting adversarial feature module.

3.4 Adversarial Distillation

We elaborate on adversarial distillation for interpreting that adversarial examples are the features satisfying a specific data distribution. A conceptual description of these experiments can be found in Fig. 1. We first train an object detection model (the teacher) on the original dataset. Then, we use the MTOG attack to

Algorithm 1 Extracting adversarial features

Input: An object detector f; an adversarial example x' and ground truth O; Threshold for score, k-means, IOGT and NMS
Output: Adversarial ground truth O'

Input x' to f and obtain $\hat{B}(x') = \{\hat{o}_1, ..., \hat{o}_n\}$, $\hat{o}_i = (\hat{b}_i^x, \hat{b}_i^y, \hat{b}_i^w, \hat{b}_i^h, \hat{C}_i, \hat{p}_i)$;
$\text{temp}_1 = \text{temp}_2 = [\,]$
for $\hat{o}_i$ in $\hat{B}(x')$ **do**
  if $\hat{p}_i >$ score **then**
    continue
  end if
  if $(\hat{b}_i^w, \hat{b}_i^h)$ not in k-means **then**
    continue
  end if
  if IOGT$(\hat{b}_i^x, \hat{b}_i^y, \hat{b}_i^w, \hat{b}_i^h) \neq 0$ **then**
    continue
    Add $\hat{o}_i$ into temp1
  end if
end for
temp2 = NMS(temp1)
O' = temp2
return O'

generate adversarial examples against the teacher and obtain the corresponding outputs of the teacher. We next use Sect. 3.3 to craft adversarial ground truth, thus making the adversarial dataset. Finally, we train the distilled model (the student) on the adversarial training set from the teacher rather than on the original training set. We find that the distilled model performs well on the adversarial test set, which indicates that adversarial examples are features satisfying a specific data distribution. Meanwhile, the student performs poorly on the original test set, which indicates that the gap between the adversarial and original data distribution results in poor generalization.

4 Experiments

4.1 Setup

Datasets. We select SAR ship detection dataset consisting 43,819 ship chips [29]. We randomly allocate the training set, validation set, and test set according to the ratio of 7: 2: 1. Meanwhile, we do the same operation on the corresponding adversarial dataset.

MTOG Attack. The maximum perturbation ϵ is set to 8 with pixel value in [0,255]. The number of iterations T is 20, the step size is 2 and the decay factor μ is 1.0. We set the coefficient $\lambda = 0.2$ empirically in order to reduce the proportion of L_{noobj} in L.

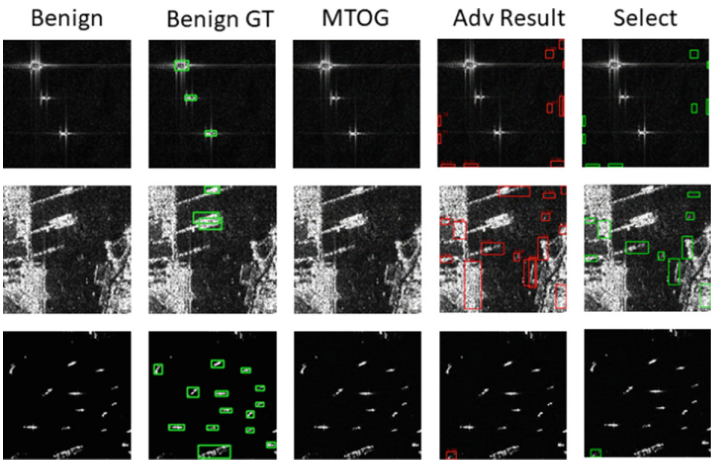

Fig. 3. Some random samples from the original SAR Ship dataset and the corresponding adversarial examples.

Extracting Adversarial Features. We set the threshold of the classification score of each candidate bounding box in the output of the model to 0.5 in order to ensure adversarial classification features $\rho-$valuable. We set the range of k-means clustering to [5, 104]. And the bounding box threshold of both IOU in NMS and IOGT is set to 0 to remove the intersecting bounding boxes.

Adversarial Distillation. We train three Yolov3-Mobilenet[1] models on the SAR ship detection dataset and its corresponding adversarial dataset, respectively. For each model, we divide the training process into two steps. At the first step, Adam optimization is used with a learning rate of 0.001 and a batch size of 16, and training epochs are 30. In the second step, the learning rate is 0.0001, the batch size is 16, and the training epochs are 20. After models are trained, we test these models on the original test set and adversarial test set. We evaluate the performance of models by mean Average Precision (mAP), and the threshold of IOU is set to 0.5. Our experiments are conducted on an Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10 GHz CPU, a GPU of NVIDIA GeForce RTX 2080 Ti with 11 GB, and 32 GB of memory.

4.2 Generating Adversarial Examples

Figure 3 shows test SAR images (left) with the detection results made by YOLOv3-Mobilenet on benign (the "Benign GT" column), the corresponding adversarial examples (the "MTOG" column) generated by MTOG attacks with the detection results made by YOLOv3-Mobilenet (the "Adv Result" column), and the adversarial example selected by Sect. 3.3 (the "SELECT" column).

[1] https://github.com/Adamdad/keras-YOLOv3-mobilenet.

Table 1. The average number of bounding boxes in extracting adversarial features module. "Ori" and "Adv" represent the average bounding boxes of original images and adversarial examples, respectively. "Score", "K-means", "IOGT", and "NMS" are defined in Sect. 3.3.

	Training	Validation	Test
Ori	1.363	1.347	1.352
Adv	13.393	13.359	13.520
Adv+Score	12.436	12.580	12.402
Adv+Score+K-means	10.840	10.954	10.789
Adv+Score+K-means+IOGT	10.343	10.452	10.289
Adv+Score+K-means+IOGT+NMS	8.528	8.557	8.490

Table 2. The mAP (%) of original object detector (Ori-model) and adversarial object detector (Adv-model) on the original test set (Ori-test) and adversarial test set (Adv-test), respectively.

	Ori-test	Adv-test	Adv-GT-test
Ori-model	86.34	83.45	0.12
Adv-model	1.19	88.51	–

From Fig. 3, we can observe that the MTOG attack fools the object detector to give many invisible objects (bounding boxes), most with high confidence and some with low confidence. After MTOG, the mAP of the detector drops to 0.12%, the results are shown in Table 2. However, some bounding boxes with natural objects still exist, some bounding boxes intersect together, and the aspect ratio of some bounding boxes does not match the k-means clustering result. After extracting adversarial features, we remove those bounding boxes with low confidence or intersect with ground truth or the aspect ratio not in k-means clustering. We keep the one with the highest confidence for the intersecting bounding boxes.

Table 1 shows the average number of bounding boxes after each step in Sect. 3.3. In Table 1, MTOG adversarial attack craft plenty of bounding boxes compared to original images. We follow the steps described in Sect. 3.3 to remove useless bounding boxes. We take the training set as an example. After an adversarial attack, the average number of bounding boxes increases from 1.363 to 13.393. After Score, it drops to 12.436. After K-means, it decreases by 1.596. Finally, the average number of bounding boxes is 8.528.

4.3 Evaluation on Adversarial Distillation

We report in Table 2 the mAP of original object detector (Ori-model) and adversarial object detector (Adv-model) on the original test set (Ori-test), adversarial test set (Adv-test) and the test set with adversarial examples and the ground truth (Adv-GT-test), respectively.

In the first column of Table 2, the data (86.34%) represents the result of training on the original dataset and evaluation on the original dataset. This data shows that YOLOv3-Mobilenet performs well on the SAR ship dataset. The data (1.19%) represents the result of training on the adversarial dataset and evaluation on the original dataset, which indicates that the gap between the adversarial and original data distribution results in poor generalization. The data (83.45%) in Table 2 is the mAP of original models evaluated on the adversarial dataset and 0.12% shows the MTOG attack successfully attack the original model. The data (88.51%) indicates that adversarial examples are features satisfying a specific data distribution, just like the original dataset. It further explains that adversarial examples are not bugs, but features, some of which are indeed valuable for localization and classification in object detection.

5 Conclusion

This paper proposes a new perspective on adversarial examples that are not aberrations but features satisfying a specific data distribution. In object detection, adversarial examples contain classification features and localization features. These features are helpful for models to generalize. We support this hypothesis by performing adversarial distillation, which constructs adversarial datasets on the teacher and trains the adversarial object detector on these datasets. We select ship detection in SAR images as an original dataset for its category is simple, and the model can better focus on localization features. We introduce the MTOG attack to generate adversarial examples to provide a basis for constructing an adversarial dataset. The experiment results show that adversarial examples are generalizable features that satisfy a specific data distribution. The model trained on the adversarial training set generalizes well on the adversarial test set. We hope that our findings can help researchers better understand the black-box deep learning models, thereby contributing to the deep development and extensive application of deep learning models.

Acknowledgements. This work was supported by the National Natural Science Foundation of China under Grant (No. 61876019, 62072037, U1936218).

References

1. Biggio, B., et al.: Evasion attacks against machine learning at test time. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) ECML PKDD 2013. LNCS (LNAI), vol. 8190, pp. 387–402. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40994-3_25
2. Bubeck, S., Lee, Y.T., Price, E., Razenshteyn, I.: Adversarial examples from computational constraints. In: International Conference on Machine Learning, pp. 831–840. PMLR (2019)
3. Carlini, N., Wagner, D.: Adversarial examples are not easily detected: bypassing ten detection methods. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 3–14 (2017)

4. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: IEEE Symposium on Security and Privacy, pp. 39–57 (2017)
5. Cheng, X., Rao, Z., Chen, Y., Zhang, Q.: Explaining knowledge distillation by quantifying the knowledge. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12925–12935 (2020)
6. Chow, K.H., Liu, L., Gursoy, M.E., Truex, S., Wei, W., Wu, Y.: TOG: targeted adversarial objectness gradient attacks on real-time object detection systems. arXiv preprint arXiv:2004.04320 (2020)
7. Chow, K.-H., Liu, L., Gursoy, M.E., Truex, S., Wei, W., Wu, Y.: Understanding object detection through an adversarial lens. In: Chen, L., Li, N., Liang, K., Schneider, S. (eds.) ESORICS 2020. LNCS, vol. 12309, pp. 460–481. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59013-0_23
8. Gilmer, J., et al.: Adversarial spheres. arXiv preprint arXiv:1801.02774 (2018)
9. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
10. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (2015)
11. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. Comput. Sci. **14**(7), 38–39 (2015)
12. Huang, L., et al.: OpenSARShip: a dataset dedicated to sentinel-1 ship interpretation. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. **11**(1), 195–208 (2017)
13. Ilyas, A., Santurkar, S., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. In: Annual Conference on Neural Information Processing Systems (2019)
14. Li, J., Qu, C., Shao, J.: Ship detection in SAR images based on an improved faster R-CNN. In: 2017 SAR in Big Data Era: Models, Methods and Applications (BIGSARDATA), pp. 1–6. IEEE (2017)
15. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
16. Liu, Y., Chen, X., Liu, C., Song, D.: Delving into transferable adversarial examples and black-box attacks. In: International Conference on Learning Representations (2017)
17. Mądry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. Stat **1050**, 9 (2017)
18. Mahloujifar, S., Diochnos, D.I., Mahmoody, M.: The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. In: AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 4536–4543 (2019)
19. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against deep learning systems using adversarial examples. arXiv preprint arXiv:1602.02697 (2016)
20. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 582–597. IEEE (2016)
21. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
22. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)

23. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
24. Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., Madry, A.: Adversarially robust generalization requires more data. Advances Neural Inf. Process. Syst. (2018)
25. Shafahi, A., Huang, W.R., Studer, C., Feizi, S., Goldstein, T.: Are adversarial examples inevitable? In: International Conference on Learning Representations (2018)
26. Song, D., et al.: Physical adversarial examples for object detectors. In: 12th USENIX Workshop on Offensive Technologies (WOOT 2018) (2018)
27. Szegedy, C., et al.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014)
28. Wang, Y., Wang, C., Zhang, H.: Combining a single shot multibox detector with transfer learning for ship detection using sentinel-1 SAR images. Remote Sens. Lett. **9**(8), 780–788 (2018)
29. Wang, Y., Wang, C., Zhang, H., Dong, Y., Wei, S.: A SAR dataset of ship detection for deep learning under complex backgrounds. Remote Sens. **11**(7), 765 (2019)
30. Wei, X., Liang, S., Chen, N., Cao, X.: Transferable adversarial attacks for image and video object detection. arXiv preprint arXiv:1811.12641 (2018)