Fooling Object Detectors: Adversarial Attacks by Half-Neighbor Masks

Yanghao Zhang[*]
University of Exeter
Exeter, EX4 4QF, UK
yanghao.zhang@exeter.ac.uk

Fu Wang[*]
Guilin Univ. of Electronic Technology
Guilin, Guangxi, 541004, China
fuu.wanng@gmail.com

Wenjie Ruan[†]
University of Exeter
Exeter, EX4 4QF, UK
w.ruan@exeter.ac.uk

ABSTRACT

Although there are a great number of adversarial attacks on deep learning based classifiers, how to attack object detection systems has been rarely studied. In this paper, we propose a Half-Neighbor Masked Projected Gradient Descent (HNM-PGD) based attack, which can generate strong perturbation to fool different kinds of detectors under strict constraints. We also applied the proposed HNM-PGD attack in the CIKM 2020 AnalytiCup Competition, which was ranked within the top 1% on the leaderboard. We release the code at https://github.com/YanghaoZYH/HNM-PGD.

CCS CONCEPTS

• **Computing methodologies** → *Neural networks*; • **Security and privacy** → *Software and application security.*

KEYWORDS

deep learning, object detector, adversarial attack, ℓ_0 constraint

1 INTRODUCTION

Object detection is one of the most fundamental computer vision tasks, which not only performs image classification [17, 19] but also identifies the locations of the objects in an image. Now object detection has been widely applied as an essential component in many applications that requires a high-level security, such as identity authentication [11], autonomous driving [5], and intrusion detection [6]. In recent years, we witness the significant progress has been made in object detection, especially by taking the advantage of deep learning models. However, deep learning based object detection systems are also demonstrated to be vulnerable to adversarial examples [6]. The adversarial example was first identified by Szegedy et al. [13], primarily on classification tasks, they showed that maliciously perturbed examples can fool a well-trained Deep Neural Network (DNN) to output wrong predictions. After that, a great number of methods have been proposed to generate adversarial examples [3, 18], notably such as First Gradient Sign Method [2] and Projected Gradient Descent (PGD) [7]. At the same time, some studies show that DNN based object detection models are also facing the same threat [4, 6, 16].

In this paper, we introduce an adversarial attack framework, called HNM-PGD, which can fool different types of object detectors under two strict constraints concurrently. Our method first identifies a mask that meets the constraints, and then generates an adversarial example by perturbing a specific area that is constrained by the mask. Adversarial examples generated in this way are guaranteed to satisfy the limitation in terms of the number of perturbed pixels and connectivity regions, while remaining a high efficiency. One key novelty in HNM-PGD lies on that it enables an automatic process without handcraft operation, which provides a practical solution for many real-world applications. As a by-product of this attack strategy, we found that some perturbations contain clear semantic information, which are rarely identified by previous studies and provide some insights regarding the internal mechanisms of object detectors.

2 BACKGROUND

2.1 Object Detection Models

Given an input example x, an object detector can described as $f(x) = z$, where z represents the output vector of the detector. Considering YOLOv4 [1] and Faster RCNN [8] as our target models, the information contained in z is slightly different, and as an adversary under white-box setting, our goal is to make target models fail to detect the objects in the given examples. Thus we focus on the target models' confidence about the existence of objects in x. For each pre-defined box, YOLOv4 directly outputs its confidence $z^{\text{conf}} : \mathbb{R}$ about there is an object inside this box. If z^{conf} is above the given threshold, YOLOv4 model views this box as a potential object container, i.e. the area that may include objects. Faster RCNN does not output z^{conf}, nevertheless, it introduces an extra background class and make predictions based on its classification result $z^{\text{cls}} : \mathbb{R}^{C+1}$, where C is the number of classes. Suppose z_i^{cls} is the maximal item in z^{cls}, if z_i^{cls} is greater than a given threshold and $i \neq C + 1$, then the corresponding box will be viewed as the potential container.

2.2 Constraints of Perturbation

In this paper, the restrictions of the adversary are *i)* the number of perturbed pixels is not more than 2% of the whole; *ii)* the number of 8-connectivity regions is not greater than 10. Except for these two constraints, there are no limitations on the magnitude of the adversarial perturbation. Because both constraints are related to the number of pixels, this belongs to the ℓ_0 norm attack.

[*]Both authors contributed equally to this research. This work is done when Fu Wang was visiting the Trustworthy AI Lab at University of Exeter.
[†]Corresponding author. This work is supported by Partnership Resource Fund (PRF) on Towards the Accountable and Explainable Learning-enabled Autonomous Robotic Systems from UK EPSRC project on Offshore Robotics for Certification of Assets (ORCA) [EP/R026173/1], and the UK Dstl project on Test Coverage Metrics for Artificial Intelligence.

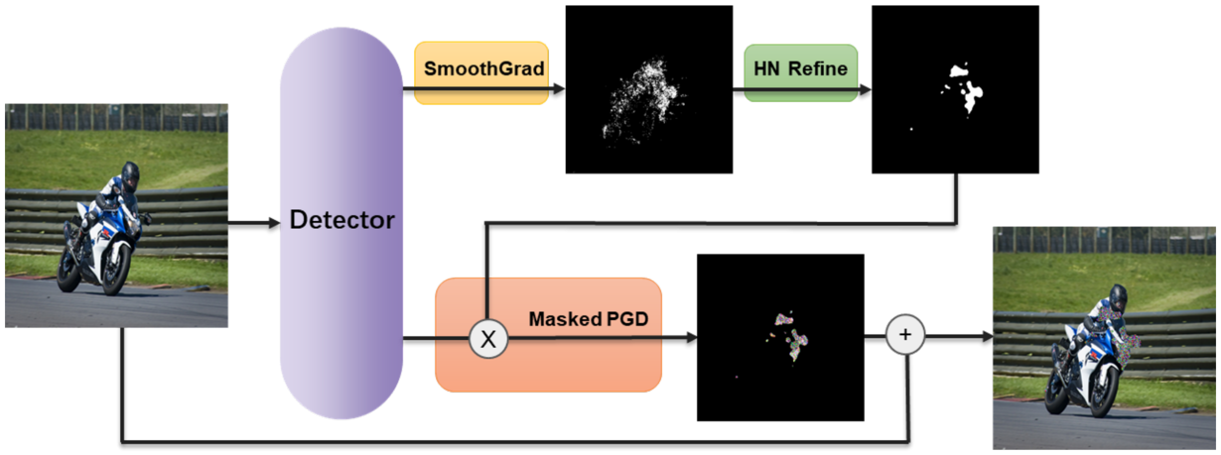

Figure 1: An illustration for the workflow of the proposed HNM-PGD.

Algorithm 1 Half-Neighbor Masked PGD (HNM-PGD)

```
Input: A given example $x$, number of random initialization $n$, control parameter $\phi$, HN kernel size $k$ and adjust step $s$, number of PGD iterations $P$, PGD step size $\alpha$
Output: $\delta$
1: $S_x = \frac{1}{n}\sum_{i=1}^{n} \nabla_x L\left(f(x+\eta_i)\right)$
2: repeat
3:     $z_{\text{resp}} = \text{Mean}(S_x) + \phi\,\text{Std}(S_x)$
4:     Initialize mask $M$ via $z_{\text{resp}}$
5:     while $k > 3$ do
6:         $M = \text{HN}(M, k)$
7:         $M = \text{HN}(M, 3)$
8:         $k = k - s$
9:     end while
10:    $\phi = \phi + 0.1$
11: until $M$ meets constraints
12: Random initialize $\delta$
13: $\delta = \delta \times M$                    ▷ ×: element-wise product
14: for $i = 1 \ldots P$ do
15:     $\delta = \delta + \alpha \cdot \text{sign}\left(\nabla_\delta L\left(f(x+\delta)\right)\right)$
16:     $\delta = \delta \times M$
17:     $\delta = \max(\min(\delta,\ 0-x),\ 1-x)$
18: end for
```

2.3 Salience Map

Salience map is a common tool to analyze and interpret DNN models' behaviors. Smilkov et al. [12] proposed SmoothGrad method to generate stable salience maps. Given a loss function L, the salience map is given by

$$S_x = \frac{1}{n}\sum_{i=1}^{n} \nabla_x L\left(f(x+\eta_i)\right), \tag{1}$$

where η_i are white noise vectors that sampled i.i.d. from a Gaussian distribution.

3 METHODOLOGY

3.1 Mask Finding

We first propose a mask generation method to locate perturbation regions for any given examples. Apparently, the size and shape of perturbation regions are critical to conduct a successful adversarial attack under the constraints. Therefore, we use salience map to capture the model's response toward each pixel in x at beginning. After compute an example's salience map, we initialize the mask via only keep pixels that the model's respond is larger than a threshold z_{resp}. To automatically carry out this initialization, we borrow the idea of standard deviation and coverage from Gaussian distribution, and compute z_{resp} via the mean and standard deviation of S_x, which can be described as $z_{\text{resp}} = \text{Mean}(S_x) + \phi\,\text{Std}(S_x)$, where ϕ is a control parameter.

To meet the pixel constraints, we follow the spirit of K Nearest Neighbor algorithm to refine the mask. Specifically, if half of a pixel's neighbors have been chosen by the current mask, then this pixel would also be chosen, otherwise it will be discarded. We employ two convolution kernels whose parameters are all 1 to conduct this Half-Neighbor (HN) procedure. The first kernel aims to reduce the number of pixels in a mask, and its size is gradually changed during iterations. The second kernel is fixed to 3×3, it can guarantee that there are no isolated pixels in the mask and reduce the number of connectivity regions (See lines 5–9 in Algorithm 1). If the mask still does not meet the constraints, the algorithm will adjust ϕ accordingly and search again.

3.2 Masked PGD Attack

Once the perturbation regions are located, we generate adversarial examples via PGD iterations. The basic idea here is summarized in Algorithm 1, where the selected regions are perturbed by a PGD adversary via conducting element-wise products between perturbation δ and mask M. The workflow of the proposed defense is shown in Fig. 1.

Due to the difference in the object detectors' output z, we need to consider YOLOv4 and Faster RCNN separately. As we discussed in section 2.1, YOLOv4 directly outputs its confidence, so Binary

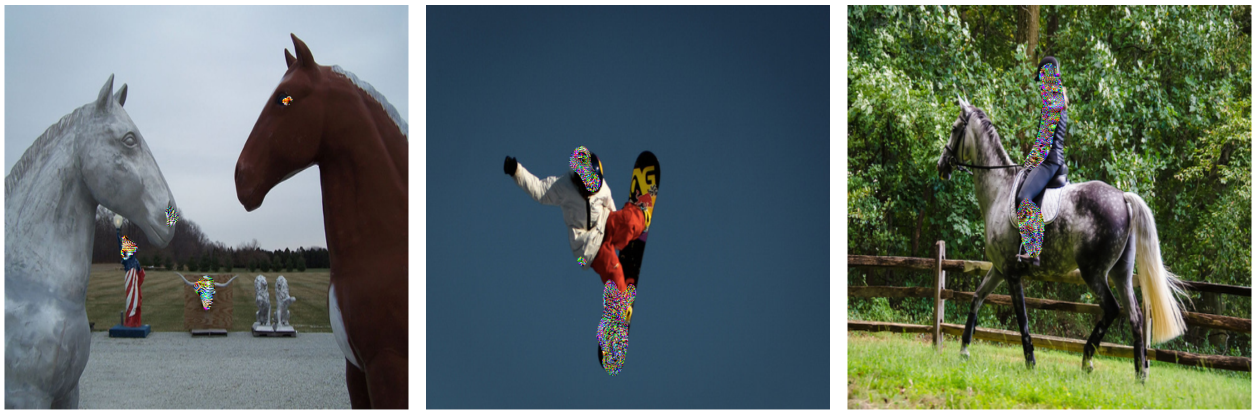
(a) (b) (c)

Figure 2: Adversarial patches examples generated by the proposed HNM-PGD, the generated patches are mostly located in the semantic part of the object, such as the horse's eye, skateboard and human's body.

Cross-Entropy (BCE) loss is a suitable option to conduct adversarial attack. Suppose there are m pre-defined boxes, and the maximal confidence is 1, BCE loss can be simplified as

$$L_{yolo}(z) = \sum_{i=1}^{m} \log z_i^{\text{conf}}, \tag{2}$$

where z is the output of a detector and $z_i^{\text{conf}} \in z$.

Different from YOLOv4, there are $C+1$ classes in Faster RCNN's classification result, including C foreground objects and 1 background class. To force the detector to classify an adversarial example into the background class, we conduct a targeted adversarial attack with a negative Cross Entropy (CE) loss, which can be written as

$$L_{frcnn}(z) = z_{C+1}^{\text{cls}} - \log\left(\sum_j \exp(z_j^{\text{cls}})\right), \tag{3}$$

where z^{cls} is the classification output of Faster RCNN detector, and $z^{\text{cls}} \in z$. Note that we can attack YOLOv4 and Faster RCNN simultaneously by simply using HN masked PGD mixmize $L_{yolo} + L_{frcnn}$.

4 EXPERIMENTS

To demonstrate our method, we select 100 images from MS COCO dataset as a toy dataset and conduct experiments for comparison on two white-box models, i.e. YOLOv4 and Faster RCNN.

4.1 Implementation Details

YOLOv4 Regarding the provided model YOLOv4, the input size is set to 608×608 while the original image has 500×500, to allow differential, we employ the function **torch.nn.Upsample** with bilinear interpolation to resolve the resize problem. Due to the approximation computation of **torch.nn.Upsample**, we need to allow more boxes to be detected to stabilize the adversarial perturbation. As YOLOv4 method only outputs the foreground objects with $z^{\text{conf}} > 0.5$, we adjust the confidence threshold from 0.5 to 0.3 during attack.

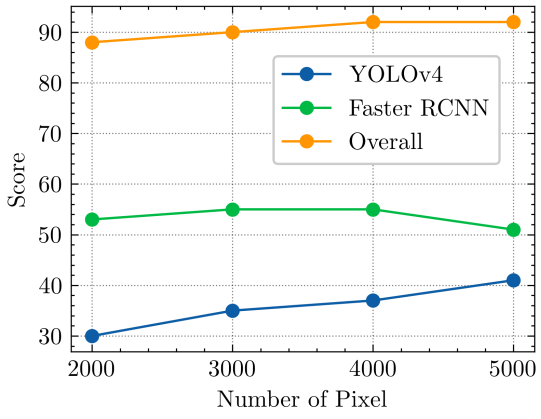

Figure 3: Performance on the toy dataset with the increasing amount of pixel.

Faster RCNN Similar to the configuration of YOLOv4, we resize the input to 800×800 with bilinear interpolation. As the permitted threshold for Faster RCNN is 0.3, which is relatively lower than YOLOv4. In practice, we assign a smaller threshold 0.1 when calculating the loss to enable more boxes to appear.

PGD Settings In this paper, the HNM-PGD is carried out with 40 steps, and the step size is 16/255.

4.2 Experimental Results

In this part, we employ the formula in the description of AnalytiCup to calculate the score, which provides a criteria to evaluate the performance of the proposed method. Our code is available on Github[1].

We produce 100 adversarial examples on the white-box models with two loss together: $L_{yolo} + L_{frcnn}$. Figure 2 gives several successful examples for the targeted models. We can observe that the

[1]https://github.com/YanghaoZYH/HNM-PGD

proposed method does locate the object correctly, and the added patches normally target on their sensitive parts. Figure 3 illustrates the overall score with the increasing upper bound of the number of pixel among 100 selected images, and the scores achieved by YOLOv4 and Faster RCNN, respectively. We find that with the increase of the quantity of pixel, Faster RCNN performs better, while this is not the case for YOLOv4. This is because the provided white-box Faster RCNN uses a low tolerate threshold, where sufficient pixel is needed for successful attack. In terms of YOLOv4, the performance fluctuates at about 53. Therefore, there is a trade-off when choosing the amount of the pixel.

We apply the same strategy and perform the adversarial attack with more steps (800) and smaller step size (4/255) for all 1000 images under the different quantity of pixel, then we pick the best result obtained on the white-box models as our solution. In the final stage of the AnalytiCup competition, we had also tried to improve the generalization of the attacking approach on the unseen model, i.e. black-box. In detail, we add some transformations (like flipping/cropping) to the input image, which is expected to not overfit the known white-box models too much. Our final score is 2414.87, ranked 17 in the competition.

5 BEYOND COMPETITION

This competition leads to a few interesting research directions. Intuitively, due to the ℓ_0 norm constraint, both location and shape of the perturbation are critical to the attacking performance. We have reviewed other top contestants' solutions and found that linear adversarial patches have a higher impact on the target model's output and use less number of pixels than blocky ones, while location is less important. This seems because blocky perturbation can only influence a relatively small range of a convolution kernel's output, while linear perturbation can cross a wider area. To verify such conjecture, we wish to adopt verification technologies on neural networks [9, 10, 15] into the object detectors and quantify the worst-case scenario of adversarial patches on object detectors. Besides, on the top of our HNM-PGD, we can also expand evaluation of existing adversarial attacks and defenses, such as [14, 18], onto object detection tasks.

6 CONCLUSION

In conclusion, we propose a PGD-based approach to attack object detectors using Half-Neighbor masks. In the proposed HNM-PGD, the automatic pipeline allows it to craft adversarial examples/patches automatically under ℓ_0 constraint, which can be applied in many applications, even physical-world attacks. On the other hand, this end-to-end attack framework also benefits further studies on defending object detectors against adversarial attacks and verifying their robustness.

ACKNOWLEDGMENTS

The work was partially supported by the Guangxi Science and Technology Plan Project under Grant AD18281065, the Guangxi Key Laboratory of Cryptography and Information Security under Grant GCIS201817. Fu Wang is supported by the study abroad program for graduate student of Guilin University of Electronic Technology under Grant GDYX2019025, and the Innovation Project of GUET Graduate Education under Grant 2020YCXS042.

REFERENCES

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934

[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*.

[3] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, and et al. 2020. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review* 37 (2020), 100270.

[4] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. 2020. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review* 37 (2020), 100270.

[5] Yann LeCun, Urs Muller, Jan Ben, Eric Cosatto, and Beat Flepp. 2005. Off-Road Obstacle Avoidance through End-to-End Learning. In *the Advances in Neural Information Processing Systems (NeurIPS)*.

[6] Jiajun Lu, Hussein Sibai, and Evan Fabry. 2017. Adversarial Examples that Fool Detectors. arXiv:1712.02494

[7] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR)*.

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems (NeurIPS)*.

[9] Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. 2018. Reachability Analysis of Deep Neural Networks with Provable Guarantees. In *In Proceedings of theInternational Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13-19 July*.

[10] Wenjie Ruan, Min Wu, Youcheng Sun, Xiaowei Huang, Daniel Kroening, and Marta Kwiatkowska. 2019. Global Robustness Evaluation of Deep Neural Networks with Provable Guarantees for the Hamming Distance. In *In Proceedings of the International Joint Conference on Artificial Intelligence, Macao, China, 10-16 August*.

[11] 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 *ACM Conference on Computer and Communications Security (SIGSAC)*.

[12] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. SmoothGrad: removing noise by adding noise. arXiv:1706.03825

[13] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*.

[14] F. Wang, L. He, W. Liu, and Y. Zheng. 2020. Harden Deep Convolutional Classifiers via K-Means Reconstruction. *IEEE Access* 8 (2020), 168210–168218.

[15] Min Wu, Matthew Wicker, Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. 2020. A game-based approximate verification of deep neural networks with provable guarantees. *Theoretical Computer Science* 807 (2020), 298–329.

[16] Bin Yan, Dong Wang, Huchuan Lu, and Xiaoyun Yang. 2020. Cooling-Shrinking Attack: Blinding the tracker with imperceptible noises. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[17] Shaoning Zeng, Bob Zhang, Yanghao Zhang, and Jianping Gou. 2018. Collaboratively weighting deep and classic representation via l2 regularization for image classification. *arXiv preprint arXiv:1802.07589* (2018).

[18] Yanghao Zhang, Wenjie Ruan, Fu Wang, and Xiaowei Huang. 2020. Generalizing Universal Adversarial Attacks Beyond Additive Perturbations. arXiv:2010.07788

[19] Yanghao Zhang, Shaoning Zeng, Wei Zeng, and Jianping Gou. 2018. GNN-CRC: discriminative collaborative representation-based classification via Gabor wavelet transformation and nearest neighbor. *Journal of Shanghai Jiaotong University (Science)* 23, 5 (2018), 657–665.