HIJACKING TRACKER: A POWERFUL ADVERSARIAL ATTACK ON VISUAL TRACKING

*Xiyu Yan[*1], Xuesong Chen[*2], Yong Jiang[†1,3], Shu-Tao Xia[1,3], Yong Zhao[2], Feng Zheng[4]*

[1]Tsinghua University, Dept. of Computer Science and Technology, China
[2]Shenzhen Graduate School of Peking University, Sch. of Electronic and Computer Engineering, China
[3]Peng Cheng Laboratory, PCL Research Center of Networks and Communications, Shenzhen, China
[4]Southern University of Science and Technology, Dept. of Computer Science and Engineering, Shenzhen

ABSTRACT

Visual object tracking has made important breakthroughs with the assistance of deep learning models. Unfortunately, recent research has clearly proved that deep learning models are vulnerable to malicious adversarial attacks, which mislead the models making wrong decisions by perturbing the input image. The threat to the models alerts us to pay attention to the model security of deep learning-based tracking algorithms. Therefore, we study the adversarial attacks against advanced trackers based on deep learning to better identify the vulnerability of tracking algorithms. In this paper, we propose to add slight adversarial perturbations to the input image by an inconspicuous but powerful attack strategy—hijacking algorithm. Specifically, the hijacking strategy misleads trackers in two aspects: one is shape hijacking that changes the shape of the model output; the other is position hijacking that gradually pushes the output to any position in the image frame. Besides, we further propose an adaptive optimization approach to integrate two hijacking mechanisms efficiently. Eventually, the hijacking algorithm results in fooling the tracker to track the wrong target gradually. The experimental results demonstrate the powerful attack ability of our method—quickly hijacking state-of-the-art trackers and reducing the accuracy of these models by more than 90% on OTB2015.

Index Terms— Hijacking, visual tracking, adversarial attack

1. INTRODUCTION

Visual object tracking has achieved important breakthrough with the assistance of recent advances in Convolutional Neural Networks (CNNs) [1, 2, 3]. For example, as the most advanced representative of Siamese network-based trackers, SiamRPN++ [4] perfectly combinate the Siamese network and deep learning model and achieving the best results on five large tracking datasets OTB2015 [5], VOT2018 [6], UAV123 [7], LaSOT [8], and TrackingNet [9].

Unfortunately, some recent researches have found that deep learning models are vulnerable to malicious adversarial attacks which are designed to mislead models with a very small perturbation [10, 11, 12]. For example, adding small perturbations on the input images of CNNs can mislead classifiers to make wrong judgments [13, 14, 15, 16]. Moreover, most recent studies manage to generate "adversarial patches" to targets that could be used to fool detectors to hide them [17, 18, 19, 20, 21]. Actually, the tracking task can be decomposed into a classification task and an estimation task [22]. Specifically, the classification subtask provides a coarse location of the target in the image by categorizing image regions into the foreground and background. The estimation subtask is then to estimate the fine state of the target, which represented by a bounding box like detection. Thus, these threats to the related models of classification and detection remind us that we should also focus on the model security of tracking algorithms which plays a significant role in the intelligent security system. To the best of our knowledge, there has been limited study on attacking against visual object tracking. [23] made the GOTURN tracker [24] consistently mistrack the people by employing a TV to display generated physical adversarial textures. Different from this work, we pay more attention to potential malicious attacks. For example, adding imperceptible perturbation on the input of tracking algorithms, which are used in surveillance systems, can make a serious risk of wrongly tracing suspicious persons. Therefore, we argue that the study of imperceptible adversarial attacks against advanced trackers deserves more efforts to further improve their security and robustness.

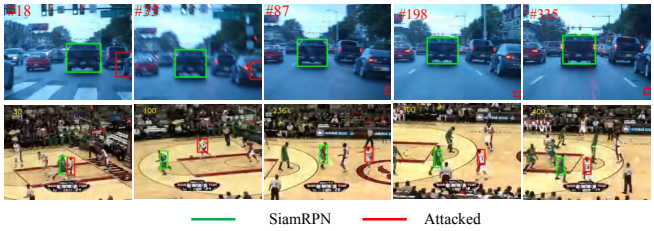

Fig. 1. Our hijacking makes the tracker—SiamRPN output incorrect bounding boxes. For the first line, our method successfully paralyzes the tracker. And the second line presents that the tracker was fooled to track a wrong target by our hijacking attack.

In this paper, we present an effective attack method to fool state-of-the-art trackers by adding adversarial perturbations on a series of consecutive frames. Specifically, we first present a hijacking algorithm against state-of-the-art trackers that gradually misleads a tracker to change the size of the prediction bounding box by minimizing the shape loss and pull the output to a wrong position by minimizing the location losses. Next, we propose an adaptive optimization strategy to jointly optimize these two losses more efficiently. With the integrated hijacking ability of shape and location, our method can fool the tracker to track the wrong target by stealth.

Eventually, we implement the hijacking attack on three state-of-the-art trackers belong to two types, including RT-MDNet [25] that represents the type of trackers with the model update, SiamRPN [2] and SiamRPN++ [4] which represent the type of trackers without a model update. The experimental results demonstrate the powerful

* Xiyu Yan and Xuesong Chen have equal contributions and they completed this work during their visit to Feng Zheng Lab in SUSTech.

† Corresponding author.

attack capabilities of our method to efficiently fool these advanced trackers. On the one hand, the hijacking fools the tracker to track the wrong object by stealth. On the other hand, the trackers hijacked could completely paralyze—the tracking accuracy on three trackers all reduced more than 90% on standard benchmark OTB2015 [5]. Fig. 1 shows the result example of successful hijacking.

In summary, the key contributions of this paper are as follows.

- We highlight the risks of adversarial attacks on tracking algorithms and propose a powerful hijacking algorithm to attack the advanced trackers by generating slight perturbation on video frames.
- We propose a novel adaptive objective optimization method to further improve the efficiency of our method.
- The experimental results prove that the proposed hijacking method efficiently and significantly reduce state-of-the-art trackers on a standard benchmark.

2. ATTACK METHODOLOGY

In this section, we first analyze the challenges of adversarial attacks on tracking. Then we introduce the proposed hijacking algorithm.

2.1. Analyses of Attack on Tracking

Given a target in any starting frame in a video, single object tracking algorithms are required to continuously predict the location and shape of the target in the subsequent frames. Here the tracking task can be particularly considered as a combination of a classification task and an estimation task. Specifically, location prediction of the target is provided by categorizing candidate boxes for foreground and background while the shape prediction depends on the estimation task by estimating the target state which often represented by a bounding box [22]. For example, the most advanced tracker SiamRPN [2] which is the representative of powerful Siamese families, quickly complete the tracking task through off-line training of two branches—classification branch and a regression branch. First, the classification branch completes the search and matching of the target from candidate boxes. Then the regression branch can accurately provide the location and shape of the target by region proposal network (RPN). Similarly, RT-MDNet [3] obtains a large number of candidate boxes with different states through intensive sampling and then completes classification of candidate boxes through online learning.

Based on such a tracking task, we analyze the adversarial attack against it. Although there are various methods proposed to attack the current classification and detection models, these methods are not suitable for tracking problem since it is a one-shot online matching problem based on video sequences. First, the characteristic of one-shot given the object of tracking in a video makes it impossible to train an adversarial patch off-line to paste into each frame like the attack on detection [17]. Thus, we should generate an adversarial perturbation frame by frame, which brings us an efficiency challenge. Second, unlike the attack on classification which merely needs to mislead the classifier given an incorrect class for an input image, the condition of a successful attack is vaguer.

Based on the analyses of the challenges for attack tracking, we are required to attack the tracker with few numbers of frames until it no longer tracks the target. Therefore, our attack is aimed at two aspects of the hijacking: location and shape of the output. To this end, our attack algorithm chooses a hijacking direction to precisely offset the location of the prediction box in the opposite direction of the object's movement. In order to make the tracker not reactivate when the target close to the output box, we also adopt the strategy of reducing shape to directly scales down the prediction box.

2.2. Hijacking Algorithm

Below we detail the proposed hijacking algorithm (see Fig. 2). Given a video sequence, our goal is to efficiently hijack bounding box from the target to a distracter $D(L_d, S_d)$ by adding adversarial perturbations on serials of frames, where L_d and S_d denote the location and shape of the distracter D respectively.

Specifically, for a video sequence, we set the hijacking direction $\vec{d}$ (towards L_d) and shape change vector $\vec{s}$ (towards S_d) in the first few frames and starting attack until stop conditions are met, after which the subsequent sequence is no longer attacked. For each attacked frame, we manage to make the predicted box take a small step toward the hijacking direction meanwhile gradually changing the shape of it. For it, we perform optimization iterations to effectively generate adversarial perturbations in a White-box manner, which means our attacker can access the paraments and outputs of the victim model.

2.2.1. Loss Functions

Given the hijacking target, we obtain an video sequence $\boldsymbol{X} = [..., \boldsymbol{x}_r, \boldsymbol{x}_{r+1}, ..., \boldsymbol{x}_s, ...]$. In these attacked frames, our objective is to generate adversarial perturbation $\Delta\boldsymbol{X} = [\Delta\boldsymbol{x}_r, \Delta\boldsymbol{x}_{\boldsymbol{x}+1}, ..., \Delta\boldsymbol{x}_s]$ on image patch subject to max perturbations range of pixel value ε.

In each attacked frame, we perform τ times iterations of the optimization objective. Corresponding to the proposed strategies, our hijacking loss function consists of two portions, namely location loss, and shape loss. Specifically, We denoted the image search region which contains T candidate boxes as $\boldsymbol{x}$. The goal is to obtain an adversarial perturbation $\Delta\boldsymbol{x}$ after iteration τ in this frame, which makes the prediction box of the tracker having a location offset along $\vec{d}$ and the shape reduction towards $\vec{s}$. For that, we first rank all T candidates by the confidence before iteration. Then, we can obtain the top 5 confidence indexes set I_c, the top 5 location indexes set I_{loc} which satisfy the requirements of $\vec{d}$ and the top 5 shape indexes set I_{shp} which meet requirements of $\vec{s}$. Thus, location loss is defined as follows:

$$\begin{aligned} \mathcal{L}_{loc} = &\sum_{i \in I_c} \{S_i(\boldsymbol{x} + \Delta\boldsymbol{x})\} - \sum_{i \in I_{loc}} \{S_i(\boldsymbol{x} + \Delta\boldsymbol{x})\}, \\ &s.t. \quad \max(\Delta\boldsymbol{x}) \leq \varepsilon. \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ and $\Delta\boldsymbol{x}$ stand for the clean image and the adversarial perturbation, $S_i(\boldsymbol{x} + \Delta\boldsymbol{x})$ denotes the perturbed confidence score of the i candidate. The purpose of the location loss is to suppress the highest confidence candidates that accurately estimate the target state while stimulating the candidates with the direction of $\vec{d}$. Besides, shape loss is defined as follows:

$$\begin{aligned} \mathcal{L}_{shp} = &\sum_{i \in I_c} \{S_i(\boldsymbol{x} + \Delta\boldsymbol{x})\} - \sum_{i \in I_{shp}} \{S_i(\boldsymbol{x} + \triangle\boldsymbol{x})\}, \\ &s.t. \quad \max(\Delta\boldsymbol{x}) \leq \varepsilon. \end{aligned} \tag{2}$$

2.2.2. Adaptive Optimization

Given the losses above, our objective is to jointly optimize $\mathcal{L}_{loc}$ and $\mathcal{L}_{shp}$. A simple joint optimization approach is adding them up and employing the standard gradient descent to optimize. However, this

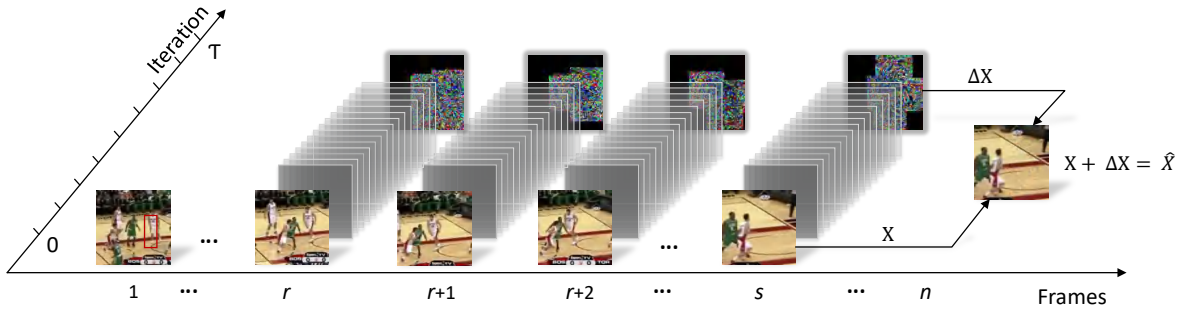

Fig. 2. Hijacking process of a video. In a video sequence, the hijacking occur on frame r to s. And in each attacked frame, an adversarial perturbation $\Delta \boldsymbol{x}$ on a clean image patch $\boldsymbol{x}$ is generated by adaptive optimizing loss through τ iterations. The adversary $\hat{\boldsymbol{x}}$ is the sum of $\boldsymbol{x}$ and $\Delta \boldsymbol{x}$ ($\hat{\boldsymbol{x}} = \boldsymbol{x} + \Delta \boldsymbol{x}$). The perturbation patch here is $15{\cdot}\Delta \boldsymbol{x}$ to show a more obvious effect.

method may result in a slow convergence rate because the selected candidate boxes are from different sets, which may suffer gradient conflicts during the optimization process.

To alleviate this problem, we develop a more sophisticated optimization strategy—an adaptive optimization method. First, we divide the optimization of each frame into two stages: joint optimization and individual optimization. For the former, we equally optimize the two losses by n (n = 5) times. Then, we can obtain the gradient changes of the two losses from iteration 1 to n. For the latter, we only optimize one loss which has a smaller gradient change. Our motivation is to adaptively find the best optimization direction for each frame through two-stage dynamic optimization, which is inspired by the fact that the second-order derivative methods can find precise directions in the optimization process. Therefore, we first approximate the second-order derivative to find a harder loss by evaluating the gradient change. Then we can make individual optimization for this difficult loss by $\tau - n$ times, namely individual optimization.

Lastly, our total objective is optimized by the proposed adaptive optimization as follow:

$$\min_{\Delta \boldsymbol{x}} \mathcal{L}_{loc} + \min_{\Delta \boldsymbol{x}} \mathcal{L}_{shp}. \tag{3}$$

3. ATTACK EVALUATION

We evaluate the results of the proposed hijacking algorithm against several state-of-the-art trackers, including RT-MDNet [25], SiamRPN [2], SiameseRPN++ [4].

3.1. Experiment Methodology

3.1.1. Dataset Chosen

Our attack approach against tracking algorithms is implemented on a standard benchmark of tracking—OTB2015 [5], which contains 100 video sequences, including long-term and short-term tracking. Specifically, the frame lengths of these 100 videos range from dozens to 3000+ frames, with a total of 58897 frames. The accuracy evaluation is based on two metrics: precision and success. The precision shows the percentage of frames within 20 pixels of the difference between the tracking result and the target. The success shows the ratio of successful frames when the IoU over the threshold 0.5.

3.1.2. Evaluation Metrics

We introduce the employed evaluation metrics to measure the adversarial attack performance on visual single object tracking —power and efficiency.

Power. To judge the power or ability of an attack, we apply the evaluation metrics of the OTB2015 benchmark introduced above—precision and success, which are just the opposite of the attack strength.

Efficiency. To measure the efficiency of the proposed method, we introduce a new "Min-Frames" parameter, which represents the minimum number of frames of successfully hijacked a video—poll the predicted bounding box to the frame edge and reduce its the shape to the set threshold. Specifically, for location hijacking, the maximum hijacking distance is from the target location to the edge of an image. A farther distance than it means that the target is out of our sight. For shape hijacking, the "Min-Frames" also represents the maximum number that an algorithm requires to perform the shape hijacking that reduces the shape of the bounding box to one fixed threshold(such as 20 pixels). Therefore, this indicator can evaluate the hijacking efficiency of the attacker. The smaller the number of "Min-Frames", the stronger the ability of the attacker to quickly hijack the output to a specified location and the desired scale.

3.1.3. Implementation Details

Our attack evaluation is implemented using Pytorch on a PC with an Intel i7, NVIDIA Tesla P40. For each attack video, we use Adam optimizer [26] to optimize the generated adversarial perturbation that inited by the same strategy as [11], with a learning rate of 0.02. The number of iterations in each attacked frame τ is set uniformly to 20 and permitted maximum perturbation ε is 16. In all experiments, we report the average result over 5 runs.

3.2. Attack Results for power and efficiency

Table 1 shows our results on the classic RT-MDNet, SiamRPN, and the most advanced SiamRPN++. As we can see, our method of hijacking paralyzes all three models successfully, almost reducing their success rate and precision to zero. This means that after completing an attack on part frames of the video, the trackers almost lose their ability to track the target. Meanwhile, the Min-frames are small, which means our method can complete a hijacking quickly (4s for 120 frames).

Table 1. The attack results against trackers on the OTB2015 in terms of success and precision.

Trackers	Precision (%)		Success (%)		Avg Min-Frames
	Org	Attacked	Org	Attacked	
SiamRPN	87.6	8.2	66.6	6.0	128
SiamRPN++(M)	86.4	7.1	65.8	5.6	194
RT-MDNet	65.2	8.6	64.6	6.8	20

Table 2. Tracking results in comparison with location loss, shape loss, and the adaptive optimization method.

SiamRPN	Precision	Success Rate	Avg Min-Frames
$\mathcal{L}_{loc}$ only	35.5%	25.6%	286
$\mathcal{L}_{shp}$ only	50.8%	39.5%	175
General opt	19.5%	12.4%	506
Adp opt	7.1%	5.6%	128

Specifically, for the online update model represented by RT-MDNet, we only use the strategy of shape hijacking can make it does not work. The reason may be that RT-MDNet adopts the online update strategy, which makes the original target information vulnerable to pollution. Therefore, once the network is attacked, the strategy may lead to the models attention transfer to one wrong target. Considering this, we restrict the Min-Frame to 20 (one update period of [25]) for a simple shape hijacking, which is different from SiamRPN(++) and experiments show that 20 frames are enough to paralyze RT-MDNet.

Compared to RT-MDNet, the attack on the Siamese-based trackers is more challenging. The off-line training strategy of the Siamese-based tracker makes the target information never being polluted in the tracking process and a large number of regression bounding boxes increase the fault tolerance capacity of the algorithms. Despite it, our method has successfully attacked SiamRPN and SiamRPN++ algorithm with a relatively larger Min-Frames.

3.3. Ablation Study

Fig. 3. The visual results to illustrate the efficiency advantage of adaptive optimization against the normal method.

We evaluate the contribution of each component of our hijacking strategy by choosing the attack results of SiamRPN. As shown in Table 2, we can see that different attack losses cause different influences on the algorithm.

We first analyze the independent hijacking strategy. First, from the power of the attack, simply hijacking the shape and location of the tracker will reduce the success rate and accuracy of the tracker. More specifically, we found that the hijacking of location was more powerful than the hijacking of the shape. Because hijacked small boxes or normal boxes may still have relatively large IoU values because there is no offset in location. Conversely, if the two boxes of the same shape are far apart, the IoU value is zero. Second, from the efficiency of the attack method, the hijacking of location is also more efficient than the hijacking of shape, reflected by the number of attacked frames. Because the shape change is a continuous regression process while the variation of location can be discrete between two frames after being attacked.

We then evaluate the effect of combining the two strategies. The experimental results show that hijacking shape and direction simultaneously is stronger than using only one strategy. Because the feature information of the target is perturbed to the greatest extent due to the combination of the shape perturbation and location perturbation, making it difficult for the siamese network to keep up with the target in later video frames. However, the joint hijacking method has the lowest attack efficiency. We think this is because there is a conflict in the gradient descent during the optimization process.

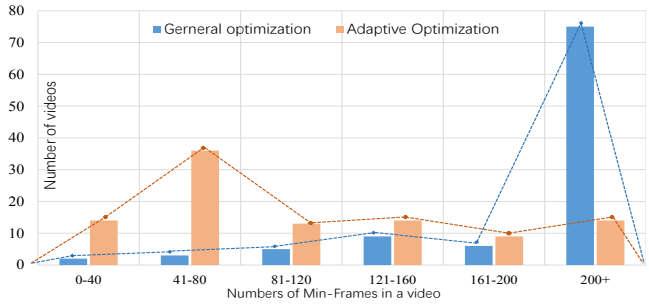

Fig. 4. Significant differences in attack efficiency between general optimization and adaptive optimization on OTB2015. The number of attack frames using adaptive optimization is greatly reduced.

Finally, the attack method using the adaptive optimization strategy achieves the best results in terms of both strength and efficiency due to its ability to quickly find the best optimization direction in the iterative process. Fig. 4 shows in detail the distribution of the Min-Frames on OTB2015. The vertices of the polyline represent the most concentrated intervals of the video distribution. We can see that for most of the video, adaptive iterative optimization can complete hijacking within 100 frames and the joint hijacking method without adaptive optimization requires about 200 frames. Fig. 3 shows the visual results of the ablation study on the adaptive optimization model. We can see the attack using adaptive optimization is more effective.

4. CONCLUSION

In this work, we explore adversarial attacks against single object tracking algorithms, which is important to the intelligent security system. We design an efficient and effective hijacking algorithm to gradually mislead the tracker to incorrectly predict the bounding box. Moreover, to relieve gradient conflict when optimizing the loss function, an adaptive optimization strategy is proposed. The evaluation results demonstrate the effectiveness and power of the proposed attack method. Our discoveries show that state-of-the-art tracking algorithms are vulnerable to malicious attacks, which reminds us to pay attention to the potential risks in intelligent monitoring systems. Our work makes some effort in this direction, and we hope that it can inspire more future research into this research perspective—attack and defense for tracking algorithms.

5. ACKNOWLEDGEMENT

This work is supported in part by the National Key Research and Development Program of China under Grant 2018YFB1800204, the National Natural Science Foundation of China under Grant 61972188, 61771273, the R&D Program of Shenzhen under Grant JCYJ201805-08152204044, the research fund of PCL Future Regional Network Facilities for Large-scale Experiments and Applications (PCL2018KP001).

6. REFERENCES

[1] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2016, pp. 850–865.

[2] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu, "High performance visual tracking with siamese region proposal network," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8971–8980.

[3] Zizhao Zhang, Yuanpu Xie, Fuyong Xing, Mason McGough, and Lin Yang, "Mdnet: A semantically and visually interpretable medical image diagnosis network," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6428–6436.

[4] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4282–4291.

[5] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 37, no. 9, pp. 1834–1848, 2015.

[6] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al., "The sixth visual object tracking vot2018 challenge results," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2018.

[7] Matthias Mueller, Neil Smith, and Bernard Ghanem, "A benchmark and simulator for uav tracking," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2016.

[8] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling, "Lasot: A high-quality benchmark for large-scale single object tracking," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5374–5383.

[9] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem, "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2018, pp. 300–317.

[10] Naveed Akhtar and Ajmal Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[11] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017.

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

[13] 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.

[14] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2574–2582.

[15] Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen, "Feature space perturbations yield more transferable adversarial examples," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7066–7074.

[16] Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, and Dacheng Tao, "Perceptual-sensitive gan for generating adversarial patches," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2019.

[17] Simen Thys, Wiebe Van Ranst, and Toon Goedemé, "Fooling automated surveillance cameras: adversarial patches to attack person detection," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 0–0.

[18] Yue Zhao, Hong Zhu, Qintao Shen, Ruigang Liang, Kai Chen, and Shengzhi Zhang, "Practical adversarial attack against object detector," *arXiv preprint arXiv:1812.10217*, 2018.

[19] Yunhan Jia, Yantao Lu, Junjie Shen, Qi Alfred Chen, Zhenyu Zhong, and Tao Wei, "Fooling detection alone is not enough: First adversarial attack against multiple object tracking," *arXiv preprint arXiv:1905.11026*, 2019.

[20] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1625–1634.

[21] Joseph Redmon and Ali Farhadi, "Yolo9000: better, faster, stronger," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7263–7271.

[22] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg, "Atom: Accurate tracking by overlap maximization," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4660–4669.

[23] Rey Reza Wiyatno and Anqi Xu, "Physical adversarial textures that fool visual object tracking," in *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 4822–4831.

[24] David Held, Sebastian Thrun, and Silvio Savarese, "Learning to track at 100 fps with deep regression networks," in *Proc. of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 749–765.

[25] Ilchae Jung, Jeany Son, Mooyeol Baek, and Bohyung Han, "Real-time mdnet," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2018, pp. 83–98.

[26] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.