Physical Hijacking Attacks against Object Trackers

Dishant Sharma[1], Harshit Kashyap[1], and Udrasht Pal[1]

International Institute of Information Technology, Hyderabad, Telangana, India
https://www.iiit.ac.in/
{dishant.sharma,harshit.kashyap,udrasht.pal}@students.iiit.ac.in

Abstract. In the field of modern autonomous systems, the combination of object detection and object tracking is crucial for accurate visual perception. While previous research has focused on attacking the object detection component of these systems, these attacks do not work on full pipelines that include object tracking. In contrast, existing attacks against object tracking lack real-world applicability or fail against Siamese trackers, a powerful class of object trackers. This paper introduces AttrackZone, a novel attack that can be physically implemented to hijack Siamese trackers by identifying regions in the environment that can be manipulated. AttrackZone exploits the heatmap generation process of Siamese Region Proposal Networks to take control of an object's bounding box, leading to potential physical consequences such as vehicle collisions and unauthorized pedestrian intrusion. The attack has been evaluated in both digital and physical domains and achieves its goals 92% of the time, with an average time requirement of 0.3-3 seconds.

Keywords: Object tracking · AttrackZone · Autonomous Driving · Siamese Network · Adversarial Attack · Video surveillance.

1 Summary

The most positive aspect of this article is that they are trying to attack a tracker which works on object tracking along with object detection. Siamese trackers can attack in both online and offline modes but the most negative aspect is the number of assumptions that they are assuming, hence making this attack very less effective in real-world scenarios.

The article explains how Object detection and object tracking are both important parts of the pipelines of different autonomous systems in various domains, such as autonomous driving pedestrian detection and mobile robot navigation.

This paper explains why the attack on object tracking is necessary for control hijacking because object detection is only half of the visual perception pipeline in autonomous systems object tracking is used to build the trajectories of objects to enhance the accuracy of object detection A visual perception pipeline for autonomous systems consists of an object detector and an object tracker Siamese system runs an object detector(Based on YOLO) to obtain a bounding box and the object class for each object of interest. These bounding boxes are then fed into an object tracker, which first computes its bounding box based on its historical results and then combines its results with the object detector. According to the author, by doing this they will reduce errors in visualization. After this siamese sends that data to the planning module to process data and make decisions for the automation system.

For tracking purposes, they are using the Siamese network which is a regional proposal network. This network classifies the image as either "background", to be ignored, or "target", to be tracked. It then uses a CNN to find the regions that correspond to a target, generating a heatmap for the target region. The heatmap restricts the search space of the tracker to only target areas.

In this attack experiment, they are using Base Siamese Networks (BSNs) SiamRPN, Distractor-aware Siamese Networks (DASNs) DaSiamRPN, and DaSiamRPN+ based on the generation of heat map Their main goal is tracker hijacking by moving in and moving out the objects:

Move-in Attack. The adversary in this attack aims to move the tracker of an object (e.g., vehicle, pedestrian, tumbleweed, animal) not in the target system's path (e.g., vehicle, drone) into its path.

Move-out Attack. In this attack, the adversary's goal is to move the tracked object out of the tracker's path. Although the object is still there, the tracker determines that it has moved out of its way because its tracker has deviated.

For experiment purpose SiamRPN, DaSiamRPN and DaSiamRPN based object trackers are used. Whole experiment is performed in three ways Emulated Attacks, Simulated Attacks on CARLA, Real-world Physical Attacks. The Real world attack is also performed in two ways Online attack, Offline attack

Results of each experiment is mentioned in the form of table below inside the critiques section

1.1 Working : AttrackZone Algorithm

According to author they are creating replica of tracking system to get the view of the victim camera and its 3D point cloud for identification of attack zones. they are taking that data either in offline or online mode Once these attack zones are determined, then exploitation of the Siamese heatmap proposal algorithm and generation of noise patterns to shift the tracker's focus to desired attack zone will happen. For this attacker use any open-source Siamese-based object tracker for generating the physical perturbations

According to author during the creation of perturbation they ensure that attack is minimally noticeable while evading established defenses (such as Kalman Filtering) and remaining adaptive to the environment then Lastly, to launch the tracker hijacking, they remotely projects the perturbations to the attack scene, e.g., by attaching a projector to an adversarial vehicle driving near the victim vehicle, or flying an adversarial drone with a mounted projector.

2 Critiques

2.1 Additional use of sensors of different classes

According to their assumption, they are trying to hijack controls of autonomous driving and video surveillance but these devices may use different sensors of more than two types like LIDAR or SONAR with camera feed so that if siamese tracker based on the camera feed deviates from actual target other sensors will stop shifting of control focus to a wrong direction. This helps minimize the chance of serious injury occurring due to a tracker hijacking attack and can detect an attack in real time.

There are several real-world technologies that use sensor fusion to enhance security and protect against control hijacking attacks on Siamese networks in autonomous driving and video surveillance. One example is the Advanced Driver Assistance Systems (ADAS) technology used in modern vehicles, which combines inputs from multiple sensors, such as cameras, LiDAR, and radar, to provide a more comprehensive understanding of the vehicle's environment and detect potential threats.

Suggestion: This approach can help detect anomalies or inconsistencies in sensor data. Although the paper mentions that sensor fusion algorithms are vulnerable to an adversary who can compromise only one of the fusion sources, this also restricts the success of our attack to some extent.

2.2 Use of Machine learning algorithm to detect anomaly

To perform control hijacking, they have accepted that 10-100 frames with sequential perturbation are necessary for a duration of 0.3-3 seconds. Therefore, we can deploy machine learning algorithms that can detect abnormal activity in the tracking system. When the perturbation is created by the attacker very rapidly and projected to the object, to miss and guide the tracking object very fast, at that moment, the change of the tracking object box creates the anomaly.

Suggestion: Anomaly detection algorithms can detect unusual tracking behavior that may be indicative of an attack. There are various algorithms with the help of this autonomous system to detect anomalies.

Support Vector Machines (SVMs): SVMs are another machine learning algorithm that can be used for anomaly detection in object tracking. In this case, the SVM would be trained on a dataset of normal tracker box positions and used to classify new positions as either normal or anomalous.

Hidden Markov Models (HMMs): HMMs are probabilistic models that can be used to model time-series data, such as object tracker box positions over time. The model can be trained on a dataset of normal object trajectories and then used to detect deviations from those trajectories.

2.3 Lack of Dataset for experiments

2.3.1 Emulated Attack Results Emulated Attack Results: They are using the complete 15 test samples from the VOT dataset for autonomous driving (66.7%) and video surveillance (33.3%). The VOT is a multi-purpose visual dataset that includes samples from various object-tracking applications. However, the VOT dataset does not provide the 3-D point cloud of a scene. That's why the whole idea of creating an automatic attack zone is not applicable here, and segmentation of images is done manually.

Source Tracker	DaSiamRPN	93.3%	100%	93.3%
	SiamRPN	93.3%	100%	93.3%
	DaSiamRPN+	93.3%	100%	93.3%
		DaSiamRPN	SiamRPN	DaSiamRPN+
		Destination Tracker		

Suggestion: If they are taking that much advantage of manual selection, they should increase the dataset. As shown by the table above, the accuracy of this attack on SiamRPN is 100%, which is not feasible and will not give a correct analysis of the attack.

2.3.2 Simulated Attacks on CARLA The dataset, created in the CARLA simulator, includes both image and LiDAR data collected in a realistic environment. As with the above experiment, if LiDAR data is already available, creating an attack zone is easier compared to creating one in real time using a 3-D point cloud. Another point to focus on in the table below is that the success rate of attacks on DaSiamRPN+ is 6.6% lower compared to other trackers. This proves that training an ML model to catch distractions frame-by-frame can make our system more robust against control hijacking.

Source Tracker	DaSiamRPN	93.3%	93.3%	86.7%
	SiamRPN	93.3%	93.3%	86.7%
	DaSiamRPN+	93.3%	93.3%	86.7%
		DaSiamRPN	SiamRPN	DaSiamRPN+
		Destination Tracker		

Suggestion: As perturbations are created to evade the Kalman filter and tested on it, we suggest using an extended version of Kalman filters like the Unscented Kalman Filter (UKF). The UKF is another nonlinear filter that represents the probability distribution of the system state using a set of sample points (called sigma points). It has been shown to be more accurate than the EKF for some nonlinear systems.

2.3.3. Real-world Experiments In the attack scenario creation, it is mentioned that in the absence of a LiDAR sensor, valid attack zones for the environments are manually mapped. They have only taken 9 offline real-world attacks into consideration, which is not sufficient for predicting the success of an attack effectively. Similarly, for online attacks, they have conducted attacks against moving vehicles and pedestrians in controlled settings with minimal traffic, such as on the highest level of an empty parking garage during daylight hours. Likewise, experiments on streets were conducted at night on residential roads with little traffic. Despite taking all these precautions and having a very small dataset with only 15 attacks, the success results mentioned in the table are not so effective.

Source Tracker	DaSiamRPN	88.9% / 88.3%	88.9% /88.3%	100% / 66.7%
	SiamRPN	88.9% / 72.7%	77.8% /90.9%	100% / 100%
	DaSiamRPN+	71.4% / 60%	85.7% /80%	100% / 80%
		DaSiamRPN	SiamRPN	DaSiamRPN+
		Destination Tracker		

Suggestion: To achieve more accurate results, they should increase the dataset by testing their attack in more scenarios. Additionally, they should conduct these tests in controlled settings with artificial replicas of different objects and obstacles to determine the accuracy of the attack.

2.4 Flaw in Assumption

Based on the assumption used in this paper, the attacker knows the location of the system running the object tracker (where the camera or cameras are mounted). With this knowledge, the attacker uses a surrogate camera (e.g., a stereo camera or camera with LiDAR) to obtain the view of the victim camera and its 3D point cloud for identifying attack zones. The author also suggests that using a drone to fly close to the victim's camera is the most convenient approach for creating a similar viewpoint. To find out the view of the victim, the author also suggests discreetly attaching a camera to the victim's car, which would capture footage from a similar perspective as the victim's.

All of the above methods may or may not give the same view as the victim camera. As the attacker should not know the exact location of the victim, creating an effective attack immediately in the real world may be challenging.

Suggestion: Instead of using the above methods to find the victim's view, the attacker should try to gain access to the victim's camera feed directly. In today's world, it is possible to gain access to a victim's camera feed without using a surrogate camera. There are various techniques that an attacker could use to achieve this, such as exploiting vulnerabilities in the camera's software or network or using social engineering techniques to gain access to the victim's device or network.

For example, an attacker could exploit a vulnerability in the victim's camera software or network to gain remote access to the camera feed. They could also use a phishing attack or other social engineering technique to trick the victim into installing malware on their device, which would allow the attacker to gain access to the camera feed.

2.5 Technology Issue

This attack targets trackers based on Siamese networks. However, the results shows that the DaSiamRPN+ algorithm used by the tracker is very effective in detecting the distractions created by the attacker. As shown in the table success rate of attack is reduced to 66.7% when DaSiamRPN is used. To overcome this issue there are two suggestion as,

Suggestions:

SiamRPN++ - To address this problem, we can use SiamRPN++ to generate physical perturbations based on the surrogate camera feed.

Kernelized Correlation Filters - Also, the latest autonomous driving and video surveillance techniques are using the Kernelized Correlation Filters tracker (KCF), which is much more efficient than the Siamese tracker. Therefore, attacks generated by the Siamese network may not be effective.

3 References

[1].shoei Nashimoto, Daisuke Suzuki, Takeshi Sugawara, and Kazuo Sakiyama. Sensor con-fusion: Defeating kalman filter in signal injection attack. In Asia Conference on Computer and Communications Security, 2018

[2].Fooling Detection alone is not Enough: Adversarial attack against multiple object tracking,2020

[3].Open Source Computer Vision. Opencv: Kalmanfilter reference. https://docs. opencv.org/3.4.1/dd/d6a/classcv$_{11}KalmanFilter.html, 2022.[Online; Accessed27-August-2022]$.

[4]$.BingShuai, AndrewG.Berneshawi, XinyuLi, DavideModolo, andJosephTighe.Siammot: Siamesemulti-objecttracking.ComputerVisionandPatternRecognition(CVPR)$, 2021.